2019-01-01

Incorporating Cybersecurity Engineering Within The Discipline Of Systems Engineering

Chuck Easttom
University of Texas at El Paso, chuckeasttom@gmail.com

INCORPORATING CYBERSECURITY ENGINEERING WITHIN THE DISCIPLINE OF SYSTEMS ENGINEERING

by

WILLIAM CHARLES EASTTOM II

Master of Science in Systems Engineering

APPROVED:

Jose F. Espiritu, Ph.D., Chair

Amit Lopes, Ph.D.

Peter Golding, Ph.D

Stephen Crites, Ph.D.
Dean of the Graduate School

INCORPORATING CYBERSECURITY ENGINEERING WITHIN THE DISCIPLINE OF SYSTEMS ENGINEERING

by

WILLIAM CHARLES EASTTOM II (CHUCK EASTTOM), M.Ed., MBA, D.Sc.

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree

MASTER OF SCIENCE

Department of Industrial, Manufacturing and Systems Engineering

THE UNIVERSITY OF TEXAS AT EL PASO

May 2019

TABLE OF CONTENTS

TABLE OF TABLES

TABLE OF FIGURES

CHAPTER 1: INTRODUCTION

1.1 Introduction

The primary subject addressed in this study is the establishment of a definition for "Cybersecurity Engineering" (CSE) It is proposed that CSE) is appropriately well-defined as a sub-discipline of Systems Engineering (SE). Adopting this proposal has three-fold important consequences:

(1) The first is the matter of defining CSE itself. Data is provided supporting the conclusion that CSE is best considered to be a sub-domain of SE. CSE is recognized to involve specific engineering processes, which supports the proposed definition for CSE;

(2) The second centers upon establishing a cybersecurity curriculum: both definition and subject matter content. Through combining data from existing curriculums along with established cybersecurity frameworks, a well-defined curriculum for cybersecurity engineering is formulated.

(3) The third involves defining a current modeling language specific to CSE.

The first two subjects are addressed within Chapter 2 of this thesis, while the third and final consequence is presented within Chapter 3.

1.2 Defining the Fields of Cybersecurity

Cybersecurity is a technical field that is growing very rapidly in complexity and importance (Easttom, 2016; Kim & Solomon, 2016). This growth includes concurrent growth in sub-disciplines such as Digital Forensics and Penetration Testing. While this rapid expansion has been beneficial to expanding knowledge and functionality of CSE, concurrently advancing the career prospects of cybersecurity practitioners, it has presented challenges for creating a functional framework for the teaching of

cybersecurity (Burley, et al., 2018) and attaining a prescriptive and accepted definition of the profession of I am proposing to be CSE.

A significant accompanying challenge is defining professional roles and position descriptions and requirements within cybersecurity (Burley, Eisenberg, & Goodman, 2014). It is suggested that we are not even clear on where in the academic pantheon the discipline of cybersecurity belongs. Recently, it has been reported that there are widescale approaches to cybersecurity practices and cybersecurity education (McGettrick, Cassel, Dark, Hawthorne, & Impagliazzo, 2014). This parallels the creation of other new disciplines, such as entrepreneurship, or information systems, and computer sciences. In some cases, cybersecurity curriculums are taught and practiced within a business management discipline framework, with a focus on policies and procedures. Thus, by contrast, in other instances, it is approached as a Computer Science sub-discipline. This disparity in even defining cybersecurity is a significant problem for both practitioners and academia and those in decision making sciences.

One result of this lack of a coherent definition of Cybersecurity is the wide range of technical backgrounds and skillsets for practitioners (Furnell, Fischer, & Finch, 2017). There are cybersecurity professionals with a strong background in Computer Science or a field of Engineering, and others with virtually no technical background included (Conklin, Cline, & Roosa, 2014). This ambiguity leads to an ill-defined or incoherent axiomatic definition of cybersecurity . Some people approach it being as a management construct, primarily focused on the formulation and implementation of appropriate security standards and policies within a business enterprise. In such an approach,

technical skills relegated to being a secondary (or even tertiary) concern, and only relevant to the implementation of security standards and policies is emphasized.

A different approach to cybersecurity – viewed from the Systems Engineering perspective – is to view it as a highly technical discipline (Craigen, Diakun-Thibault, & Purse, 2014). In considering this view, policies and procedures are still a part of cybersecurity, but they remain ancillary to technical skills. This approach focuses on the technical aspects of cybersecurity, and the technical skillset of practitioners. In this view, cybersecurity practitioners are likely to have a strong Computer Science background, requiring a degree in computer science or a similarly related demanding technical discipline (electrical and computer engineering fields with emphasis on computer systems, for example). This thesis embraces the technical perspective and viewpoint of cybersecurity but additionally provides more specificity and refinement to that definition.

This disparity of approaches to cybersecurity has a significant impact on cybersecurity curriculum. It is unclear what precisely is the curriculum for a cybersecurity degree. This is markedly different from other related fields. For example, it is immediately clear what is included in a management information systems degree as compared to a computer science degree, or a computer engineering degree. The standardization of essential curriculum is beneficial to both students selecting a degree and to employers hiring a graduate of a particular degree program. Thus, standardizing cybersecurity as both a profession and as an academic discipline is a critical task.

1.3 Cybersecurity Education

The problem is to find a clear definition of the field of cybersecurity. Unlike many other fields, there is ambiguity in defining cybersecurity and this ambiguity impacts

university cybersecurity curriculum. There exists significant disparity in the specific curriculum that is taught under the auspices of cybersecurity. There is a need to clearly define cybersecurity and to outline the appropriate curriculum guidelines for this discipline.

In order to formulate a response to the current issues in cybersecurity education, it is first necessary to fully understand the current state of cybersecurity education. This understanding can be facilitated by examining the curriculum in a sample of cybersecurity academic programs. To illustrate the disparity in content for cybersecurity degree programs, table 1 summarizes the content of several Master of Cybersecurity programs.

Table 1 Overview of Cybersecurity Graduate Programs

University	Program Description
Capitol Technology University	Viewed as an engineering discipline with courses designated as information assurance engineering. Curriculum focused on technical aspects.
University of Tulsa	Viewed as a mixture of technical and business disciplines. Some technical courses, along with management and legal courses.
John Hopkins	Very technical including significant mathematics and computer science courses.
University of Texas at Dallas	This program is actually a Master of Business Administration in cybersecurity. Thus, the curriculum is very business oriented, and not as technically focused.
Florida Institute of Technology	This program is actually a Master of Business Administration in cybersecurity. Thus, the curriculum is very business oriented, and not as technically focused.
University of Maryland	This program focuses on the requirements of specific industry certifications. Thus, it has a moderate amount of technical rigor.

The data in table 1 illustrates a problem that permeates cybersecurity programs. It is not even clear what is included in a cybersecurity degree. Even those programs that have the term engineering in the degree name, may not reflect an actual engineering curriculum. This is a departure from traditional engineering disciplines such as electrical engineering, mechanical engineering, and systems engineering, wherein the curriculum is relatively consistent across universities and is defined by ABET (Accreditation Board for Engineering and Technology). Table 2 summarizes programs that are specifically master's degrees in cybersecurity engineering.

Table 2 Cybersecurity engineering programs

University	Program Description
University of Southern California	This program has a technical focus, but there are no courses in mathematics or engineering (i.e. requirements engineering, systems engineering, etc.) as part of the curriculum.
University of Maryland	This program has a technical focus, but there are no courses in mathematics or engineering (i.e. requirements engineering, systems engineering, etc.) as part of the curriculum. There are also no mathematics or engineering requirements for entry into the program.
Texas A&M College Station	This program does require an undergraduate degree in engineering or computer science for admission. Technical course focus, but without specific courses in engineering (i.e. requirements engineering, systems engineering, etc.)
Embry Riddle	This program does require an undergraduate degree in engineering or computer science for admission. The curriculum includes a software engineering course.

The data in table 1 and table 2 illustrate the wide range of curriculum in a cybersecurity program, even in a cybersecurity engineering program. This poses a dilemma for both students and employers. It is difficult for a student to determine which program is appropriate for his or her goals. This is substantially different than other areas of academia. As was previously mentioned, it is well understood that a management information systems degree is business oriented, whereas a computer science degree will be more technical, and a computer engineering degree will have specific engineering courses. A student can easily determine which degree is appropriate for his or her career goals. Employers can also view the degree name alone, without additional details about the program, and have a reasonable expectation of what skills were learned. However, this is not the case with cybersecurity.

Even if one embraces the term cybersecurity engineering, this does not actually solve the problem of establishing what the curriculum entails. As table 2 illustrated, there is significant disparity in cybersecurity engineering curriculum. This is markedly different from other engineering disciplines that are well defined. There is a need for more than a terminology change, there needs to be a well-defined pedagogical approach. There is a need for a formal definition of cybersecurity engineering, both as a profession and as an academic endeavor. One focus of this current paper is in defining cybersecurity engineering as a specific discipline. As will be explored in later sections, it is proposed that cybersecurity engineering be specifically viewed as a subset of systems engineering.

Ambiguity in defining cybersecurity as a discipline or as an academic domain has a deleterious effect on both the teaching and practice of the profession (Beuran, Chinen, Tan, & Shinoda, 2016; Burley, Eisenberg, & Goodman, 2014). Establishing a clear

definition, along with specific curriculum guidance is essential for cybersecurity to mature as a profession. The nature of cybersecurity is such that it is appropriately viewed as an engineering discipline. More specifically, cybersecurity can be effectively treated as a sub-discipline of systems engineering.

In order to facilitate the establishment of cybersecurity as a formal engineering discipline, several activities must be accomplished. The first is to review the nature of engineering in general, and systems engineering in particular, in order to establish what a well-defined foundation for a cybersecurity engineering discipline. This includes applying common engineering principles and techniques to cybersecurity. Secondly, current cyber security curriculum needs to be reviewed and modified to create a cybersecurity engineering curriculum framework. That framework must be clear and applicable to any university. Finally, cybersecurity specific engineering methods and techniques need to be developed.

1.4 Literature review

Conklin, White, Cothren, Davis, & Williams (2015) describe the fact that it is difficult to define cybersecurity. One of the primary points that is made early in their book is the difficulty in defining computer security and related terms such as information assurance. The author's do continue on to describe NIST standards, as well as security models. However, these models don’t define cybersecurity. Rather they simply provide specific recommendations for specific issues. Their book identifies a gap in the current literature.

Craigen, Diakun-Thibault, & Purse (2014) also discussed the problem with defining what cybersecurity is. Their study reviewed a wide range of literature and determined that the term cybersecurity is used very broadly with a great deal of variableness in

definitions. They determined that cybersecurity is an inherently interdisciplinary field. The authors proposed the following definition for cybersecurity

"Cybersecurity is the organization and collection of resources, processes, and structures used to protect cyberspace and cyberspace-enabled systems from occurrences that misalign de jure from de facto property rights."

