CONCEPTUAL SYSTEMS SECURITY ANALYSIS AERIAL REFUELING CASE STUDY

THESIS

Martin "Trae" Span III, Captain, USAF

AFIT-ENV-MS-18-M-237

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A.
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government. This material is declared a work of the U. S. Government and is not subject to copyright protection in the United States.

AFIT-ENV-MS-18-M-237

CONCEPTUAL SYSTEMS SECURITY ANALYSIS AERIAL REFUELING CASE STUDY

THESIS

Presented to the Faculty

Department of Systems Engineering and Management

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Systems Engineering

Martin “Trae” Span III, BS

Captain, USAF

March 2018

DISTRIBUTION STATEMENT A.
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT-ENV-MS-18-M-237

CONCEPTUAL SYSTEMS SECURITY ANALYSIS AERIAL REFUELING CASE STUDY

THESIS

Martin “Trae” Span III, BS

Captain, USAF

Committee Membership:

Lt. Col Logan O. Mailloux, PhD
Chair

Dr. Robert F. Mills, PhD
Member

Col William Young, PhD
Member

Abstract

In today's highly interconnected and technology reliant environment, systems security is rapidly growing in importance to complex systems such as automobiles, airplanes, and defense-oriented weapon systems. While systems security analysis approaches are critical to improving the security of these advanced cyber-physical systems-of-systems, such approaches are often poorly understood and applied in ad hoc fashion. To address these gaps, first a study of key architectural analysis concepts and definitions is provided with an assessment of their applicability towards complex cyber-physical systems. From this initial work, a definition of cybersecurity architectural analysis for cyber-physical systems is proposed. Next, the System Theory Theoretic Process Analysis approach for Security (STPA-Sec) is tailored and presented in three phases which support the development of conceptual-level security requirements, applicable design-level criteria, and architectural-level security specifications in alignment with the systems engineering standard ISO/IEC/IEEE 15288 and the newly released NIST SP 800-160 Systems Security Engineering publication.

This work uniquely presents a detailed case study of a conceptual-level systems security analysis of a notional aerial refueling system based on the tailored STPA-Sec approach. This case study provides a detailed security analysis with emphasis on how to conduct STPA-Sec early in the development process to more accurately elicit, understand, and define security requirements in three levels of increasing detail: 1. Initial security requirements are systematically elicited; 2. Design for security requirements are

identified; and 3. Build-to security specifications are formally defined. This work is critically important for advancing the science of systems security engineering by providing a standardized approach for understanding security, safety, and resiliency requirements in complex systems with traceability and testability.

Acknowledgments

I am certainly blessed to have this opportunity; I give credit to God for his favor throughout my degree program and express my gratitude to my family and especially my wife for her support.

I would like to thank Lt. Col Logan Mailloux for his investment in me and this research. This thesis would not be nearly as polished or 'word smithed' without your abundant time and effort. Your technical writing prowess and passion for research is admirable. Additional thanks go to Colonel Young for the 'Jedi' training sessions in STPA-Sec. I appreciate your mentorship and admire how you accomplish as much as you do.

Martin "Trae" Span III

Table of Contents

Page

Abstract iv
Table of Contents vii
I. INTRODUCTION 1
General Issue 1
Problem Statement 2
Research Context 2
Investigative Questions 3
Methodology 4
Assumptions/Limitations/Scope 4
The Way Ahead 5
Bibliography 7
II. Literature Review 9
Description: 9
Publication Details: 9
III. Methodology 18
Description: 18
Publication Details: 18
IV. Case Study 33
Description: 33
Publication Details: 33
V. Conclusions and Recommendations 48
Conclusions of Research 48
Significance of Research 53

Recommendations for Future Research.....54

Bibliography.....55

CONCEPTUAL SYSTEMS SECURITY ANALYSIS AERIAL REFUELING CASE STUDY

I. INTRODUCTION

General Issue

In today's highly interconnected and technology reliant environment, systems security is rapidly growing in importance. As the Internet of Things continues to grow, the centrality of cyber-physical devices to modern life is increasingly important. Thus, security (and safety) is now an emergent property of cyber-physical systems, where their software and real-time networks require continuous interaction [1]. For example, the 2017 Ford F-150, a fairly common vehicle, has over 150 million lines of code distributed across dozens of computing devices with software providing essential functionality [2], [3]. Moreover, intelligent adversaries are challenging traditional assumptions that cyber-physical systems are secure due to their relative isolation and uniqueness with recent examples including the widely publicized hacking demonstration against a Jeep Cherokee [4], claims of hacking a commercial airliner [5], and comprehensive reports of vehicular attack paths [6].

In light of these growing threats, the United States Department of Defense (U. S. DoD) has made recent changes to expand traditional IT-focused security approaches and mandate security assessments for major weapon systems (MWS) [7], [8], [9], [10], [11], [12]. These policies dictate that acquisition programs integrate security efforts into existing systems engineering processes, and work to ensure security considerations hold equal footing with other requirements and design trade-offs at major program reviews.

Although, these DoD mandates are in place, a well-received streamlined executable approach for MWS cybersecurity analysis is yet to be defined. This research explores these problems and presents a case study of interest to the USAF.

Problem Statement

While systems security analysis approaches are critical to improving the security of complex systems-of-systems, such approaches are often poorly understood and applied in ad hoc fashion. Across the defense industry, the DoD, and the Air Force program offices have differing approaches to this problem. Most of the system security approaches surveyed are focused on realized systems with limited solution space as they begin analysis at the physical system solution where the cost to design for security and resiliency is greatly increased. Moreover, the lack of measurable and verifiable security and resiliency requirements in early, pre physical form, system design has plagued the development of complex weapons systems often leaving mission owners with little choice between restricted functionality and expensive bolt on security features, often at the detriment of mission effectiveness.

Research Context

Emphasizing the importance of this problem, the US Congress included a mandate and provided significant funding in the National Defense Authorization Act of 2016 Section 1647 [8] to assess all of major weapons systems across all services for cyber vulnerabilities. In response, the Air Force stood up the Cyber Resiliency Office for Weapons Systems (CROWS) to develop and execute the Air Force Cyber Campaign Plan's (CCP) two overall goals: 'Bake In' cyber resiliency to new weapons systems, and,

in line with the congressional mandate, mitigate ‘critical’ vulnerabilities in fielded systems [13]. The CROWS office has emphasized the importance of integrating cyber into systems engineering through training of a cyber savvy acquisition force. Cybersecurity for weapons systems in no longer a task just for the IT professional, but should be integrated in systems engineering efforts. It must be specified in early requirements to be included in the design trades with all other mission needs. In light of the growing cybersecurity threat, a clear necessity has emerged for a streamlined executable early systems based approach.

This work directly contributes to the CROWS and CCP Line of Action 3‘Train Cyber Workforce’ by providing a case study example of a conceptual security analysis to elicit security requirements for future major weapons systems.

Investigative Questions

1. What is Cybersecurity Architectural Analysis?
2. What methods exist for conducting Cybersecurity Architectural Analysis?
3. What are the key characteristics for Cybersecurity Architectural Analysis and how do they map to current approaches for complex cyber-physical systems?
4. How can STPA-Sec be tailored to enable the development of security requirements and design criteria?
5. How executable is STPA-Sec for USAF warfighting Systems?
6. What recommendations can be made to increase the utility and ease the use of STPA-Sec?

Methodology

This research effort is conducted in two parts: First, this work surveys multiple approaches used to perform complex system security analysis (i. e. , cybersecurity architectural analysis) with specific focus to those relevant to the DoD and complex weapon systems. This research informs tailored definitions and desirable characteristics for conducting systems security. From this survey, STPA-Sec was chosen for further study and experimentation.

Second, we conduct a case study offering a complete and thorough example of STPA-Sec on a notional next generation aerial refueling platform. STPA-Sec is a promising approach for performing conceptual systems security analysis based on a methodology of systems theory dating back to Leveson's original systems safety STAMP work which has been well received within the safety, aeronautical, and systems engineering communities [14], [15]. STPA-Sec is an extension of her methodology to the security domain and has been shown to effectively address security issues in complex cyber-physical systems [16]. This case study evaluates STPA-Sec's utility to perform conceptual systems security analysis for United States Department of Defense Major Weapon Systems.

Assumptions/Limitations/Scope

To maximize the audience of this document, this thesis excludes sensitive or classified sources and information. The scope of this research effort is limited to system security approaches with publically available and citable documentation to enable the broadest distribution. Conclusions drawn on the effectiveness of the surveyed methods

are limited to SME inputs and author discretion as completion of a case study with each approach for direct comparison is beyond the scope of this research effort and would in fact take years to complete with little value. Mission specific details have been obfuscated and generalized for widest distribution.

The Way Ahead

With the progressive nature of the research questions, this thesis will follow a scholarly, or k-paper, format. In Chapter II, the publication "Cybersecurity Architectural Analysis for Complex Cyber-Physical Systems" provides an extended background and literature review for this thesis. Of note, the work details key terms for systems security and surveys several architectural analysis approaches from government and industry. Moreover, it suggests a working definition of Cybersecurity Architectural Analysis, and identifies desirable characteristics for effective architectural analysis.

In Chapter III, the publication "A Systems Security Analysis Approach for Understanding, Defining, and Specifying Security Requirements for Complex Cyber-Physical Systems" introduces a tailored approach for STPA-Sec, a promising methodology for conceptual systems security analysis. In addition, an abbreviated case study for a space vehicle is presented to provide insight into the processes and benefits of an STPA-Sec analysis.

In Chapter IV, the journal publication "Conceptual Systems Security Analysis with Aerial Refueling Case Study" provides a detailed overview of the methodology and data used for the case study portion of this thesis. A tailored approach of STPA-Sec is presented building upon the foundation from Chapter III and expanded with additional

insight and recommendations for execution of STPA-Sec for a complex cyber-physical system. This is presented as a case study for a next generation aerial refueling platform.

Chapter V summarizes research questions with an emphasis on impact to the DoD, presents conclusions, and identifies future work.

Bibliography

[1] Y. Liu, Y. Peng, B. Wang, S. Yao and Z. Liu, "Review on cyber-physical systems," *IEEE/CAA Journal of Automatica Sinica,* vol. 4, no. 1, pp. 27-40, 2017.

[2] R. Saracco, "Guess What Requires 150 Million lines of Code," EIT Digital, 13 Jan 2016. [Online]. Available: https://www. eitdigital. eu/news-events/blog/article/guess-what-requires-150-million-lines-of-code/. [Accessed Feb 2017].

[3] R. Charette, "IEEE Spectrum: This Car Runs on Code," 1 February 2009. [Online]. Available: http://spectrum. ieee. org/transportation/systems/this-car-runs-on-code. [Accessed 1 June 2017].

[4] A. Greenberg, "Wired," Wired Magazine, 21 July 2015. [Online]. Available: https://www. wired. com/2015/07/hackers-remotely-kill-jeep-highway/. [Accessed 25 April 2017].

[5] E. Perez, "CNN," CNN, 18 May 2015. [Online]. Available: http://www. cnn. com/2015/05/17/us/fbi-hacker-flight-computer-systems/. [Accessed 25 April 2017].

[6] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham and S. Savage, "Comprehensive Experimental Analyses of Automotive Attack Surfaces," USENIX Security Symposium, 2011.

[7] K. Baldwin, J. Miller, P. Popick and J. Goodnight, "The United States Department of Defense Revitalization of System Security Engineering Through Program Protection," in *IEEE Systems Conference*, 2012.

[8] United States Congress, "Nation Defense Authorization Act 2016 Section 1647," 25 November 2015. [Online]. Available: https://www. congress. gov/114/plaws/publ92/PLAW-114publ92. pdf. [Accessed 1 June 2017].

[9] Department Of Defense, "DoDI 8500. 01 Cybersecurity," 2014.

[10] Department of Defense, "DoDI 8510. 01 Risk Management Framework (RMF) for

DoD Information Technology (IT)," 2014.

[11] Department of Defense, "Defense Acquisition Guidebook Chapter 9 Program Protection," 5 April 2017. [Online]. Available: https://www. dau. mil/tools/dag/Pages/DAG-Page-Viewer. aspx?source=https://www. dau. mil/guidebooks/Shared%20Documents%20HTML/Chapter%209%20Program%20Protection. aspx. [Accessed 1 June 2017].

[12] Department Of Defense, "DoD Program Manager's Guidebook for Integrating the Cybersecurity Risk Management Framework (RMF) into the System Acquisition Lifecycle," 30 October 2015. [Online]. Available: https://acc. dau. mil/adl/en-US/721696/file/81323/Cybersecurity%20Guidebook%20v1. 10%20signed. pdf. [Accessed 1 June 2017].

[13] USAF Public Affairs, "AF Looks to Ensure Cyber Resiliency in Weapons Systems," 4 Jan 2017. [Online]. Available: http://www. af. mil/News/Article-Display/Article/1041426/af-looks-to-ensure-cyber-resiliency-in-weapons-systems-through-new-office/. [Accessed 2 Feb 2018].

[14] N. Leveson, "A new accident model for engineering safer systems," *Safety science,* vol. 42, no. 4, pp. 237-270, 2004.

[15] Massachusetts Institute of Technology, "MIT Partnership for a Systems Approach to Safety," 27 March 2017. [Online]. Available: http://psas. scripts. mit. edu/home/stamp-workshop-2017/. [Accessed 8 June 2017].

[16] N. Leveson and J. Thomas, "An STPA Primer," 9 September 2013. [Online]. Available: http://sunnyday. mit. edu/STPA-Primer-v0. pdf. [Accessed 12 August 2017].

II. Literature Review

Description:

Chapter II is a self-contained conference paper that provides an extended background and literature review. It provides key terms and concepts in support of familiarizing the System Engineer with systems security analysis for complex system cybersecurity.

This work answers research question 1 of the thesis providing a definition for Cyber Architectural Analysis. It fulfills question 2 through surveying existing architectural analysis approaches, and answers question 3 by identifying key characteristics and providing a mapping of utility for the practitioner.

Publication Details:

Title: Cybersecurity Architectural Analysis for Complex Cyber-Physical Systems

Publication: Submitted to Cyber Defense Review(CDR) journal

Cybersecurity Architectural Analysis for Complex Cyber-Physical Systems

Martin "Trae" Span, Logan O. Mailloux, Michael R. Grimaila
Air Force Institute of Technology
Wright-Patterson Air Force Base, Ohio

***Abstract*—In the modern military's highly interconnected and technology-reliant operational environment, cybersecurity is rapidly growing in importance. Moreover, cybersecurity is no longer limited to traditional computer systems and IT networks, as a number of highly publicized attacks have occurred against complex cyber-physical systems such as automobiles and airplanes. While architectural analysis approaches are critical to improving cybersecurity, these approaches are often poorly understood and applied in ad hoc fashion. This work addresses these gaps by answering the questions: 1. "What is cybersecurity architectural analysis?" and 2. "How can architectural analysis be used to more effectively support cybersecurity decision making for complex cyber-physical systems?" First, a readily understandable description of key architectural concepts and definitions is provided which culminates in a working definition of "cybersecurity architectural analysis," since none is available in the literature. Next, we survey several architectural analysis approaches to provide the reader of an understanding of the various approaches being used across government and industry. Based our proposed definition, the previously introduced key concepts, and our survey results, we establish desirable characteristics for evaluating cybersecurity architectural analysis approaches. Lastly, each of the surveyed approaches is assessed against the characteristics and areas of future work are identified.**

Keywords—cybersecurity; architectural analysis; system architecture; systems security engineering; complex system security

I. Introduction

The cybersecurity threat is one of the most serious economic and national challenges we face as a nation – economic prosperity in the 21st century depends on cyber [1]. Cyber attacks have grown in frequency and complexity, and it is now commonplace to hear of widespread cyber attacks on personal computers, webservers, and even large company and government personnel databases [2]. Moreover, as the Internet of Things (IoT) continues to grow, the centrality of cyber-physical devices to modern life is increasingly important [3]. Previously, cyber-physical systems such as automobiles and airplanes were relatively simplistic. Astonishingly, the 2017 Ford F-150, a relatively common vehicle, has over 150 million lines of code [4], demonstrating the complexity of modern systems when software is at the core of functionality [5]. For these cyber-enabled systems, adversaries are challenging traditional assumptions that systems are secure due to their relative isolation and uniqueness. Recent examples include a widely publicized hacking demonstration against a Jeep Cherokee [6], claims of hacking a commercial airliner [7], and comprehensive reports of vehicle vulnerabilities [8]. In light of this growing threat, it is critical to analyze modern weapon systems for cybersecurity vulnerabilities as directed by United States Congress [9].

Recent United States (U.S.) Department of Defense (DoD) policy updates have expanded the traditional IT security approaches and mandated cybersecurity assessments for cyber-enabled weapon systems [9], [10], [11], [12]. These revisions dictate that acquisition programs integrate cybersecurity efforts into existing systems engineering processes, and work to ensure cyber considerations hold equal footing with other requirements and design trade-offs at major acquisition milestones [13].

For highly complex systems, including U.S. DoD weapon systems, architectural analysis is a critical enabler to effective cybersecurity; however, architectural analysis approaches are often poorly understood and applied in ad hoc fashion. This work addresses these gaps by answering the questions:

1. "What is cybersecurity architectural analysis?"
2. "How can architectural analysis be used to more effectively support cybersecurity decision making for cyber-physical systems?"

This paper examines and proposes answers to the above questions. In Section II, we provide a readily understandable discussion of key concepts and definitions. Section III expands on this foundation and surveys several cybersecurity architecture analysis approaches from government and industry. In Section IV, desirable characteristics for architectural analysis for cybersecurity are identified and mapped to the approaches from Section III. Lastly, Section V summarizes key findings and identifies promising follow-on research areas for increasing the effectiveness of cybersecurity architectural analysis of unprecedented systems, specifically modern complex cyber-physical systems.

II. Foundational Concepts and Definitions

This section provides a brief historical context for system-level architectural analysis and, more formally, discusses key definitions for cybersecurity architectural analysis.

A. Brief History of System Architecture

Much of the seminal work in the field of architecture analysis was accomplished by Zachman, who proposed the first system architecture—a logical construct for integrating the complexities of modern information systems [14]. Similarly to the varying levels of abstraction in physical construction plans, Zachman argued that system architectures should be composed of many perspectives in varying levels of detail. Moreover, he insisted that these perspectives (or views) be synchronized across the system, forming one integrated architecture.

Sowa expanded Zachman's work to form the Information Systems Architecture (ISA) framework [15]. Shown in Fig. 1, the ISA employs six interrogatives (what, how, where, who, when, and why) across five levels of detail (scope, business, system, technology, and detailed representations) as a means of expressing relationships to guide complex system development [16]. In this way, the ISA offers a simplified approach to compare and elaborate on the desired capabilities, requirements, components, and functions in an integrated enterprise-level model which enables effective decision making. Note, not all 30 conceptual graphs are required; thus, the ISA is also tailorable. Since its inception, the ISA (commonly known as the Zachman Framework) has been a popular choice for system architects—it has been widely used by system architects for decades, while several other system-level frameworks have incorporated or adopted its tenets [17].

B. *Key Definitions*

Here we discuss definitions for key terminology used in this work (i.e., "cybersecurity," "architecture," and "analysis"). First, the term "cybersecurity" should be addressed because it is generally the most poorly understood (see sidebar in [18]). Within the U.S. DoD, cybersecurity is formally defined as:

> *The prevention of damage to, protection of, and restoration of electronic systems to ensure its availability, integrity, authentication, confidentiality, and nonrepudiation* [19].

Despite being often cited, this definition tends to cause confusion because it is packed with domain-specific IT jargon: availability ensures the system is usable as anticipated; integrity is the protection from unauthorized modification; confidentiality is keeping data private; authentication is a validation of the claimed identity; and nonrepudiation is the ability to prove that an action has taken place. While seemingly comprehensive, the U.S. DoD definition is somewhat hindered with legacy terminology; a more practical (i.e., a working) definition of cybersecurity might simply seek to protect critical systems against cyber-based threats [20].

The next key term to define is "architecture" (note, we interpret "architecture" synonymously with "system architecture" and/or "system-level architecture"). Perhaps the most classically understood definition of architecture is provided by Maier and Rechtin:

> *Structure in terms of components, connections, and constraints of a product, process, or element* [21].

This definition offers a holistic view of the system of interest to include technological aspects as well as non-technological aspects, such as processes. In the simplest terms, an architecture merely provides a means for viewing the system of interest from different perspectives. Conversely, in a somewhat physically-driven characterization, ISO/IEC/IEEE 42010 provides the following definition for architecture:

> *The fundamental organization of a system, embodied in its components, their relationship to each other and to the environment, and the principles governing its design and evolution* [22].

Somewhat surprisingly, the U.S. DoD provides a very progressive definition of system architecture:

> *A set of abstractions (or models) that simplify and communicate complex structures, processes, rules, and constraints to improve understanding and implementation* [23].

In addition to being readily understandable, this definition alerts the reader to the intrinsic value offered by such architectures in that they serve to simplify communication with, and improve understanding of, key stakeholders (not just engineers). Moreover, this definition implies that architectures are intended to improve the system's implementation. While these value-rich aspects of the definition are a bit atypical, they are useful for helping others to understand what an architecture is and does.

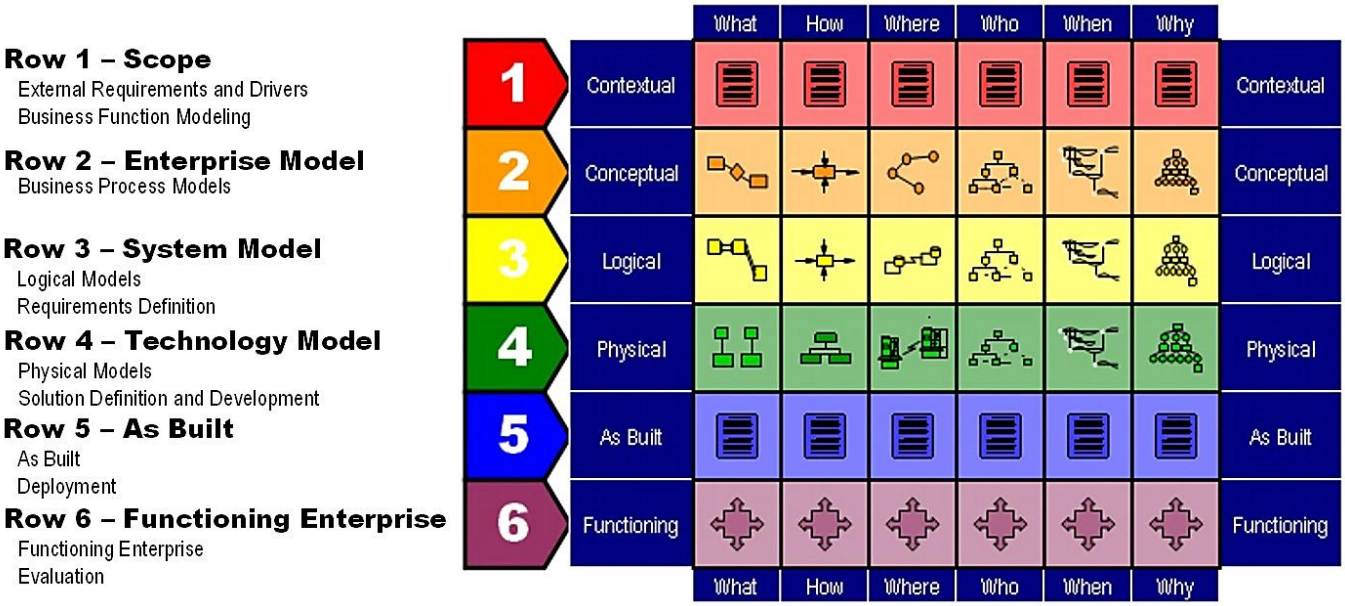


Fig 1. The Zachman Framework for Enterprise Architecture [24].

Lastly, the task of identifying a formal definition of "analysis" within the context of a "system architecture" proved more difficult than previous definitions. Often a systems architecture will center on an integrated model of entities and the relationships between them; architectural models serve as a vehicle to bring order, and thus understandability, to the growing complexity associated with complex systems. An architecture-focused definition may read as such

> *Architectural analysis is the activity of discovering important system properties using conceptual and physical models of the system of interest* [25].

However, an architecture's purpose is to increase understanding and facilitate better engineering choices [17]. This two-fold purpose is acutely stated by Crawley *et al.*:

> *Architectural analysis focuses on understanding both the architecture's function and form for the purpose of supporting decision making* [26].

It is worth noting the closely related concept of architecture trade-off analysis, which focuses on evaluating and comparing alternative architecture-level designs and attributes (e.g., modifiability, security, performance, reliability, etc.) [27].

C. Cybersecurity Architectural Analysis Working Definition

Ultimately, architectural analysis identifies trade-off points among system attributes and facilitates communication among stakeholders (e.g., customers, developers, operators, maintainers). System-level architectural analysis requires consideration of various missions, essential functions, potential components, and desirable attributes, which help to clarify and refine stakeholder needs and, later, requirements. Moreover, integrated architectural analysis provides a robust framework for ongoing and concurrent system design and analysis.

