An Assessment Methodology and Models for Cyber Systems

A Dissertation
Presented in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy
with a
Major in Computer Science
in the
College of Graduate Studies at
The University of Idaho
by
Jennifer Guild

Major Professor: Jim Alves-Foss, Ph.D.
Committee Members: George W. Dinolt, Ph.D.;
Paul Oman, Ph.D.; Clinton Jeffery, Ph.D.
Department Administrator: Fredrick Sheldon, Ph D.

August 2016

Authorization to Submit Dissertation

This Dissertation of Jennifer Guild, submitted for the degree of Doctor of Philosophy with a Major in Computer Science and titled "An Assessment Methodology and Models for Cyber Systems," has been reviewed in final form. Permission, as indicated by the signatures and dates given below, is now granted to submit final copies to the College of Graduate Studies for approval.

Major Professor: _____ Date _____
Jim Alves-Foss, Ph.D.

Committee Members: _____ Date _____
George W. Dinolt, Ph.D.

_____ Date _____
Paul Oman, Ph.D.

_____ Date _____
Clinton Jeffery, Ph.D.

Department
Administrator: _____Date _____
Frederick Sheldon, Ph.D.

Abstract

All computer systems or systems of computers are composed of some combination of three basic components; hardware, firmware, and software. These systems are assessed to determine our confidence in their level of robustness, where robustness is the characterization of strength of a security function, mechanism, service, or solution, and the assurance that it is implemented and functioning correctly. Most experienced assessors are aware that the level of robustness required for each system is dependent upon dynamic factors such as operational environment, threat source interest, and mission criticality. This dissertation provides a methodology and mathematical models to assess systems.

The models, and the results they yield, provide an equal level of understanding for those that implement them, as well as those that interpret their results. The methodology provides an objective characterization of the system by providing the mechanisms to map the evidence of the assessment findings to mathematical models. It is very important to understand that the methodology presented in this dissertation is not to be a checklist or a formula to grade systems. Instead, it is meant provide an objective characterization of the system.

Acknowledgments

I would like to thank Dr. George Dinolt from the U.S. Naval Postgraduate School, without whose guidance and advice I could not have written a dissertation at this level. I would like to thank Dr. Jim Alves-Foss for his patience and guidance while I wrote this dissertation at the same time as formulating national policy.

Dedication

I would like to thank my family for their encouragement and tolerance during my pursuit of this achievement. I would like to dedicate this work to my husband, Jim, whose patience and support during these years made this effort possible.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

Most people automatically interpret the strength of the security capabilities of a computer system, heretofore referred to as a system, based upon their knowledge, experience, cultural background, and the association of that system to its functionality. The average United States (US) metropolitan citizen is confident their bank has significantly stronger security measures in place than the free wireless at the local Starbucks. That characterization of the strength of a security service and the confidence that it is implemented and functioning correctly is referred to as robustness, whereas assurance is defined as just the measure of that confidence [CNS10]. A security service is a capability that supports one or more security requirements (confidentiality[1], integrity[2], availability[3]), with an example being authentication[4] [CNS10].

There are certain systems, when instantiated as a federal banking system and National Security Systems[5] (NSS) [CNS10] require greater levels of robustness so as to not allow an unauthorized person or system access to the system being protected. Such access could result in damage to our financial markets (stock market crash due to "software glitch") [Chi12], federal banks being unable to conduct day-to-day business (cyber attack on Georgian banks) [Mar08], or power grids going black (transformer failure causes failure of key computer) [Win12].