This definition does not appear to have any advantage over previous definitions. It does not provide any guidance on the actual practice of cybersecurity. Furthermore, even if one completely adopts this definition, it provides no insight into what should be included in cybersecurity curriculum. The study by Craigen, Diakun-Thibault, & Purse (2014) further illustrated the confusion in the cybersecurity field over such a fundamental task as defining what the field is.

While a definition of cybersecurity may have not yet been formalized, the field of cybersecurity is replete with guidelines and standards regarding specific security issues (Conklin, White, Cothren, Davis, & Williams, 2015). These mitigation guidelines are effective for their intended purpose, which is to provide specific guidance in mitigating specific security concerns. However, these mitigation guidelines do not provide an overall definition for the field. And they do not provide any pedagogical guidance for the curriculum required to train students to enter this profession.

A fundamental step in understanding the deficiencies in cybersecurity education is to review the current state of cybersecurity education and training. This includes both training in the context of an academic degree, as well as corporate and industrial training. A thorough understanding of the current state of cybersecurity education will facilitate an exploration of mechanisms to correct those deficiencies.

The National Initiative for Cybersecurity Education (NICE) published by the National Institute of Standards (NIST) provides guidelines on cybersecurity education. Shoemaker, Kohnke, and Sigler (2016) authored a comprehensive guide to the NICE initiative. What was of most interest to this current study is the emphasis on engineering in the Shoemaker, Kohnke, and Sigler (2016) book. The authors mention engineering 324 times. Particular emphasis is placed on secure software engineering. Even more relevant to this current study is the fact that systems engineering is mentioned 28 times. Clearly, the authors viewed engineering as a key aspect of cybersecurity and they also perceive a relationship between cybersecurity and systems engineering.

While Shoemaker, Kohnke and Sigler (2016) emphasized engineering as an aspect of cybersecurity, they stopped short of proposing a specific engineering-oriented curriculum. Rather, the authors focused on the need for engineering skills within cybersecurity training and practice. However, their work, as well as the National Initiative for Cybersecurity Education itself points to the need for significant overhauls in cybersecurity curriculum.

The primary purpose of the NICE standards is to identify knowledge areas that are relevant to cybersecurity. The degree to which these are implemented within the curriculum is not addressed. However, these knowledge, skills, and abilities (KSA's) identified by the NICE standard can be beneficial. These are defined in NIST Special Publication 800-181 (NIST, 2017).

While the primary focus of this current study is on cybersecurity education within the United States, it is an issue that has international impact. There are problems in cybersecurity education in a number of countries (Bellasio, 2018; Caelli & Lue, 2018;

Lehto, 2018). Other nations, including the Dominican Republic have also expressed issues with their cybersecurity education (Dawson & Nuñez, 2018). Studies have also suggested a need to overhaul cybersecurity education and training in Japan (Beuran, Chinen, Tan, & Shinoda, 2018). Similar studies have found issues with cybersecurity education in Saudi Arabia (Alsmadi & Zarour, 2018). What these studies demonstrate is that there are issues with cybersecurity education across the globe.

Caelli and Lue (2018) examined both graduate and undergraduate coursework in Australia. Their study was broad based and concluded that such programs are in adequate. To quote the authors of the study "this study submits that in Australia available courses are few and are acknowledged as not meeting market demands for skilled cybersecurity professionals."

This particular study (Caelli & Lue, 2018) references the Core Body of Knowledge (CBOK) of the Australian Computer Society and its 5 sub sections on cybersecurity management. One key point of their study is that there are existing standards for cybersecurity, but that such standards are not well integrated into relevant academic curriculum. The authors examined both undergraduate and master's programs. The study did not examine doctoral programs.

Lehto (2018) examined cybersecurity education in Finland. This study began by providing an overview of the Finnish education system. This included a discussion of graduate education (both doctoral and master's degree) and undergraduate degrees. However, it also described vocational and primary education systems in Finland. The point of this article was to highlight the importance of cybersecurity education.

Furthermore, the authors advocated integration of cybersecurity education at more levels of the Finnish education system.

The issues with cybersecurity are also found throughout Europe. Bellasio, et al. (2018) examined developing cybersecurity capabilities in industry. While this study did focus primarily on industry, there was a discussion of cybersecurity education. A section of the 324-page report was devoted specifically to training and education. Section D.3.2, entitled "Framework for cybersecurity education" posited a generalized framework for security in formal education. The framework is applicable to primary and secondary education as well as university education. The authors did not put forward a specific curriculum recommendation, but rather a recommendation for bringing together relevant stakeholders who can form such a curriculum.

Rashid, et al., (2018) took a more fundamental and critical look at cybersecurity education. Rather than focus on the deficiencies in specific curriculum, the authors focused on the more fundamental question of the cybersecurity body of knowledge. The authors examined the Cybersecurity Body of Knowledge (CyBOK). The primary focus of their study was to emphasize a need to both enhance the CyBOK and to ensure that curriculum is aligned with CyBOK.

The CyBOK itself is a nascent standard that is still under development (CyBOK, 2018). This standard looks at specific knowledge areas (KA's). The KA's are: Law & Regulation, Privacy & Online Rights, Human Factors, Risk Management & Governance, Security Operations & Incident Management, Forensics, Adversarial Behaviors, and Malware & Attack Technologies. However, notably absent from this list of KA’s is any

addressing of issues involving secure design, secure systems engineering, or related engineering tasks.

McGettrick, Cassel, Dark, Hawthorne, & Impagliazzo (2014) describe a workshop hosted by the Association of Computing Machinery (ACM) Education Board. One of the starting assumptions for this workshop was "Cybersecurity is currently an immature and ill-defined subject and not a true discipline since it lacks some of the criteria normally applied to disciplines." The workshop concluded that very technical graduate programs are needed to provide the level of cybersecurity professionals needed. This workshop highlighted the need for specific curriculum guidelines but stopped short of clearly defining a cybersecurity engineering curriculum.

The lack of specific definitions of the profession has direct impact on the practice of the profession. As one example, Easttom (2018) discusses the cybersecurity sub-field of penetration testing. This sub-discipline is currently practiced in an essentially ad hoc manner. While there are standards to help guide the practice, they are rarely part of cybersecurity curriculum. Furthermore, it is quite rare to see techniques that are common to engineering, being practiced in cybersecurity in general and in penetration testing in particular. As one example, requirements traceability matrices are not commonly utilized in penetration testing.

Furnell, Fischer, & Finch (2017) discuss the increasing demand for cybersecurity professionals. In their study they describe the fact that the need for cybersecurity professionals is outpacing the supply. However, the authors also discuss that identifying the appropriate skillset and educational background can be problematic. In essence, employers are not always certain what qualifications they should be looking for.

The primary issue with Furnell, Fischer, & Finch's study (2017) is the need for more cybersecurity professionals. However, they also note the lack of a coherent definition for what skills are needed. This is directly related to the ambiguity in defining cybersecurity as a profession, or as an academic discipline. It is not possible for an employer to simply state that what is required is a degree in cybersecurity. It is not even clear what such a degree would include.

As early as 2013, Caldwell (2013) was discussing the cybersecurity skills gap. According to his paper, it would take up to 20 years to address this cybersecurity skills gap. The primary focus on Caldwell's study was to identify specific shortages and how those affected organizations. However, it should be apparent that shortages are related to academic curriculum. One cannot fill shortages simply by seeking out employees with a specific degree, if the curriculum for that degree is not well established.

It has been posited that cybersecurity is an engineering discipline (Bayuk, 2011; Landwehr, 2012). However, such assertions have generally been made without clearly defining how to conduct cybersecurity as an engineering discipline. In order to realize cybersecurity engineering as a clearly defined engineering discipline, there are preliminary steps required. The first is to define what engineering is. The term engineering is widely, and often inappropriately used. Too often 'engineering' is used for any process that involves technology. Therefore, it is important to clearly define engineering. Then the second issue is to define cybersecurity engineering as a separate discipline within engineering.

Engineering is a wide field with many sub-disciplines (Martínez-Caro & Campuzano-Bolarín, 2011). Computer science emerged as a separate discipline from electrical

engineering but traces its roots to electrical engineering. One might argue that cybersecurity is in turn emerging from computer science. However, this current paper posits cybersecurity as an engineering discipline, and in fact that natural outgrowth of a specific engineering discipline, that of systems engineering.

Systems engineering is an interdisciplinary field that incorporates engineering and management functions in order to design and manage complex systems (Buede & Miller, 2016). The entire systems lifecycle of any system of interest is addressed. Cybersecurity is, in large part, concerned with developing systems that are sufficiently secure or enhancing current systems so that they are sufficiently secure. Sufficient security is defined as having reduced risk below an acceptable threshold (Conklin, White, Cothren, Davis, & Williams, 2014).

There are a variety of systems engineering activities that are relevant to cybersecurity. However, one facet of systems engineering that stands out as most applicable to cybersecurity is the thought process. Systems engineering is dominated by systems thinking. The essence of systems thinking is, as the name suggests, viewing the entire systems and the subsystems interactions, rather than viewing components in isolation (Dickerson & Mavris, 2016). This is the appropriate approach for cybersecurity. The systems in question involved diverse computers, networks, operating systems, and even human elements of the system.

Cybersecurity traditionally involves dealing with each security concern individually. For example, the security of a given web server might be addressed (Conklin, et al., 2015). Or perhaps a specific security vulnerability such as malware might be the focus of

a security effort (Easttom, 2016). However, these isolated approaches to cybersecurity are incomplete. A systems approach to cybersecurity would yield more robust results.

Another area in which a system engineering approach would be appropriate for cybersecurity is the discipline of penetration testing. Penetration testing is a sub discipline of cybersecurity concerned with testing the security of a given system by actually attempting to breach that security. Despite having been practiced for many years, penetration testing is still a nascent discipline. It is often practiced in a rather ad hoc manner (Bonderud, 2014; Easttom, 2018; Puri, 2018) rather than a systematic approach. This lack of a systematic approach can be corrected by applying systems engineering techniques to penetration testing.

Failure analysis is another area in which a systems engineering approach would significantly enhance cybersecurity. Currently, failure analysis in cybersecurity is primarily conducted in power and energy systems (Jauhar, et al., 2015; Wang, et al., 2016). Formal failure analysis is relatively unknown in other areas of cybersecurity. However, systems engineering has robust methods for verification and validation of systems (Engel, 2010; Rainey & Tolk, 2015).

Systems engineering has also produced a new subdiscipline entitled reliability engineering (Kapur & Pecht, 2015; Modarres, Kaminskiy, & Krivtsov, 2016). The focus of reliability engineering is an emphasis on dependability in the entire life cycle of a product. The goal is for each component in a system to maintain a high level of dependability. The goal of cybersecurity is a reliable, dependable system. It is natural to look to reliability engineering for guidance on proper cybersecurity engineering.

Kapur & Pecht, (2015) explain that reliability is defined as the probability of success. In cybersecurity this can be defined as the probability that a given counter measure will successfully prevent a particular breach or attack. Reliability engineering is concerned with quantifying risk. Furthermore, reliability engineering is focused on measuring and defining levels of success and failure (Kapur & Pecht, 2015).