Specific to the cyber domain, architectural analysis should be used to understand cyber dependencies within the functions and form of the system to enable well-informed decisions. This type of structured analysis brings an otherwise unmanageable amount of information under control in support of system security requirements [28]. Architectural analysis enables system-level programmatic risk management by providing context and functional mapping to the various physical elements of the system. Thus, cybersecurity architectural analysis allows appropriate security mitigations to be applied where needed with rigorous justification.

After considering seminal definitions in the area, and working through the various architectural analysis approaches discussed in Section III, we present a working definition of cybersecurity architectural analysis for consideration:

> *The activity of discovering and evaluating the function and form of a system to facilitate cybersecurity decisions.*

This definition identifies two key activities, discovery and evaluation, while simultaneously catering to both new development (i.e., a focus on desired capability through functionality) and legacy systems (i.e., a focus on existing system solutions). For new developments, discovery typically implies exploring the business or mission problem space to further understand the desired capability through functional analysis. For existing systems, this process is often conducted in the reverse, mapping critical subsystems back to critical functions which support important business operations or mission execution. It is also worth noting that cybersecurity architectural analysis should also help with identifying and understanding how security requirements support the desired capability, which also provides traceability that is often lacking in systems security efforts.

As part of the broader system definition and development effort, cybersecurity architectural analysis should help inform engineering tradeoffs and decision making such as those processes and activities described in ISO/IEC/IEEE 15288.

III. Current Cybersecurity Architectural Analysis Approaches

In this section, we survey architectural analysis approaches and assess their applicability for complex system cybersecurity. Within the U.S. DoD (and its major defense contractors), several approaches (i.e., methods, processes, and tools) have been developed to secure and assess the cybersecurity of complex systems and systems-of-systems. While providing a detailed case study for each approach surveyed in this work would be ideal for a robust assessment, it is simply not feasible as some approaches take months if not years to complete. This survey is based on publicly available literature and presentations that focus specifically on architectural analysis for weapon systems.

The predecessor for many cybersecurity architectural analysis approaches is compliance-based Information Assurance (IA), which focuses almost exclusively on applying security controls to computer networks and IT systems. For complex systems this approach is inadequate as demonstrated by several high profile security breaches [29]. This inadequacy has driven the development of many of the approaches described in this work.

A. Department of Defense Architectural Framework (DoDAF)

The integrated architecture currently in use by the U.S. DoD is the DoD Architecture Framework (DoDAF). Its purpose is to manage complexity to enable key decisions through organized information sharing [23]. However, in DoDAF, like many other architecture frameworks, security (or cybersecurity) is not specifically addressed [30]. James Richards, in his work *Using the Department of Defense Architecture Framework to Develop Security Requirements* [28], proposes a methodology for using DoDAF to derive security requirements. He outlines a process of first building an architectural model of the enterprise, focusing on a core set of views including the OV-5b operational activity model, the DIV-2 logical data model, and the OV-3 operational resource flow matrix. These critical views are used to model security-relevant processes, data, business rules, and communications. Next, he suggests comparing views for compliance and then assessing and refining the architecture. The overall purpose of Richards' approach is to use DoDAF to expose or derive security requirements [28]. This approach has not been widely adopted but his work demonstrates utility for complex cyber-physical systems.

B. Unified Architecture Framework (UAF)

In contrast to the U.S. DoD unique solution DoDAF, industry has developed the Unified Architecture Framework (UAF) [31]. Based on industry need, the UAF includes a formal security domain amongst the more common architectural views. The UAF security domain includes views for security taxonomy, structure, connectivity, processes, constraints, and traceability. More specifically, it uses SysML class diagrams to identify data types and map them to protections and security controls. As an integrated architecture, it allows security-relevant elements to be mapped to system resources and operations. UAF also capitalizes on the success of MBSE efforts to depict and analyze the security properties of a SoI via an executable architecture. Note, UAF is in the final stages of development, so its utility has yet to be fully realized; however, the some pathfinder examples of proposed security views demonstrate utility for conducting cybersecurity architectural analysis of complex cyber-physical systems [32].

C. Publically Avilable Industry Efforts

Major defense contractors often use custom architectural analysis approaches to design and evaluate their system architectures with respect to cybersecurity. Although it is likely that most large U.S. DoD contractors are working solutions in this area; at the time of this survey, the authors were only exposed to efforts from Raytheon, Northrop Grumman, and Lockheed Martin. Note, Raytheon's Cyber Resiliency Architecture Framework (CRAF) was the only approach with a detailed open source publication available. Limited information is available on Northrop and Lockheed's approaches.

Raytheon developed CRAF using a DoDAF reference architecture with extensions for specific cyber resilience mappings and metrics [33]. The goal of CRAF is to assess and identify gaps in cyber resiliency by mapping systems, subsystems, and components against prioritized capabilities to identify resilience requirements for important mission scenarios.

Using failure modes and effects analysis, Northrop Grumman created a risk-based assessment methodology using an integrated architecture modeled in the new UAF to identify cyber risks for their systems [32]. This approach is still under development and is one of the first systems security efforts based on the upcoming UAF standard security views from the Object Management Group (OMG).

Lockheed Martin has created a custom solution titled the Secure Engineering Assurance Model (SEAM) [34]. SEAM is a tailored systems security engineering approach to integrate security into every solution they deliver. This framework provides tailored security considerations and checklists for each program area.

D. Risk Management Framework (RMF) for Cybersecurity

In response to increasing risks against critical infrastructure and information technology systems, the US government enacted the Federal Information Security Management Act of 2002 which established minimum information security requirements for federal information systems, and charged the National Institute of Standards and Technology (NIST) with developing security standards and guidelines to address these growing risks [35]. In response to this requirement, NIST created the Risk Management Framework (RMF) which provided a structured yet flexible process for applying these standards and guidelines [36]. Accordingly, RMF is the mandated approach for addressing cybersecurity in the U.S. DoD [11]. In general, this approach applies a prescriptive risk-based methodology to cybersecurity with the goal of identifying, mitigating, and eliminating system vulnerabilities to protect systems from unauthorized access, use, disclosure, disruption, modification, or destruction. Within the United States Air Force, the Air Force Life Cycle Management Center is tasked with conducting RMF for legacy weapon systems (designated as the Platform IT (PIT) systems) [37]. This PIT assessment and authorization process consists of six steps described in the next paragraph [13].

First, the team must categorize the PIT system according to the information displayed, processed, stored, and transmitted along with the classification of the information and associated technologies. Second, security controls are selected (or assigned) based on the impact resulting from the loss of said information (i.e., criticality analysis) [12]. The third step is implementing said controls with consideration for cybersecurity requirements across the entire system development life cycle—although security controls have been historically applied to IT systems, many have been tailored for PIT systems with prescribed overlays [37]. The fourth step is key to the RMF process and assesses the effectiveness of applied security controls through threat mapping and vulnerability analysis. On a related note, much of the security work conducted today is exclusively focused on this step. Based on the identified vulnerabilities, the fifth step is to produce a risk assessment and mitigation plan, which is then briefed to the Authorization Official for authorization. The sixth step of the RMF process is continuous monitoring of the system with respect to cybersecurity. As the system and threat environment evolve over time, security control effectiveness needs to be continuously assessed while keeping in mind future changes and cybersecurity impact.

The RMF is the mostly widely implement approach of those surveyed as it is mandatory for DoD information systems to receive an authorization to operate. While this approach has mitigated vulnerabilities, many site its perceived difficulty, steep learning curve, and IT centric focus as currently implemented as critiques in its utility for complex cyber-physical systems.

E. Avionics Cyberspace Vulnerability Assessment and Mitigation (ACVAM) and Cyber Hardening Efforts

The United States Air Force Research Laboratory (AFRL), in conjunction with the Air Force Institute of Technology's Center for Cyberspace Research, developed an Avionics Cyberspace Vulnerability Assessment and Mitigation (ACVAM) Workshop [38]. This weapon-system-specific workshop teaches a thorough analysis approach by systematically identifying and assessing all external inputs and communications paths to and from a weapon system (i.e., an exhaustive boundary analysis of the system's architecture). The major activities include gathering information, identifying and analyzing access points, finding and analyzing susceptibilities,

anticipating attacks, and applying and recommending mitigations and protections. The ACVAM approach requires extensive Subject Matter Expert (SME) involvement, access to design documents, and detailed operator insight to discover susceptibilities and determine appropriate mitigations to increase mission assurance by eliminating or reducing vulnerability to cyberattacks [39].

Additionally, AFRL is developing more specific cyber hardening tools and resiliency instructions [40]. While specific details are not publicly available, the cyber hardening approach was recently briefed to the defense community at large [39]. In general, this approach describes avionics cyber hardening and resiliency concepts and suggests ways to protect avionics and related systems from cyber-attack. Moreover, this approach encourages engineers to 'think avionics cyber' using three tenets of cyber protection: focus on what's critical; restrict access to the critical; and detect, react, and adapt [41]. These approaches provide a robust analysis but require technically savvy domain experts to execute, which restricts its utility for a larger group of complex systems.

F. Attack Path Analysis via Automotive Example

Historically, attack path analysis has served the security community well [42]. In a great example from the automotive domain, Checkoway *et al.* provide a practical attack path analysis and comprehensive discussion which solidifies the importance of threat modeling as a cybersecurity architectural analysis technique [8]. While this specific example is automobile centric, many similarities are shared between cyber-physical systems. More specifically, the work details a four-step method of analyses. First, threat model characterization is accomplished through identification of external attack vectors and attack surfaces. Second, vulnerability analysis addresses the accessibility, criticality, and exploitability of potential vulnerabilities. Third, a threat assessment attempts to gauge the attacker's motivation by answering the question of what utility a given attack path has for the attacker. Finally, the approach suggests mitigation actions by synthesizing similarities among vulnerabilities to provide pragmatic recommendations for enhancing the system's cybersecurity.

G. System Theory Process Analysis for Security (STPA-Sec)

In recent work, MIT's System Theory Process Analysis (STPA) approach for safety was extended to focus on security-related concerns, known as STPA-Sec [43]. The goal of this approach is to ensure mission-critical functions are maintained in the face of disruption(s). Starting from a strategic viewpoint, system developers and users can proactively shape the operational environment by controlling specified mission critical system risks. This top-down approach elevates the security problem from guarding the system (or network) against all potential attack paths to a higher-level problem of assuring the system's critical functions. The STPA-Sec steps include: identifying unacceptable losses, identifying system hazards (vulnerabilities), drawing the system functional control structure, and identifying unsafe or insecure CAs [43]. This method has been embraced by defense and commercial industries with several favorable case studies [44].

H. Functional Mission Analysis for Cyber (FMA-C)

The DoD has adopted Functional Mission Analysis for Cyber (FMA-C) as an approach to secure operational computer networks [45]. FMA-C is being taught to thousands of airmen in an effort to assure critical cyber systems and reduce vulnerabilities. While the structure and content of FMA-C is similar to STPA-Sec, its application is tailored to As-Is Information Technology infrastructures. In practice, USAF Mission Defense Teams apply FMA-C to fielded cyber systems to identify mission critical vulnerabilities. It has proved to be a useful tool for understanding and mitigating risks in traditional cyber (i.e. ICT) domains.

I. Other Notable Methodologies

As previously noted, other methodologies and frameworks for systems-level security analysis are sure to exist which are not covered in this work. A few notable works focused on mission assurance are available here [46], [47], [48] and on software here [49], [50].

IV. Desirable Characteristics for Conducting Cybersecurity Architectural Analysis

This section identifies desirable characteristics for cybersecurity architectural analysis and cross-examines the approaches discussed in Section III.

A. Cybersecurity Architectural Analysis Characteristics

The first characteristic is definitional in nature and classifies approaches as either top-down or bottom-up. Those defined as top-down start with analysis at the function level with identification and examination of critical missions and/or capabilities—sometimes operations depending on how the approach is being applied. As is typical of architecting for new systems (and sometimes upgrades), higher-level functional analysis leads to further functional decomposition and allocation to a more specific form (e.g., lower subsystems, elements, or components). These approaches lend themselves to the identification of stakeholder security needs, early trade-offs, thorough security requirements definition, and integration of more holistic security solutions [27].

Conversely, bottom-up approaches begin with the form in mind (i.e., the physical or technological solution) and often focus on perimeter security through boundary analysis [51]. While this approach successfully identifies vulnerabilities in networked components, it is often less useful for protecting systems from intelligent adversaries. For example, Bayuk and Horowitz [52] surmise that perimeter defense tactics are largely ineffective, and conclude that a top-down, risk-based systems engineering approach to system security should be used instead.

The next key characteristic is whether the approach should be driven by threats or vulnerabilities. Prior research suggests that the foundation for improving system security starts with an analysis of potential threats, which leads to more appropriate security requirements for implementation [42]. This is intuitive; without first understanding the adversary—system-specific threats (and their rapid agility)—it is difficult, or impossible, to defend against them. Understanding and modeling the threat becomes a critical prerequisite for generating and developing secure systems [53]. Once the model has been developed and

validated, vulnerability analysis is the logical follow-on. With the threats understood, the system architecture can be analyzed for vulnerable access points through techniques such as attack path analysis and/or red teaming.

While acknowledging the rapidly changing nature of threats, the exercise of red teaming and brainstorming potential attack paths is a helpful critical thinking exercise for ensuring sound cybersecurity practices. Moreover, threat modeling and vulnerability analysis typically form the foundation for cybersecurity architectural analysis. While threat modeling alone does not ensure cybersecurity, rigorous threat modeling and vulnerability analysis are helpful for ensuring the security of realized systems. However, more focus should be applied to providing security solutions and not just focused on identifying problems.

In today's highly contested cyberspace environment, documentation-based engineering is largely ineffective against dynamic adversaries [42]. Developing a successful response to a dynamic adversary necessitates the tools and methods used to develop countermeasures be, in kind, dynamic. In response to these complexities, Model-Based Systems Engineering (MBSE) offers an integrated modeling approach capable of mapping desired capabilities to functions (and even components), as well as providing traceability and fit-for-purpose views to enable more effective decision-making [54]. In a recent effort, Apvrille and Roudier proposed SysML-Sec, an injection of security considerations into SysML in an effort to foster integration between system designers and security experts [55]. SysML-Sec and more generally MBSE approaches enable security-focused computer simulations of a potential system architecture. These executable architectures provide tremendous value by providing insights into early design trade-off analysis [56]. While MBSE requires significant initial investment in tools and training, it significantly increases the depth of possible architectural analysis especially in executable architectures.

B. Assesment of Architectural Analysis Approaches

Table I provides a consolidated assessment (i.e., a mapping) of the proposed architectural analysis characteristics to the surveyed approaches from Section III. This mapping seeks to provide a consolidated reference for differentiating approaches to inform the user and assist in selecting an appropriate cybersecurity architectural approach which meets the stakeholders' needs. Consideration is given to each approaches' usability, scalability, and tool availability. The ideal approach will also easily facilitate modeling and simulation studies to perform early design feasibility studies and support trade-off analysis (i.e., MBSE).

In general, bottom-up approaches are relatively systematic; however, historically they have not produced secure systems and tend to scale poorly. Top-down approaches have the benefit of being more scalable, but they often require a high level of tool proficiency to effectively model (thus, the potential of MBSE to systems security is largely missed). While vulnerability analysis is inherent in every approach, a threat-based approach is less so. This aspect is important because effectively safeguarding unprecedented, complex systems requires more than a good architectural tool or technique – a holistic engineering approach that embraces all aspects of security (e.g., people, processes, policy, technology, feasibility, cost, etc.) is required [57], [58].

TABLE I: ARCHITECTURAL APPROACHES TO CHARACTERISTICS MAPPING.

	Top Down	Bottom Up	Threat Driven	Vul. Based	MBSE Integrated	MBSE Executable	Tool Based
DoDAF + Richards	X[1]			X	X	X[4]	X
CRAF	X[1]		X	X	X	X	X
UAF Security	X			X	X	X[4]	X
ACVAM		X	X	X			
STPA-Sec	X[2]			X			
RMF		X[5]	X	X	X[3]		

1. Promotes a top-down approach after mission functions are identified (i.e., does not include mission thread analysis).
2. Approach begins at a higher level than other approaches examined (i.e., includes mission thread analysis) and includes lower level analysis.
3. Suggests using MBSE, but not required and often not considered.
4. Would require pairing with additional modeling & simulation plugin.
5. RMF is intended to be a top-down approach but is often applied bottom-up using security control compliance based on system type.

V. CONCLUSIONS AND FUTURE WORK

The practice of architectural analysis is not new; however, in the context of complex cyber-physical systems, the role of architectural analysis with respect to cybersecurity is not well understood. Moreover, given cybersecurity's widespread interest, it was surprising to find a general lack of understanding or consistency regarding what it means to conduct architectural analysis for cybersecurity while surveying the literature. Thus, this work briefly surveys key architectural analysis concepts and provides a timely and widely applicable working definition of "cybersecurity architectural analysis" for the community to consider. Next, a survey of several cybersecurity architectural analysis approaches from industry and government is provided, along with an assessment of their applicability for complex cyber-physical systems according to several desirable characteristics. These results help practitioners and researchers understand how to achieve more effective cybersecurity architectural analysis efforts in order to develop secure systems according to stakeholders needs.

While there are several promising cybersecurity architectural approaches, each with unique aspects to be more fully explored, standardized approaches such as UAF paired with MBSE hold promise and have a wider acceptance than some alternatives. In the near term, the authors have chosen to explore STPA-Sec to more fully understand its utility as a relatively simple architectural analysis approach to assist in the development of safe, secure, and resiliency military systems. Specifically, the authors are executing a detailed case study for a next-generation aircraft refueling system. This case study focuses on understanding the utility of the STPA-Sec approach for eliciting cybersecurity and resiliency requirements when developing complex military systems (i.e., unprecedented cyber-physical systems of systems). Ultimately, continued research in this field will enable more effective and efficient cybersecurity architectural analysis for complex systems regardless of application domain.

DISCLAIMER

The views expressed in this paper are those of the authors and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the United States Government.

ACKNOWLEDGMENTS

This work was supported by the U. S. Air Force, Air Force Institute of Technology, Cyberspace Center for Research, Wright-Patterson Air Force Base, Ohio, United States of America.

REFERENCES

[1] White House, "Remarks by the President on Securing our Nation's Cyber Infrastructure," White House Press, 2009.

[2] P. Singer and A. Friedman, Cybersecurity and Cyberwar, New York: Oxford, 2014.

[3] Y. Liu, Y. Peng, B. Wang, S. Yao and Z. Liu, "Review on cyber-physical systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 1, pp. 27-40, 2017.

[4] R. Saracco, "Guess What Requires 150 Million lines of Code," EIT Digital, 13 Jan 2016. [Online]. Available: https://www.eitdigital.eu/news-events/blog/article/guess-what-requires-150-million-lines-of-code/. [Accessed Feb 2017].

[5] R. Charette, "IEEE Spectrum: This Car Runs on Code," 1 February 2009. [Online]. Available: http://spectrum.ieee.org/transportation/systems/this-car-runs-on-code. [Accessed 1 June 2017].

[6] A. Greenberg, "Wired," Wired Magazine, 21 July 2015. [Online]. Available: https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/. [Accessed 25 April 2017].

[7] E. Perez, "CNN," CNN, 18 May 2015. [Online]. Available: http://www.cnn.com/2015/05/17/us/fbi-hacker-flight-computer-systems/. [Accessed 25 April 2017].

[8] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham and S. Savage, "Comprehensive Experimental Analyses of Automotive Attack Surfaces," USENIX Security Symposium, 2011.

[9] United States Congress, "Nation Defense Authorization Act 2016 Section 1647," 25 November 2015. [Online]. Available: https://www.congress.gov/114/plaws/publ92/PLAW-114publ92.pdf. [Accessed 1 June 2017].

[10] Department Of Defense, "DoDI 8500.01 Cybersecurity," 2014.

[11] Department of Defense, "DoDI 8510.01 Risk Management Framework (RMF) for DoD Information Technology (IT)," 2014.

[12] Department of Defense, "Defense Acquisition Guidebook Chapter 9 Program Protection," 5 April 2017. [Online]. Available: https://www.dau.mil/tools/dag/Pages/DAG-Page-Viewer.aspx?source=https://www.dau.mil/guidebooks/Shared%20Documents%20HTML/Chapter%209%20Program%20Protection.aspx. [Accessed 1 June 2017].

[13] Department Of Defense, "DoD Program Manager's Guidebook for Integrating the Cybersecurity Risk Management Framework (RMF) into the System Acquisition Lifecycle," 30 October 2015. [Online]. Available: https://acc.dau.mil/adl/en-US/721696/file/81323/Cybersecurity%20Guidebook%20v1.10%20signed.pdf. [Accessed 1 June 2017].

[14] J. A. Zachman, "A Framework for Information Systems Architecture," *IBM Systems Journal 26*, vol. 26, no. 3, pp. 276-292, 1987.

[15] J. F. Sowa and J. A. Zachman, "Extending and Formalizing the Framework for Information Systems Architecture," *IBM Systems Journal*, vol. 31, no. 3, pp. 590-616, 1992.

[16] J. Zachman, "The Zachman Framework Evolution," 1 April 2011. [Online]. Available: https://www.zachman.com/ea-articles-reference/54-the-zachman-framework-evolution. [Accessed 11 May 2017].

[17] A. Tang, J. Han and P. Chen, "A Comparative Analysis of Architecture Frameworks," *IEEE Computer Society: Proceedings of the 11th Asia-Pacific Software Engineering Conference*, vol. 4, no. 1530-1362, pp. 1-8, 2004.

[18] C. Paulsen, "Cybersecuring Small Businesses," *IEEE Computer*, vol. 49, no. 8, pp. 92-97, 2016.

[19] Department Of Defense, "DoDI 8500.01 Cybersecurity," 2014.

[20] G. Hurlburt, "Good Enough Security: The Best We'll Ever Have," *IEEE Computer*, pp. 98-101, 2016.

[21] M. W. Maier and E. Rechtin, The Art of Systems Architecting, CRC Press, 2009.

[22] ISO/IEC/IEEE 42010, "Systems and Software Engineering: Architecture Description," 2011.

[23] Department of Defense, "Department of Defense Architecture Framework," 2010.

[24] J. Zachman, "Wikipedia," 5 May 2010. [Online]. [Accessed 10 May 2017].

[25] R. N. Taylor, N. Medvidovic and E. Dashofy, Software architecture: foundations, theory, and practice., Wiley Publishing, 2009.

[26] E. Crawley, B. Cameron and D. Selva, System Architecture, Hoboken: Pearson, 2016.

[27] R. Ross, M. McEvilley and J. Oren, "NIST Special Publication 800-160: Systems Security Engineering," National Institute of Standards and Technology, Washington DC, 2016.

[28] J. E. Richards, "Using the Department of Defense Architecture Framework to Develop Security Requirements," 2014. [Online]. Available: sans.org. [Accessed Feb 2017].

[29] P. Singer and A. Friedman, Cybersecurity and Cyberwar, New York: Oxford, 2014.

[30] L. Ertaul and J. Hao, "Enterprise Security Planning with Department of Defense Architecture Framework (DODAF)".

[31] Object Management Group, "Unified Architecture Framework Profile," OMG, 2016.

[32] T. Hambrick and M. Tolbert, "Unified Architecture Framework Profile-Systems Engineering Method for Security Architectures-NMWS 17," 21 May 2017. [Online]. Available: https://nmws2017.com/agenda. [Accessed 21 May 2017].

[33] S. Hassell, "Using DoDAF and Metrics for Evaluation of the Resilience of Systems, Systems of System, and Networks Against Cyber Threats," *INCOSE INSIGHT*, vol. 18, no. 1, pp. 26-28, 2015.

[34] P. Nejib and D. Beyer, "Secure Engineering Assurance Model," 11 June 2014. [Online]. Available: http://www.incose.org/docs/default-source/enchantment/140611beyernajib-lockeedseam.pdf?sfvrsn=2. [Accessed 8 June 2017].

[35] "E-Government Act of 2002. Pub. L. No. 107-347, 116 Stat. 2899," 17 12 2002. [Online]. Available: https://www.congress.gov/107/plaws/publ347/PLAW-107publ347.pdf. [Accessed 30 01 2018].

[36] R. Ross, "Managing Enterprise Security Risk with NIST Standards," *Computer*, pp. 88-91, 20 08 2007.

[37] AFLCMC/EZAS, "Aircraft Cybersecurity Risk Management Framework," 19 May 2014. [Online]. Available: http://www.mys5.org/Proceedings/2014/Day_2_S5_2014/2014-S5-Day2-12_VanNorman.pdf. [Accessed May 2017].

[38] Air Force Institute of Technology Center for Cyberspace Research, "Avionics Cyberspace Vulnerability Assessment and Mitigation (ACVAM) Workshop," Air Force Research Laboratory, 1 December 2015. [Online]. Available: https://www.afit.edu/ccr/page.cfm?page=1184&tabname=Tab2A. [Accessed 1 May 2017].

[39] Air Force Research Laboratory, "Air Force Research Lab Avionics Vulnerability Assessment and Mitigation Efforts," in *Ohio Cyber Dialogue with Industry*, Dayton, 2017.