---

[1] Per CNSS 4009 confidentiality is the property that information is not disclosed to system entities (users, processes, devices) unless they have been authorized to access the information.
[2] Per CNSS 4009 integrity is the property whereby an entity has not been modified in an unauthorized manner.
[3] Per CNSS 4009 availability is the property of being accessible and useable upon demand by an authorized entity.
[4] Per CNSS 4009, authentication is the process of verifying the identity or other attributes claimed by or assumed of an entity (user, process, or device), or to verify the source and integrity of data
[5] Per CNSSI 4009: Any information system (including any telecommunications system) used or operated by an agency or by a contractor of any agency, or other organization on behalf of an agency, the function, operation, or use of which: I. involves intelligence activities; II. involves cryptologic activities related to national security; III. Involves command and control of military forces; IV. involves equipment that is an integral part of a weapon or weapon system; or V. subject to subparagraph (B), is critical to the direct fulfillment of military or intelligence missions; or is protected at all times by procedures established for information that have been specifically authorized under criteria established by an Executive Order or an Act of Congress to be kept classified in the interest of national defense or foreign policy. (B). Does not include a system that is to be used for routine administrative and business applications (including payroll, finance, logistics, and personnel management applications). (Title 44 U.S. Code Section 3542, Federal Information Security Management Act of 2002.)

These higher robustness systems have security services and mechanisms that provide the most stringent protection and rigorous security countermeasures. Such systems contain the most valuable information (confidentiality), require a high level of confidence in their level of availability, and/or the accuracy of their data (integrity). Obviously, any loss of this data could cause grave cyber damage, and therefore economic damage, to US individuals, businesses, and the government, as well as physical harm to our troops or civilians.

Most people do not seem to require the same level of robustness for their personal systems as those required by the banking industry or the US Government (USG), though many people have a substantial amount of their personal and financial lives residing on systems. As technology has progressed, so has our understanding of systems. A system is no longer just the desktop computer, but includes mobile devices (such as mobile phones, tablets, and wearable devices), newer automobiles that have embedded Bluetooth, GPS, 802.11, cellular technologies, and commercial aircraft systems. Commercial aircraft are not only heavily dependent upon GPS, but also have satellite communication systems that support them. The software of a commercial aircraft itself, not necessarily the navigational or communication systems in exclusivity, is required to comply with the Federal Aviation Authority (FAA) DO-178C[6] requirements, which includes formal methods as an complementary testing methodology [Gig12] for system assurance.

Formal methods are mathematical modeling and analysis techniques for the specification development, verification, and validation of systems used to prove whether or not expected properties are met [Coh86]. The model must use mathematically defined syntax and semantics. In the case of aircraft, as noted above, formal analysis is conducted to prove the reliability of the safety of flight systems, where in NSS, the formal analysis is conducted to prove robustness of the security designs. Formal analysis also provides mathematical proofs of compliance between the mathematical model and its properties, such that a model never asserts a property to be true when it may not be true.

---

[6] DO-178C, Software Considerations in Airborne Systems and Equipment Certification is the document by which the FAA determines the safety of software-based aerospace systems. http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20100020981.pdf

Whether the system is a vehicle, mobile phone, or a laptop, if it is to be instantiated by or connected to US Government systems, its level of robustness and assurance must be assessed. A robustness or assurance assessment is a process or methodology in which system artifacts are identified, collected as evidence, and assessed against a single instantiation of a system (referred to as a model), to determine the level of risk to US Government by the instantiation or operation of this system. In current US Government system assessment methodologies, the assessment itself is a composition of technical testing conducted on a lab-based system instantiation and then testing conducted on the instantiated system in an operational (live) environment.

This composition, or lack of decomposition, is key shortcoming of the current processes because the assessment approach, evidence, and results have the operational requirements, vulnerabilities, constraints, countermeasures, and threat assessments of that single instantiation imposed onto all subsequent implementers, which hinders reciprocity[7].

Threats, vulnerabilities, risk, and impact must be considered at the conceptual design stage to increase assurance of a system. An independent Information System Security Engineer (ISSE) that is involved in the system development processes starting at design conception, can increase the measure of confidence in the assurance of the system by identifying applicable supplementary artifacts, and through the use of subject matter expertise, increase the quality of all assurance evidence. Such evidence should provide sufficient confidence such that formal methods will only be required for security critical aspects of systems.