The integration of concepts from reliability engineering, into cybersecurity engineering, would clearly enhance cybersecurity. Having specific metrics for reliability, coupled with established methods for quantifying reliability would provide cybersecurity with more clearly defined data. In order for any discipline to truly become an engineering discipline, there must be a mathematical basis to the discipline. Reliability engineering provides such a basis for cybersecurity engineering.

Requirements engineering is thoroughly described by Laplante (2017). This sub-discipline of systems engineering is also applicable to cybersecurity engineering. Laplante provides detailed analysis of a number of techniques in requirements engineering. These include domain analysis, prototyping, and use cases. All of these techniques could also enhance cybersecurity engineering.

Laplante (2017) also provides thorough coverage of the ISO/IEEE 29148 standard. The 29148 standard is the standard for requirements engineering for software and systems engineering. The standard describes the iterative and recursive application of requirements processes needed for effective requirements engineering.

A review of the current literature shows the current state of cybersecurity is ill defined, and often performed in an ad hoc manner. Engineering, regardless of the particular sub discipline is concerned with precision, modeling, testing, and related engineering

principles. Bringing sound engineering principles and approaches to cybersecurity should have a positive effect on the field of cybersecurity. What is needed are clear guidelines on what constitutes cybersecurity. One goal of this paper is to provide such guidelines.

CHAPTER 2: TECHNICAL APPROACH

The issues with cybersecurity have been explored in the first chapter of this study. In order to address the issues with defining cybersecurity engineering are addressed two methods are applied. The first is to provide a definition of cybersecurity engineering the is clear and meaningful. The second approach is to define modeling used for cybersecurity. While there are tools that can be applied to any engineering discipline, specific disciplines often have their own tools. For example, Unified Modeling Language (UML) was created specifically for software engineering. Its application to other engineering disciplines such as aerospace or mechanical is not recommended and generally ineffective. This illustrates the need for modeling tools and languages that are tailored to specific engineering disciplines. Cybersecurity engineering is no exception.

The first approach will specify particular engineering techniques and processes that should be part of cybersecurity engineering. This provides clear guidance for anyone responsible for developing cybersecurity engineering curriculum. This approach will be specific with practical examples provided.

The second approach takes the existing SysML modeling language and both modifies and extends it for cybersecurity. Certainly, SysML can be utilized now in cybersecurity, however modifying SysML generates a new modeling language that is more particularly designed for cybersecurity. This modeling language is outlined as the second approach to cybersecurity engineering.

2. 1 Defining Cybersecurity Engineering

The first issue is to define cybersecurity engineering as a discipline. In order to accomplish that goal, it is first necessary to define engineering. The Accreditation Board

for Engineering and Technology defines engineering as "The profession in which a knowledge of the mathematical and natural sciences gained by study, experience, and practice is applied with judgment to develop ways to utilize, economically, the materials and forces of nature for the benefit of mankind" (ABET, 2018).

This definition indicates that any engineering discipline must be predicated on knowledge of mathematical and natural sciences. However, as has been discussed in the introduction and literature review, this is frequently not the case with cybersecurity. Even when the term engineering is used, there is often no application of engineering principles. That definition of engineering demands that cybersecurity engineering has a mathematical and scientific foundation.

Traditional engineering disciplines include mechanical, electrical, civil, and chemical (Vanderbilt, 2018). In the 20th century that list was expanded to include aerospace, bio, nuclear, computer and other types of engineering. The past 50 years has seen a rise in the field of systems engineering. What all these diverse fields of engineering have in common, is that they all are predicated on same engineering principles. Those principles begin with rigorous design, based on application of mathematics and natural sciences. Put more succinctly, engineering is primarily concerned with a mathematical and scientific approach to design. That systematic approach to design carries on into development and testing, and in fact throughout the systems life cycle. Even the rigorous design is in turn predicated on a scientific and methodical approach to requirements engineering (Laplante, 2017).

Based on an understanding of engineering definitions and principles, it should be clear that in order to make cybersecurity engineering a true engineering discipline, there are

elements of the practice and teaching of cybersecurity that must be changed. The most efficient way to effect such changes is to model cybersecurity engineering after some existing engineering discipline. It may seem appropriate to choose computer engineering or software engineering as templates for cybersecurity engineering. However, cybersecurity engineering inherently involves a symbiosis of a wide range of systems. Cybersecurity is not limited to computers. There are human factors, policies, and legal issues (Bauer & Van Eeten, 2009) that are foreign to computer engineering and software engineering.

Cybersecurity involves diverse computer systems, human processes, varying operating systems, and network communications. Each of these components are themselves systems. Cybersecurity could appropriately be labeled a system of systems. Therefore, systems engineering is the appropriate template for cybersecurity engineering. One of the proposals in this current study is that cybersecurity be formalized as a sub discipline of systems engineering.

Before cybersecurity engineering can be defined as a sub-discipline of systems engineering, it is critical to first establish a clear understanding of what systems engineering is. The International Council on Systems Engineering (INCOSE) defines systems engineering as follows:

> "Systems Engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering

> the complete problem: Operations, Performance, Test, Manufacturing, Cost & Schedule, Training & Support, Disposal." (INCOSE, 2019).

Systems engineering is, by definition, an interdisciplinary engineering discipline. It brings together diverse fields of engineering and includes project management activities (Wasson, 2015). Systems engineering is concerned with a given system, or system of systems, throughout the system life cycle. This begins with the concept phase and continues through system disposal (Dickerson & Mavris, 2016). This is an appropriate approach for cybersecurity engineering as well.

The first area to consider is requirements engineering. In systems engineering this is used to define the requirements for the system to be developed. The process is to begin with the informal, and often vague articulation of requirements as per the stakeholders, and to process that into specific and actionable system requirements (Laplante, 2017). In cybersecurity, requirements engineering is a critical component that is often overlooked. Many cybersecurity projects are done simply because they meet minimum requirements for some regulatory requirement or because they are common cybersecurity tasks (Easttom, 2018). Formalized requirements engineering is not a common occurrence in cybersecurity. This indicates that one benefit of formally defining cybersecurity engineering is that requirements engineering can then be integrated into cybersecurity projects and curriculum.

While the engineering processes apply to all aspects of cybersecurity, it can be instructive to consider a specific example to illustrate the application of requirements engineering. For that purpose, one can consider a penetration test. Currently penetration testing is often done in an ad hoc manner. The process is often an unrelated set of

activities used to attempt to breach the target system. The penetration tester then documents the success or failure of such attempts. This informal process leads to uneven testing. Some systems as well as some portions of systems are thoroughly tested. Other systems and components may be inadequately tested, or not tested at all. In penetration testing the requirements engineering process can be used to define the specific requirements for a particular penetration test. It is often the case that the client has only vague ideas about what a penetration test is, or what they want to accomplish.

The requirements engineering activities begin with requirements elicitation. This is a process wherein stakeholders and engineers meet to discuss requirements. As the name suggests, the engineers elicit requirements from the stakeholders (Dick & Hull, 2017). The requirements initially gathered are then analyzed. During the requirements analysis phase UML diagrams, user stories, and other techniques may be used to clarify the requirements. Often requirements analysis is then followed with system modeling (Laplante, 2017). Modeling can be done with UML or SysML modeling, or tools such as MATLAB. The idea is to explore the requirements that have been gathered. Next the requirements are specified and validated.

In requirements engineering the systems engineer uses techniques to elicit requirements from the client or other stakeholders. This is a process that can be readily tailored to cybersecurity engineering. As one example, this can be applied to penetration testing, a subset of cybersecurity. For penetration testing requirements engineering can involve several techniques:

(1) Review past incidents the client organization has had and incidents that have occurred in the same industry. Extrapolate from those specific requirements and seek the client's agreement on those requirements.

(2) Use-case diagrams are common in systems engineering. They provide a very easy to understand model that even a very non-technical stakeholder can understand. Penetration testers can use mis-use cases to model potential misuses of the client's systems. These mis-use cases can include insider threats, external attackers, and even accidental security violations. Mis-use cases are described in detail later in this study, in the section on SecML.

(3) Review specific requirements from relevant regulatory bodies and industry standards. Many standards, such as the Payment Card Industry Data Security Standard (PCI-DSS), define specific penetration testing requirements (Easttom, 2018).

Once requirements have been gathered and approved by the stakeholder, those requirements should form the foundation of the penetration test. In systems engineering a bi-directional requirements matrix is a common tool for tracing requirements. For penetration testing this will trace every requirement to at least one specific test that was conducted, and every single test should trace back to a specific requirement. This ensures that all requirements were met in the penetration test, and that all tests conducted were necessitated by one or more specific requirements. Figure 1 displays a simplified requirements matrix for a penetration test.

	A	B	F	G	H	I	J	K	L	M	N	O	P	Q
1			Requirements											
2														
3			A.1-SQL Injection	A.2-SMB Flaw	B.1-Malware Delivery	B.2-Wireless Security								
4	**Req. ID**	**Pen Testing Activity**												
5	1	All web login screens will be tested for SQL injection manually	X											
6	2	All web login screens will be tested for SQL injection using at least one automated tool.	X											
7	3	Each server will be probed using at least three separate Metasploit attacks on SMB		X										
8	4	An inoccuous msvenom payload will be sent to at least one machine per sub-net			X									
9	5	An inoccuous script virus will be sent to at least one machine per subnet			X									

Figure 1 Requirements Bidirectional Traceability Matrix

Clearly an actual penetration test would have many more activities and requirements. But this figure demonstrates the usefulness of the requirements bidirectional traceability matrix when applied to penetration testing. The primary issue in this example is to integrate requirements engineering into the penetration test. The specific requirements will vary depending on the specific needs for that particular penetration test.

Once the requirements are established, it is necessary to plan the actual penetration test. Systems engineering provides several effective tools to aid in planning. One such tool is the Work Breakdown Structure (WBS). The WBS is a diagram that takes a large process and breaks it down into smaller, manageable pieces. This is useful for ensuring all tasks have been planned. It also breaks the project into smaller tasks to facilitate both scheduling and budgeting. A simplified Work Breakdown Structure for a single server is shown in figure 2.