[40] K. Osborn, "BattleSpace IT - Air Force: An F-16 could be vulnerable to cyber attack," Defense Systems, 18 October 2016. [Online]. Available: https://defensesystems.com/articles/2016/10/18/cyber.aspx. [Accessed May 2017].

[41] J. Hughes and G. Cybenko, "Three tenets for secure cyber-physical system design and assessment," in *SPIE Defense+ Security*, 2014.

[42] J. Cleland-Huang, T. Denning, T. Kohno, F. Shull and S. Weber, "Keeping Ahead of Our Adversaries," *IEEE Software*, vol. 33, no. 3, pp. 24-28, 2016.

[43] W. Young and N. G. Leveson, "An Integrated Approach to Safety and Security Based on Systems Theory," *Communications of the ACM*, vol. 57, no. 2, pp. 31-35, 2014.

[44] Massachusetts Institute of Technology, "MIT Partnership for a Systems Approach to Safety," 27 March 2017. [Online]. Available: http://psas.scripts.mit.edu/home/stamp-workshop-2017/. [Accessed 8 June 2017].

[45] Air Force Cyber College, "Top-down Purpose-based Cybersecurity," 2015. [Online]. Available: https://www.sans.org/summit-archives/file/summit-archive-1492176717.pdf. [Accessed 01 January 2018].

[46] G. Hastings, L. Montella and J. Watters, "MITRE Crown Jewels Analysis," The MITRE Corporation, 2009.

[47] H. G. Goldman, "Building secure, resilient architectures for cyber mission assurance," The MITRE Corporation, 2010.

[48] C. Alberts and A. Dorofee, "Mission Assurance Analysis Protocol (MAAP): Assessing Risk in Complex Environments," 2005.

[49] C. Alberts, C. Woody and A. Dorofee, "Introduction to the Security Engineering Risk Analysis (SERA) Framework," 2014.

[50] Software Engineering Institute, "Security Engineering Risk Analysis (SERA)," CERT- Carnegie Mellon University, 1 November 2015. [Online]. Available: https://www.cert.org/cybersecurity-engineering/research/security-engineering-risk-analysis.cfm?. [Accessed 1 May 2017].

[51] R. Anderson, Security Engineering, 2nd ed., Indianapolis, Indiana: Wiley Publishing, Inc, 2008.

[52] J. Bayuk and B. Horowitz, "An Architectural Systems Engineering Methodology for Addressing Cyber Security," *Systems Engineering*, vol. 14, no. 3, pp. 294-304, 2011.

[53] A. Shostack, Threat modeling: Designing for security, John Wiley & Sons, 2014.

[54] A. Ramos, J. Ferreira and J. Barceló, "Model-based systems engineering: An emerging approach for modern systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42 , no. 1, pp. 101-111, 2012.

[55] L. Apvrille and Y. Roudier, "Towards the Model-Driven Engineering of Secure yet Safe Embedded Systems," *Electronic Proceedings in Theoretical Computer Science*, vol. 148, no. 2, pp. 15-30, 2014.

[56] J. A. Estefan, "Survey of Model-Based Systems Engineering (MBSE) Methodologies," International Counsel On Systems Engineering (INCOSE), 2008.

[57] J. Eloff and M. Eloff, "Information Security Architecture," *Computer Fraud and Security*, vol. 11, pp. 10-16, 2005.

[58] T. Patterson, "Holistic Security: Why Doing More Can Cost You Less and Lower Your Risk," *Computer Fraud and Security*, pp. 13-15, 2003.

III. Methodology

Description:

Chapter III is a self-contained conference paper that details the methodology, STPA-Sec. It presents a tailored approach for STPA-Sec as a conceptual analysis of a complex cyber physical system. A simple example for a space system is presented. The background of this research introduced in Chapter II is re-presented for context.

This work answers research question 4 by presenting a tailored approach for STPA-Sec exemplified through an abbreviated space system example.

Publication Details:

Title: A Systems Security Analysis Approach for Understanding, Defining, and Specifying Security Requirements for Complex Cyber-Physical Systems

Publication: Submitted to IEEE International Conference on Cyber Security and Protection of Digital Services

A Systems Security Analysis Approach for Understanding, Defining, and Specifying Security Requirements for Complex Cyber-Physical Systems

[1]Martin "Trae" Span, [1*]Logan O. Mailloux, [1]Paul M. Beach, [1]Robert F. Mills and [2]William "Bill" Young
[1]Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio
[2]Syracuse University, Syracuse, New York
[*]Correspondence: logan.mailloux@us.af.mil

***Abstract*—In today's highly interconnected and technology reliant environment, systems security is rapidly growing in importance. Moreover, security is no longer limited to traditional computer systems and IT networks, as a number of highly publicized attacks have occurred against complex cyber-physical systems such as automobiles and airplanes. While systems security analysis approaches are critical to improving the security of these systems-of-systems, such approaches are often poorly understood and applied in ad hoc fashion. This work addresses such gaps by detailing a relatively straight forward security analysis approach for understanding, defining, and specifying security requirements. First, a readily understandable description of key architectural analysis concepts and definitions is provided along with an assessment of their applicability to complex cyber-physical systems. Next, a variant of the System Theory Process Analysis approach for Security (STPA-Sec) is detailed in three phases which support development of functional-level security requirements, applicable design-level criteria, and architectural-level security specifications in alignment with the stated systems and software engineering processes in ISO/IEC/IEEE 15288 and the recently released NIST SP 800-160. This work is important for advancing the science of systems security engineering by providing a viable systems security analysis approach for eliciting and capturing traceable security, safety, and resiliency requirements and criteria that can be designed-for, built-to, and formally verified.**

Keywords—security; systems security analysis; system architecture; systems security engineering

I. INTRODUCTION

The cybersecurity threat is one of the most serious challenges in the 21st century. Over the past decade, attacks against Information and Communication Technologies (ICT) have grown considerably in frequency and complexity, and it is now commonplace to hear of widespread attacks against personal computers, webservers and services, Internet of Things (IoT) devices, and even critical government databases. Moreover, the security of cyber-physical devices is becoming increasingly important as these devices take on central roles in nearly every aspect of modern life. Previously, cyber-physical systems such as automobiles and airplanes were complicated, but not interactively complex. Security (and safety) is now an emergent properly of cyber-physical systems, where their software and real-time networks require previously isolated components to continuously interact [1]. For example, the 2017 Ford F-150, a fairly common vehicle in the United States, has over 150 million lines of code distributed across dozens of computing devices with software providing its essential functionality [2], [3]. Moreover, adversaries are challenging traditional assumptions that cyber-physical systems are secure due to their relative isolation and uniqueness with recent examples including the widely publicized hacking demonstration against a Jeep Cherokee [4], claims of hacking a commercial airliner [5], and comprehensive reports of vehicular attack paths [6].

In light of these growing threats, it is critical for security professionals to have appropriate tools and techniques for performing systems security engineering and analysis. For example, the United States Department of Defense (U.S. DoD) which historically values systems security, has made several recent changes to expand traditional IT-focused security approaches and mandate security assessments for cyber-physical weapon systems [7], [8], [9], [10], [11], [12]. These policies dictate that acquisition programs integrate security efforts into existing systems engineering processes, and work to ensure security considerations hold equal footing with other requirements and design trade-offs at major program reviews. However, it is not easy to understand what constitutes a "secure" system, nor how to specify effective security criteria as stated in Good's 1986 challenge essay [13]:

> The first thing we need in this process is the ability to state computer security requirements clearly and precisely… so that a competent professional can study it for a reasonably short amount of time and, say, "Oh, yes, I agree. If you build that particular system to that particular requirement, it's secure enough for that particular purpose."

These security requirements are critically important because they establish the foundation upon which analysis and evidences are used to "judge whether a system is 'secure'" [13].

For cyber-physical systems (i.e., highly complex systems-of-systems), architectural analysis is often viewed as a critical enabler for systems security analysis; however, these approaches are typically focused on lower-level security decisions. Moreover, they are often poorly understood and applied in ad hoc fashion. To addresses these gaps this work suggests a relatively straightforward systems security analysis approach for understanding, defining, and specifying security requirements for complex cyber-physical systems. First, a brief discussion of architectural analysis concepts is provided in Section II along with a working definition of "cybersecurity architectural analysis" since none exists in the literature. Section III surveys several systems-oriented security architecture analysis approaches from government and industry, while Section IV assesses how these approaches can be used to more effectively support a holistic systems security analysis approach

for a given System of Interest (SoI). Based on these results, a variant of the System Theory Process Analysis approach for Security (STPA-Sec) is detailed in Section V along with an example in Section VI. Of note, our suggested conceptual-level focused systems security analysis approach is tailorable and comprises three phases of increasing detail which result in security requirements, architectural security considerations, and design-level security criteria. Of great importance, these three phases align with the established systems and software engineering processes in ISO/IEC/IEEE 15288 and the recently released NIST Special Publication 800-160, *Systems Security Engineering* [14], [15].

Lastly, Section VII summarizes key findings and identifies promising follow-on research areas for increasing the rigorous application of systems security engineering and accompanying analysis approaches for developing secure, safe, and resilient systems – those which can be more easily understood, defined, specified, implemented, and verified. Because of the Authors' affiliation with the U.S. DoD, as well as, the DoD's unparalleled investment in systems security, much of the work presented in this paper has an obvious U.S. DoD perspective; however, this should not hinder the contribution of this work to other domains as our intention is to promote and advance the science of security with respect to understanding the effective application of systems security engineering processes, activities, and tasks regardless of the SoI or application domain.

II. Foundational Concepts and Definitions

This section provides historical context for discussing system-level architectural analysis and, more formally, discusses key definitions for understanding architectural analysis for security (i.e., cybersecurity architectural analysis). Much of the seminal work in the field of architectural analysis was accomplished by Zachman, who proposed the first system architecture – a logical construct for capturing, presenting, and integrating the complexities of modern information systems *[16]*. Akin to the multitude of perspectives intrinsic to construction blueprints (e.g., structural, plumbing, electrical, etc.), Zachman argued that system architectures should be composed from many perspectives with varying levels of detail. Furthermore, he insisted that these perspectives (or views) be synchronized across the system, forming one integrated architecture.

Sowa expanded Zachman's work to form the Information Systems Architecture (ISA) framework shown in Fig. 1 [17]. Across the ISA framework the interrogatives (what, how, where, who, when, and why) are explored with six perspectives (shown as rows) as a means of expressing relationships at varying levels of detail to guide complex system development [18]. Thus, the ISA establishes a baseline to enable effective decision making for new system developments by comparing and elaborating on desired capabilities, requirements, components, and functions in a single well-integrated, enterprise-level model (note, not all 36 views are required; thus, the ISA is also tailorable). Since its inception, the ISA – commonly known as "the Zachman Framework" – has been a popular choice for system architects; it has been widely used for decades, while several other system-level frameworks have incorporated or adopted its tenets [19].

A. Key Definitions

As a prerequisite for understanding and assessing the utilization of architectural analysis approaches for security decision making, a brief discussion of key definitions is offered. First, please note that we use the term "cybersecurity" in this work as it now seems to be the prevailing pseudonym for "security" regardless of intentionality or context (see sidebar in [20]). While several competing definitions exist, within the U.S. DoD, cybersecurity is formally defined as [21]:

> The prevention of damage to, protection of, and restoration of electronic systems to ensure its availability, integrity, authentication, confidentiality, and nonrepudiation.

While comprehensive, the U.S. DoD's definition for the purpose of analyzing cyber-physical SoI is somewhat hindered because it is saturated with domain-specific IT jargon, where: availability ensures the system is usable as anticipated; integrity is the protection from unauthorized modification; confidentiality is keeping data private; authentication is a validation of the claimed identity; and non-repudiation is the ability to prove that an action has taken place [22]. A more practical systems-oriented description of security might simply seek to prevent hazardous functionality which leads to potentially unsafe system states with unacceptable losses [23].

The next key term to define is "architecture" where we interpret "architecture" synonymously with "system architecture" and variations thereof. As a well-known standard ISO/IEC/IEEE 42010 provides a rather straightforward definition for system architecture [24]:

> The fundamental organization of a system, embodied in its components, their relationship to each other and to the environment, and the principles governing its design and evolution.

While this definition is readily understandable, it is somewhat focused on the SoI's physical realization and does not capture the desired holistic nature necessary for performing systems security engineering and analysis [25]. Another classically understood definition is provided by Maier and Rechtin [26]:

> Structure in terms of components, connections, and constraints of a product, process, or element.

This definition offers a more holistic view of the SoI to include both technological and non-technological aspects. It also provides a means for viewing the system from different perspectives with multiple levels of abstraction as purported by Zachman. With a full life cycle focused perspective, the U.S. DoD provides a progressive definition for consideration [27]:

> A set of abstractions (or models) that simplify and communicate complex structures, processes, rules, and constraints to improve understanding and implementation.

Although somewhat conceptual, this definition alerts the reader to the intrinsic value offered by architectural efforts in that they serve to improve communication amongst multiple team members and increase understanding of complex systems with the goal of improving the SoI's implementation (i.e., its physical realization).

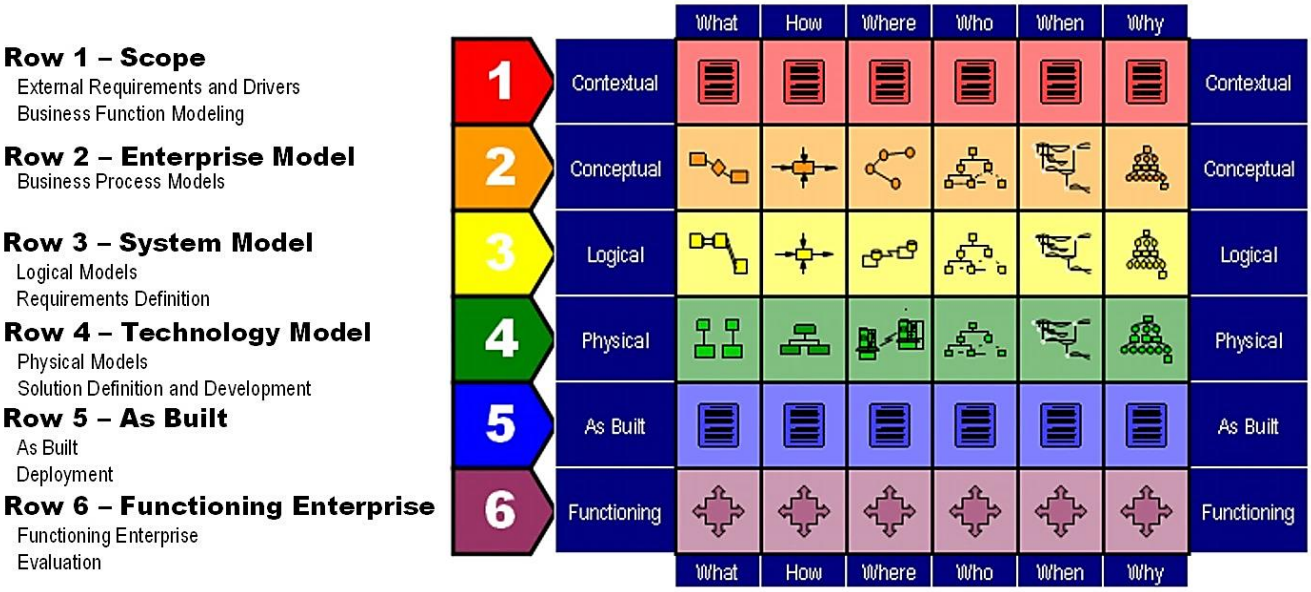


FIG 1. THE ZACHMAN FRAMEWORK FOR ENTERPRISE ARCHITECTURE [27].

Lastly, the task of identifying a formal definition of "analysis" within the context of a systems architecture proved more difficult than anticipated. This is because system architectures often center on an integrated model of entities and the relationships between the entities to facilitate order and understandability, not analytical capability. One such definition reads as follows [28]:

> Architectural analysis is the activity of discovering important system properties using conceptual and physical models of the system of interest.

However, an architecture's purpose is to increase understanding and facilitate trade-offs focused on evaluating and comparing alternative designs and attributes (e.g., security, performance, reliability, etc.) [15], [19]. This two-fold purpose is acutely captured by Crawley *et al.* [29]:

> Architectural analysis focuses on understanding both the architecture's function and form for the purpose of supporting decision making.

This definition reflects both the architecture's inherent ability to manage complexity and enable analysis in support of more effective decision making. Ultimately, architectural analysis requires consideration of various missions, essential functions, components, and desired system attributes which help to clarify and refine stakeholder needs and system requirements. Thus, architectural analysis should identify trade-off points among competing options and enable more effective communication between various stakeholders (e.g., customers, developers, operators, maintainers).

B. Proposed Cybersecurity Architectural Analysis Definition

Architectural analysis provides a means for understanding cyber (i.e., ICT) dependencies within the functions and form of the desired system. This type of structured analysis brings an otherwise unmanageable amount of information under control in support of well-informed system security engineering decisions [30]. More holistically, architectural analysis enables pragmatic risk management by providing context and functional mapping to the various physical elements of the SoI. Thus, cybersecurity architectural analysis facilitates the application of appropriate security mitigations where needed with rigorous justification.

After considering the seminal definitions presented above (and others not discussed) and working to understand the architectural analysis approaches for cybersecurity discussed in Section III, we propose a working definition of cybersecurity architectural analysis for consideration:

> The activity of discovering and evaluating the function and form of a desired system to facilitate cybersecurity decisions.

This definition is easily understandable and addresses both the conceptual-level analysis associated with new system development (i.e., studying the desired functionality) and lower-level considerations for existing systems (i.e., comparing the desired functionality to the existing form).

III. CYBERSECURITY ARCHITECTURAL ANALYSIS APPROACHES

In this section, we survey architectural analysis approaches for complex system cybersecurity. Within the U.S. DoD (and its major defense contractors), several approaches (i.e., methods, processes, and tools) have been developed to secure and assess the cybersecurity of complex systems and systems-of-systems. This survey is based on publicly available literature and presentations that focus specifically on architectural analysis for weapon systems. The predecessor for many cybersecurity architectural analysis approaches is compliance-based Information Assurance (IA), which focuses almost exclusively on applying security controls to computer networks and IT

systems. For complex systems this approach is inadequate as demonstrated by several high profile security breaches [31].

A. Department of Defense Architectural Framework (DoDAF)

The integrated architecture currently in use by the U.S. DoD is the DoD Architecture Framework (DoDAF). Its purpose is to manage complexity to enable key decisions through organized information sharing [27]. However, in DoDAF, like many other architecture frameworks, security (or cybersecurity) is not specifically addressed [32]. James Richards, in his work *Using the Department of Defense Architecture Framework to Develop Security Requirements* [30], proposes a methodology for using DoDAF to derive security requirements. He outlines a process of first building an architectural model of the enterprise, focusing on a core set of views including the OV-5b operational activity model, the DIV-2 logical data model, and the OV-3 operational resource flow matrix. These critical views are used to model security-relevant processes, data, business rules, and communications. Next, he suggests comparing views for compliance and then assessing and refining the architecture. The overall purpose of Richards' approach is to use DoDAF to expose or derive security requirements [30].

B. Unified Architecture Framework (UAF)

In contrast to the U.S. DoD unique solution DoDAF, industry has developed the Unified Architecture Framework (UAF) [33]. Based on industry need, the UAF includes a formal security domain amongst the more common architectural views. The UAF security domain includes views for security taxonomy, structure, connectivity, processes, constraints, and traceability. More specifically, it uses SysML class diagrams to identify data types and map them to protections and security controls. As an integrated architecture, it allows security-relevant elements to be mapped to system resources and operations. UAF also capitalizes on the success of MBSE efforts to depict and analyze the security properties of a SoI via an executable architecture. Note, UAF is in the final stages of development, so its utility has yet to be fully realized; however, the proposed security views appear to be useful for conducting cybersecurity architectural analysis [34].

C. Publically Avilable Industry Efforts

Major defense contractors often use custom architectural analysis approaches to design and evaluate their system architectures with respect to cybersecurity. Although it is likely that most large U.S. DoD contractors are working solutions in this area; at the time of this survey, the authors were only exposed to efforts from Raytheon, Northrop Grumman, and Lockheed Martin. Note, Raytheon's Cyber Resiliency Architecture Framework (CRAF) was the only approach with a detailed open source publication available. Limited information is available on Northrop and Lockheed's approaches.

Raytheon developed CRAF using a DoDAF reference architecture with extensions for specific cyber resilience mappings and metrics [35]. The goal of CRAF is to assess and identify gaps in cyber resiliency by mapping systems, subsystems, and components against prioritized capabilities to identify resilience requirements for important mission scenarios.

Using failure modes and effects analysis, Northrop Grumman created a risk-based assessment methodology using an integrated architecture modeled in the new UAF to identify cyber risks for their systems [34]. This approach is still under development and is one of the first systems security efforts based on the upcoming UAF standard security views from the Object Management Group (OMG).

Lockheed Martin has created a custom solution titled the Secure Engineering Assurance Model (SEAM) [36]. SEAM is a tailored systems security engineering approach to integrate security into every solution they deliver. This framework provides tailored security considerations and checklists for each program area.

D. Risk Management Framework (RMF) for Cybersecurity

In response to increasing risks against critical infrastructure and information technology systems, the US government enacted the Federal Information Security Management Act of 2002 which established minimum information security requirements for federal information systems, and charged the National Institute of Standards and Technology (NIST) with developing security standards and guidelines to address these growing risks [37]. In response to this requirement, NIST created the Risk Management Framework (RMF) which provided a structured yet flexible process for applying these standards and guidelines [38]. Accordingly, RMF is the mandated approach for addressing cybersecurity in the U.S. DoD [10]. In general, this approach applies a prescriptive risk-based methodology to cybersecurity with the goal of identifying, mitigating, and eliminating system vulnerabilities to protect systems from unauthorized access, use, disclosure, disruption, modification, or destruction. Within the United States Air Force, the Air Force Life Cycle Management Center is tasked with conducting RMF for legacy weapon systems (designated as the Platform IT (PIT) systems) [39]. This PIT assessment and authorization process consists of six steps described in the next paragraph [12].

First, the team must categorize the PIT system according to the information displayed, processed, stored, and transmitted along with the classification of the information and associated technologies. Second, security controls are selected (or assigned) based on the impact resulting from the loss of said information (i.e., criticality analysis) [11]. The third step is implementing said controls with consideration for cybersecurity requirements across the entire system development life cycle—although security controls have been historically applied to IT systems, many have been tailored for PIT systems with prescribed overlays [39]. The fourth step is key to the RMF process and assesses the effectiveness of applied security controls through threat mapping and vulnerability analysis. On a related note, much of the security work conducted today is exclusively focused on this step. Based on the identified vulnerabilities, the fifth step is to produce a risk assessment and mitigation plan, which is then briefed to the Authorization Official for authorization. The sixth step of the RMF process is continuous monitoring of the system with respect to cybersecurity. As the system and threat environment evolve over time, security control effectiveness needs to be continuously assessed while keeping in mind future changes and cybersecurity impact.

E. Avionics Cyberspace Vulnerability Assessment and Mitigation (ACVAM) and Cyber Hardening Efforts

The United States Air Force Research Laboratory (AFRL), in conjunction with the Air Force Institute of Technology's Center for Cyberspace Research, developed an Avionics Cyberspace Vulnerability Assessment and Mitigation (ACVAM) Workshop [40]. This weapon-system-specific workshop teaches a thorough analysis approach by systematically identifying and assessing all external inputs and communications paths to and from a weapon system (i.e., an exhaustive boundary analysis of the system's architecture). The major activities include gathering information, identifying and analyzing access points, finding and analyzing susceptibilities, anticipating attacks, and applying and recommending mitigations and protections. The ACVAM approach requires extensive Subject Matter Expert (SME) involvement, access to design documents, and detailed operator insight to discover susceptibilities and determine appropriate mitigations to increase mission assurance by eliminating or reducing vulnerability to cyberattacks [41].

Additionally, AFRL is developing more specific cyber hardening tools and resiliency instructions [42]. While specific details are not publicly available, the cyber hardening approach was recently briefed to the defense community at large [41]. In general, this approach describes avionics cyber hardening and resiliency concepts and suggests ways to protect avionics and related systems from cyber-attack. Moreover, this approach encourages engineers to 'think avionics cyber' using three tenets of cyber protection: focus on what's critical; restrict access to the critical; and detect, react, and adapt [43].

F. Attack Path Analysis via Automotive Example

Historically, attack path analysis has served the security community well [44]. In a great example from the automotive domain, Checkoway *et al.* provide a practical attack path analysis and comprehensive discussion which solidifies the importance of threat modeling as a cybersecurity architectural analysis technique [6]. While this specific example is automobile centric, many similarities are shared between cyber-physical systems. More specifically, the work details a four-step method of analyses. First, threat model characterization is accomplished through identification of external attack vectors and attack surfaces. Second, vulnerability analysis addresses the accessibility, criticality, and exploitability of potential vulnerabilities. Third, a threat assessment attempts to gauge the attacker's motivation by answering the question of what utility a given attack path has for the attacker. Finally, the approach suggests mitigation actions by synthesizing similarities among vulnerabilities to provide pragmatic recommendations for enhancing the system's cybersecurity.