An ISSE does not imply the use or requirement of formal methods. The large majority of ISSEs, as well as the Authorizing Officials (AOs) (those US Government officials that accept the risk of instantiating the system) have learned to assess systems and risk while on-the-job (OTJ). In fact, an ISSE or AO that has a theoretical education or background in Computer Science (CS), Computer Engineering (CE), Electrical Engineering (EE), or Mathematics is unusual. That key aspect of not

[7] Reciprocity is the mutual recognition of the validity of the robustness and risk among a community, in this case the community is the US Department of Defense (DoD), US Intelligence Community (IC), and remainder of the US Government.

having a consistently educated workforce is a primary driver for creating an easily understandable and implementable assessment methodology that provides an objective mathematical model.

The work presented in this dissertation addresses the concepts of assessment decomposition, assessment modeling, and a robustness assessment methodology. The work also suggests techniques and methodologies to provide greater evidence of assurance to those that make risk decisions (AOs). The combination provides a consistent methodology, which will reduce the time it takes to conduct assessments by allowing future assessments to build on past assessments, as well as provide cost saving by preventing duplicate assessments.

The remainder of this chapter outlines the fundamental concepts of assessment and assessed systems as they pertain to this dissertation. Section 1.1 provides the basic considerations of assessments. Section 1.2 describes the current lack of assessment methodologies and techniques this research addresses. In Section 1.3, the motivation for and justification of this work are presented, as well as the contributions of this work. Section 1.4 presents an overview of the remainder of the dissertation.

1.1 Basic Considerations

All systems are composed of some combination of three basic components; hardware, firmware, and software. Hardware is the only component that is required by all systems, and some may argue that firmware is just hardware with software included. This characterization includes distributed computing systems, and all manner of systems from wearable devices to a next generation aircraft carrier (CVN 78) [Nav12] that is a system of systems.

There are many types of environments in which NSS operate beyond the standard desktop PC and laptop, and therefore must be assessed considering that operational environment. U.S. federal agency operational environments fall into one of two major categories, vehicle and stationary/land based. A vehicle is anything maneuverable on land, on or through water, in air, or in space, such as a quad-copter, smart watch, or satellite. Conversely, anything stationary is categorized as land

based and includes but is not limited to workstations (laptop or desktop) and Network Operating Centers (NOC). All of these environments can contain more than one system at one or more robustness levels.

Robustness can be defined as the confidence of a system to operate as intended throughout its lifecycle: ensuring essential services, coping with faults, failures, unexpected interactions and malicious activities [HRC11]. Currently, there are two[8] assessment methodologies in use by the US Government that have three robustness levels:

- Low robustness is common commercial practice
- Medium robustness is the usage of best wide spread practices and tools, to include attack surface definition, threat modeling, requirements tracking, design analysis, code correspondence and configuration management
- High robustness is the usage of state-of-the-art best practices, strict engineering practice, formal methods, simplicity of architecture and design and tools, to include attack surface definition, threat modeling, requirements tracking, design analysis, code correspondence and configuration management

Another assessment methodology is based upon the effects of a potential impact versus requirements/security implementation. The National Institute of Standards and Technologies (NIST) Special Publication 800-37, also has 3 categories, which are defined in NIST Federal Information Processing Standards (FIPS) Publication 199:

- Low-Impact - A system in which all three security objectives of confidentiality, integrity, and availability are assigned a FIPS 199 potential impact value of low[9] [Nat10].
- Moderate-Impact - A system in which at least one security objective (i.e., confidentiality, integrity, or availability) has a potential impact value of moderate[10] [Nat10].
- High-Impact - A system in which at least one security objective (i.e., confidentiality, integrity, or availability) has a potential impact value of high[11] [Nat10].

---

[8] NIAP-CCEVS (Basic, Medium, High), DOD Instruction 8500.2 (Basic, Medium, High) http://www.dtic.mil/whs/directives/corres/pdf/850002p.pdf.

[9] The loss of confidentiality, integrity, or availability could be expected to have a limited adverse effect on organizational operations, organizational assets, or individuals.

[10] The loss of confidentiality, integrity, or availability could be expected to have a serious adverse effect on organizational operations, organizational assets, or individuals.