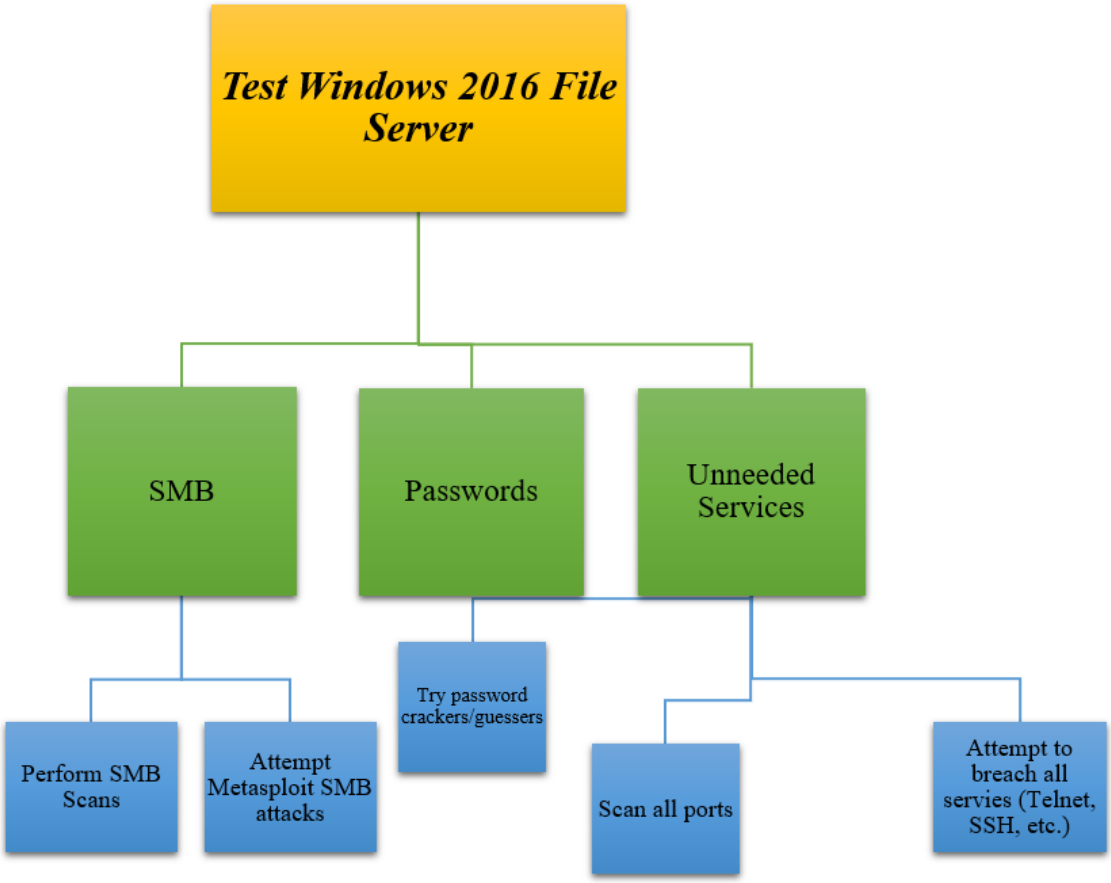


Figure 2 WBS for a single server

Simply by applying requirements engineering and a Work Breakdown Structure to a penetration test, one will achieve a more systematic test. This can improve efficacy of penetration testing as well as streamline efficiency. This cannot be accomplished, however, if the penetration tester is not educated on these fundamental concepts from systems engineering.

Beyond the particular example of planning and executing a penetration test, is the broader issue of designing any security system. Such design principles apply to any security implementation such as deploying a new intrusion detection system (IDS), implementing new network policies, or developing a honey pot (decoy system). The

current trend in cybersecurity is to perform tasks with minimal if any planning. This is another area wherein systems engineering can enhance cybersecurity.

As was previously discussed, elements from reliability engineering can also be applied to cybersecurity engineering. By integrating the established methodologies for measuring reliability, cybersecurity engineering has a ready-made set of metrics. At its core, reliability engineering is about risk management. And that is also the ultimate goal of cybersecurity.

Any integration of systems engineering with cybersecurity will have to integrate specific standards. The ISO 15288 standard defines the system development lifecycle. This same lifecycle should be applied to developing security in any environment. Thus, when implementing a new intrusion detection system, or in implementing new network policies, one should follow the ISO 15288 system development lifecycle. That standard includes the following clauses:

1. Clause 6.4.1 - Stakeholder Requirements Definition Process
2. Clause 6.4.2 - Requirements Analysis Process
3. Clause 6.4.3 - Architectural Design Process
4. Clause 6.4.4 - Implementation Process
5. Clause 6.4.5 - Integration Process
6. Clause 6.4.6 - Verification Process
7. Clause 6.4.7 - Transition Process
8. Clause 6.4.8 - Validation Process
9. Clause 6.4.9 - Operation Process
10. Clause 6.4.10 - Maintenance Process
11. Clause 6.4.11 - Disposal Process

This defines the process for developing or acquiring any system, beginning with defining requirements. This process is commonly taught in introductory systems engineering courses but may be foreign to many cybersecurity practitioners.

Understanding the system development lifecycle is essential to development of any system, including cybersecurity systems.

Any cybersecurity system must begin with the requirements engineering process. One of the hallmarks of requirements engineering is requirements elicitation. This is a process whereby the engineer elicits requirements from stakeholders. The premise being that the stakeholders may not be aware of what can be done, nor what should be done. The engineer's expertise is needed to elicit a set of requirements. This is particularly applicable to cybersecurity. It is very likely that the stakeholders are not well versed in cybersecurity and will not effectively arrive at a complete list of requirements, without some assistance from an engineer.

Another tool from systems engineering that can be of significant benefit in cybersecurity engineering is the use case diagram. This was originally part of UML and is now incorporated into SysML. The use case diagram shows a range of users, including other systems, and how they will interact with the system of interest. The details of the use case are expounded upon later in this current study in the section on SecML. However, the simple concept of modeling how users interact with a system can be very useful in defining system functionality. It is also an effective tool for communicating with stakeholders during requirements elicitation.

The tools discussed in this section are just a few of the techniques that systems engineering uses which can be applied to penetration testing. It can be advantageous for any penetration tester to take a course in systems engineering. Educational institutions may wish to consider adding a systems engineering course to cybersecurity curriculum.

At a minimum, the cybersecurity engineer should at least become familiar with the INCOSE handbook.

The tools presented in this section are only a sample of the tools utilized in systems engineering. As important as the tools, are the concepts of systems engineering. For example, system modeling and simulation is common in systems engineering, but not commonly done in cybersecurity. Defining cybersecurity engineering as a sub-discipline of systems engineering requires that modeling and simulation be included in both the practice of cybersecurity engineering, and in the curriculum.

Modeling and simulation provide a useful mechanism for testing systems in a variety of scenarios. For example, if one is developing a system to counter Denial of Service (DoS) attacks, then it would be useful to simulate a DoS attack to determine how the system will respond. MATLAB is already widely used to model network traffic (Dronjuk, Nazarkevych, & Fedevych, 2014;Jiang, Xu, & Xu, 2015). It is therefore appropriate to utilize MATLAB to model network traffic-based attacks. However, this modeling is not common in cybersecurity. This is one example of the application of modeling and simulation to cybersecurity engineering.

MATLAB is a tool that is commonly used in systems engineering and in other engineering disciplines. This tool has been applied to a wide range of engineer disciplines, including aerospace engineering (Mohseni & Mittal, 2014) and bioengineering (Wilkinson, et al., 2017). This tool should be included in cybersecurity engineering as well. The versatility of this modeling trip makes it an effective tool in many diverse engineering disciplines.

Reliability analysis is another important component of systems engineering, and one that would be well applied to cybersecurity engineering. Reliability engineering and analysis is the process of determining how reliable a given system is. Often in cybersecurity, systems are implemented without knowledge of their reliability. Reliability engineering includes a number of techniques, and formulas for determining reliability.

One hallmark of all engineering disciplines is that of quantifiable data. It is necessary to have objective metrics in order to make informed decisions. Reliability engineering includes a number of formulas that can assist in acquiring such metrics. Fortunately, many of these do not require advanced mathematical knowledge.

The Mean Squared Deviation formula is relatively simple and provides insight into how any system deviates from expectations. This formula is shown in figure 3.

$$\mathrm{MSD} = \frac{1}{\mathrm{n}} \sum_{i=2}^{n} (y_i - T)^2$$

Figure 3 Mean Squared Deviation Formula

The T is the target value. Adjusting the settings of controllable inputs allows one to alter the MSD. This is a relatively simple formula from reliability engineering that can be applied to the reliability of any cybersecurity system. This would be particularly useful in evaluating the efficacy of intrusion detection systems (IDS) and anti-virus software. The MSD formula could be coupled with modeling and simulation to fine tune the cybersecurity system before it is put into operation.

The MSD formula naturally leads to the MPE formula. The mean percentage error (MPE) is the mean number of errors from modeling. In other words, what is the mean

error of the model verses actual values. This is critical in modeling as it can be used to evaluate the efficacy of the model itself. The MPE formula is shown in figure 4.

$$\mathrm{MPE} = \frac{100\%}{n} \sum_{t=1}^{n} \frac{a_t - f_t}{a_t}$$

Figure 4 Mean Percentage Error Formula

In addition to useful formulas for cybersecurity engineering, reliability engineering contains concepts that are applicable to cybersecurity. For example, the concept of mean time between failures (MTBF) estimates the mean time before a component will fail. For an anti-virus software solution this could be the mean time before a given file is mis-identified as being a virus, or not being a virus.

Another concept from reliability engineering is the mean time to repair (MTTR). Continuing with the example of an anti-virus software suite, what is the mean time after a virus is not caught, for a system to recover from the virus infection. This data would allow the cybersecurity engineer to objectively evaluate the cybersecurity system in question.

As was outlined in this section of the current study, cybersecurity engineering is appropriate viewed as a subdiscipline of systems engineering. By integrating elements from other domains within systems engineering, cybersecurity can be elevated to a true engineering discipline. This requires integration of reliability engineering and requirements engineering into cybersecurity. Furthermore, implementing robust and effective modeling techniques. The end result of these efforts is a formal cybersecurity engineering discipline.

2.2 Cybersecurity Engineering Curriculum

It is not sufficient to define cybersecurity engineering, unless a specific curriculum is developed. As with all engineering disciplines, there needs to be a defined curriculum to train students to enter the profession. While there certainly can exist some degree of variability in curriculum, for cybersecurity engineering to be an academic discipline, there must be some fundamental elements present in the curriculum. As an example, electrical engineering curriculum can vary at different institutions but always includes circuit design course(s), signal processing course(s), semiconductor course(s), and a sequence of calculus courses including ordinary differential equations. For cybersecurity engineering to be a true academic discipline it must have similar characteristics common to all curriculum.

The previously discussed ABET definition of engineering, requires a foundation in science and mathematics. The question becomes, what mathematical foundation is most relevant to cybersecurity? Unless one intends to delve deeply into cryptography, advanced number theory is unlikely to be necessary. However, discrete mathematics offers a number of tools that are relevant to cybersecurity. Discrete mathematics includes set theory, combinatorics, and graph theory (Epp, 2010). Graph theory has been utilized to model network traffic as well as digital forensics investigations (Deo, 2017; Easttom, 2018b). Thus at least an introductory course in discrete mathematics should be included in cybersecurity engineering or required as an elective.

Currently, cybersecurity professionals often overlook mathematical methods and modeling. However, basic calculus can be applied to cybersecurity (Adams, et al., 2013; Mermigas, Patsakis, & Pirounias, 2013). Differentiation is primarily concerned with rates

of change. Network flow studies can be utilized in understanding network attacks. Calculus is not as clearly applicable to day to day cybersecurity operations as is graph theory, therefore the level of calculus required in a given program could be optional dependent on the program. It should be required that at least a basic calculus course is offered in the curriculum or as a pre-requisite.