G. System Theory Process Analysis for Security (STPA-Sec)

In recent work, MIT's System Theory Process Analysis (STPA) approach for safety was extended to focus on security-related concerns, known as STPA-Sec [45]. The goal of this approach is to ensure mission-critical functions are maintained in the face of disruption(s). Starting from a strategic viewpoint, system developers and users can proactively shape the operational environment by controlling specified mission critical system risks. This top-down approach elevates the security problem from guarding the system (or network) against all potential attack paths to a higher-level problem of assuring the system's critical functions. The STPA-Sec steps include: identifying unacceptable losses, identifying system hazards (vulnerabilities), drawing the system functional control structure, and identifying unsafe or insecure CAs [45]. This method has been embraced by defense and commercial industries with several favorable case studies [46].

H. Functional Mission Analysis for Cyber (FMA-C)

The DoD has adopted Functional Mission Analysis for Cyber (FMA-C) as an approach to secure operational computer networks [47]. FMA-C is being taught to thousands of airmen in an effort to assure critical cyber systems and reduce vulnerabilities. While the structure and content of FMA-C is similar to STPA-Sec, its application is tailored to As-Is Information Technology infrastructures. In practice, USAF Mission Defense Teams apply FMA-C to fielded cyber systems to identify mission critical vulnerabilities. It has proved to be a useful tool for understanding and mitigating risks in traditional cyber (i.e. ICT) domains.

I. Other Notable Methodologies

As previously noted, other methodologies and frameworks for systems-level security analysis are sure to exist which are not covered in this work. A few notable works focused on mission assurance are available here [48], [49], [50] and on software here [51], [52].

IV. Characteristics for Effective Cybersecurity Architectural Analysis

This section identifies desirable characteristics for cybersecurity architectural analysis and cross-examines the approaches discussed in Section III.

A. Cybersecurity Architectural Analysis Characteristics

The first characteristic is definitional in nature and classifies approaches as either top-down or bottom-up. Those defined as top-down start with analysis at the function level with identification and examination of critical missions and/or capabilities—sometimes operations depending on how the approach is being applied. As is typical of architecting for new systems (and sometimes upgrades), higher-level functional analysis leads to further functional decomposition and allocation to a more specific form (e.g., lower subsystems, elements, or components). These approaches lend themselves to the identification of stakeholder security needs, early trade-offs, thorough security requirements definition, and integration of more holistic security solutions [15].

Conversely, bottom-up approaches begin with the form in mind (i.e., the physical or technological solution) and often focus on perimeter security through boundary analysis [53]. While this approach successfully identifies vulnerabilities in networked components, it is often less useful for protecting systems from intelligent adversaries. For example, Bayuk and Horowitz [54] surmise that perimeter defense tactics are largely ineffective, and conclude that a top-down, risk-based systems engineering approach to system security should be used instead.

The next key characteristic is whether the approach should be driven by threats or vulnerabilities. Prior research suggests that the foundation for improving system security starts with an analysis of potential threats, which leads to more appropriate security requirements for implementation [44]. This is intuitive; without first understanding the adversary—system-specific threats (and their rapid agility)—it is difficult, or impossible, to defend against them. Understanding and modeling the threat becomes a critical prerequisite for generating and developing secure systems [55]. Once the model has been developed and validated, vulnerability analysis is the logical follow-on. With the threats understood, the system architecture can be analyzed for vulnerable access points through techniques such as attack path analysis and/or red teaming.

While acknowledging the rapidly changing nature of threats, the exercise of red teaming and brainstorming potential attack paths is a helpful critical thinking exercise for ensuring sound cybersecurity practices. Moreover, threat modeling and vulnerability analysis typically form the foundation for cybersecurity architectural analysis. While threat modeling alone does not ensure cybersecurity, rigorous threat modeling and vulnerability analysis are helpful for ensuring the security of realized systems. However, more focus should be applied to providing security solutions and not just focused on identifying problems.

In today's highly contested cyberspace environment, documentation-based engineering is largely ineffective against dynamic adversaries [44]. Developing a successful response to a dynamic adversary necessitates the tools and methods used to develop countermeasures be, in kind, dynamic. In response to these complexities, Model-Based Systems Engineering (MBSE) offers an integrated modeling approach capable of mapping desired capabilities to functions (and even components), as well as providing traceability and fit-for-purpose views to enable more effective decision-making [56]. In a recent effort, Apvrille and Roudier proposed SysML-Sec, an injection of security considerations into SysML in an effort to foster integration between system designers and security experts [57]. SysML-Sec and more generally MBSE approaches enable security-focused computer simulations of a potential system architecture. These executable architectures provide tremendous value by providing insights into early design trade-off analysis [58]. While MBSE requires significant initial investment in tools and training, it significantly increases the depth of possible architectural analysis especially in executable architectures..

B. Mapping of Characteristics to Approaches

Table I provides a mapping of the proposed architectural analysis characteristics to the surveyed approaches from Section III. This mapping seeks to provide a consolidated reference for differentiating approaches to inform the user and assist in selecting an appropriate cybersecurity architectural approach which meets the stakeholders' needs. Consideration is given to each approaches' usability, scalability, and tool availability. The ideal approach will also easily facilitate modeling and simulation studies to perform early design feasibility studies and support trade-off analysis (i.e., MBSE).

In general, bottom-up approaches are relatively systematic; however, historically they have not produced secure systems and tend to scale poorly. Top-down approaches have the benefit of being more scalable, but they often require a high level of tool proficiency to effectively model (thus, the potential of MBSE to systems security is largely missed). While vulnerability analysis is inherent in every approach, a threat-based approach is less so. This aspect is important because effectively safeguarding unprecedented, complex systems requires more than a good architectural tool or technique – a holistic engineering approach that embraces all aspects of security (e.g., people, processes, policy, technology, feasibility, cost, etc.) is required [59], [60].

While providing a detailed case study for each approach surveyed in this work would be ideal, it is simply not feasible as some approaches take months if not years to complete. Thus, we chose to further explore STPA-Sec as relatively simple and effective means for performing systems security analysis for a complex SoI. For example, STPA-Sec can be accomplished without extensive training and can be accomplished in a matter of days rather than months [61]. While STPA-Sec does not formally use MBSE tools, this decision was primarily driven by STPA-Sec's top-down approach which begins in the conceptual phase of the system life cycle—earlier than other approaches considered (e.g. starting with the Business or Mission Analysis technical processes in ISO/ISE/IEEE 15288) [61].

TABLE I: ARCHITECTURAL APPROACHES TO CHARACTERISTICS MAPPING.

	Top Down	Bottom Up	Threat Driven	Vul. Based	MBSE Integrated	MBSE Executable	Tool Based
DoDAF + Richards	X[1]			X	X	X[4]	X
CRAF	X[1]		X	X	X	X	X
UAF Security	X			X	X	X[4]	X
ACVAM		X	X	X			
STPA-Sec	X[2]			X			
RMF		X[5]	X	X	X[3]		

1. Promotes a top-down approach after mission functions are identified (i.e., does not include mission thread analysis).
2. Approach begins at a higher level than other approaches examined (i.e., includes mission thread analysis) and includes lower level analysis.
3. Suggests using MBSE, but not required and often not considered.
4. Would require pairing with additional modeling & simulation plugin.
5. RMF is intended to be a top-down approach but is often applied bottom-up using security control compliance based on system type.

V. SYSTEMS SECURITY STPA-SEC ANALYSIS APPROACH

STPA-Sec is a promising approach for performing conceptual systems security analysis based on a methodology of systems theory dating back to Leveson's original systems safety work which has been well received within the safety, aeronautical, and systems engineering communities [23], [46]. STPA-Sec is an extension of this methodology to the security domain and has been shown to effectively address security issues in complex cyber-physical systems [63]. In this section and the one that follows, we seek to highlight some of STPA-Sec's utility in facilitating early security and resiliency requirements generation with traceability to the stakeholders' mission.

In Table II, we present the STPA-Sec approach organized into three levels of systems security analysis: Conceptual,

Architectural, and Design. Of great importance, these levels align well with the established systems and software engineering processes in ISO/IEC/IEEE 15288, and the recently released NIST SP 800-160 [14], [15]. This systems-oriented approach begins at the highest level of abstraction with the Business/mission Analysis (BA) and Stakeholder Needs and requirements definition (SN) processes to define early security requirements. Next, the System Requirements definition (SR) and Architecture Definition (AR) processes explore potential architectures and the desired system capability from a functional level which results in architectural specific "design-to" criteria. In the third phase, potential general forms of the system are considered and analyzed in the Design definition (DE) process which results in specific "build-to" criteria. Table II provides a general assessment of each phase's difficulty and duration, as well as, a listing of the number of STPA-Sec steps each of which are elaborated upon below.

TABLE II: SYSTEMS SECURITY ANALYSIS STPA-SEC OVERVIEW.

	Systems Security Oriented STPA-Sec Phases		
	Concept Analysis	***Architectural Analysis***	***Design Analysis***
Purpose	Determine Security Requirements	Determine Design-To Criteria	Determine Build-To Criteria
ISO/IEC/IEEE Process	BA/SN	SR/AR	DE
Difficulty	Easy	Easy	Moderate
Duration	Hours	Days	Weeks
STPA-Sec Steps	4 Steps	5 Steps	5 Steps

A. Concept Analysis

As shown in Table III, the four conceptually-oriented STPA-Sec steps start with mission-level analysis to prevent the system from entering hazardous system states that could lead to unacceptable losses and mission failure. Beginning systems security analysis at the mission-level allows security engineers to more accurately understand the stakeholders' needs and maximizes the engineering trade space as system goals are transformed into constraints (i.e., early safety, security, and resiliency requirements).

TABLE III: STPA-SEC CONCEPT ANALYSIS.

Step	**Description**
1. Define the SoI's purpose and goal	Capture the mission statement and key activities of the system: 1) A system to: (What) 2) By Means of: (How) 3) In Order to: (Why)
2. Identify unacceptable losses	Define high level, intolerable system outcomes to key stakeholders (e.g., loss of life, injury, damage to equipment, reputation, mission, etc.).
3. Identify hazards	Identify system states that when coupled with worst case conditions lead to an unacceptable loss.
4. Develop system security constraints	Develop mission-informed security constraints that prevent the system from entering hazardous states. These constraints are synonymous with early safety, security, and resiliency functional requirements.

The first step of STPA-Sec defines the SoI's mission in terms of a purpose and goal from the stakeholders' perspective (akin to the BA and SN processes). This is done in a relatively straightforward fashion with emphasis on stakeholder involvement by standardizing the mission statement into three parts: 1) A system to 2) by means of 3) in order to. The first phrase "A system to" is meant to capture the primary purpose of the system (i.e., the What) in a few words. The "by means of" identifies the activities or processes the system uses to achieve its purpose (i.e., the How). Lastly, the "In order to" identifies the goal, or what mission the system contributes to (i.e., the Why). Accurately defining the desired system's purpose and goal requires involvement from key stakeholders such as mission owner(s), operators, and users. Moreover, correctly defining the mission (or business case) provides a baseline for prioritizing and performing security tradeoffs within an operationally-focused paradigm.

The second step of STPA-Sec identifies unacceptable losses. An unacceptable loss is an specific, unacceptable outcome as defined by mission and system owners (i.e., the key stakeholders). An unacceptable loss should be specific and at a high level. The system losses should identify what is of highest value to the stakeholders and differentiate from what is nice to have/desired. Unacceptable losses can be mission, personnel, or equipment loss; common unacceptable losses include loss of life and loss of mission essential equipment. Any outcome that a key stakeholder is concerned about should be identified. For example, loss of reputation or loss of critical data are examples of unacceptable losses that can be addressed through STPA-Sec. Given the importance of these unacceptable losses to the mission system and stakeholders, they provide key information to drive requirements for safety, survivability, and security.

The third step identifies hazards that can contribute to these unacceptable losses. STPA defines a hazard as a system state (or set of conditions) that together with a worst-case set of environmental conditions will lead to an unacceptable loss [63]. The hazards identified should be within the system boundary and not themselves an environmental condition or external actor. As a general rule, hazards should be abstracted up to the highest level possible and in most cases the list of hazards will be fewer than 10. Identifying hazards can also serve to refine and clarify the list of unacceptable losses, as each hazard should be mapped to one or more unacceptable losses (otherwise it is not a hazard or the list of unacceptable losses is incomplete) [63].

Controlling hazards is STPA-Sec's conceptual mechanism for delivering system cybersecurity. The mechanism is based on Leveson's STAMP model that associates the high-level unacceptable losses as arising from control deficiencies across the system rather than component failures. The control deficiencies manifest as problems between components (interactions) rather than simple mechanical failures. The latter have been the traditional cause of failures in mechanical systems, but the former provides much more utility when developing contemporary large, software-intense systems. Hazardous states are a necessary precondition to loss. For example, if the unacceptable loss is two aircraft colliding, then the associated hazard could be generalized as failing to maintain safe separation between the aircraft. If safe separation is

maintained, the two aircraft should never collide. The need for safe separation can be identified based on the first part of STPA-Sec. Ensuring safe separation is maintained (a control function) throughout operations is an engineering problem and can be addressed through systems engineering (secure systems engineering specifically). There are several different ways the violation might occur. The air traffic control system might be hacked or attacked. One or both of the aircraft might be subjected to a cyber attack. Likewise, there are any number of mitigations that can be used to address the hazard. However, at this stage of the System Engineering process, the hazardous functionality (safe separation not enforced) is already identified. In other words, loss prevention functionality (safe separation enforcement) is now identified and the remainder of the engineering process can focus on developing a suitable architecture to enforce this functionality. This approach allows engineers to handle safety and security in the same manner that all other emergent system properties are addressed. The systems approach does not preclude the need for reliable components. STPA-Sec still identifies scenarios involving component failure, but it also highlights complex, highly interactive scenarios involving management decisions, operations processes and operators. These other factors are also contributory to many losses. Therefore ensuring safe and secure operations must go beyond a focus on technology.

The fourth step of the concept analysis phase is to develop system security constraints that prevent the SoI from entering one of the previously identified hazardous states. These constraints are restrictions placed on the system (and implemented via the security architecture) to bound operation within acceptable parameters. In this way, the first four steps of STPA-Sec begin to specify acceptable and non-acceptable system states which can eventually be formally tested and verified when the architecture is developed. The insights gained through STPA-Sec can also be used to inform and improve early MBSE efforts. These safety and security focused constraints also provide stakeholder-focused traceability for safety, security, and resiliency requirements which are important for system survivability (a critical issue in U.S. DoD systems [64]).

B. Architectural Analysis

STPA-Sec Architectural Analysis is a continuation of the conceptual phase and examines the SoI at the functional level (rather than a form specific implementation as is often the case in cybersecurity analyses). This approach maintains the largest trade space for potential solutions and helps ensure the desired system functionality is implemented without unnecessary architectural and design constrictions.

Table IV details the necessary steps to perform STPA-Sec architectural analysis where a functional control model is used to represent the SoI. The functional control structure can be accomplished at various levels of abstraction such that an entire system is represented as a single model or it can be decomposed into multiple sub-models used to more specifically understand the control the SoI's functions. This tailorable approach uses functional decomposition to more thoroughly understand critical mission-essential relationships between key actors and processes represented as Control Actions (CA). The phase begins with identification of all required CAs followed by an analysis of their criticality and how they contribute to preventing the SoI from entering hazardous states (akin to the SR process). The output of STPA-Sec Architectural analysis identifies potentially hazardous or unsecure CAs for a given system architecture (akin to the AR process). In essence these CAs form system level security requirements given a system architecture.

TABLE IV: STPA-SEC ARCHITECTURAL ANALYSIS.

Step	Description
1. Identify model elements	Identify actor(s), controller(s), and controlled process(es) for the SoI at the desired level of abstraction.
2. Identify each elements' responsibilities	Capture the description and actions planned to be taken for the model elements identified.
3. Model control relationships	Organize the model elements to pictorial show the relationships between elements in a functional control structure.
4. Identify Control Actions (CA)	Captures (in verb form) the actions necessary for each element to execute their responsibilities.
5. Complete the CA analysis table	This table systematically enumerates which hazards are caused by each CA identified in step 4.

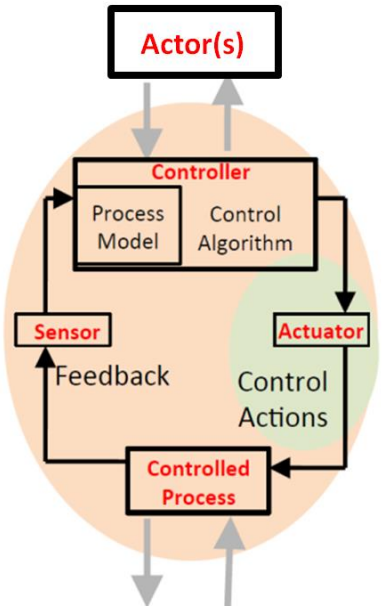
The first step of this phase identifies each of the key model elements including actors, controllers, actuators, sensors, and controlled processes. Figure 2 illustrates a basic control structure example. Starting from the bottom of the model, the controlled process(es) is(are) the previously identified (Concept Analysis Step 1) activities the system uses to achieve its purpose. Beginning with these "How" activities, the primary task in step 1 of STPA-Sec architectural analysis is identifying the controller and actors responsible for performing the process.

Once model elements are identified, then responsibilities for actors and controllers are populated in step 2. This step involves capturing these important responsibilities and assigning them to the appropriate actor for each controlled process (or action). These responsibilities can be identified from operational or system documentation, and in particular discussions with users, system SME's, and other stakeholders. The third step primarily organizes the previously identified model elements into a functional control structure (i.e., a model) as shown in Figure 2. Step 4 identifies CAs for the system. A CA is a terse verb (action) statement capturing the execution of a function (or task) necessary to control the subject process. Populating the list of CAs begins with pairing down the responsibilities previously identified in step 2 into CAs that an actor or controller performs to manage the controlled process. CAs exist at various levels of abstraction based on the desired depth of analysis. Often it is best to start at the highest level of abstraction for the initial functional architectural analysis and then move to potentially hazardous actions as previously identified.

The bulk of STPA-Sec architectural analysis resides in the fifth step – populating a CA analysis table (an example is provided in Section VI). This step requires a thorough analysis of each CA identified in step 4 and enumerates what, if any, hazardous conditions can be created by the systems' actions. More specifically, each CA is evaluated across four scenarios:

the CA is not provided; the CA is provided; the CA is provided too late, too early, or out of sequence; and the CA is stopped too soon or applied too long. This analysis clearly identifies when CAs need to be applied and not applied in order to prevent unsafe or unsecure hazardous states from occurring during system operation. This step provides an initial "design-to" criteria which is further decomposed and studied during design.

FIG 2. STPA BASIC CONTROL STRUCTURE. FROM [63].



C. *Design Analysis*

The STPA-Sec Design Analysis phase studies the specifics of a CA using relatively simple process models and scenarios. These process models enumerate the decision logic, key variables, and acceptable variable values associated with each CA in a systematic and straight forward fashion. Additionally, this analysis identifies which feedback mechanism is responsible for those process model variable values (e.g., a sensor or computing mechanism). STPA-Sec design analysis focuses on more thoroughly understanding and specifying the CAs which prevent the SoI from entering potentially hazardous and unsecure states. The steps of design analysis are captured in Table V.

Step 1 of STPA-Sec Design Analysis develops process model descriptions. During this step, it is advantageous to first generate process model descriptions for CAs determined as potentially hazardous from the completed STPA-Sec Architectural Analysis. This is because a complex system may have a large (and potentially overwhelming) number of process model descriptions. Each process model should briefly describe the scenario of interest and focus on when to execute a given CA with details such as identifying the specific system elements and potential responses to CAs. Additionally, the process model should include assumptions about the controlled process.

In step 2, Process Model Variables (PMVs) are described as various conditions which indicate a system state (i.e., one of a number of discrete states the SoI could exist in). These conditions and states are then enumerated in step 3 to ensure all potential PMV values are properly understood to specify potentially hazardous systems states. Step 4 identifies the sensors which are responsible for generating said PMV values (i.e., data) to include conventional sensors, personnel, computer systems, etc. This step also lends insight into which feedback sensors are critical to monitor for potentially hazardous states and enforce CAs. Steps 1-4 produce a list of preliminary design considerations to include detailed CAs and PMVs which specify functional logic to inform subsystem and component implementation and verification.

Step 5 of STPA-Sec design analysis is the generation of causal scenarios where the impact of environmental conditions (previously explored during conceptual analysis) are examined to more specifically understand and assess how losses may occur. Akin to tabletop red teaming, causal scenario generation is typically conducted by system experts, well-qualified users, and threat analysts with the goal of identifying plausible scenarios (or conditions combined with effects outside the system boundary) that violate or breach a constraint. In a general sense, this final step also serves to provide validation for the thoroughness of the entire STPA-Sec analysis effort. In this way, changes or additional constraints are often identified as part of the causal scenarios when attempting to 'break' the SoI.

TABLE V: STPA-SEC DESIGN ANALYSIS.

Step	Description
1. Develop process model descriptions	Describes the decision logic ("in plain English") for executing a given CA.
2. Identify Process Model Variables (PMV)	PMVs are measurable indicators of the conditions that trigger a CA.
3. Specify PMV values	PMV values are all the possible values a PMV can be assigned both acceptable and hazardous.
4. Identify PMV sensors	Identifies which sensors provide PMV values to the actors and controller for decision making.
5. Carry out causal scenarios	Brainstorm how a specific implementation of the system may be compromised. Identifies critical CAs and validates the thoroughness of the model, CAs, and constraints.

VI. AUTONOMOUS RESUPPPLY SPACE VEHICLE CASE STUDY

In this section, we provide a simplified case study highlighting STPA-Sec's utility for eliciting and defining safety, security, and resiliency requirements, as well as, specifying design and implementation criteria which can be formally verified. This section demonstrates STPA-Sec's high-level flow and outputs, and is not intended to be a comprehensive analysis (additional examples available here [46]). This case study was initially conducted as an ad hoc working group consisting of engineers, operators, and mission owners, the details of which have been obfuscated by discussing a notional autonomous resupply space vehicle (i.e., a complex cyber-physical system). Additionally, this discussion extends a recent satellite STPA analysis performed by Thomas [65].

Iterative analysis of STPA-Sec's various phases is encouraged and somewhat limited to the expertise and skillset of those generating the scenarios. It is highly recommended to

generate these scenarios with expertise from system operators, SME's, and with those familiar with likely threats because STPA-Sec asks operators and SMEs context-specific, direct questions rather than general, ambiguous questions. For example, an operator might be asked why they might activate an ABORT command for the loss of communications with the vehicle (or perhaps better yet, how long would they reasonably wait before they activated the ABORT command if communication had been lost). Perhaps more importantly, the very inclusion of the ABORT functionality can be discussed and presented as a security trade before the functionality is designed into the architecture. Hypothetically, the ABORT command might be used to terminate the mission and return the vehicle to earth before resupply had been completed. The functionality might be included to prevent damage to the vehicle or ISS in the case of communications loss. However, the functionality might also be used by an attacker to cause mission failure. Through early discussions with operators and stakeholders, alternative means might be devised to ensure the safety of the vehicle in the case of a loss of communications. The approach described is fundamentally different than current security approaches that would focus on assuring the functionality after it was already included in the architecture. STPA-Sec facilitates a discussion on whether or not particular functionality should be included and allows an early assessment as to the degree engineers believe the functionality can be assured. Obviously, if functionality costs more to secure than it adds to mission completion, serious consideration should be given to not including (instantiating) the functionality into the architecture.

Obviously, not every scenario can be imagined, but the STPA-Sec produces a set of hazardous scenarios that will lead to the unacceptable losses if they occur under the worst case environmental conditions (that are beyond the ability of the engineering team to control). Operator insights are used to develop and improve the architecture. The top-down approach ensures that operators and SMEs only focus on scenarios that lead to losses of interest to stakeholders. The approach is also much more focused than simply asking operators, "what might go wrong." Not every adverse situation will lead to a loss that stakeholders care about. Proposing solutions to a security problem that does not concern the key stakeholders adds unnecessary cost and complexity to the system.

A. Concept Analysis Phase

To begin the analysis, we first clarified and documented the system's intended mission by defining the system's purpose and goal with several key stakeholders. High-level documentation such as ConOps, OpCons, gap analysis, mission needs statements, and any use cases are reviewed with data parsed into the framework depicted below. This phase is essential for properly understanding the scope of the security problem (i.e., context) while simultaneously not over limiting the solution space. For example, from a holistic systems perspective technical and procedural security solutions should be equally considered.

A system to do (what): Autonomously resupply the International Space Station (ISS)
By means of (how): Launching, Flying proper trajectories, docking, and returning to Earth
In order to (why): Maintain operations on the ISS in a cost-effective manner while minimizing risk to astronauts

Next, we considered any unacceptable losses with respect to the stated system's mission. Initial stakeholder discussions (facilitated by the initial mission statement) resulted in a number of lower level losses which were consolidated into a list of three high-level unacceptable losses. While the list can be more or less detailed, three losses satisfies our needs to ensure they were sufficiently different:

1. Loss or significant damage to Vehicle or ISS
2. Loss of vehicle cargo
3. Death or Injury to astronauts

The third step includes identifying hazards which may contribute to or result in these unacceptable losses. This is done by considering the functionality necessary for the completion of the mission (in this case: Launch, Flight, Docking and Return). Hazards can contribute to multiple losses, but must contribute to at least one unacceptable loss. The goal at this point is not to identify every possible undesirable outcome. For instance, temporary losses of communication with the vehicle are undesirable, however, at this point we are focusing only on hazards that can lead to outcomes that stakeholders have agreed are unacceptable and must be mitigated (such as loss of the cargo).