1.2 Background

In the U.S. Department of Defense (DoD) and the U.S. Intelligence Community (IC), most systems process, store, and transmit information within a single security domain. At a minimum, this domain is based on a DoD classification level [Cli95], of which there are four:

- Unclassified (U)
- Confidential (C)
- Secret (S)
- Top Secret (TS)

A security domain may also include compartment(s), such as the IC's Sensitive Compartmented Information (SCI). A compartment is bound to a DoD classification level. An example of a classification and compartment together is TS//SCI, which indicates a Top Secret classification level with an added compartment of SCI. There are also handling caveats, such as For Official Use Only (FOUO) or Controlled Unclassified Information (CUI), in addition to the classification level and compartment labeling. The primary data labels for the DoD are the classification level and compartment.

There are three primary DoD networks; the Non-secure Internet Protocol Router Network (NIPRNet), the Secret Internet Protocol Router Network (SIPRNet), and Joint Worldwide Intelligence Communication System (JWICS) (Unclassified, Secret, and TS//SCI respectively). Depending upon the job description of military personnel or DoD civilian, those personnel may have one or more computers at their workspace, with each computer connected to a network of a different security domain. This footprint is a significant drain on financial and power resources, requires increased heating and cooling (HVAC) when compared to a single computer, and for vehicle-mounted systems represents a greater weight requirement.

---

[11] The loss of confidentiality, integrity, or availability could be expected to have a severe or catastrophic adverse effect on organizational operations, organizational assets, or individuals.

Most employees that work in multiple security domains would prefer a single workstation to access all of their networks. For the employer, having a single workstation capable of handling multiple networks with an acceptable level of assurance would reduce their power consumption and HVAC requirements. This is essential in areas of higher population concentration, such as Washington, DC, as well as vehicles such as submarines.

For the past 30+ years, computer scientists, mathematicians, and electrical engineers have been designing, developing, and instantiating systems that could process, store, or transmit data of different security domains, i.e. classifications and compartments. The initial focus, long ago, was for a system that could securely process and store multiple classifications of data. These systems that processed and stored data at Multiple Levels of Security (MLS), were complete systems, such as Boeing MLS LAN (aka Boeing Secure Network Server (SNS)) [Sto89] [SAI07].

1.2.1 History of Assessment Methodologies

The Trusted Computer System Evaluation Criteria (TCSEC) was the first applied assessment methodology in which the DoD provided basic Information Assurance[12] (IA) (now known as Cybersecurity within the DoD) requirements for assessing the effectiveness of IA controls built into computer systems being considered for the processing, storage and retrieval of sensitive or classified information [DOD85]. This assessment mechanism had seven levels of assurance, with A1 being the highest with formalisms required as evidence, which the Boeing MLS LAN achieved. The focus of TCSEC IA requirements and controls were operating systems (OS) with the top few levels specifying MLS systems. As computer systems and networks evolved, that focus required a new assessment methodology to be considered in order to assess additional technologies being introduced into systems.

The Common Criteria Evaluation and Validate Scheme (CCEVS) was the follow on assessment methodology to TCSEC, and was managed by National Information Assurance Partnership (NIAP), a joint endeavor between the NIST and the National

[12] Per CNSSI 4009: Information Assurance measures that protect and defend information and information systems by ensuring their availability, integrity, authentication, confidentiality, and nonrepudiation. These measures include providing for restoration of information systems by incorporating protection, detection, and reaction capabilities.

Security Agency (NSA). The CCEVS was a two-part scheme. First, an impartial security assessment of a specific Information Technology (IT) product (referred to as a Target of Evaluation (TOE)) was conducted against a specific Protection Profile (PP). Secondly, an independent party validates evidence of the assessment. This process was to provide consistency of assessments and promote comparable results. The focus of the CCEVS assessment is on information IT products [NIA111] not just OSes. The CCEVS assessment methodology conforms to the International Common Criteria for Information Technology Security Evaluation (ICCITSE) [NIA111], thus establishing the first internationally accepted IA assessment methodology. Both CCEVS and ICCITSE also include seven predefined levels of assurance.