Based on viewing cybersecurity engineering as a sub discipline of systems engineering, the first portion of the required curriculum should be foundational systems engineering courses. Those foundational courses must provide the necessary background in requirements and reliability engineering, as well as systems engineering concepts and tools. The focus of this current study is for a master's degree program, and the proposed curriculum reflects that fact.

As this proposed discipline is posited as a sub discipline of systems engineering, the logical place to begin is to extract commonalities among systems engineering programs. Table three details the core requirements for several systems engineering master's degree programs.

Table 3 Common Systems Engineering Curriculum

University	Program Description
UT El Paso	Systems Engineering Fundamentals & Architecture Program and Systems Engineering Management Integration, Validation & Testing of Complex Systems
Penn State	Engineering Analysis I Technical Project Management Systems Verification Validation and Testing Deterministic Models and Simulation Systems Optimization Systems Engineering Requirements Engineering
John Hopkins	Introduction to Systems Engineering Management of Systems Projects Software Systems Engineering System Conceptual Design System Design and Integration System Test and Evaluation
Iowa State	Engineering Management Theory Decision Analysis in Systems Design Systems Engineering and Project Management Requirements Engineering Systems Engineering and Analysis.
Southern Methodist University	Systems Engineering Process Integrated Risk Management Systems Reliability and Availability Analysis Systems Integration and Test Systems Reliability Engineering Systems Engineering Design.

Examining the course requirements for each of these universities reveals some variability in the curriculum requirements, but also demonstrates some commalities. Every systems engineering master's program has at least some topics in common. Most programs include the following

- An introductory systems engineering fundamentals course
- A requirements engineering course (or similar course)
- A Testing or validation and verification course.

Therefore, these three courses should be considered core to any cybersecurity engineering course. Beyond these being common to systems engineering degree programs, the three are logical choices. A general, fundamentals of systems engineering is required for students to gain a broad understanding of the field. Requirements engineering has already been described in this study as relevant to cybersecurity, and thus a course in requirements engineering is a logical component of any degree program. Finally, the entire nature of cybersecurity is related to testing and validation. Thus, a course in validation and verification is important. It is important that cybersecurity engineers understand how to effectively test and quantify the results of that testing.

Beyond the systems engineering courses, a cybersecurity engineering degree program must obviously include cybersecurity specific courses. As was already discussed in this study, there is an extreme level of variability in this regard. However, by examining the curriculum in only those programs that are highly technical, one can extract the core required courses. That is shown in table four.

Table 4 Cybersecurity Courses

University	Program Description
Capitol Technology University	Introduction to Information Assurance Operating System Principles for Information Assurance Legal Aspects of Computer Security and Information Privacy Computer Forensics and Incident Handling Malicious Software Vulnerability Mitigation Perimeter Protection Internal Protection Principles of Cyber Security
University of Maryland	4 courses chosen from Software Engineering TCIP/IP Networking Advanced TCIP/IP Networks Reverse Software Engineering Analytics for Decision Support Data Science Network Data Science Software Defined Networking.
Texas A&M College Station	Foundations of Computing Foundations of Cybersecurity Engineering Cybersecurity Law and Policy Then 2 electives chosen from Digital Forensics Engineering Data Analytics for Cybersecurity Cybersecurity Risk Management Software Security
Embry Riddle	System Exploitation and Penetration Testing Computer Security Software Security Assessment Applied Cryptography Software Engineering Discipline System Safety and Certification

As with the cybersecurity programs, there is some variation in course requirements. However, a few commonalities stand out. Several of the programs require some sort of advanced network course or have that as a pre-requisite. Several programs have some

software engineering or secure programming course. Most programs have a general security requirement.

In addition to viewing the courses provided at a sample of universities, the aforementioned NICE KSA's can provide a guideline for appropriate cybersecurity curriculum. There are 630 KSA's identified in NIST SP 800-181 (NIST, 2017). I would be neither practical, nor beneficial to simply list all 630 here. However, a few bear particular attention as they represent critical topics that may not be given the attention, they deserve in existing cybersecurity programs.

KSA 608 is "Knowledge of Unix/Linux and Windows operating system structures and internals". This knowledge area requires a good, in depth understanding of both Windows and Unix/Linux. Students from a strong computer science or engineering background may already have these skills, but cybersecurity often attracts students from diverse academic backgrounds. Any cybersecurity program should include, at least as an elective, an operating systems course.

There are several KSA's that are related to cyber threat intelligence including KSA 623, 603, 593, and 571, as well as other KSA's (NIST, 2017). This indicates that threat intelligence is an essential part of the NICE framework. However, none of the programs examined in this study had a single cyber threat intelligence course even as an elective. This is an area that is being neglected in many current cybersecurity programs. It should be included as at least an elective in any cybersecurity engineering program. And in order for a program to be compliant with the NICE framework, this material should at least be introduced in one of the required courses. It would be possible to include a unit on cyber threat intelligence in a general cybersecurity course.

Strong networking knowledge is emphasized in several KSA's such as KSA 33, 34, 57, 61, 599, KSA 600, 614, and 618 (NIST, 2017). Fortunately, many current cybersecurity programs already emphasize strong networking knowledge in their curriculum. This must be continued in cybersecurity engineering. Ideally, basic networking would be a pre-requisite for any cybersecurity engineering graduate program. Then the program itself could offer advanced networking courses as electives.

Programming skills are also emphasized in the NICE framework. For example, KSA 79, 80, 81, 82, 396, 372, and 197 emphasize programming (NIST, 2017). A cybersecurity engineering program should require at least fundamental programming knowledge as a pre-requisite. Then the program can include secure programming as a requirement, based on the student having basic programming knowledge. Specific KSA's require not just programming, but software engineering. And KSA 140 specifically requires secure programming. However, in the curriculum proposed in this current study includes systems engineering courses. Coupling systems engineering with programming would provide an understanding of software engineering.

Other portions of the NICE framework concern general technology. For example, KSA 109 requires general knowledge of computer hardware (NIST, 2017). KSA 114 describes a general knowledge of electronic devices. KSA 116 regards knowledge of file extensions. These, and related KSA's, require general knowledge of computer hardware and operating systems. This is best addressed as a pre-requisite.

The NICE framework specifically requires knowledge of forensics. KSA 133 discusses general forensics. However, other KSA's get more specific on particular forensic topics. For example, KSA 132 is about system file forensics (NIST, 2017). KSA

156 describes knowledge of legal rules and court procedure. These, and related KSA's require that cybersecurity curriculum includes digital forensics curriculum.

Based on the requirements of systems engineering and the requirements of cybersecurity, it is relatively simple to define a cybersecurity curriculum. Viewing such a curriculum in light of the NICE framework provides even more support for defining a formal cybersecurity engineering curriculum. Based on this data, the core security courses proposed would be:

- Introduction to cybersecurity
- Secure programming
- Digital forensics

These three cybersecurity courses coupled with the previously identified systems engineering courses, provides a core of curriculum of six courses, or 18 credit hours consisting of:

- An introductory systems engineering fundamentals course
- A requirements engineering course (or similar course)
- A Testing or validation course.
- Introduction to cybersecurity
- Secure programming
- Advanced networking

The remaining 18 hours (to form a complete 36 credit hour master's degree program) should involve electives from cybersecurity courses, engineering courses, and computer science courses. The electives approved should be courses that emphasis strong technical

skills. Courses in penetration testing, operating systems, malware analysis, advanced networking, and similar topics would be appropriate.

As previously mentioned, the focus in this current study is on master's degree curriculum. Thus, it is also appropriate to define pre-requisites for such a program. As has been discussed previously in this study, basic programming and networking skills are commonly required for cybersecurity. Furthermore, some basic mathematical skills are required to perform true cybersecurity engineering. With these facts in mind, the proposed pre-requisites for a cybersecurity master's degree program would be:

- One programming course
- One computer network course
- Discrete mathematics
- A general computer hardware and operating systems course
- At least one course in calculus

These five courses would be the absolute minimum to ensure a student had the necessary skills to successfully navigate the master's curriculum. It should be noted that the NICE framework also requires general mathematics knowledge in KSA 52 (NIST, 2017). This provides even more evidence of the importance of mathematics in cybersecurity and cybersecurity engineering.

These pre-requisites must be present in order to effectively teach cybersecurity engineering at the graduate level. Individual universities may require more, but certainly none should require less than these pre-requisites. It should also be obvious that the more technical background the student has, the more successful he or she is likely to be in a graduate cybersecurity program.

This section has outlined a general overview for a cybersecurity curriculum, with cybersecurity engineering defined as a subset of systems engineering. By combining the well-established engineering principles of systems engineering, with cybersecurity knowledge such as the NIST SP 800-181, a robust cybersecurity program can be established. Furthermore, cybersecurity engineering can be established as a clear and well-defined engineering discipline.

CHAPTER3: SECML

Systems engineering utilizes a number of approaches to modeling systems and sub systems. Modeling is an integral part of design and testing. In fact, there is an entire field of model-based systems engineering (Borkey & Bradley, 2019). One of the primary modeling methods utilized in systems engineering is system modeling language SysML (Liu, 2015). SysML is an extension of the earlier Unified Modeling Language (UML). UML was created by the Object Management Group (OMG) in order to design software. SysML extends that to modeling a wide range of systems. SysML includes 9 diagrams, some of these are taken directly from UML, others were created for SysML.

Software engineering also includes a number of other modeling languages. For example, there are Domain Specific Modeling languages (Walker, Parreiras, & Staab, 2014), such as framework-specific modeling language (FSML). FSML is used in object-oriented programming. There are multiple, specific modeling languages for a wide range of software engineering applications (Combemale, et al., 2016).

As has been discussed, there are specific modeling techniques and even modeling languages for particular engineering disciplines. If cybersecurity engineering is to be truly defined as a separate engineering discipline, it would also benefit from its own modeling language. This would facilitate modeling that is tailored to cybersecurity needs.

An important part of systems engineering is modeling. In fact, there is an entire sub-discipline of systems engineering concerned with modeling (Borky & Bradley, 2019; Liu, 2015). The concept is to facilitate better understanding of a system at any stage in the system development life cycle. Being able to simulate and model system behavior can even be used during requirements gathering (Buede & Miller, 2016).

It is clear the modeling is an integral part of engineering, and particularly systems engineering (Liu, 2015; Rainey & Tolk, 2015). It is also true that modeling has been used, in a limited fashion in some aspects of cybersecurity (Jauhar, et al., 2015; Wang, et al., 2016).

What is being proposed in this thesis, is a modification to SysML in order to facilitate modeling in security. This modeling is termed SecML and is used to model security needs. The SecML definition uses some SysML and UML diagrams, and adds a few new diagrams. The concept is to provide a modelling language that is specific to cybersecurity. Software engineering uses UML, systems engineering SysML, it is only natural that cybersecurity engineering should have a modeling language specific to the domain.