Another important point is that all the hazards must be under the control of the systems engineers designing the system. Space weather would not be considered a hazard, however, the system exceeding operating temperature tolerances (such as might occur from space weather) would be a hazard requiring engineers to design a suitable environmental control system. Detailed causal scenarios such as space weather extremes are handled later in the scenario development phase (as are generic cyber attacks). Likewise, operator error is not a hazard per se, but operator failure to issue a docking command when required would be a detailed causal scenario. Like maintaining adequate environmental control onboard the vehicle, the focus is maintaining desired functionality (i.e., an operator issuing proper commands under the proper conditions).

Table VI lists a high-level set of hazards for the example. scenario. This is done in a simple table with the losses listed across the top and potential hazards along the left-hand side. The preliminary analysis is not complete at this point. Early systems engineering analysis and conceptual design should focus on functionality not the particular form the architecture will take. One limitation of "baking in" cyber security has been the dependence on having detailed information about the particular "form" or architecture under consideration (e.g Operating systems). STPA-Sec enables security analysis without such detailed information. In the simplified example,

the basic functionality the resupply vehicle requires to complete the mission was specified during the initial phase of the analysis. The vehicle must have functionality to launch, fly trajectory, dock, and return. Instantiating this functionality in a vehicle that is secure and resilient is the ultimate goal of the security engineering effort.

TABLE VI: HAZARDS AND LOSSES MAPPING.

		Losses		
		L1: Loss of Vehicle or ISS	L2: Significant Damage to ISS or Vehicle	L3: Loss of Vehicle payload
Hazards	H1: Failure to Maintain Safe Separation between ISS and Vehicle	X	X	
	H2: Exceed Safe Closure rate between vehicle and ISS	X	X	
	H3: Payload Environment not maintained within limits			X

B. Architectural Analysis Phase

Figure 3 depicts a high-level functional control structure that engineers can use to better define and reason about how the set of 4 basic functionalities can be assured, meaning that their behavior can be bounded within acceptable limits. The simple block diagram consists of a ground system, an automated control and data handling system aboard the vehicle, an environmental system, a movement control system, and "other subsystems." We previously determined that one of the unacceptable losses is damage to the vehicle and/or ISS. A hazard under the control of the designers is exceeding the safe closure rate between the vehicle and the ISS during docking (one of the high-level functions). The closure rate might occur accidentally or it might occur as the result of a cyber attack on some part of the system. However, an early but critical design choice is how the docking functionality will be implemented.

The functional structure represents two different architectures (with one potentially easier to assure against disruptions than the other). Perhaps locking out the ground segment from providing commands during the docking maneuver might be advisable. If such separation were desired, then any C2 attacks on the ground segment would not disrupt the docking functionality. However, severing C2 with the ground segment during the docking process eliminates the possibility of ground controller intervention if an unexpected situation arises. Security and resilience are not the only architectural trades impacting the connectivity of the ISS, ground segment, and the vehicle. Cost and complexity are also factors. Inclusion of only two of the three might reduce cost and complexity. Likewise, perhaps the ground only requires an ABORT function. The point is not to decide these factors solely on the basis of the STPA-Sec analysis, but the analysis, as it proceeds, enables and facilitates a security discussion during the early trades where the key decisions are made instead of asking security engineers to "secure" an architecture that may be unsecureable.

Figure 4 decomposes the initial depiction in Figure 3 to facilitate deeper analysis. The simplified example now turns to increasing understanding on how particular hazards can be mitigated in the design to follow and the requirements for achieving this end. Mitigating H2 is examined in more detail. Closure rate and position are controlled by the maneuver control system. The engineering challenge is defining a functional structure that will maintain closure rates within desired limits.

Figure 4 further decomposes the functionality into a control loop model consisting of the sensing functions, the controlling functions, and the thruster or actuation functions. The STPA-Sec analysis provides a verifiable set of commands (or CAs) necessary to perform all mission functions. In a complete STPA-Sec analysis, a full enumeration of controls actions required for each basic functionality (launching, flying, docking, and returning) would be captured. In this example for brevity we have selected a single CA for further analysis. For docking, the CA list would include an INCREASE THRUST command. The INCREASE THRUST command is necessary to perform the docking process, however, if applied the wrong time (or continued too long) it can create H2. The next portion of the analysis identifies the context under which issuing the INCREASE THRUST command might create a hazard or situation where failure to issue the command creates a hazard. This analysis is accomplished through the Control Action Analysis Table, Table VII.

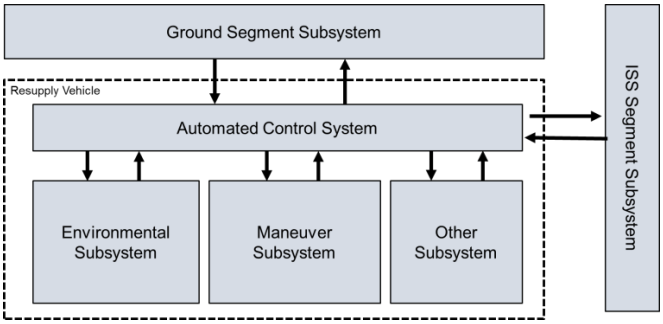


FIG 3. HIGH LEVEL STRUCTURE WITH ISS CONTROL OF DOCKING. FROM [63].

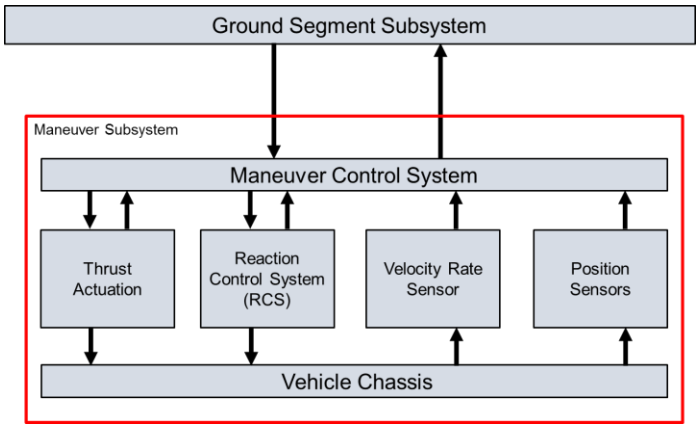


FIG 4. REFINED CONTROL STRUCTURE. FROM [63].

If an attacker wishing to damage the ISS or vehicle was somehow able to issue the INCREASE THRUST command when the vehicle was already at max desirable speed or when the vehicle was in close proximity to the station, a hazardous situation (H1 or H2) develops. The hazardous situation does not necessarily mean a loss will occur, but the hazardous situation is a necessary precondition for the loss to occur. Thus, if the functionality can be controlled within the desired limits in the face of intentional and unintentional disruptions, then a case can be made for the system's resilience and security. Issuing the INCREASE THRUST command when the vehicle is at max acceptable speed in close proximity to the ISS places the system in a hazardous state. The CA analysis table is the culminating product of the architecture analysis capturing critical analysis for informing the system design further explored in the design analysis phase of STPA-Sec.

TABLE VII: STPA-SEC CONTROL ACTION (CA) ANALYSIS.

Control Action	**Not providing causes Hazard**	**Providing Causes Hazard**	**Too Early/too late, wrong order**	**Stopping too soon/applying too long**
Increase Thrust	CA-ACS-# 1a: Not providing INCREASE THRUST command is hazardous if thrust is required to slow closure rate to safe speed [H-1, H-2]	CA-ACS-# 1b: Providing INCREASE THRUST command is hazardous if already at max speed or when in close proximity to the ISS [H-1, H-2]	CA-ACS-# 1c: Providing INCREASE THRUST command **too late** is hazardous if already at max speed or when in close proximity to the ISS [H-1, H-2]	CA-ACS-# 1d: Providing INCREASE THRUST command **too long** is hazardous if already at max speed or when in close proximity to the ISS [H-1, H-2]

C. Design Analysis Phase (With a Focus on Causal Scenarios)

In the design phase, steps 1-4, seek to understand why and how a particular command (or CA) might be issued. This is done through consideration of the process model description, PMV, and acceptable PMV values. This allows developers and security specialists to specify an acceptable range of PMV values which can be formally modelled and tested as desired. For example, if the space vehicle's thrust is the PMV, acceptable values may range from 10-50 km/s. In order to comprehensively specify and secure the SoI's behavior (functionality) PMVs and their acceptable values need to be captured for each system's function and respective CAs.

Lastly, in step 5, we introduce four high-level causal scenarios. The first scenario is that the Attitude Control thrust system might apply thrust when it is not required based on missing, faulty or incomplete feedback. The second scenario is that the Attitude Control System might receive the proper feedback information on current speed, but might have an internal logic error in either interpreting the input data to determine the actual state or issuing the wrong command based on the correct state. The third high-level scenario is that the thruster might issue the incorrect amount of thrust based on input from the ACS. The fourth scenario involves some type of component failure (stuck or broken nozzle).

It is important to note that each of the high level scenarios can be decomposed further as the design continues and architectural decisions are made. Based on the abbreviated discussion, designers might decide to include a mechanical backup to the onboard vehicle autonomy. The early analysis highlights where the early weight of security engineering effort should be focused. The system's positioning and environmental control functionality have significant impact. At a functional level (i.e., independent of a particular physical solution), security and resiliency requirements can be defined for each hazard. For example, to prevent operator error, additional engineering considerations should be incorporated into the desired system, operational procedures, and training programs. Likewise, the identification of critical information (and information processing) may necessitate additional redundancy costs. Note, this is done at a functional level without a specific architecture. Thus, while this workshop was conducted with a specific system in mind, the results are generally applicable to types (or categories) of systems, defense related or not.

VII. CONCLUSIONS AND FUTURE WORK

This work addresses the gap in systems security engineering and analysis approaches for the development of secure systems by suggesting a simple, yet clear means for specifying measurable and verifiable security design requirements. Moreover, the suggested approach is tailorable with three levels of details, all of which provide traceability to the system owner's needs. The utility of STPA-Sec systems security analysis approach is demonstrated with an autonomous resupply space vehicle example which elevates the security problem from guarding the SoI against potential attack paths to the higher-level systems problem of assuring the SoI's critical functions. Moreover, this approach makes the security problem readily understandable and is mapped to standardized systems security engineering processes as described in ISO/IEC/IEEE 15288 and NIST 800-160.

In particular, this work seeks to improve the science of systems security engineering with a focus on understanding and

defining security requirements. Future work includes completing a detailed study of a next-generation military refueling system to assess the utility of STPA-Sec for conceptual analysis of complex military systems to include a means for eliciting cyber resiliency requirements. Ultimately, continued research in this field enables more secure and defensible systems to be fielded.

DISCLAIMER

The views expressed in this paper are those of the authors and do not reflect the official policy or position of the U. S. Air Force, the Department of Defense, or the U.S. Government.

ACKNOWLEDGMENTS

This work was supported by the U. S. Air Force Research Laboratory, Space Vehicles Directorate, Kirtland Air Force Base, NM.

REFERENCES

[1] Y. Liu, Y. Peng, B. Wang, S. Yao and Z. Liu, "Review on cyber-physical systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 1, pp. 27-40, 2017.

[2] R. Saracco, "Guess What Requires 150 Million lines of Code," EIT Digital, 13 Jan 2016. [Online]. Available: https://www.eitdigital.eu/news-events/blog/article/guess-what-requires-150-million-lines-of-code/. [Accessed Feb 2017].

[3] R. Charette, "IEEE Spectrum: This Car Runs on Code," 1 February 2009. [Online]. Available: http://spectrum.ieee.org/transportation/systems/this-car-runs-on-code. [Accessed 1 June 2017].

[4] A. Greenberg, "Wired," Wired Magazine, 21 July 2015. [Online]. Available: https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/. [Accessed 25 April 2017].

[5] E. Perez, "CNN," CNN, 18 May 2015. [Online]. Available: http://www.cnn.com/2015/05/17/us/fbi-hacker-flight-computer-systems/. [Accessed 25 April 2017].

[6] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham and S. Savage, "Comprehensive Experimental Analyses of Automotive Attack Surfaces," USENIX Security Symposium, 2011.

[7] K. Baldwin, J. Miller, P. Popick and J. Goodnight, "The United States Department of Defense Revitalization of System Security Engineering Through Program Protection," in *IEEE Systems Conference*, 2012.

[8] United States Congress, "Nation Defense Authorization Act 2016 Section 1647," 25 November 2015. [Online]. Available: https://www.congress.gov/114/plaws/publ92/PLAW-114publ92.pdf. [Accessed 1 June 2017].

[9] Department Of Defense, "DoDI 8500.01 Cybersecurity," 2014.

[10] Department of Defense, "DoDI 8510.01 Risk Management Framework (RMF) for DoD Information Technology (IT)," 2014.

[11] Department of Defense, "Defense Acquisition Guidebook Chapter 9 Program Protection," 5 April 2017. [Online]. Available: https://www.dau.mil/tools/dag/Pages/DAG-Page-Viewer.aspx?source=https://www.dau.mil/guidebooks/Shared%20Documents%20HTML/Chapter%209%20Program%20Protection.aspx. [Accessed 1 June 2017].

[12] Department Of Defense, "DoD Program Manager's Guidebook for Integrating the Cybersecurity Risk Management Framework (RMF) into the System Acquisition Lifecycle," 30 October 2015. [Online]. Available: https://acc.dau.mil/adl/en-US/721696/file/81323/Cybersecurity%20Guidebook%20v1.10%20signed.pdf. [Accessed 1 June 2017].

[13] D. I. Good, "The Foundations of Computer Security We Need Some," 29 September 1986. [Online]. Available: http://www.ieee-security.org/CSFWweb/goodessay.html. [Accessed 03 January 2018].

[14] ISO/IEC/IEEE, "Systems and software engineering — System life cycle processes, Third Edition," Geneva, Switzerland, 2015.

[15] R. Ross, M. McEvilley and J. Oren, "NIST Special Publication 800-160: Systems Security Engineering," National Institute of Standards and Technology, Washington DC, 2016.

[16] J. A. Zachman, "A Framework for Information Systems Architecture," *IBM Systems Journal 26*, vol. 26, no. 3, pp. 276-292, 1987.

[17] J. F. Sowa and J. A. Zachman, "Extending and Formalizing the Framework for Information Systems Architecture," *IBM Systems Journal*, vol. 31, no. 3, pp. 590-616, 1992.

[18] J. Zachman, "The Zachman Framework Evolution," 1 April 2011. [Online]. Available: https://www.zachman.com/ea-articles-reference/54-the-zachman-framework-evolution. [Accessed 11 May 2017].

[19] A. Tang, J. Han and P. Chen, "A Comparative Analysis of Architecture Frameworks," *IEEE Computer Society: Proceedings of the 11th Asia-Pacific Software Engineering Conference*, vol. 4, no. 1530-1362, pp. 1-8, 2004.

[20] C. Paulsen, "Cybersecuring Small Businesses," *IEEE Computer*, vol. 49, no. 8, pp. 92-97, 2016.

[21] Department Of Defense, "DoDI 8500.01 Cybersecurity," 2014.

[22] G. Hurlburt, ""Good enough" security: The best we'll ever have," *Computer*, vol. 49, no. 7, pp. 98-101, 2016.

[23] N. Leveson, "A new accident model for engineering safer systems," *Safety science*, vol. 42, no. 4, pp. 237-270, 2004.

[24] ISO/IEC/IEEE 42010, "Systems and Software Engineering: Architecture Description," 2011.

[25] L. O. Mailloux, M. A. McEvilley, S. Khou and J. M. Pecarina, "Putting the "systems" in security engineering: an examination of NIST special publication 800-160," *IEEE Security & Privacy*, vol. 14, no. 4, pp. 76-80, 2016.

[26] M. W. Maier and E. Rechtin, The Art of Systems Architecting, CRC Press, 2009.

[27] Department of Defense, "Department of Defense Architecture Framework," 2010.

[28] R. N. Taylor, N. Medvidovic and E. Dashofy, Software architecture: foundations, theory, and practice., Wiley Publishing, 2009.

[29] E. Crawley, B. Cameron and D. Selva, System Architecture, Hoboken: Pearson, 2016.

[30] J. E. Richards, "Using the Department of Defense Architecture Framework to Develop Security Requirements," 2014. [Online]. Available: sans.org. [Accessed Feb 2017].

[31] P. Singer and A. Friedman, Cybersecurity and Cyberwar, New York: Oxford, 2014.

[32] L. Ertaul and J. Hao, "Enterprise Security Planning with Department of Defense Architecture Framework (DODAF)".

[33] Object Management Group, "Unified Architecture Framework Profile," OMG, 2016.

[34] T. Hambrick and M. Tolbert, "Unified Architecture Framework Profile-Systems Engineering Method for Security Architectures-NMWS 17," 21 May 2017. [Online]. Available: https://nmws2017.com/agenda. [Accessed 21 May 2017].

[35] S. Hassell, "Using DoDAF and Metrics for Evaluation of the Resilience of Systems, Systems of System, and Networks Against Cyber Threats," *INCOSE INSIGHT*, vol. 18, no. 1, pp. 26-28, 2015.

[36] P. Nejib and D. Beyer, "Secure Engineering Assurance Model," 11 June 2014. [Online]. Available: http://www.incose.org/docs/default-source/enchantment/140611beyernajib-lockeedseam.pdf?sfvrsn=2. [Accessed 8 June 2017].

[37] "E-Government Act of 2002. Pub. L. No. 107-347, 116 Stat. 2899," 17 12 2002. [Online]. Available: https://www.congress.gov/107/plaws/publ347/PLAW-107publ347.pdf. [Accessed 30 01 2018].

[38] R. Ross, "Managing Enterprise Security Risk with NIST Standards," *Computer*, pp. 88-91, 20 08 2007.

[39] AFLCMC/EZAS, "Aircraft Cybersecurity Risk Management Framework," 19 May 2014. [Online]. Available: http://www.mys5.org/Proceedings/2014/Day_2_S5_2014/2014-S5-Day2-12_VanNorman.pdf. [Accessed May 2017].

[40] Air Force Institute of Technology Center for Cyberspace Research, "Avionics Cyberspace Vulnerability Assessment and Mitigation (ACVAM) Workshop," Air Force Research Laboratory, 1 December 2015. [Online]. Available: https://www.afit.edu/ccr/page.cfm?page=1184&tabname=Tab2A. [Accessed 1 May 2017].

[41] Air Force Research Laboratory, "Air Force Research Lab Avionics Vulnerability Assessment and Mitigation Efforts," in *Ohio Cyber Dialogue with Industry*, Dayton, 2017.

[42] K. Osborn, "BattleSpace IT - Air Force: An F-16 could be vulnerable to cyber attack," Defense Systems, 18 October 2016. [Online]. Available: https://defensesystems.com/articles/2016/10/18/cyber.aspx. [Accessed May 2017].

[43] J. Hughes and G. Cybenko, "Three tenets for secure cyber-physical system design and assessment," in *SPIE Defense+ Security*, 2014.

[44] J. Cleland-Huang, T. Denning, T. Kohno, F. Shull and S. Weber, "Keeping Ahead of Our Adversaries," *IEEE Software*, vol. 33, no. 3, pp. 24-28, 2016.

[45] W. Young and N. G. Leveson, "An Integrated Approach to Safety and Security Based on Systems Theory," *Communications of the ACM*, vol. 57, no. 2, pp. 31-35, 2014.

[46] Massachusetts Institute of Technology, "MIT Partnership for a Systems Approach to Safety," 27 March 2017. [Online]. Available: http://psas.scripts.mit.edu/home/stamp-workshop-2017/. [Accessed 8 June 2017].

[47] Air Force Cyber College, "Top-down Purpose-based Cybersecurity," 2015. [Online]. Available: https://www.sans.org/summit-archives/file/summit-archive-1492176717.pdf. [Accessed 01 January 2018].

[48] G. Hastings, L. Montella and J. Watters, "MITRE Crown Jewels Analysis," The MITRE Corporation, 2009.

[49] H. G. Goldman, "Building secure, resilient architectures for cyber mission assurance," The MITRE Corporation, 2010.

[50] C. Alberts and A. Dorofee, "Mission Assurance Analysis Protocol (MAAP): Assessing Risk in Complex Environments," 2005.

[51] C. Alberts, C. Woody and A. Dorofee, "Introduction to the Security Engineering Risk Analysis (SERA) Framework," 2014.

[52] Software Engineering Institute, "Security Engineering Risk Analysis (SERA)," CERT- Carnegie Mellon University, 1 November 2015. [Online]. Available: https://www.cert.org/cybersecurity-engineering/research/security-engineering-risk-analysis.cfm?. [Accessed 1 May 2017].

[53] R. Anderson, Security Engineering, 2nd ed., Indianapolis, Indiana: Wiley Publishing, Inc, 2008.

[54] J. Bayuk and B. Horowitz, "An Architectural Systems Engineering Methodology for Addressing Cyber Security," *Systems Engineering*, vol. 14, no. 3, pp. 294-304, 2011.

[55] A. Shostack, Threat modeling: Designing for security, John Wiley & Sons, 2014.

[56] A. Ramos, J. Ferreira and J. Barceló, "Model-based systems engineering: An emerging approach for modern systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42 , no. 1, pp. 101-111, 2012.

[57] L. Apvrille and Y. Roudier, "Towards the Model-Driven Engineering of Secure yet Safe Embedded Systems," *Electronic Proceedings in Theoretical Computer Science*, vol. 148, no. 2, pp. 15-30, 2014.

[58] J. A. Estefan, "Survey of Model-Based Systems Engineering (MBSE) Methodologies," International Counsel On Systems Engineering (INCOSE), 2008.

[59] J. Eloff and M. Eloff, "Information Security Architecture," *Computer Fraud and Security*, vol. 11, pp. 10-16, 2005.

[60] T. Patterson, "Holistic Security: Why Doing More Can Cost You Less and Lower Your Risk," *Computer Fraud and Security*, pp. 13-15, 2003.

[61] W. Young and R. Porad, "System-Theoretic Process Analysis for Security (STPA-SEC): Cyber Security and STPA," 27 March 2017. [Online]. Available: http://psas.scripts.mit.edu/home/wp-content/uploads/2017/04/STAMP_2017_STPA_SEC_TUTORIAL_as-presented.pdf. [Accessed 21 January 2018].

[62] N. Leveson and J. Thomas, "An STPA Primer," 9 September 2013. [Online]. Available: http://sunnyday.mit.edu/STPA-Primer-v0.pdf. [Accessed 12 August 2017].

[63] Defense Acquisition University, "System survivability key performance parameter," 23 May 2017. [Online]. Available: https://dap.dau.mil/acquipedia/Pages/ArticleDetails.aspx?aid=626ace5f-fc1f-4638-9321-fe4345451558. [Accessed 1 June 2017].

[64] J. Thomas, "Basic STPA – Exercisies," April 2017. [Online]. Available: http://psas.scripts.mit.edu/home/wp-content/uploads/2017/04/Thomas-Basic-STPA-exercises.pdf. [Accessed 31 January 2018].

IV. Case Study

Description:

Chapter IV is a self-contained journal article that provides additional information on the methodology and provides the details and results of the case study. This work provides a detailed description of STPA-Sec and provides a complete and thorough example for an aerial refueling platform analysis at the conceptual level. This work expands on Chapter III's introduction to STPA-Sec elaborating each activity in further detail.

This work further answers research question 4 through detailing the tailored STPA-Sec approach used in the case study. The majority of the work answers research questions 5 and 6, as it provides a detailed and independent evaluation of STPA-Sec's utility for eliciting, defining, and understanding security and resiliency requirements for advanced cyber-physical systems. This work uniquely offers a complete and thorough example of STPA-Sec by studying a notional next generation aerial refueling platform, and provides recommendations on its utilization for conceptual systems security analysis for United States Department of Defense Major Weapon Systems.

Publication Details:

Title: Conceptual Systems Security Analysis With Aerial Refueling Case Study

Publication: Pending submission to IEEE Access

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, March 1 2018 version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Conceptual Systems Security Analysis with Aerial Refueling Case Study

Martin "Trae" Span[1], Member, IEEE, Logan O. Mailloux[1], Member, IEEE, Robert F. Mills[1], Senior Member, IEEE, and William "Bill" Young[2]

[1] Air Force Institute of Technology Wright-Patterson Air Force Base, Ohio
[2] Syracuse University, Syracuse, New York

Corresponding author: Logan O. Mailloux (e-mail: logan.mailloux@us.af.mil).