The National Security Telecommunications and Information Systems Security Policy (NSTISSP) No. 11, which is the National Information Assurance Acquisition Policy requires that all IA [CNS10] and IA Enabled [CNS10] devices have an assurance assessment (also known as a vulnerability assessment) through one of the following processes [NST00].

- The ICCITSE Mutual Recognition Arrangement [NST00]
- The NIAP Evaluation and Validation Program [NST00]
- The NIST Federal Information Processing Standard (FIPS) validation Program [NST00]

In addition to the CCEVS assessment methodology, there are other assessment processes for the assessment of single level NSS:

- DoD –
    - DoD Instruction (DoDI) 8510.01, Risk Management Framework (RMF) for DoD Information Technology
    - DoD Information Assurance Certification and Accreditation Process (DIACAP) Instruction (DODI 8510-01) which uses the IA controls in DODI 8500.2 (predecessor to RMF)
    - DoD Information Technology Security Certification and Accreditation Process (DoD Instruction 5200.40) which used the IA controls in DODI 8500.2 (predecessor to DIACAP)

- IC – Director of Central Intelligence Directive (DCID) 6/3 [Dir99]
- Other Federal Agencies - National Information Assurance Certification and Accreditation Process (NIACAP)

The DIACAP and NIACAP both recognize systems that are not directly connected to the DoD Information Network (DoDIN), formerly known as Global Information Grid (GIG), or similar network. These systems are referred to as Platform Information Technology (PIT) [13], and are not required to complete all of the IA controls for those assessment methodologies. This is because these systems are special purpose and essential to real time, mission capabilities. While PIT do not and cannot conform to standardized configurations, they are assessed against an applicable subset of DIACAP, and in the very near future, RMF.

Another set of unique systems, Cross Domain Solutions (CDS), which are implemented in the DoD and IC, also required/requires an assessment approach other than the standard approaches. Previously, the DoD and IC had separate methodologies to assess CDSs, which are systems that process, store, or transmit more than a single security domain. A CDS provides the ability to access and/or transfer information between security domains. The DoD methodologies were known as Secret and Below Interoperability (SABI) and Top Secret and Below Interoperability (TABI). The SABI and TABI processes required a foundation of an OS evaluated to a minimum of CCEVS Evaluated Assurance Level (EAL) 4 against the Labeled Security Protection Profile (LSPP). The previous IC CDS process was known as the Top Secret/Sensitive Compartmented Information and Below Interoperability (TSABI). The basis for TSABI was DCID 6/3, which did not mandate, but most assessors required, a foundation of an OS evaluated to a minimum of EAL 4 against the LSPP. In July of 2006, the Unified Cross Domain Management Office (UCDMO) was established by policy to consolidate these methodologies to form the DoD/IC CDS Process and establish a culture of reciprocity between the DoD and IC for CDS assessments [Bai08].

[13] Platform IT refers to computer resources, both hardware and software, that are physically part of, dedicated to, or essential in real time to the mission performance of special purpose systems. Derived from DoDD 8500.1, Paragraph E2.1.16.4, Platform IT.

1.2.2 State of Current Assessment Methodologies

As of June 2010, NIAP is no longer a partnership between NIST and NSA. CCEVS is now solely managed and staffed by the NSA [NIA111]. The only products being accepted into assessment are those that satisfy one of the following:

- Those products claiming compliance with an existing U.S. approved Protection Profile [NIA11].
- If an approved profile does not exist, the product can only be evaluated to Common Criteria EAL 2 [NIA11].

NIAP is also undergoing a transformation. EALs and robustness will no longer be specified [NIA12]. Protection profiles are being created for the Commercial Solutions for Classified program (CSFC) [NSA12] and the assurance requirements will be based upon what is achievable for a technology [NSA12].