3.1 Misuse Diagram

The first SecML diagram, and the easiest to understand, is the misuse case diagram. Both SysML and UML utilize use case diagrams in order to understand how users interact with a given system. Users also include other systems that might use a given system. For security purposes, the most concern is on how an attacker might misuse a system. Therefore, it is logical to diagram misuse cases. The essence of an attack is misusing a system.

A typical use case diagram is shown in figure 5.

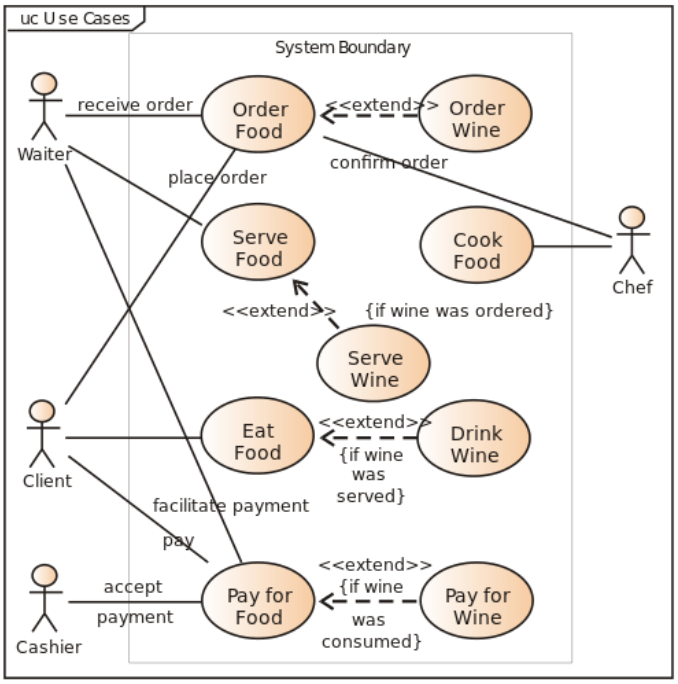


Figure 5 Typical Use Case Diagram

In the traditional use case diagram, some relatively simple elements are utilized. The image that appears to be a stick figure is used to represent any user of the system (Dennis, Wixom, & Tegarden, 2015). This can, of course be a human user. However, another system can in fact be a user, and will still be depicted with the stick figure. Activities that are done in the system are labeled ovals. The connection between a user and a system action is represented via a line. Furthermore, when one activity extends another, that relationship is demonstrated with the dotted line and the <<extends>> label.

The UML use case diagram has been widely used to model specific uses of a system of interest (Dennis, Wixom, & Tegarden, 2015). Its utility derives from the ease of understanding it. The diagram elements are self-evident, and easily understood. That is one reason why this particular UML diagram is useful in communicating with non-technical stakeholders.

The concept of mis-use cases already exists. However, the modification here is to have formal notation for the misuse. The logical starting point is the UML use-case diagram that is then modified to demonstrate misuses. This involves adding/modifying some symbolism. For the SecML mis-use case diagram the following notations are added. The notation is shown in figure four.

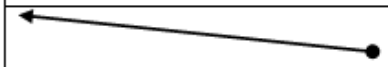
 Abuser	This is an abuser (i.e. attacker, misuser, etc.). The typical user diagram is used, enclosed in an oval.
	A mis-use upload activity is represented by an arrow from the abuser to the upload target. The base of the arrow is a round circle.
	A mis-use download activity is represented by an arrow from the abuser to the upload target. The base of the arrow is a round circle.
	This indicates a counter measure in place

Figure 6 Misuse case Diagram elements

An exemplary use-case diagram is shown in figure 7. The diagram shows a normal user and an abuser misusing the system. It also shows which activities have some mitigation provided to address misuse, and which do not yet have any mitigating factors.

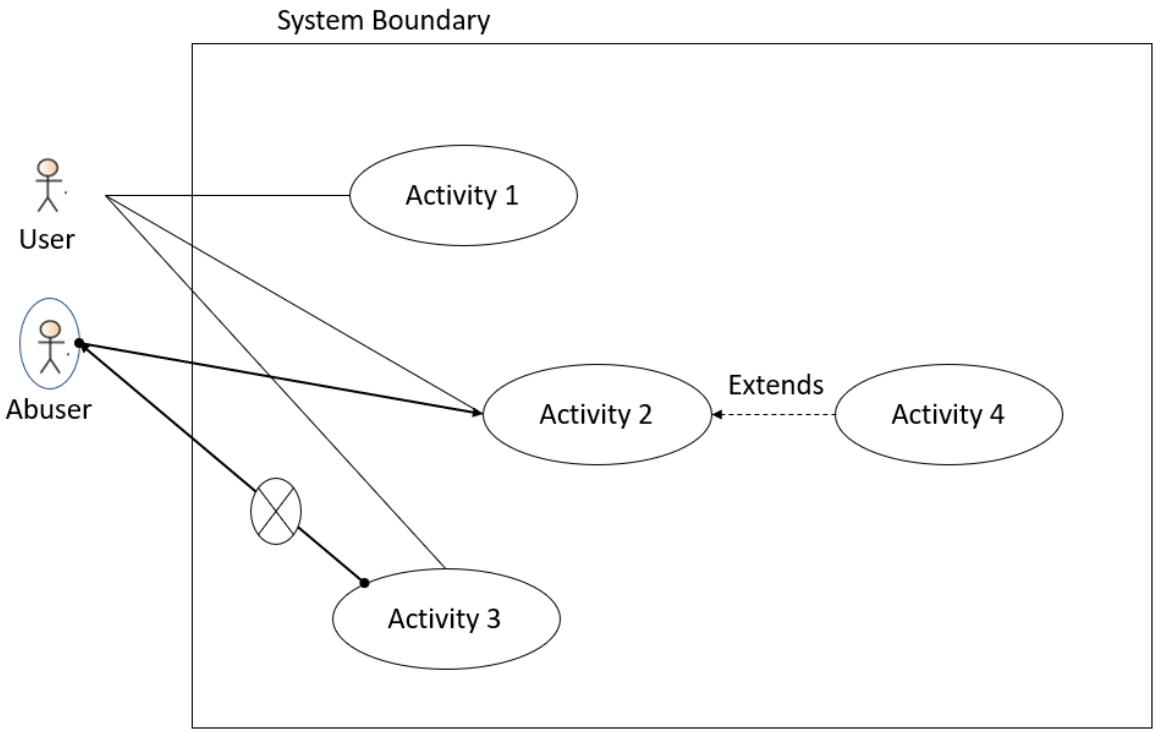


Figure 7 Misuse case diagram example

The diagram can be enhanced with additional notation. For example, the indication of a counter measure could be further described, indicating what the counter measure is. A number is also indicated on the counter measure symbol indicating that there are multiple counter measures. The number of mitigating factors would be an integer inside the circle with the X. This is shown in figure 8.

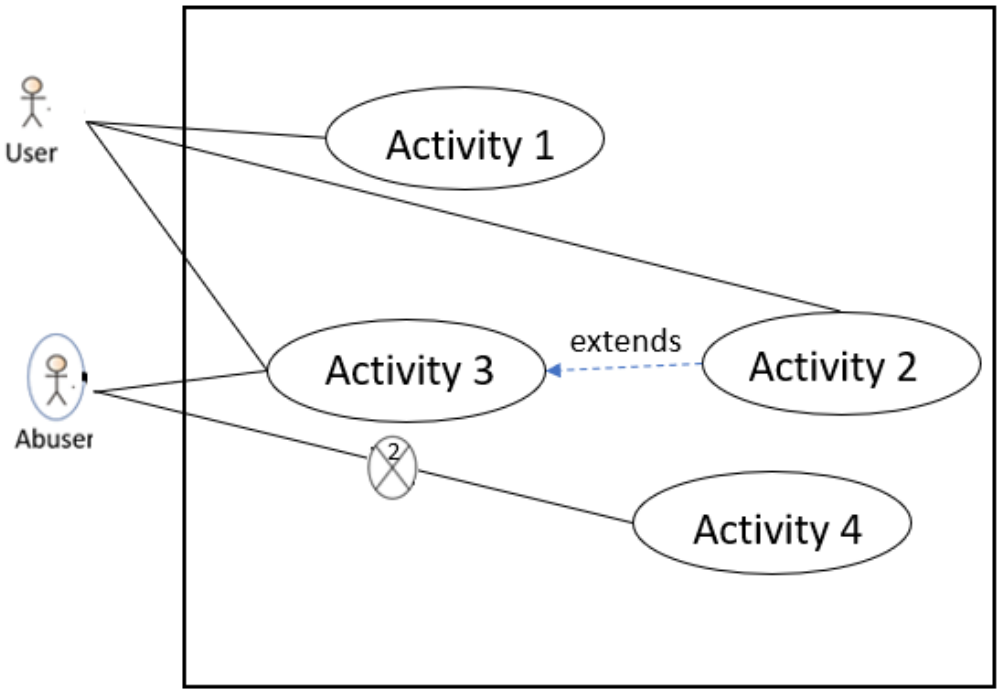


Figure 8 Misuse case expanded

Figure 8 would then be enhanced by explanatory notes about the countermeasures employed. For example, the countermeasures could be 1) Policy against downloading attachments; 2) anti-virus. This is designed to counter viruses sent from an abuser/attacker to a victim. The misuse case diagram allows the cybersecurity engineer to model how the attacker would misuse the system, and what counter measures are currently in place. More importantly, by modeling all misuse cases, it will become obvious which attack vectors have adequate mitigation measures, and which do not.

3.2 Security Sequence Diagram

In SysML a sequence diagram shows how objects interact over time (Borky & Bradley, 2019). In SecML the sequence diagram is being used in in almost the same

manner as it is in SysML. The current UML/SysML sequence diagram is shown in figure 9.

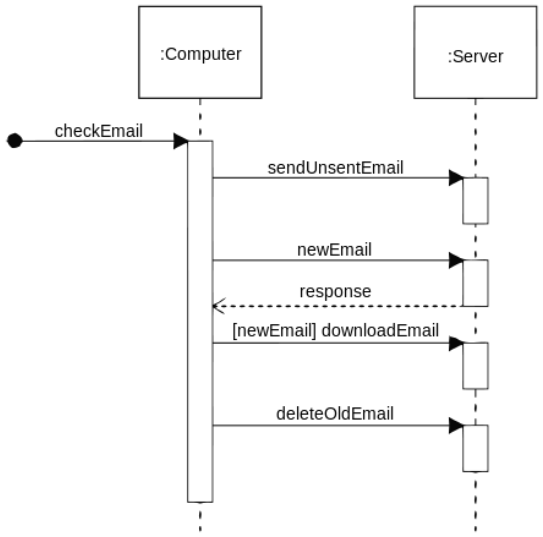
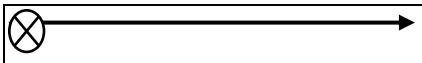


Figure 9 Current Sequence Diagram

The sequence diagram demonstrates a sequence of actions (Liu, 2015). However, these models were mean to diagram intended activities, not attacks. To modify the sequence diagram involves adding/modifying some symbolism. For the SecML, the security sequence diagram the following notations are added.