ABSTRACT In today's highly interconnected and technology-reliant environment, cybersecurity is rapidly growing in importance. Cybersecurity is no longer limited to traditional computer systems and IT networks, as a number of highly publicized attacks have occurred against complex cyber-physical systems such as automobiles and airplanes. While numerous vulnerability analysis and architecture analysis approaches are in use, these approaches are often focused on realized systems with limited solution space. An effective approach to understand security and resiliency requirements early in the system acquisition cycle is needed. One such approach, System Theory Process Analysis for Security (STPA-Sec), addresses the cyber-physical security problem from a systems viewpoint at the conceptual level early in the program when the solution trade-space is largest rather than merely examining components and adding protections in production and sustainment. This work provides a detailed and independent evaluation of STPA-Sec's utility for eliciting, defining, and understanding security and resiliency requirements for advanced cyber-physical systems. This work uniquely offers a complete and thorough example of STPA-Sec by studying a notional next generation aerial refueling platform, and demonstrates STPA-Sec's utility to perform conceptual systems security analysis for United States Department of Defense Major Weapon Systems.

INDEX TERMS Conceptual Analysis, Cybersecurity, Security, Security Engineering, Security Requirements, STPA-Sec, Systems Engineering, Systems Security Engineering

I. INTRODUCTION

In today's highly interconnected and technology reliant environment, systems security is rapidly growing in importance. As the Internet of Things continues to grow, the centrality of cyber-physical devices to modern life is increasingly important. Thus, security (and safety) is now an emergent property of cyber-physical systems, where their software and real-time networks require continuous interaction [1]. The cyber threat is one of the most serious economic and national security challenges we face as a nation; America's economic prosperity in the 21st century depends on cybersecurity [2].

In light of growing cyber threats, the United States Department of Defense (U. S. DoD) has made recent changes to expand traditional IT-focused security approaches and mandate security assessments for major weapon systems (MWS) [3], [4], [5], [6], [7], [8]. These policies dictate that acquisition programs integrate security efforts into existing systems engineering processes, and work to ensure security considerations hold equal footing with other requirements and design trade-offs at major program reviews. Although, these DoD mandates are in place, a well-received streamlined executable approach for MWS cybersecurity analysis is yet to be defined.

The challenge of cybersecurity is a "wicked problem" where the problem is twofold: first the problem itself must be defined. Then the solution or actions required to get from as-is to to-be must be determined. Specific to security, it is trying to secure systems via an unknown solution and defending against an unknown evolving threat. Nested, interactive complexity and the socio-technical aspects make cybersecurity a wicked problem [9].

Traditional security approaches are typically conducted at a component level and the results aggregated together into a system analysis. This analysis fails to capture emergent properties of the system that arise from complex interactions. This research addresses these problems through executing a

2169-3536 © 2017 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

conceptual security analysis in a case study of interest to the USAF.

This paper has three main goals: First, it provides a complete and thorough example of STPA-Sec for a complex aerial refueling system. Second, it offers initial tips and recommendations for future practitioners. Third it demonstrates the utility of STPA-Sec for complex MWS security analysis.

This work presents a background of STPA-Sec in Section II. Section III provides an introduction to the case study and presents a tailored STPA-Sec approach. Sections IV-VI detail the steps and analysis performed for each phase of an STPA-Sec analysis. First the purpose of the phase will be introduced, followed by a description of the steps. The case study example is presented along with rationale to assist a practitioner in accomplishing an STPA-Sec analysis for their complex SoI. Section VII provides an assessment of STPA-Sec's utility, and section VIII provides a brief summary and conclusion.

II. BACKGROUND

System Theoretic Process Analysis, STPA-Sec, is a promising methodology for performing secure systems or security analysis. STPA-Sec applies a systems engineering approach providing engineers the largest trade space for developing secure solutions. STPA-Sec elevates the security problem from guarding the system against all potential attack paths to the higher-level problem of assuring the system's critical functions. Because STPA is a top-down, system engineering approach to system safety and security, it can be used early in the system development process to generate high-level safety and security requirements and constraints. These high-level requirements can be refined using STPA to guide the system design process and generate detailed safety and security requirements for individual components [10].

STPA-Sec is an extension of Nancy Levinson's systems safety work: STPA and System Theoretic Accident Model and Process (STAMP) [11]. This work has been well received within the safety and systems engineering community [12]. It is founded on systems theory, analyzing the system as a whole rather than a sum of the parts to capture emergent properties common in complex systems. It asserts safety and security are emergent properties resulting from relationships among the parts of the system. STAMP and STPA define the safety problem as a control problem and leverage control theory to design an effective control structure that reduces or eliminates adverse events [13]. STPA-Sec is an extension of this methodology from the safety domain to security and has been shown to effectively address security through the dissertation work of its founder, Colonel William Young [14]. STPA-Sec has demonstrated promise for facilitating early security and resiliency requirements generation with traceability to stakeholder prioritized safety and security needs. STPA-Sec has proved its utility for cybersecurity in the defense industry and the DoD. The DoD has adopted STPA-Sec as Functional Mission Analysis for Cyber, FMA-C. FMA-C is a version of STPA-Sec owned by the USAF and has been tailored to meet the USAF mission need [15]. FMA-C is being taught to thousands of airmen in an effort to assure critical cyber systems and reduce vulnerabilities. While the structure and content of FMA-C is very similar to STPA-Sec, its application has been tailored to As-Is Information Technology infrastructure. In practice, USAF Mission Defense Teams apply FMA-C on fielded cyber systems to identify mission critical vulnerabilities. The practical application of FMA-C to IT central systems has scoped its focus to this mission need. STPA-Sec, specifically in the tailored approach presented in this paper, enables analysis of a conceptual MWS prior to a design solution. It elevates the analysis to highest level employing systems engineering through systems theory focused on complex interactions in cyber physical systems.

While John Thomas and Nancy Leveson's STPA primer [10] is an excellent resource for instructions on implementing STPA, its chapter for STPA-Sec has not yet been released. Of the STPA-Sec analyses completed to date, most are either simplified for presentation purposes or limited distribution due to sensitive and proprietary system information. This work seeks to provide an academic but relevant and complete example (includes all phases and steps of STPA-Sec) to promote utility and enable greater understanding for future practitioners.

III. TAILORED STPA-SEC APPROACH

This work details a tailored STPA-Sec approach as performed for a high level case study analysis of a notional next generation refueling military aircraft, titled the KC-X.

The data used to conduct this case study is sourced from publically available acquisition documentation prepared by the United States Air Force, USAF, for the next generation tanker, KC-X [16] [17] and supplemented by the authors as required. The documentation is written prior to selecting a contractor or finalizing a design. It includes the following information: First, an overview of the operational scope required with a mapping to joint service requirements is presented. Next, an overview of the intended mission and required capabilities is presented. System activities, required functions, and other needs are listed. Finally draft performance parameters are presented for evaluation of materiel solutions. Mission specific details have been obfuscated and generalized for widest distribution.

A tailored approach of STPA-Sec was first presented in the authors' previous work [18]. This approach is composed of three levels (phases) of analysis: conceptual, architecture, and design as depicted in figure 1. The overall tailored STPA-Sec approach begins at the highest level of abstraction with concept analysis, focused on elaboration of the system purpose and goal. Next, it descends into architecture analysis of the system from a functional level. Finally, potential forms of the system are considered and analyzed in the design analysis steps. A detailed explanation of each phase and step

of this approach is presented along with the case study results and recommendations in sections IV-VI.

STPA-Sec is not intended to be implemented as a checklist approach; it should be iterative both within each phase and in progression to lower levels of detail providing verification and validation. Refinement to previous steps is expected and encouraged throughout the analysis.

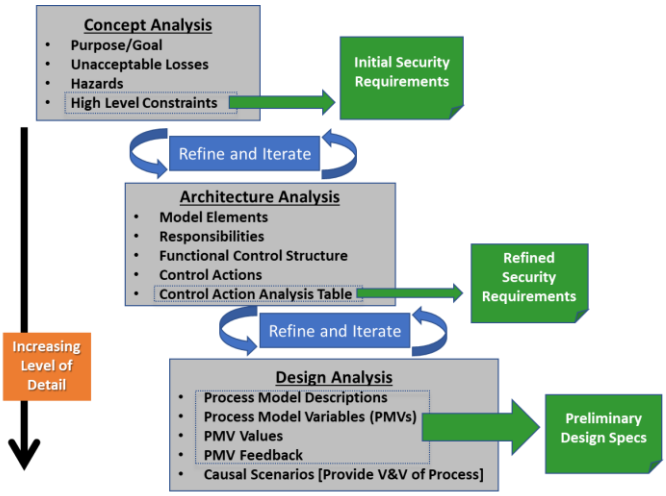


FIGURE 1 TAILORED STPA-SEC OVERVIEW.

IV. CONCEPT ANALYSIS

As shown in Table II, the four conceptually-oriented STPA-Sec steps begin with mission-level analysis to prevent the system from entering hazardous system states that could lead to unacceptable losses and mission failure. Beginning systems security analysis at the mission-level allows security engineers to more accurately understand the stakeholders' needs and maximizes the engineering trade space as system goals are transformed into constraints (i.e., early safety, security, and resiliency requirements).

TABLE II: STPA-SEC CONCEPT ANALYSIS.

Step	Description
1. Define the SoI's purpose and goal	Capture the mission statement and key activities of the system: 1) A system to: (What) 2) By Means of: (How) 3) In Order to: (Why)
2. Identify unacceptable losses	Define high level, intolerable system outcomes to key stakeholders (e.g., loss of life, injury, damage to equipment, reputation, mission, etc.).
3. Identify hazards	Identify system states that when coupled with worst case conditions lead to an unacceptable loss.
4. Develop system security constraints	Develop mission-informed security constraints that prevent the system from entering hazardous states. These constraints are synonymous with early safety, security, and resiliency functional requirements.

1. Purpose and Goal

The first step of STPA-Sec defines the SoI's mission in terms of a purpose and goal from the stakeholders' perspective. Stakeholder involvement is extremely important to an accurate mission statement. For a military environment, the mission commander is the ideal person to provide this input with support from his key staff. STPA-Sec's purpose and goal is very strategic and the exact word choices dictate future analysis steps.

The format of the mission statement is standardized into three parts: 1) A system to 2) by means of 3) in order to. The purpose and goal for the KC-X example are shown in table III. The first phrase "A system to" is meant to capture the primary purpose of the system (i.e., the What) in a few words. For the KC-X, since input directly from the mission commander was not available, the authors distilled pages worth of mission information from other documentation into a short concise statement, provide worldwide aerial refueling. This statement was directly captured from the source documentation 'mission statement', but paired down from the documentation version. It included a specification to refuel both US and coalition aircraft at the strategic, operational, and tactical levels. For an STPA-Sec analysis it is beneficial to capture the core of the mission in as few words as necessary.

The second portion of the purpose and goal, "by means of", identifies the key activities or processes the system uses to achieve its purpose (i.e., the How). This is often the most difficult step as the verbs chosen here become the controlled processes further analyzed in architectural and design analysis. For the KC-X and complex systems in general, narrowing down to a small set of verbs to cover the broad spectrum of key activities is not an easy task. In this case an OV-5 was available, but its organization did not lead to direct plug and chug of mission activity summary tasks. The OV-5 grouped activities by ground and air, so it required reorganization into functional groupings in an effort to roll up the 20 some functions into 3 high-level activities. STPA-Sec does not specify how many key activities is appropriate, but it would be difficult to conduct the analysis with 20 key activities and likely would be very repetitive. Based on a functional grouping of the activities presented in the OV-5, the key activities for the KC-X are: Flying, Refueling, and Mission Planning. These high level, yet simple and practical functional activities rolled up tasks in the OV-5 such as take-off, navigate en-route, and participate in MSN networks into a more useful and paired down set of key activities. There was some debate over mission planning, but it was included since this is a security based analysis. Additionally, in the context of known attack surfaces, we determined the activity of mission planning, which is highly reliant on computer systems and data from external sources, should be included as a key activity for this analysis.

Lastly, the "in order to" identifies the goal, or what mission the system contributes to (i.e., the Why). Capturing

the mission the system supports sounds easy but is often difficult in practice as it is difficult to determine the correct level of mission the system supports. It can be an easy trap to say, 'enables the Air Force mission', or 'supports the warfighter', but these statements lack appropriate specificity. For the KC-X, a down select occurred between: fulfilling the national defense strategy, meeting the quadrennial defense review, achieving the Joint Capability Areas, and prosecuting the USAF's 7 warfighting missions. The decision was made to combine the most relevant of the above in a coherent goal focused on the AF mission decomposed into the specific primary missions enabled through the KC-X system.

Accurately defining the desired system's purpose and goal can be challenging. Best results are produced with involvement from key stakeholders such as mission owner(s), operators, and users. Moreover, correctly defining the mission (or business case) provides a baseline for prioritizing and performing security tradeoffs within an operationally-focused paradigm.

TABLE III: KC-X PURPOSE AND GOAL.

Purpose	A System to	Provide worldwide aerial refueling
Method	By Means of	Flying, Refueling, and Mission Planning
Goal	In order to	Enable the Air Force Mission to meet Joint Capability Areas via refueling and airlift: Force Enable, Force Extend, Force Multiply

2. Unacceptable Losses

The second step of STPA-Sec identifies unacceptable losses. An unacceptable loss is a specific, unacceptable outcome as defined by mission and system owners (i.e., the key stakeholders). Unacceptable losses should be rolled up to the highest level. The system losses should identify what is of utmost value to the stakeholders differentiating from what is nice to have/desired. Unacceptable losses can be mission, personnel, or equipment loss; common unacceptable losses include loss of life and loss of mission essential equipment. However, losses are not just limited to these, loss of reputation or loss of critical data are examples of unacceptable losses that can be addressed through STPA-Sec. To determine unacceptable losses, any outcome that a stakeholder is concerned about should be identified and documented. Then those losses can be re-examined and grouped together or rolled up into the highest level of losses.

Unacceptable Losses

L1: Death or Human injury
L2: Damage to or loss of aircraft
L3: Unable to Complete Primary Mission(s)

For the KC-X example, the losses identified are shown above. In the author's review of other STPA work these were common unacceptable losses shared across other analyses for complex systems [18]. While an STPA or STPA-Sec analysis is not limited to these three losses, the three identified here should be applicable to many complex systems. More specifically, these losses can be generalized such that they are not aircraft or even military specific with slight modifications. L1 is broadly applicable and more generally corresponds to high value asset or operator loss, L2 more generally is the system of interest (SoI) loss, and L3 accounts for functional losses. As a note, since this is a military system, in wartime, L2 may be loosened to allow for a certain number of airframes or a certain amount of damage to become acceptable to ensure the mission can be completed in contended airspace.

Given the importance of unacceptable losses to the mission system and stakeholders, unacceptable losses provide critical information to follow on STPA-Sec steps, resulting in requirements for safety, survivability, and security that are traced back to prevention of these losses.

3. Hazards

The third step identifies hazards that can contribute to cause an unacceptable loss. STPA defines a hazard as a system state (or set of conditions) that together with a worst-case set of environmental conditions will lead to an unacceptable loss [10]. Environmental conditions can be events such as weather but are defined as any condition impacting the system that is outside of the system boundary (conditions the system has no control over).

The hazards identified should be within the system boundary and not themselves an environmental condition or external actor. A hazard for an aircraft is not a mountain or weather because the designer of the aircraft has no control over the weather or the location of a mountain. Instead the hazard may be the aircraft getting too close to the mountain or the aircraft being in an area of bad weather. For an STPA-Sec analysis the hazard is written as violation of altitude/clearance from terrain (H2). The resulting hazard could be exasperated by many environmental conditions: weather, turbulence, improper ATC guidance, loss of navigation, ect. For all these conditions, the hazard as written is still valid and system design choices can be selected to mitigate a larger set of potentially unsafe scenarios (leading to unacceptable losses), thus covering a broad scope of identified and unidentified environmental conditions. An example of a design choice informed by this hazard is purposefully designing a redundant altimeter system informed by multiple sources of altitude, including radar. This design choice would likely prevent the collision with the mountain (unacceptable loss) independent of which environmental condition put the aircraft on a collision course.

For the KC-X example, four hazards are identified in table IV. Specific emphasis was placed on writing them independent of environmental conditions. Hazards 1 and 2 are written to scope broad groups of likely environmental challenges the aircraft will encounter (weather, improper navigation, turbulence, pilot error) into controllable activities that combined with these worst case environmental conditions would likely result in a loss. Hazard 3 is included to capture the hazardous potential of a poorly designed radar warning or equivalent system in an effort to stress the importance of design tradeoffs for system survivability. Hazard 4 could arguably be omitted, but was retained to

emphasize the importance of reliability and security for those systems deemed mission critical (i.e. the flight management system or refueling control subsystem).

TABLE IV: KC-X HAZARDS.

	Hazard to Loss Cross Walk Table	L1 Death or Human injury	L2 Damage to or loss of aircraft	L3 Unable to Complete Mission(s)
H1	Flying to Close too other aircraft/out of position	X	X	X
H2	Violation of Altitude/clearance from terrain	X	X	X
H3	Unable to evade enemy threats	X	X	X
H4	Msn critical systems not functional when required			X

As a general rule, hazards should be abstracted up to the highest level possible and in most cases the list of hazards should be fewer than 10. In practice it is often easier to collect a larger list of hazards and then after review the list can be combined to group similar hazards and roll up others to target a list less than 10. Identifying hazards can also serve to refine and clarify the list of unacceptable losses, as each hazard should be mapped to one or more unacceptable losses. Each hazard can map to one or more losses. But, if a hazard is not mapped to an unacceptable loss then it is either not a hazard or the list of unacceptable losses is incomplete [10].

Controlling hazards is STPA-Sec's conceptual mechanism for delivering system cybersecurity. The mechanism is based on Leveson's STAMP model that associates the high-level unacceptable losses as arising from control deficiencies across the system rather than component failures. The control deficiencies manifest as problems between components (interactions) rather than simple mechanical failures. The latter have been the traditional cause of failures in mechanical systems, but the former provides much more utility when developing contemporary large, software-intense systems. Hazardous states are a necessary precondition to loss. For example, if the unacceptable loss is two aircraft colliding, then the associated hazard could be generalized as failing to maintain safe separation between the aircraft. If safe separation is maintained, the two aircraft should never collide. The need for safe separation is identified through STPA-Sec. Ensuring safe separation is maintained (a control function) throughout operations is an engineering problem and can be addressed through systems engineering (secure systems engineering specifically). There are several different ways the violation might occur. The air traffic control system might be hacked or attacked. One or both of the aircraft might be subjected to a cyber-attack. Likewise, there are any number of mitigations that can be used to address the hazard. However, at this conceptual stage of the system engineering process, the hazardous functionality (safe separation not enforced) is already identified. In other words, loss prevention functionality (safe separation enforcement) is now identified and the remainder of the engineering process can focus on developing a suitable architecture to enforce this functionality. This approach allows engineers to handle safety and security in the same manner that all other emergent system properties are addressed. The systems approach does not preclude the need for reliable components. STPA-Sec still identifies scenarios involving component failure, but it also highlights complex, highly interactive scenarios involving management decisions, operations processes and operators. These other factors are also contributory to many losses. Therefore ensuring safe and secure operations must go beyond a focus on technology.

4. Constraints

The fourth step of the concept analysis phase is developing system security constraints that prevent the SoI from entering one of the previously identified hazardous states. These constraints are restrictions placed on the system (and implemented via the security architecture) to bound operation within acceptable parameters. These constraints are the output of the stakeholder inputs for steps 1-3 as the measurable result of the analysis. These constraints inform early security requirements that are directly traceable to key mission needs through controlling the hazard and preventing its associated unacceptable loss(es). The constraints identified for the KC-X example are shown in table V.

TABLE V: KC-X CONSTRAINTS.

	Constraint	Hazard Mapped to
1	A/C must maintain minimum safe separation distance	H1
2	Must have minimum mission critical safety systems functional to attempt AR	H1
3	A/C must maintain minimum safe altitude limits	H2
4	Must have minimum mission critical safety systems functional for terrain flight	H2
5	Must maintain integrity of mission critical warning and deterrence systems	H3
6	Msn critical systems must be available when required to perform primary msn	H4

In the same way hazards are mapped to losses, each constraint should be mapped to one or more hazard. Conversely each hazard should be mapped to at least one constraint. If a constraint cannot be created for a given hazard, it is likely that hazard is outside the system boundary and thus an environmental condition.

Through our analysis we found most constraints mapped to a single hazard however some hazards had multiple constraints. Constraints can be simple statements re-writing a given hazard in the form of a restriction against operation in the hazardous condition. KC-X constraints 1, 3, and 6 are examples of this. Constraints 2 and 4 are written to restrict activities if mission critical safety systems are not

operational, and encourage redundancy and robustness of design for the functions deemed mission critical.

The first four steps of STPA-Sec begin to specify acceptable and non-acceptable system states which can eventually be formally tested and verified when the architecture is developed. At the completion of this conceptual phase of analysis the list of constraints is high level and not necessarily security specific, applying more broadly to safety, security, and resiliency. As the STPA-Sec analysis is continued into the next phases these can and will be refined to further specify measurable and verifiable requirements for safe and secure system operation. The insights gained through STPA-Sec can also be used to inform and improve early MBSE efforts. The refined safety and security focused constraints provide stakeholder-focused traceability for safety, security, and resiliency requirements which are important for system survivability (a critical issue in U.S. DoD systems [19]).

V. Architectural Analysis

STPA-Sec architectural analysis is a continuation of the conceptual phase and examines the SoI at the functional level (rather than a form specific implementation as is often the case in cybersecurity analyses). Approaching the analysis from a functional rather than physical implementation maintains the largest trade space for potential solutions and helps ensure the desired system functionality can be implemented without unnecessary architectural and design constrictions.

Table VI details the necessary steps to perform STPA-Sec architectural analysis. The majority of this analysis involves the creation of a Functional Control Structure (FCS) representing the SoI. The FCS can be created at various levels of abstraction such that an entire system is represented as a single model, as shown in figure 2. Or the FCS can decomposed into multiple sub-models, figures 4-6, used to more specifically understand the execution of the SoI's key activities. As a reminder, the key activities were defined in the concept analysis phase step 1 as the verbs composing the method, or 'by means of' section of the SoI purpose and goal. STPA uses functional decomposition to thoroughly understand critical relationships between actors and processes represented as Control Actions (CA). After producing an FCS, STPA-Sec enumerates all required CAs followed by an analysis of their criticality, how they contribute to preventing the SoI from entering hazardous states. The output of STPA-Sec architectural analysis identifies potentially hazardous or unsecure CAs for a given system architecture. These CAs help further refine system level security requirements given a system architecture.

TABLE VI: STPA-SEC ARCHITECTURE ANALYSIS.

Step	Description
1. Identify Model Elements	Identify actor(s), controller(s), and controlled process(es) for the SOI at the desired level of abstraction.
2. Identify Each Model Elements Responsibilities	Capture the description and actions planned to be taken for the model elements identified.
3. Draw the FCS	Provide a visual functional-level depiction of the SoI. Depicts the model elements and control relationships between them.
4. Identify Control Actions (CA)	Captures (in verb form) the actions necessary for each element to execute its responsibilities.
5. Complete the Control Action Analysis Table	This table systematically enumerates which hazards are caused by each CA identified in step 4.

1. Model Elements

The first step of this phase begins populating the FCS with model elements. The FCS model elements include actors, controllers, actuators, sensors, and controlled processes. A standard format of an FCS is shown in figure 3. Starting from the bottom of the model, the controlled process(es) is(are) the previously identified (Concept Analysis Step 1) activities the system uses to achieve its purpose. Beginning with these "How" verb activities, the primary task in step 1 of STPA-Sec architectural analysis is identifying the controller and actors responsible for performing the process. Actors are the operators managing the process and providing inputs to the system. The controller is system specific, but often in high level FCS models is merely represented as a computer. For the KC-X example the actors are aircrew and the controller is the flight computer as shown in Figure 2.

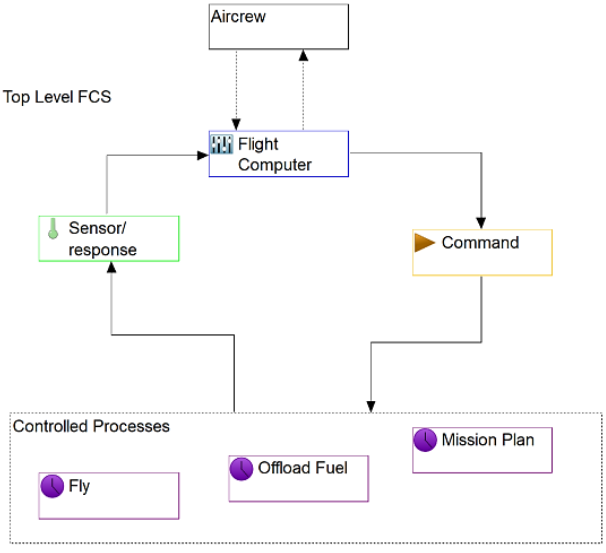


FIGURE 2 KC-X TOP LEVEL FUNCTIONAL CONTROL STRUCTURE.