	This is an unauthorized sequence, such as sending a spoofed email or sending malware. The same sequence is used but beginning with a circle and an X in it.

An exemplary modified sequence diagram is shown in figure 4.

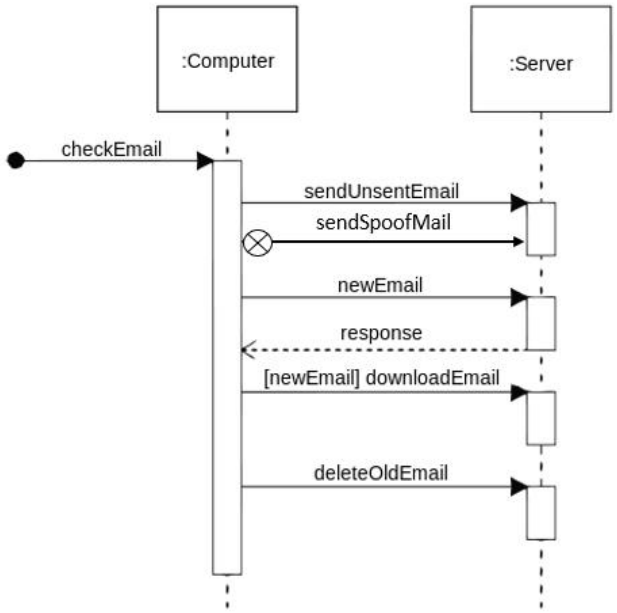


Figure 10 Modified Sequence Diagram

Like the traditional SysML sequence diagram, the messages are still written with the message name above and the directionality of the message. However, the modification for SecML allows one to differentiate between normal operations and unauthorized actions. Seeing unauthorized and authorized actions as they actually occur in a system, provides a very effective understanding of system operations.

The modified sequence diagram provides an overview of the sequence of events in any cyber-attack. This allows the cybersecurity engineer to model various attacks. Coupling the security sequence diagram with a misuse case diagram provides an effective overview of the attack vector in question.

3.3 Data Interface Diagram

This diagram type is created specifically for SecML. It models the flow of data into and out of any system. The concept it to look at any system or sub system and diagram all interfaces for data to flow into and out of the system of interest. Any place that data can flow is an area for security concerns. Data flowing outward can lead to data exfiltration. Data flowing inward can lead to malware being introduced to the system.

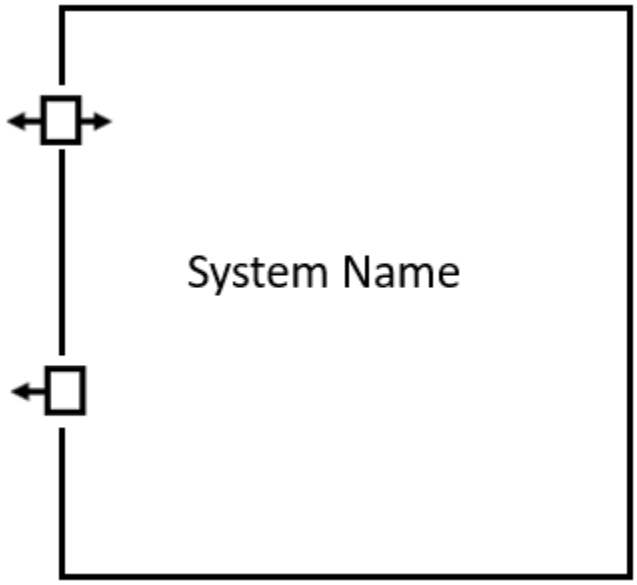
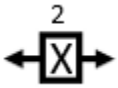


Figure 11 Data Interface Diagram

The specific elements of the Data Interface Diagram are shown in table 5.

Table 5 Data Interface Diagram Elements

Element	Description
	Direction of communication flow.
	A box represents a specific interface to a system or subsystem.
	Interface with only outbound communication.
	Interface with both inbound and outbound communication.
	An interface with an X in it is an interface that has some counter measure implemented for attacks.
	In interface with a counter measure and a number, indicates there are multiple counter measures.

This diagram is intentionally simple. The goal is to make the process one that cybersecurity engineers can efficiently use with minimal training required. The concept it to ensure that all data flow points have been identified, and that mitigation measures have been identified. This diagram is used to examine the system of interest and to determine what, if any mitigation strategies have been put into place for each data interface. This is essentially a limited interface diagram.

3.4 Security Block Diagram

Unified Modeling Language, which was the basis for SysML has a component diagram. In UML, component diagrams are used to identify components in software and to model how they connect. For example, UML contains assembly connectors that model a connection when one component requires another component. The delegation connector links an external component.

SysML has a related diagram, the Block Definition Diagram. This is used in SysML to model structural elements of the system of interest, in the form of blocks. The blocks are

then shown with their composition and classification. An exemplary block definition diagram is shown in figure 12.

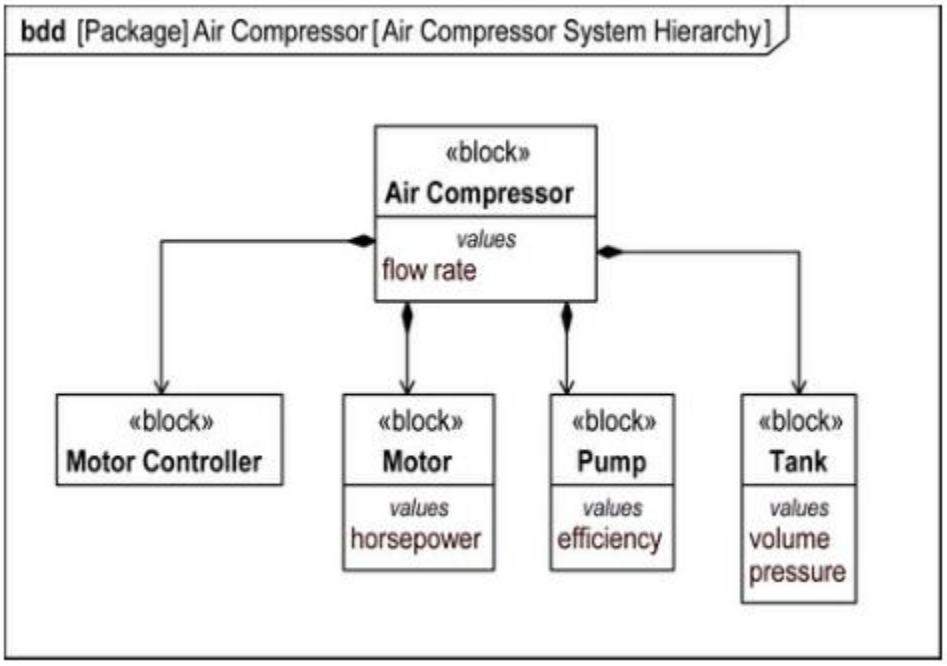


Figure 12 Exemplary Block Definition Diagram

For the purposes of security this should be modified. The concept is to identify any system or component and examine the internal components, and the flow of data in those components. This allows the cybersecurity engineer to examine a given component and view its internal blocks for security issues. The basic modified block diagram is shown in figure 13.

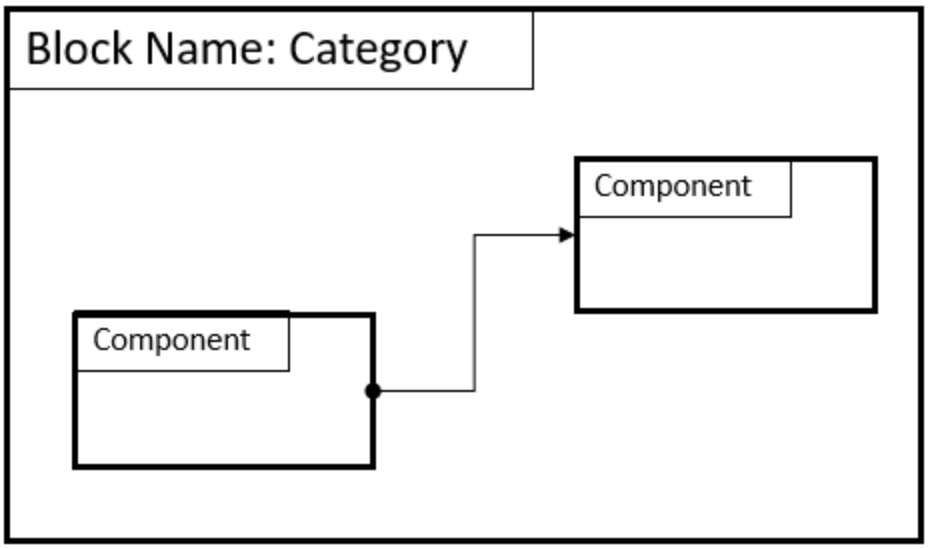


Figure 13 Modified Block Diagram

The categories for the security block diagram are simplified, and shown in the following table:

Table 6 Security Block Diagram Categories

Category	**Description**
Data Device	A computer, workstation, server, tablet or similar device that stores and processes data.
Connecting Device	A device whose primary function is to facilitate network connections. This includes routers, hubs, and switches.
Security Device	A device whose primary function is some security feature such as a firewall, IDS/IPS, etc..

These categories are chosen because they are useful in defining the security features on a given network or system of interest. From a cybersecurity engineering perspective, these three classifications of devices are able to fully describe the security elements of the system of interest.

In this section the foundations of a security modeling language have been described. This modeling language, SecML is based on the pre-existing SysML. It may be that further research leads to enhancements to these models, and the addition of new models to SecML. As with all modeling language, it is expected that SecML will be revised and expanded.

Conclusions and Future Work

This current study outlined two primary areas for applying systems engineering to cybersecurity in order to define cybersecurity as a formal engineering discipline. The first area is to clearly define what cybersecurity engineering is. That section of this study described the application of systems engineering to cybersecurity. Furthermore, this section outlined recommended curriculum for cybersecurity engineering.

The second section of this current study outlined a new modeling language for cybersecurity engineering. This modeling language, named SecML (Security Modeling Language) is based on SysML. Some SysML diagrams are modified for use in SecML, other diagrams were created specifically for SecML. There are also a small number of SysML diagrams that are used in SecML in the same way they are used in SysML.

This current studies data can be expanded in two critical ways. The first is for additional researchers to review the curriculum recommendations provided in this study and perhaps edit them. The goal is to arrive at a cybersecurity engineering curriculum that is as well defined as other engineering disciplines such as electrical engineering and mechanical engineering.

Another area to expand this work would be to refine and expand the SecML modeling language. Existing modeling language such as UML and SysML have been refined over

time. It is expected that SecML will also be refined and improved over time. This could include additional diagram elements, or refinements to existing elements in SecML.