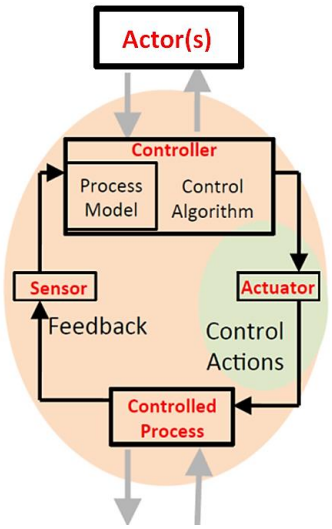


FIGURE 3 STPA BASIC CONTROL STRUCTURE. FROM [10].

2. Responsibilities

Once model elements are identified, then responsibilities for the model elements are populated in step 2. This step captures the actions required of each element for executing the activity or controlled process. These responsibilities can be identified from operational or system documentation, and from particular discussions with users, system SME's, and other stakeholders. Once a list of responsibilities is populated, they are assigned to the appropriate actor for each controlled process (or action).

This step proved to be particularly challenging for the KC-X example likely due to the analysis being completed at the conceptual level. It was difficult to extract responsibilities from source documentation without restricting the results to a physical form implementation. In other words, responsibilities are often generated based on what that actor or controller did in the previous system. Most responsibilities listed were merely duty descriptions of crew members based on previous tanker aircraft operations. Utilization of these responsibilities would restrict the trade space of the future system. For the KC-X, the documentation listed a boom operator as responsible for accomplishing refueling and more specifically providing alignment cues for the receiver aircraft. While a valid responsibility, it assumes the KC-X solution relies on a human for boom operation and navigation cues. For a next generation platform, it is just as likely this task of navigation cues could be automated and performed by a computer system. In summary, STPA practitioners should exercise caution when assigning responsibilities that the solution space is not unnecessarily scoped down from thinking only how the task is currently executed. Identifying model element responsibilities is a key prerequisite activity to defining control actions in step 4 and is often revisited after the control actions are defined. Additionally, the documentation parsing effort in search of responsibilities may inform additional model elements as new actors or controllers may be discovered.

3. Draw the Functional Control Structure

The third step organizes the previously identified model elements into a functional control structure (i.e., a model) by adding control relationships. Step 1 already populated the model elements (boxes) for the FCS. Control relationships (and the FCS as a whole) depict who/what is issuing commands (controller), who/what is executing the commands (actuator), and who/what is providing feedback (sensor). Figure 3 illustrates these basic elements as previously described in step 1 and shows the basic relationships between them. At a high level of analysis organizing and depicting an FCS can seem trivial, but figure 5 depicts the KC-X mission planning example's more complex relationships. Often in complex systems multiple actors can issue commands to the controller, thus additional relationships need to be modeled. In the KC-X mission planning activity, both the pilot and an external mission planning software, i.e. JMPS, have direct inputs to the flight computer. The flight computer may auto-read a mission plan from a cartridge or the pilot may manually enter components of the plan. When decreasing levels of abstraction the FCS will become much more complex. However, the primary objective of creating a FCS is to manage the complexity of the system by accurately depicting key elements and control relationships.

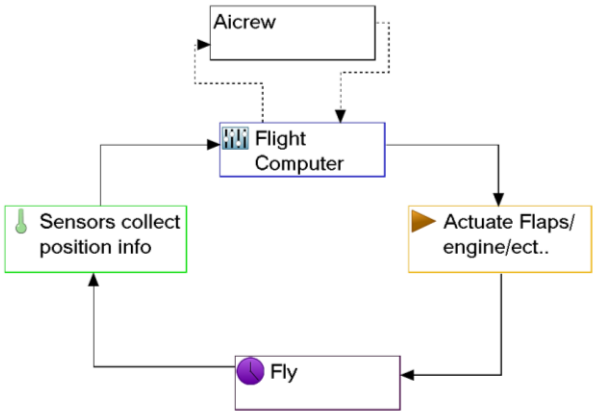


FIGURE 4 KC-X KEY ACTITIVY: FLY FCS

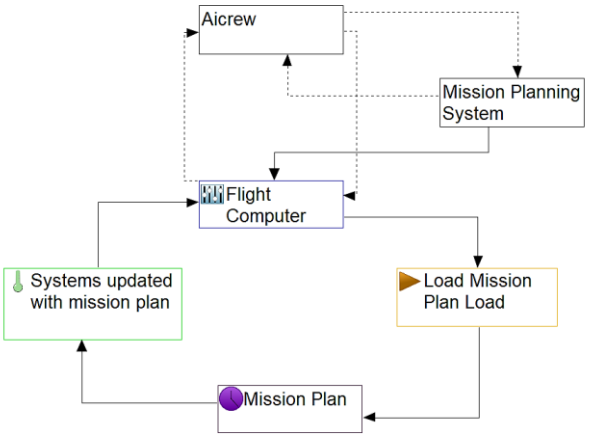


FIGURE 5 KC-X KEY ACTITIVY: MISSION PLAN FCS.

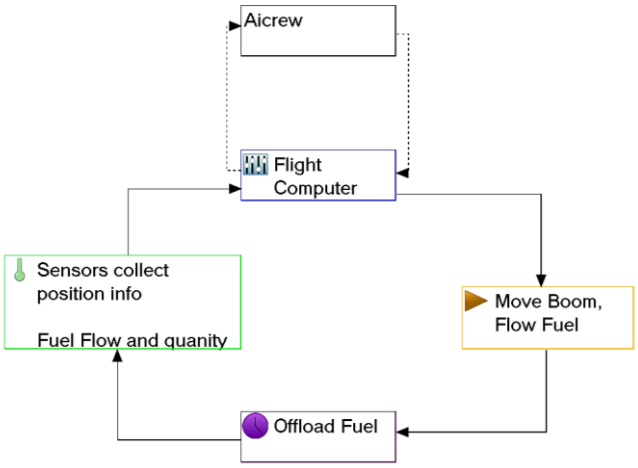


FIGURE 6 KC-X KEY ACTITIVY: OFFLOAD FUEL FCS.

4. Control Actions (CA)

Step 4 identifies CAs for the system. A CA is a terse verb (action) statement capturing the execution of a function (or task) necessary to control the subject process. This step in conjunction with the CA analysis performed in step 5 is some of the most important (and challenging) actions performed in this phase. Populating the list of CAs begins with pairing down the responsibilities previously identified in step 2 into terse action statements that an actor or controller performs to manage the controlled process. STPA-Sec is not intended to be a linear process. The nonlinearity of effort is evidenced for the KC-X while trying to accomplish this step and step 2 (responsibilities). The primary challenge is determining what level of abstraction is appropriate for the CAs as CAs exist at many levels of abstraction. The level of abstraction should be dictated by the overall level of detail for the analysis.

CAs should be identified for each activity, thus each FCS. The CA's identified for the KC-X example are shown in table VII. This step proved to be the most difficult for the authors to conduct for this conceptual analysis. Many potential CA's were considered, but the authors struggled to generate them at the highest level of abstraction.

Specifically for the *fly* key activity, multiple control actions were considered, quickly highlighting the challenge of determining the correct level of abstraction. CA1 and CA2 for example, started from a much larger list of many activities required to fly (in order of decreasing level of abstraction): navigate route, change heading, increase bank angle, maneuver yoke, increase speed, increase throttle position setting. The list of potentially valid control actions for the activity *fly* is extensive. After brainstorming this list, similar activities were aggregated as much as possible and the three activities of Position Maintenance, Velocity Maintenance, and Communicate were chosen to capture the much larger subset of potential control actions. For an analysis of a preliminary system design it may be more appropriate to capture the CAs at a lower level of abstraction and thus a much longer list, but for this example, the highest level of abstraction was chosen. The mission plan FCS had many potential tasks under consideration for the control action. CAs 7 and 8 demonstrate the link between control actions and responsibilities. Responsibilities identified through parsing documentation are present in the description section of the control action list. While not exhaustive of all potential tasks and responsibilities, it provides a good description of efforts executed in the Mission Plan FCS. For this FCS the authors chose to aggregate the list of potential actions into the CAs: Prepare OPS and Distribute OPS.

TABLE VII: KC-X CONTROL ACTIONS.

Control Action	Activity	Performer	Description
1. Position Mx	Fly	Aircrew/ Computer	Adjust position- heading change, takeoff, land, climb, descend. Computer included for autopilot functions
2. Velocity Mx	Fly	Aircrew/ Computer	Change Velocity- accelerate, decelerate, climb, descend. Computer included for autopilot functions
3. Communicate	Fly	Aircrew/ Computer	Radio and digital(i.e. ACARS, IFF) to other A/C , ATC and ground assets. Access and communicate in net centric environment.
4. Precontact	Offload Fuel	Aircrew/ Computer	Instructing both crews on proper position to begin AR. Solution independent to allow for human direction or computer aided position information
5. Contact	Offload Fuel	Aircrew/ Computer	Receiver connected to begin refueling. Solution Independent of human vs. computer to allow automation as desired
6. Breakaway	Offload Fuel	Aircrew/ Computer	Command to disengage either when complete or in case of emergency. Solution Independent of human vs. computer to allow automation as desired
7. Prepare OPS	Mission Plan	Aircrew/ external mission planning system	Reviews mission tasking, intel, and weather. Interacts with external mission planning system to create mission plan file
8. Distribute OPS	Mission Plan	Aircrew/ Computer	Aircrew inserts cartridge into jet, also provides crew briefings and coordination for mission plan. Computer distributes mission plan files to A/C systems

Of significance to a conceptual STPA-Sec is the performer for the CAs. In the KC-X example, specific, sometimes painful, effort was made to remain as solution agnostic as possible. KC-X CAs 4-6 provide an example: in aerial refueling, operators are familiar with precontact, contact, and breakaway commands as issued by the boom operator. Since this analysis is for a future solution, specifying aircrew as the performer would limit the solution

space. It is feasible for a future computer controlled system to provide some of these CAs. Or the future KC-X may implement a hybrid solution with both humans and computers. As much as possible efforts should be made not to restrict the trade space of potential solutions with prior operation biases when conducting a conceptual analysis.

It can be challenging to understand completeness for this step, but a good indicator of a sufficient CA list is when all the activities for a given FCS can be completed with the listed CAs. This becomes evident when executing causal scenarios (discussed in the design analysis phase step 5). If the scenario can be executed on the given FCS with the identified CAs then it's likely the CAs listed are sufficient. This again demonstrates that STPA-Sec is truly nonlinear, and must be iterative.

5. Control Action Analysis Table

The bulk of STPA-Sec architectural analysis results reside in the fifth step – populating a CA analysis table, table VIII. This step requires a thorough analysis of each CA identified in step 4 table VII and enumerates what, if any, hazardous conditions can be created by the system's actions.

Each control action is evaluated across four scenarios. The first scenario asks what happens if the CA is not provided. In many cases this ends up being a hazardous scenario. This is the most likely scenario to result in a hazard as most CAs are designed to be executed for safe, secure, and efficient system operation. For the KC-X, all but CA5 identify potential hazards when a CA is not provided. CA 5s exception is discussed in the following paragraph. CA 1 and 2 provide an easy example of not providing a CA causing a hazard. If position maintenance and velocity maintenance, the two CAs required to execute the flying functions of the aircraft, are not provided and the aircraft is in a critical phase of flight where pilot or computer input is required, a hazardous scenario will result. Since an aircraft cannot execute its mission without performing the flying function, it is not surprising that not providing CA's 1 and 2 could result in H1, H2, and H3.

The second scenario asks what hazards can occur if the CA is provided. At first this scenario may sound counterintuitive as one may question why CAs would exist that are hazardous when performed. KC-X CA 5, Contact, provides an example of a potential hazard when the CA is provided. With a combination of unsafe environmental conditions, most specifically the receiver out of position, it can be hazardous to execute the contact CA. If the CA is issued when the refueler or receiver aircraft is out of position, H1 and a resulting collision would likely occur resulting in a loss.

The third scenario analyzes the result of a CA provided too late, too early, or out of sequence. This scenario is relevant for specifically timed activities where a violation of that timing or sequence is hazardous. For the KC-X CA 6, Breakaway, provides an example of this. Breakaway is issued during refueling when the aircraft enter an unsafe position. If the breakaway CA is provided too late, the hazard of flying too close or out of position (H1) is likely to occur and lead to an unacceptable loss.

The fourth scenario analyzes if the CA is stopped too soon or applied too long. KC-X CA3 provides an illustration of a hazard for this scenario. If communication is stopped too soon and incomplete a hazard could reasonably occur. For example, if radio instructions are provided from the refueler to the receiver to descend and decrease speed, but communication is clipped before the decrease speed command is provided, when the maneuver begins, if only 1 aircraft decreases speed the H1 hazard is likely to occur.

This CA analysis table captures a multitude of potentially hazardous states to aid in further design criteria to increase the robustness of the system against these failure modes. This step provides refinement of early security constraints by informing more specific requirements to bound the execution of the key activities to prevent hazardous scenarios and their associated losses. This data informs an initial "design-to" criteria which is further developed during STPA-Sec design analysis. However, the CA analysis table does not incorporate probabilities of occurrence. Severity is only captured in its identification of which system hazards are likely to be induced in that scenario. Further analysis beyond STPA-Sec could be conducted on the likelihood of the scenarios occurring and the expected specific consequence to assess the risk level and criticality of mitigations.

The downside to the high level CA's identified in Step 4 is they lend themselves to less specific analysis for the CA Analysis Table. It also produced very similar scenario impacts that would likely be more diverse with a lower level set of CAs. The results of this effort accomplished on a lower level set of CAs may provide more actionable hazard insights to inform more specific security requirements and design constraints. However, conclusions are still available from this higher level CA analysis. For the KC-X example, the results of the CA analysis table illuminate the importance of the security and reliability of *fly* CAs 1-3 as 3 out of 4 scenarios result in hazards H1-H3 and ultimately unacceptable losses if omitted or executed improperly.

TABLE VIII: KC-X CONTROL ACTION ANALYSIS TABLE.

CA#	Control Action	Not providing causes Hazard	Providing Causes Hazard	Too Early/too late, wrong order	Stopping too soon/applying too long
EXAMPLE		Not Providing CA-1 is Hazardous if (CONDITIONS) [(Hazards associated) H1, ect]			
1	Position Mx (Aircrew)	Not Providing Position MX is Hazardous if in a critical phase of flight [H1, H2, H3]		Position MX is Hazardous if done too early or too late in a critical phase of flight [H1, H2, H3]	Position MX is Hazardous if stopped to soon or applied to long in a critical phase of flight [H1, H2, H3]
2	Velocity Mx	Not Providing Velocity MX is Hazardous if in a critical phase of flight [H1, H2, H3]		Velocity MX is Hazardous if done too early or too late in a critical phase of flight [H1, H2, H3]	Velocity MX is Hazardous if stopped to soon or applied to long in a critical phase of flight [H1, H2, H3]
3	Communicate	Not Providing Communication is Hazardous if in a critical phase of flight(takeoff, landing, joining refueler) [H1, H3]		Communication too late is Hazardous if in a critical phase of flight(takeoff, landing, joining refueler) [H1, H3]	Communication stopped too soon (clipped transmission) is Hazardous if in a critical phase of flight [H1, H3]
4	Precontact	Not Providing Precontact is Hazardous as a A/C could be out of position and damage equipment [H1,H4]		The wrong sequence for Precontact is Hazardous if in a critical phase of refueling setup [H1,H4]	
5	Contact		Providing Contact is hazardous if attempted during an unsafe position [H1]	Providing Contact out of sequence is hazardous if attempted during an unsafe position [H1]	
6	Breakaway	Not providing Breakaway is hazardous if unsafe position occurs [H1]		Not providing Breakaway on time is hazardous if unsafe position occurs [H1]	
7	Prepare OPS	Not providing Prepare OPS is hazardous in almost all scenarios (no planned route, no deconflicts, no mission plan loaded on systems…) [H1,H2,H3,H4]			
8	Distribute OPS	Not providing Distribute OPS is hazardous in almost all scenarios (no filed flight plan, no crew briefing, no mission plan loaded on systems…) [H1,H2,H3,H4]	Providing Distribute OPS is hazardous when malware or intentionally incorrect information is distributed to systems [H1,H2,H3,H4]		

VI. DESIGN ANALYSIS

The STPA-Sec design analysis phase studies the specifics of a control action using relatively simple process models and scenarios. These process models enumerate the decision logic, key variables, and acceptable variable values associated with each CA in a systematic and straight forward fashion. Additionally, this analysis identifies which feedback mechanism is responsible for those process model variable values (e.g., a sensor or computing mechanism). STPA-Sec design analysis enables a more thorough understanding and specification for CAs which prevent the SoI from entering potentially hazardous and unsecure states. The steps of design analysis are captured in table IX.

This phase was not fully elaborated for the KC-X example. The focus of this effort demonstrates utility to USAF warfighting systems through illustrating all STPA-Sec steps in a case study example. When conducting a conceptual analysis as for the KC-X, the high level of abstraction chosen for CAs does not lend itself to a fully

elaborated design analysis. This section describes and illustrates the steps of design analysis but does not include an analysis of each control action.

TABLE IX: STPA-SEC DESIGN ANALYSIS.

Step	Description
1. Develop Process Model Descriptions	Describes the decision logic ("in plain English") for executing a given CA.
2. Identify Process Model Variables (PMV)	Process Model Variables are measurable indicators of the conditions that trigger a CA.
3. Identify PMV Values	PMV values are all the possible values a PMV can be assigned both acceptable and hazardous.
4. Identify PMV Feedback	Identifies which sensors provide PMV values to the actors and controller for decision making.
5. Carry out Causal Scenarios	Brainstorm how a specific implementation of the system may be compromised. Identifies critical CAs and validates the thoroughness of the model, CAs, and constraints.

1. Process Model

Step 1 of STPA-Sec design analysis develops process model descriptions. Process model descriptions describe the decision logic that defines how and when the controller executes CAs. Each process model should briefly describe the scenario of interest and focus on when to execute a given CA with details identifying the specific system elements and potential responses to CAs. Additionally, the process model should include assumptions about the controlled process.

During this step, it is advantageous to first generate process model descriptions for CAs determined as potentially hazardous from the completed STPA-Sec architectural analysis (step 5 CA analysis table). This approach is recommended as a complex system may have a large (and potentially overwhelming) number of process model descriptions. In accordance with this recommendation, for the KC-X example an abbreviated set of CAs was chosen for design analysis to illustrate the steps.

TABLE X: KC-X PROCESS MODEL DESCRIPTIONS.

Control Action	Key Activity	Process Model Description / Decision Logic
1. Position Mx	Fly	Execute Position Mx during critical phases of flight
2. Velocity Mx	Fly	Execute Velocity Mx during critical phases of flight
6. Breakaway	Refuel	Issue Breakaway when unsafe position

2-4. Process Model Variables, Values, and Feedback

In step 2, Process Model Variables (PMVs) are described as various conditions which indicate a system state. For the KC-X example the Breakaway CA, as shown in table XI, was chosen to illustrate the steps of STPA-Sec design analysis. Since Breakaway is issued when an unsafe position occurs between the refueler and receiver, an appropriate process model variable is separation distance.

The discrete states a SoI could exist in are then enumerated in step 3 as PMV values. It is critical these values are properly understood to specify potentially hazardous systems states. For the KC-X example with separation distance as the process model variable, we sought to establish values that were simple yet enclosed the entire range of values. Rather than choosing a unit of distance, whose range could be infinite, we chose constrained values that informed the action: in bounds, out of bounds, and unknown. For the KC-X Breakaway CA, this limited set of distinguishable PMV values is more useful than an unbounded range of separation distance in feet. These allow for simpler discrete logic commands to be developed, (in bounds- do not issue CA, out of bounds- issue CA, unknown- issue CA).

Step 4 identifies the sensors which are responsible for generating the PMV values (i.e., data) to include conventional sensors, personnel, computer systems, etc. For the KC-X the sensors providing this feedback were an altimeter warning system, proximity warning system, and the aircrew visual cues. This step lends insight into which feedback sensors are critical to monitor for potentially hazardous states and enforce CAs.

TABLE XI: KC-X FULL PROCESS MODEL DESCRIPTION.

CA	Process Model Description	Process Model Variables	Process Model Variable Values	Feedback Information
Breakaway	Issue Breakaway when unsafe position	Separation Distance	In bounds, out of bounds, unknown	Altimeter warning, proximity warning, eyeball

Steps 1-4 produce a list of preliminary design considerations to include detailed process models and PMVs which specify functional logic. This informs subsystem and component implementation as preliminary design specifications.

5. Causal Scenarios

Step 5 of STPA-Sec design analysis is the generation of causal scenarios where the impact of environmental conditions (previously explored during conceptual analysis) are examined to more specifically understand and assess how losses may occur. Akin to tabletop red teaming, causal scenario generation is typically conducted by system experts, well-qualified users, and threat analysts with the goal of identifying plausible scenarios (or conditions combined with effects outside the system boundary) that violate or breach a constraint.

Figure 7 presents a tool for provoking thought in forming causal scenarios for an STPA-Sec analysis.

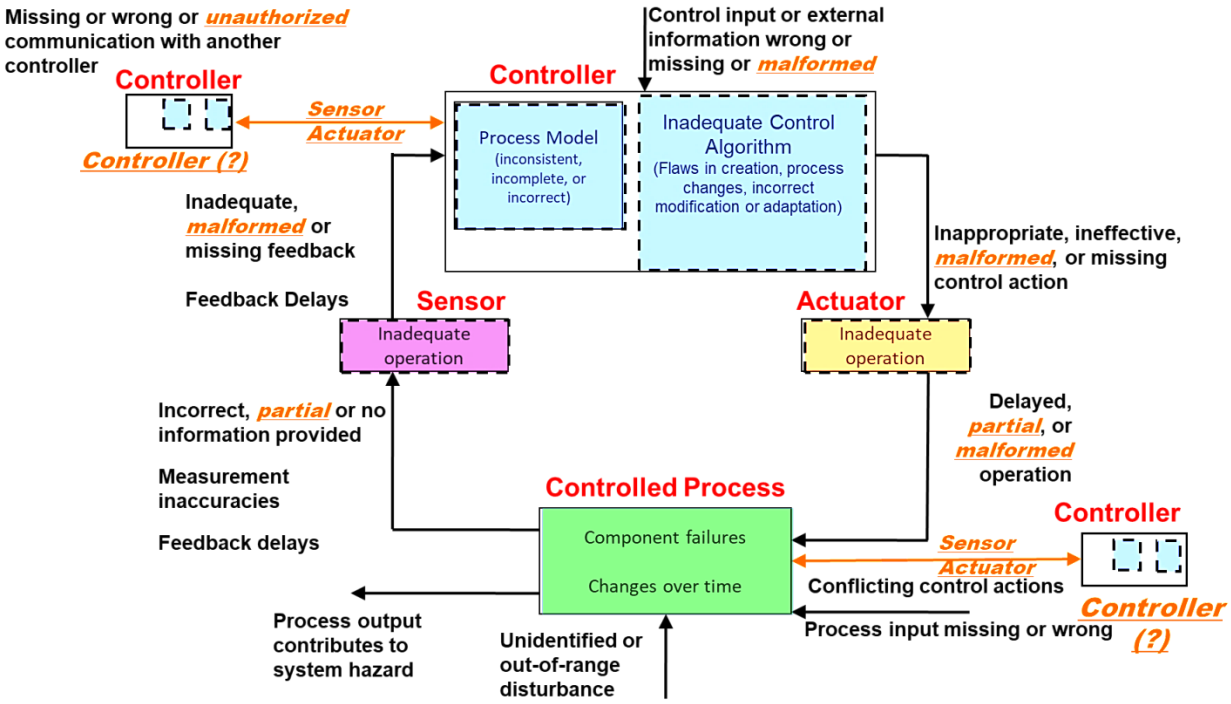


FIGURE 7 STPA-SEC CAUSAL SCENARIOS FROM [20].

For the KC-X example, a causal scenario analysis was generated for the breakaway CA. Breakaway is an excellent example of a well-designed PMV as regardless of the environmental condition thought up in a causal scenario, the system has a robust design to respond. PMV feedback will indicate in bounds, out of bounds, or unknown. No matter the external condition, one of these states is valid and will be indicated. For all scenarios where the feedback assigns a value of out of bounds or unknown a breakaway will be issued. This means that turbulence, improper receiver position, poor refueler maneuvering, engine malfunction, and any other environmental causal scenarios would still not break this process model as designed. Even for the unknown scenarios, ROE's will be in place to attempt to determine the system state and then if still undetermined, issue the breakaway command. This specific example of the breakaway CA was chosen to illustrate a well-designed process model; often causal scenarios will illustrate the potential for an undesired impact and require rework of previous steps to rectify the issue.

In a general sense, this final step serves to provide verification and validation for the thoroughness of the STPA-Sec analysis effort. In this way, changes or additional constraints are often identified as part of the causal scenarios when attempting to 'break' the SoI.

VII. STPA-SEC UTILITY ASSESSMENT

This section presents a subjective assessment of STPA-Sec's utility for complex weapon systems. Table XII provides a summarized assessment for each phase.

TABLE XII: STPA-SEC UTILITY ASSSESSMENT.

	Systems Security Oriented STPA-Sec Phases		
	Concept Analysis	***Architectural Analysis***	***Design Analysis***
Purpose	Determine Security Requirements	Determine Design-To Criteria	Determine Build-To Criteria
Difficulty	Easy	Moderate	Moderate-High
Level of Domain Expertise Req'd	Novice	Advanced	Expert
Level of STPA Expertise Req'd	Low	High	Moderate
Amount of STPA instructional materials available	Numerous	Some	Few
Duration	Hours	Days	Weeks
Number of Steps	4 Steps	5 Steps	5 Steps

The conceptual analysis phase provides the greatest return on time investment. This phase is easy to execute with very little STPA-Sec knowledge. Additionally, the most amount of STPA instructional material and examples are available for this effort. We recommend it is best accomplished in a small working group of key stakeholders. Establishing an agreed upon purpose and goal along with unacceptable losses from the key stakeholders is very powerful for shaping system requirements and enabling traceability. STPA-Sec concept analysis prevents the common pitfall of 'securing the wrong thing'.

STPA-Sec architectural analysis is more challenging and time consuming than conceptual analysis especially when

the system is decomposed to lower functional levels. The level of abstraction chosen for this analysis highly influences the time required. The functional control structure and list of CAs as executed for the KC-X example were kept at a high level, if these were decomposed to lower levels of abstraction, the lists of CAs and the CA analysis table could easily become 5-10 times its current length. While this effort would require more domain expertise, the additional information would inform more detailed design-to criteria while offering the STPA-Sec benefit of traceability to key mission activities and prevention of unacceptable losses. While not as clearly articulated in instructional material, the STPA primer and other slideshow tutorials do adequately address how to perform this architectural analysis. However, with the exception of fictionalized educational examples, the architectural analysis for actual systems is often proprietary. As such, most real system examples found do not share this full analysis. These steps often are the hardest to execute for the STPA-Sec novice, and the lack of fully detailed real world system examples adds to this difficulty.

STPA-Sec design analysis is the most detailed phase of STPA-Sec and can require the most amount of time if each CA is fully elaborated via process models. The practitioner will require a much more robust understanding of the SoI as PMVs, potential PMV values, and the feedback sensors are often more technical than the previous analysis. However, this analysis pays dividends in its ability to provide early design specifications for components. These specifications are not only presented in clear decision logic, but are traceable back to key stakeholder mission goals. This phase was the least detailed for the KC-X example as the authors had limited system knowledge. However, the proof of concept was demonstrated and instructions and recommendations for execution were presented. Very few of the STPA materials found fully detail this phase of execution. While the steps are not as complex to execute as the architectural analysis phase, design analysis requires the greatest domain expertise along with the most amount of time investment for completion. Additionally, the detailed analysis completed in this phase often drives refinements and changes to previous phases specifically through the causal scenario exercise.

VIII. CONCLUSION

This work presents a thorough case study example of an STPA-Sec analysis for a next generation aerial refueling system. This work contributes a detailed explanation of each step and practical tips for STPA-Sec's execution illustrated through the KC-X example. The primary purpose of this work is not to present a breakthrough security analysis of the KC-X, but to provide a consolidated resource to enable the future practitioner to execute an STPA-Sec for their SoI.

This work contributes a tailored approach of STPA-Sec to MWS, specifically USAF aircraft. While this tailored approach is organized into phases and steps for execution, it is not intended to form a checklist for security. Many security methods are executed in this fashion and do not result in highly secure systems. STPA-Sec encourages an iterative analysis where steps are expected to drive changes to previous results in an effort to further refine and specify security requirements.

STPA-Sec demonstrates utility for eliciting, defining, and understanding security and resiliency requirements for advanced cyber-physical systems. Further research could expand this example to incorporate SME's from a system program office to increase the level of detail and evaluate the specific requirements generated. Additionally, alternate examples for other services or system types such as space systems could be analyzed.

DISCLAIMER

The views expressed in this paper are those of the authors and do not reflect the official policy or position of the U. S. Air Force, the Department of Defense, or the U.S. Government.

ACKNOWLEDGMENT

This work was supported by the U. S. Air Force Institute of Technology Center for Cyberspace Research, Wright Patterson Air Force Base, OH.

REFERENCES

[1] Y. Liu, Y. Peng, B. Wang, S. Yao and Z. Liu, "Review on cyber-physical systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 1, pp. 27-40, 2017.

[2] White House, "Remarks by the President on Securing our Nation's Cyber Infrastructure," White House Press, Washington DC, 2009.

[3] K. Baldwin, J. Miller, P. Popick and J. Goodnight, "The United States Department of Defense Revitalization of System Security Engineering Through Program Protection," in *IEEE Systems Conference*, 2012.

[4] United States Congress, "Nation Defense Authorization Act 2016 Section 1647," 25 November 2015. [Online]. Available: https://www.congress.gov/114/plaws/publ92/PLAW-114publ92.pdf. [Accessed 1 June 2017].

[5] Department Of Defense, "DoDI 8500.01 Cybersecurity," 2014.

[6] Department of Defense, "DoDI 8510.01 Risk Management Framework (RMF) for DoD Information Technology (IT)," 2014.

[7] Department of Defense, "Defense Acquisition Guidebook Chapter 9 Program Protection," 5 April 2017. [Online]. Available: https://www.dau.mil/tools/dag/Pages/DAG-Page-Viewer.aspx?source=https://www.dau.mil/guidebooks/Shared%20Documents%20HTML/Chapter%209%20Program%20Protection.aspx. [Accessed 1 June 2017].

[8] Department Of Defense, "DoD Program Manager's

Guidebook for Integrating the Cybersecurity Risk Management Framework (RMF) into the System Acquisition Lifecycle," 30 October 2015. [Online]. Available: https://acc.dau.mil/adl/en-US/721696/file/81323/Cybersecurity%20Guidebook%20v1.10%20signed.pdf. [Accessed 1 June 2017].

[9] W. Young, "System Theoretic Process Analysis for Security," Massachusetts Institute of Technology, Boston, 2014.

[10] N. Leveson and J. Thomas, "An STPA Primer," 9 September 2013. [Online]. Available: http://sunnyday.mit.edu/STPA-Primer-v0.pdf. [Accessed 12 August 2017].

[11] N. Leveson, "A new accident model for engineering safer systems," *Safety science,* vol. 42, no. 4, pp. 237-270, 2004.

[12] Massachusetts Institute of Technology, "MIT Partnership for a Systems Approach to Safety," 27 March 2017. [Online]. Available: http://psas.scripts.mit.edu/home/stamp-workshop-2017/. [Accessed 8 June 2017].

[13] N. Leveson, "Engineering a Safer and More Secure World," MIT, 2011.

[14] W. Young, "A New Approach to Security Analysis Based on Systems Theory," Masssachusets Institute of Technology, Boston, 2014.

[15] Air Force Cyber College, "Top-down Purpose-based Cybersecurity," 2015. [Online]. Available: https://www.sans.org/summit-archives/file/summit-archive-1492176717.pdf. [Accessed 01 January 2018].

[16] US Air Force, "KC-X Statement of Objectivies (SOO)," 2010.

[17] US Air Force, "System Requirements Document (SRD) for the KC-X," 2010.

[18] Massachusetts Institute of Technology, "MIT Partnership for a Systems Approach to Safety," 1 January 2017. [Online]. Available: https://psas.scripts.mit.edu/home/. [Accessed 12 July 2017].

[19] Defense Acquisition University, "System survivability key performance parameter," 23 May 2017. [Online]. Available: https://dap.dau.mil/acquipedia/Pages/ArticleDetails.aspx?aid=626ace5f-fc1f-4638-9321-fe4345451558. [Accessed 1 June 2017].

[20] W. Young and R. Porada, "System-Theoretic Process Analysis for Security (STPA-SEC): Cyber Security and STPA," 27 March 2017. [Online]. Available: http://psas.scripts.mit.edu/home/wp-content/uploads/2017/04/STAMP_2017_STPA_SEC_TUTORIAL_as-presented.pdf. [Accessed 21 January 2018].

MARTIN "TRAE" SPAN III is a Systems Engineering Masters Student at the Air Force Institute of Technology (AFIT), Wright-Patterson AFB, Ohio USA. He commissioned in 2012 after graduating from the United States Air Force Academy. As a Captain in the United States Air Force (USAF), he serves as a developmental engineer and holds department of defense certifications in systems engineering science and technology management, test & evaluation, and program management. He has served the USAF as a developmental test engineer responsible for planning and executing complex weapon system test and evaluation. He is a member of IEEE and the Tau Beta Pi honor society. Capt. Span's research interests include systems engineering and systems security engineering. He can be contacted at: martin.span.1@us.af.mil

LOGAN O. MAILLOUX (BS 2002, MS 2008, Ph.D. 2015) is an Assistant Professor at the Air Force Institute of Technology (AFIT), Wright-Patterson AFB, Ohio USA. He is commissioned as Lieutenant Colonel in the United States Air Force (USAF) and serves as a computer developmental engineer. He is a Certified Information System Security Professional (CISSP), Certified Systems Engineering Professional (CSEP), and holds department of defense certifications in cyberspace operations, systems engineering science and technology management, test & evaluation, and program management. He is a member of IEEE, INCOSE, and ITEA professional societies, as well as, HKN and TBP honor societies. He has served the USAF as a cyberspace operations expert responsible for planning and executing network defense exercises, documenting and training computer security best practices, performing test and evaluation of enterprise resource planning solutions, and maintaining distributed simulation infrastructure. Lt Col Mailloux's research interests include systems security engineering, quantum key distribution, cyber-physical systems, and complex information systems. He can be contacted at: Logan.Mailloux@afit.edu.

ROBERT F. MILLS is a Professor of Electrical Engineering in the Department of Electrical and Computer Engineering, Air Force Institute of Technology (AFIT), Wright-Patterson AFB OH. His teaching and research areas include systems engineering, digital avionics, cyber warfare, electronic warfare, computer networks and digital communications systems. He is a member of Eta Kappa Nu and Tau Beta Pi, and is a Senior Member of the IEEE.

WILLIAM "BILL" YOUNG JR is a Colonel in the USAF and currently commands the 53rd Electronic Warfare Group at Eglin AFB. He earned his PhD from the Engineering Systems Division at Massachusetts Institute of Technology's School of Engineering. He commissioned in 1991 after graduating from the United States Air Force Academy. He earned his wings from Specialized Undergraduate Navigator Training (SUNT). He is a Distinguished Graduate of the US Air Force Weapons School and is a 2006 graduate of the USAF School of Advanced Air and Space Studies (SAASS). He is an Instructor Electronic Warfare Officer with more than 2,400 flying hours in the EA-6B and B-52, including 240 combat hours during Operation ENDURING FREEDOM. In addition to a PhD in Engineering Systems, he possesses a Bachelor of Science degree in Engineering Science and four Masters Degrees.

V. Conclusions and Recommendations

Conclusions of Research

This work is written to advance the specialty discipline of system security engineering and, specifically targets practitioners within the DoD and supporting contractors. The first half of this work, presented in the two publications for chapters II and III provides a readily understandable summary of current cybersecurity architectural analysis approaches, and introduces a tailored approach of STPA-Sec which differentiates itself from other approaches in its use of systems engineering to conduct security analysis at a functional and conceptual level. The second half of this work presented in chapter IV as a journal paper titled, "Conceptual Systems Security Analysis with Aerial Refueling Case Study" thoroughly describes conceptual STPA-Sec analysis through a case study with practical recommendations which enable systems engineers to conduct conceptual analyses for future complex warfighting systems.

This section provides conclusions of this effort and demonstrates completeness through summaries of answers to the six research questions addressed in this thesis. Additionally, the significance of this research is highlighted and recommendations for future research are presented.

1. What is Cybersecurity Architectural Analysis?

Originally this question was not included as a research question since the definition was assumed to be available as a part of the literature survey in support of this work. However, while conducting the literature review, frustratingly absent from published literature was a description of architecture analysis for security. Definitions of

system architecture are available; there are very few definitions for architectural analysis, and none were in context of a security analysis. It became apparent that proposing a definition of cybersecurity architectural analysis was in fact a research contribution. In addition to the proposed definition being tailored to the security domain, it more importantly highlights an emphasis on functional level analysis rather focusing solely on form common across the few other descriptions available of architectural analysis activities. Cybersecurity architectural analysis is "the activity of discovering and evaluating the function and form of a system to facilitate cybersecurity decisions."

2. *What methods exist for conducting Cybersecurity Architectural Analysis?*

Chapter II contributes the primary literature review for this work, and presents a consolidated introduction to complex cyber-physical system architectural analysis approaches. Chapter II Section III surveys ten approaches relevant for conducting cybersecurity architectural-level analysis on complex systems. While literature is available on the approaches surveyed, no seminal work on architecture analysis for cyber, and none specifically relevant to complex weapon systems, was discovered. Significant time and effort was invested to explore which approaches are being used in the USAF and DoD followed by the collection and review of available documentation. This included attending training courses, conferences, and reviewing published analysis results.

3. *What are the key characteristics for Cybersecurity Architectural Analysis and how do they map to current approaches for complex cyber-physical systems?*

In Chapter II Section IV distinguishing characteristics for cybersecurity architectural analysis approaches are presented. These characteristics include: bottom-up

vs top down approach, threat driven analysis, vulnerability based analysis, MBSE enabled (used an integrated architecture), and if a software tool was required. These characteristics are determined through detailed literature review, proposal of the definition, and the review of each approach's published documentation, instructional materials, and system examples available. At the end of Section IV, Table I presents a consolidated mapping of approaches to characteristics as a simplified resource to assist the security practitioner in understanding the surveyed approaches.

TABLE I: ARCHITECTURAL APPROACHES TO CHARACTERISTICS MAPPING.

	Top Down	**Bottom Up**	**Threat Driven**	**Vul. Based**	**MBSE Integrated**	**MBSE Executable**	**Tool Based**
DoDAF + Richards	X[1]			X	X	X[4]	X
CRAF	X[1]		X	X	X	X	X
UAF Security	X			X	X	X[4]	X
ACVAM		X	X	X			
STPA-Sec	X[2]			X			
RMF		X[5]	X	X	X[3]		

1. Promotes a top-down approach after mission functions are identified (i.e., does not include mission thread analysis).
2. Approach begins at a higher level than other approaches examined (i.e., includes mission thread analysis) and includes lower level analysis.
3. Suggests using MBSE, but not required and often not considered.
4. Would require pairing with additional modeling & simulation plugin.
5. RMF is intended to be a top-down approach but is often applied bottom-up using security control compliance based on system type.

4. How can STPA-Sec be tailored to enable the development of security requirements and design criteria?

This question is answered first in Chapter III Section V with the description of a tailored three phase STPA-Sec approach. Of note, this work is not taking credit for the development of STPA-Sec, as it was developed by Dr. William Young [1], but re-organizes the steps and separated the overall execution into conceptual, architectural, and design phases to increase the utility and usability of STPA-Sec and align work with

the ISO/IEC/IEEE 15288 Systems Engineering Processes [2]. Chapter III section III further describes this tailored approach and provides an illustrative graphic in Figure 1, reproduced here. Sections IV-VI demonstrate this approach for STPA-Sec providing examples of each step in a thorough KC-X case study.

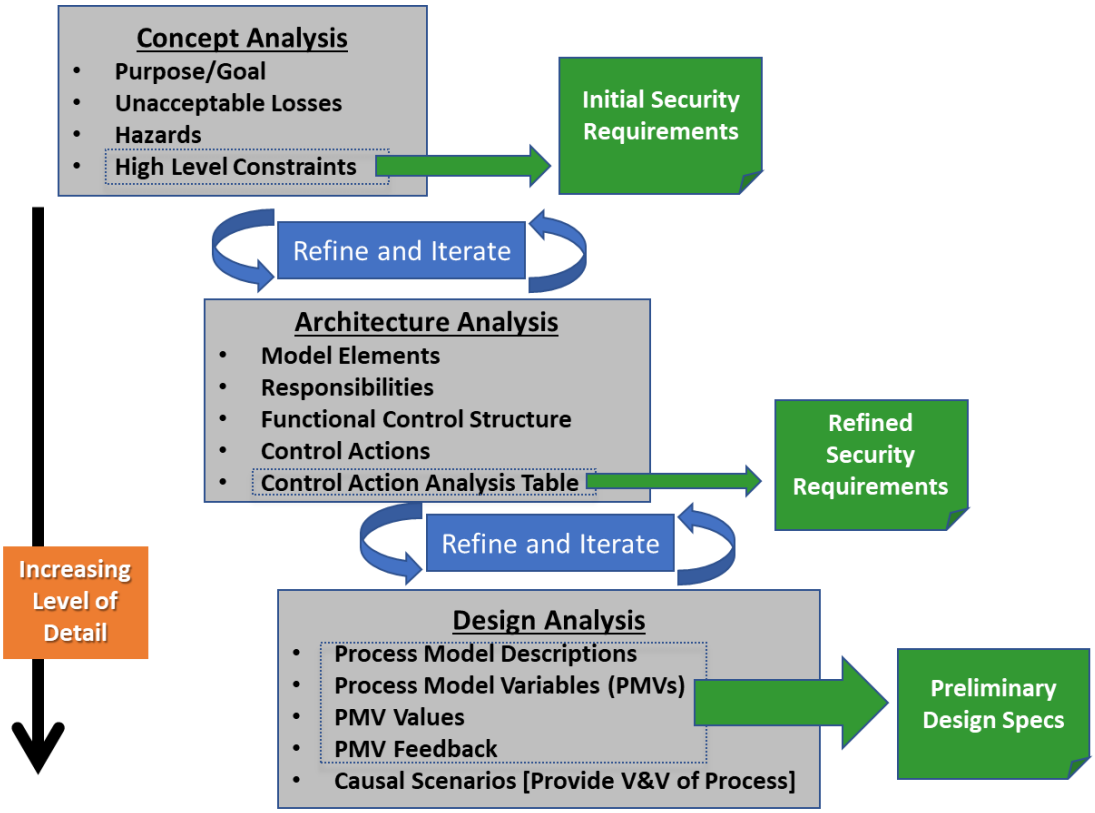


FIGURE 1 TAILORED STPA-SEC OVERVIEW.

5. How executable is STPA-Sec for USAF warfighting Systems?

This question is answered in Chapter IV through the case study of a next generation aerial refueling platform, KC-X. Sections IV-VI detail each phase and step of STPA-Sec as applied to the KC-X case study. Section VII provides an assessment of

STPA-Sec's utility specifically in Table XII and its narrative. While certain steps are identified as challenging, the results of the KC-X case study demonstrate STPA-Sec is scalable, readily executable, and useful for analysis of USAF warfighting systems.

TABLE XII: STPA-SEC UTILITY ASSSESSMENT.

	Systems Security Oriented STPA-Sec Phases		
	Concept Analysis	Architectural Analysis	Design Analysis
Purpose	Determine Security Requirements	Determine Design-To Criteria	Determine Build-To Criteria
Difficulty	Easy	Moderate	Moderate-High
Level of Domain Expertise Req'd	Novice	Advanced	Expert
Level of STPA Expertise Req'd	Low	High	Moderate
Amount of STPA instructional materials available	Numerous	Some	Few
Duration	Hours	Days	Weeks
Number of Steps	4 Steps	5 Steps	5 Steps

6. What recommendations can be made to increase the utility and ease the use of STPA-Sec?

This question is answered throughout the details of the case study presented in chapter III. The recommendations presented throughout this work attempt to describe and detail STPA-Sec to potential practitioners who are largely unfamiliar with the approach, offering tips and recommendations for its implementation. Recommendations and specific execution tips for the practitioner are presented in the description of the steps throughout the KC-X case study. For example, Chapter IV Section V step four provides recommendations for determining the appropriate level of abstraction for control actions in a SoI.

Significance of Research

With the increasing reliance on technology in warfighting systems, specifically complex aircraft, in conjunction with recent published vulnerabilities of cyber-physical systems [3], cybersecurity is of critical importance. The security problem is no longer limited to IT networks; as such, approaches developed for identifying computer network vulnerabilities have fallen short at securing complex cyber-physical systems against intelligent adversaries. Cybersecurity analysis approaches are needed to identify potential vulnerabilities and understand and define security requirements that can be designed to and formally verified. Moreover, the next major armed conflict is likely to have a significant cyber component [4].

The need for MWS built to operate in a highly contested cyberspace environment is understood at the highest levels of U. S. leadership with mandates and funding appropriated in the National Defense Authorization Act of 2016 Section 1647 [5]. Within the U. S. Air Force, the Cyber Resiliency Office for Weapons Systems (CROWS) office was specifically stood up to address these larger cyber concerns for both fielded and new weapons systems through the Air Force Cyber Campaign Plan (CCP). This thesis directly contributes to the CROWS and CCP by assisting in the training of a cyber savvy acquisition force, LOA 3.

Lastly, this work provides a widely distributable STPA-Sec case study specific to a USAF aircraft. Perhaps more importantly, it provides a consolidated and distilled reference with recommendations to enable practitioners to execute conceptual systems security analysis.

Recommendations for Future Research

Future research may continue to expand the detail provided for the KC-X case study. This work used the case study as an example to support the detailed description of STPA-Sec, analysis of its utility, and to provide recommendations for the future practitioner. While the STPA-Sec analysis was largely informed and guided by its founder, Dr. William Young, and a few other knowledgeable practitioners, it provides a limited scope of detailed analysis.

Additionally, while key stakeholder inputs should be already incorporated in the source documentation used for the case study, an ideal STPA-Sec is best executed through a working group session with the mission commanders and key SMEs. The absence of this working group does not invalidate the results and certainly does not detract from the case study's illustrations of the STPA-Sec process, but it does leave opportunity for future work. Specific to the KC-X example, the analysis in the architecture and design phases could be extended further by decomposing the key activities and control actions into lower, more detailed levels. Additionally, further process model analysis could be completed along with more causal scenarios.

The final recommendation for future research is working to integrate MBSE into the cybersecurity architectural analysis efforts. This is an intended area for the UAF to address, but more research and examples will need to be completed to realize its utility for complex airborne warfighting systems. This research area is highly recommended for future students as it could combine the effectiveness of conceptual analysis methods, like STPA-Sec, with the benefits of integrated architecture and executable models.

Bibliography

[1] W. Young and R. Porad, "System-Theoretic Process Analysis for Security (STPA-SEC): Cyber Security and STPA," 27 March 2017. [Online]. Available: http://psas. scripts. mit. edu/home/wp-content/uploads/2017/04/STAMP_2017_STPA_SEC_TUTORIAL_as-presented. pdf. [Accessed 21 January 2018].

[2] ISO/IEC/IEEE, "Systems and software engineering — System life cycle processes, Third Edition," Geneva, Switzerland, 2015.

[3] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham and S. Savage, "Comprehensive Experimental Analyses of Automotive Attack Surfaces," USENIX Security Symposium, 2011.

[4] P. Singer and A. Friedman, Cybersecurity and Cyberwar, New York: Oxford, 2014.

[5] United States Congress, "Nation Defense Authorization Act 2016 Section 1647," 25 November 2015. [Online]. Available: https://www. congress. gov/114/plaws/publ92/PLAW-114publ92. pdf. [Accessed 1 June 2017].

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 074-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to an penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
22-03-2018	Master's Thesis	August 2016 – March 2018

TITLE AND SUBTITLE	**5a. CONTRACT NUMBER**
Conceptual Systems Security Analysis Aerial Refueling Case Study	**5b. GRANT NUMBER**
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	**5d. PROJECT NUMBER**
Span, Martin III, Captain, USAF	**5e. TASK NUMBER**
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)	**8. PERFORMING ORGANIZATION REPORT NUMBER**
Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/ENV) 2950 Hobson Way, Building 640 WPAFB OH 45433-7765	AFIT-ENV-MS-18-M-237
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSOR/MONITOR'S ACRONYM(S)**
Air Force Institute of Technology Center for Cyberspace Research 2950 Hobson Way, Building 640 WPAFB OH 45433-7765	AFIT/CCR
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

DISTRUBTION STATEMENT A. APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

This material is declared a work of the U. S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT

In today's highly interconnected and technology reliant environment, systems security is rapidly growing in importance to complex systems such as automobiles, airplanes, and defense-oriented weapon systems. While systems security analysis approaches are critical to improving the security of these advanced cyber-physical systems-of-systems, such approaches are often poorly understood and applied in ad hoc fashion. To address these gaps, first a study of key architectural analysis concepts and definitions is provided with an assessment of their applicability towards complex cyber-physical systems. From this initial work, a definition of cybersecurity architectural analysis for cyber-physical systems is proposed. Next, the System Theory Theoretic Process Analysis approach for Security (STPA Sec) is tailored and presented in three phases which support the development of conceptual-level security requirements, applicable design-level criteria, and architectural-level security specifications.

This work uniquely presents a detailed case study of a conceptual-level systems security analysis of a notional aerial refueling system based on the tailored STPA-Sec approach. This work is critically important for advancing the science of systems security engineering by providing a standardized approach for understanding security, safety, and resiliency requirements in complex systems with traceability and testability.

15. SUBJECT TERMS

Cybersecurity; Systems Security Engineering; STPA-Sec; Security Engineering; Systems Engineering

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Logan O. Mailloux, Lt. Col, AFIT/ENV
U	**U**	**U**	**UU**	65	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-6565, ext 3348 (logan. mailloux@afit. edu)

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18