If cybersecurity engineering is to be truly embraced as an engineering discipline, at some point it will require its own standards. The current study suggests the integration of systems engineering standards such as ISO/IEEE 15288 and ISO/IEEE29148. However, for the discipline to fully mature as a complete engineering discipline, at some point one or more cybersecurity engineering standards should be developed. This is an appropriate area for further study.

REFERENCES

ABET (2018). *Accreditation Board for Engineering and Technology*. Retrieved from www.abet.org January 10, 2019.

Adams, M. D., Hitefield, S. D., Hoy, B., Fowler, M. C., & Clancy, T. C. (2013). Application of cybernetics and control theory for a new paradigm in cybersecurity. *arXiv preprint arXiv:1311.0257*.

Alsmadi, I., & Zarour, M. (2018, April). Cybersecurity Programs in Saudi Arabia: Issues and Recommendations. *In 2018 1st International Conference on Computer Applications & Information Security (ICCAIS)* (pp. 1-5). IEEE.

Bauer, J. M., & Van Eeten, M. J. (2009). Cybersecurity: Stakeholder incentives, externalities, and policy options. *Telecommunications Policy*, *33*(10-11), 706-719.

Bayuk, J. L. (2011). Systems security engineering. *IEEE Security & Privacy, 9*(2), 72-74.

Bellasio, J., Flint, R., Ryan, N., Sondergaard, S., Monsalve, C. G., Meranto, A. S., & Knack, A. (2018). *Developing Cybersecurity Capacity.* Retrieved from https://www.rand.org/pubs/research_reports/RR2072.html

Beuran, R., Chinen, K. I., Tan, Y., & Shinoda, Y. (2016). Towards effective cybersecurity education and training.

Bonderud, D. (2014). *Don't monkey around: Why ad-hoc testing is a hacker's best friend.* Retrieved February 02, 2019 from https://www.veracode.com/blog/2014/09/dont-monkey-around-why-ad-hoc-testing-is-a-hackers-best-friend

Borky, J., & Bradley, T. (2019). *Effective model-based systems engineering*. New York City, New York: Springer Publishing.

Buede, D. M., & Miller, W. D. (2016). *The engineering design of systems: models and methods*. Hoboken, New Jersey: John Wiley & Sons.

Burley, D. L., Eisenberg, J., & Goodman, S. E. (2014). Would cybersecurity professionalization help address the cybersecurity crisis? *Communications of the ACM, 57*(2), 24-27.

Burley, D., Bishop, M., Kaza, S., Gibson, D. S., Buck, S., Parrish, A., & Mattord, H. (2018). Special Session: Joint Task Force on Cybersecurity Education. *In Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 918-919). ACM.

Caelli, W. J., & Liu, V. (2018). Cybersecurity education at formal university level: An Australian perspective. In *Journal for the Colloquium for Information Systems Security Education* (Vol. 5, No. 2, pp. 26-44). CISSE.

Caldwell, T. (2013). Plugging the cyber-security skills gap. *Computer Fraud & Security, 2013*(7), 5-10.

Combemale, B., France, R., Jézéquel, J. M., Rumpe, B., Steel, J., & Vojtisek, D. (2016). *Engineering modeling languages: Turning domain knowledge into tools*. Chapman and Hall/CRC.

Conklin, W. A., White, G., Cothren, C., Davis, R., & Williams, D. (2015). *Principles of computer security*. McGraw-Hill Education Group.

Conklin, W. A., Cline, R. E., & Roosa, T. (2014). Re-engineering cybersecurity education in the US: an analysis of the critical factors. *In System Sciences (HICSS), 2014 47th IEEE Hawaii International Conference on* (pp. 2006-2014).

Craigen, D., Diakun-Thibault, N., & Purse, R. (2014). Defining cybersecurity.

Technology Innovation Management Review, 4(10).

CyBOK (2018). https://www.cybok.org/.

Dawson, M., & Nuñez, P. M. T. (2018). Issues in Cybersecurity: Security Challenges and Problems in the Dominican Republic. *Land Forces Academy Review*, *23*(3), 173-180.

Dennis, A., Wixom, B. H., & Tegarden, D. (2015). *Systems analysis and design: An object-oriented approach with UML*. Hoboken, New Jersey: John Wiley & Sons.

Deo, N. (2017). *Graph Theory with Applications to Engineering and Computer Science*. Mineola, New York: Dover Publications

Dick, J., and Hull, E. (2017). *Requirements Engineering*. New York City, NY: Springer Press.

Dickerson, C., & Mavris, D. N. (2016). *Architecture and principles of systems engineering*. Boca Raton, FL: CRC Press.

Dronjuk, I., Nazarkevych, M., & Fedevych, O. (2014). Asymptotic method of traffic simulations. *In Distributed Computer and Communication Networks* (pp. 136-144). Springer, Cham.

Easttom, C. (2016). *Computer security fundamentals, 3rd Edition*. New York City, New York: Pearson Press.

Easttom, C. (2018). *Penetration testing fundamentals: A hands-on guide to reliable security audits*. New York City, New York: Pearson Press.

Easttom, C. (2018). "A Systems Approach To Indicators Of Compromise Utilizing Graph Theory". *2018 IEEE International Symposium on Technologies for Homeland Security*.

Engel, A. (2010). *Verification, validation and testing of engineered systems*. Boca Raton, FL: John Wiley & Sons.

Epp, S. S. (2010). *Discrete mathematics with applications.* Cengage learning. Boston, MA: Cengage learning.

Furnell, S., Fischer, P., & Finch, A. (2017). Can't get the staff? The growing need for cyber-security skills. *Computer Fraud & Security*, *2017*(2), 5-10.

INCOSE (2019). *International Council on Systems Engineering*. Retrieved 03 February, 2019 from https://www.incose.org

Jauhar, S., Chen, B., Temple, W. G., Dong, X., Kalbarczyk, Z., Sanders, W. H., & Nicol, D. M. (2015, November). Model-based cybersecurity assessment with nescor smart grid failure scenarios. *In Dependable Computing (PRDC), 2015 IEEE 21st Pacific Rim International Symposium on* (pp. 319-324).

Jiang, D., Xu, Z., & Xu, H. (2015). A novel hybrid prediction algorithm to network traffic. *Annals of telecommunications-annales des télécommunications*, *70*(9-10), 427-439.

Kapur, K. C., & Pecht, M. (2014). *Reliability engineering.* Boca Raton, Fl : John Wiley & Sons.

Kim, D., & Solomon, M. G. (2016). *Fundamentals of information systems security*. Burlington, MA: Jones & Bartlett Publishers.

Landwehr, C. E. (2012). Cybersecurity: From engineering to science. *Developing a blueprint for a science of cybersecurity*, *2*.

Laplante, P. A. (2017). Requirements engineering for software and systems. Boca Raton, FL: Auerbach Publications.

Lehto, M. (2018). Cyber Security Education and Research in the Finland's Universities and Universities of Applied Sciences. In *Cyber Security and Threats: Concepts, Methodologies, Tools, and Applications* (pp. 248-267). IGI Global.

Liu, D. (2015). *Systems engineering: design principles and models*. Boca Raton, FL: CRC Press.

Martínez-Caro, E., & Campuzano-Bolarín, F. (2011). Factors affecting students' satisfaction in engineering disciplines: traditional vs. blended approaches. *European Journal of Engineering Education*, *36*(5), 473-483.

McGettrick, A., Cassel, L. N., Dark, M., Hawthorne, E. K., & Impagliazzo, J. (2014, March). Toward curricular guidelines for cybersecurity. *In Proceedings of the 45th ACM technical symposium on Computer science education* (pp. 81-82). ACM.

Mermigas, D., Patsakis, C., & Pirounias, S. (2013, January). Quantification of information systems security with stochastic calculus. *In Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop* (p. 47). ACM.

Modarres, M., Kaminskiy, M. P., & Krivtsov, V. (2016). *Reliability engineering and risk analysis: a practical guide*. Boca Raton, FL: CRC Press.

Mohseni, K., & Mittal, R. (Eds.). (2014). Synthetic jets: fundamentals and applications. CRC Press.

NIST (2017). *National Initiative for Cybersecurity Education (NICE) Cybersecurity Workforce Framework.* Retrieved February 1, 2019 from https://csrc.nist.gov/publications/detail/sp/800-181/final

Puri, S. (2018). *Is Ad Hoc testing reliable?* Retrieved February 02, 2019 from https://www.thinksys.com/qa-testing/ad-hoc-testing-reliable/

Rainey, L. B., & Tolk, A. (Eds.). (2015). *Modeling and simulation support for system of systems engineering applications*. Boca Raton, Fl: John Wiley & Sons.

Rashid, A., Danezis, G., Chivers, H., Lupu, E., Martin, A., Lewis, M., & Peersman, C. (2018). Scoping the Cyber Security Body of Knowledge. *IEEE Security & Privacy, 16*(3), 96-102.

Safa, N. S., Von Solms, R., & Futcher, L. (2016). Human aspects of information security in organizations. *Computer Fraud & Security*, *2016*(2), 15-18.

Shoemaker, D., Kohnke, A., & Sigler, K. (2016). *A Guide to the National Initiative for Cybersecurity Education (NICE) Cybersecurity Workforce Framework (2.0)*. Boca Raton, FL: CRC Press.

Topham, L., Kifayat, K., Younis, Y. A., Shi, Q., & Askwith, B. (2016). Cyber Security Teaching and Learning Laboratories: A Survey. *Information & Security*, *35*(1), 51.

Vanderbilt University (2018). *Summary of Engineering Disciplines.* Retrieved 05 February 2019 from https://engineering.vanderbilt.edu/ge/es140/Otherlinks/Summary.php.

Wang, Y. N., Lin, Z. Y., Liang, X., Xu, W. Y., Yang, Q., & Yan, G. F. (2016). On modeling of electrical cyber-physical systems considering cyber security. *Frontiers of Information Technology & Electronic Engineering, 17*(5), 465-478.

Walter, T., Parreiras, F. S., & Staab, S. (2014). An ontology-based framework for domain-specific modeling. *Software & Systems Modeling, 13*(1), 83-108.

Wasson, C. S. (2015). *System engineering analysis, design, and development: Concepts, principles, and practices*. John Wiley & Sons.

Wilkinson, D. C., Alva-Ornelas, J. A., Sucre, J., Vijayaraj, P., Durra, A., Richardson, W., ... & Gomperts, B. N. (2017). Development of a Three-Dimensional Bioengineering Technology to Generate Lung Tissue for Personalized Disease Modeling. *Stem cells translational medicine*, *6*(2), 622-633.

VITA

William C. Easttom II (Chuck Easttom) already holds a Doctor of Science in Cybersecurity (Dissertation topic was "A Comparative Study Of Lattice Based Algorithms For Post Quantum Computing") as well as two other master's degrees. He has 16 patented inventions and 27 published books. Dr. Easttom is a Senior Member of the IEEE, a Senior Member of the ACM, and a Distinguished Speaker of the ACM.