> Per NIAP: "Based on over 10 years of experience with Common Criteria assessments, the NIAP program has concluded consistent and repeatable assessment results require a Protection Profile with tailored assurance activities developed in partnership with vendors and the other Common Criteria Schemes, defined as a Technical Community. The changes in policy are the natural result of understanding the assurance that can be achieved with different types of technologies and the limitations of what can be achieved through the assessment of vendor products. Although EAL4 has become the defacto standard for assessment, the generic EAL4 requirements are not relevant, achievable and repeatable in all cases. Given this false label of assurance, the credibility of NIAP and the Common Criteria in general has been negatively affected. To restore the CC brand, it is necessary to restrict assessments to technology specific Protection Profiles with achievable, repeatable and testable requirements and assurance activities." [NIA09]

Interestingly enough, the members of the ICCITSE did not come to those same conclusions and continue to work with EALs and robustness. The changes to these policies are rippling through the DoD and IC. As NIAP no longer specifies robustness,

the DoD and IC must update all processes and procedures to remove those references, including DODI 8500.2. Another policy change is that NSTISSP No. 11 has been suspended, and the impact of this has not yet been fully realized. As of the Fall of 2015, the DoD and IC continue to use ICCITSE member conducted assessments of MLS OSes, so the impact of these policy changes has not yet become apparent.

The LSPP was the Protection Profile against which all MLS OSes were evaluated. While that profile has expired, US Government programs continue to instantiate systems evaluated by ICCITSE labs using this profile. LSPP was to be replaced by the US Government Directory Protection Profile for Medium Robustness Environments but this profile is no longer on the US Government Approved Protection Profiles list [Inf07] The good news is that NIAP insists that all previously evaluated products will remain certified for the stated version of the product, although major updates will invalidate the certification.

The U.S. Government Protection Profile for Separation Kernels in Environments Requiring High Robustness (SKPP) is no longer listed and NSA will not evaluate any additional products against that profile. As such, CCEVS no longer accepts protection profiles for high assurance products. Therefore, essentially, there are no approved methodologies for use in the DoD to evaluate medium and high robustness OSes. Currently, there are three groupings of authority for the assessment of NSS: DoD, IC, and the U.S. federal government agencies that are not part of the DoD or IC.

Previously, each of these authorities levied different IA requirements, security controls, and risk management approaches on NSS. During the last several years, there has been an effort called Certification and Accreditation Transformation to create a single set of IA requirements and controls (NIST Special Publication (SP) 800-53) for all three authorities. The effort is still a work in progress. Some possible reasons why this transformation effort is still ongoing:

- Not all entities within the DoD and IC have implemented the single set of controls
- The controls do not allow for assessment of a product during design or development.
- The effort was mandated before the IA requirements and controls were finalized.

- The implementation of the requirements and controls were limited to a completed desktop solution in an administrative operational environment.

The environments of operation and the sensitivity of the data are two items influencing the differences in risk management approaches. There are clusters of authorities under these three groups of authorities that are continuing to press the concept of a single assessment methodology among the three groups. However, several decisions regarding the CCEVS are greatly affecting this combined approach.

The Certification and Accreditation (C&A) Transformation effort by the DoD and IC attempted to create an atmosphere of reciprocity amongst tenant organizations for the assessments of those requirements and controls. The Committee on National Security Systems Instruction (CNSSI) 1253 was the policy initiating the C&A transformation by consolidating the security controls within DCID 6/3, DODI 8500.2, and the NIST SP 800-53 into a single repository of security controls contained within NIST SP 800-53 Revision 4. NIST SP 800-37 defines a single C&A process that consolidates DCID 6/3, DODI 8500.2, and NIST SP 800-53 processes.

The transformation effort has yet to produce the expected reciprocity of assessments among the different authority groups. Although the NIST SP 800-53 Revision 4 security controls are designed to be customizable, so that newer technologies could be assessed using these controls, the focus of the controls is on workstation systems, which again does not cover all environments. However, NIST SP 800-53 is being updated to include assurance and controls for assessments of more than just workstations.

As previously mentioned, another transformation occurred several years ago regarding the processes to assess CDS. The UCDMO consolidated three processes to form the DoD/IC CDS Process: SABI, TABI, and TSABI.

One of the goals of the combined process is to move CDS from individual, isolated environments of operation into a cloud-based environment, allowing for greater control and visibility into data movement among the differing security domains. Previously, if an isolated CDS was connected to special purpose systems, such as a shipboard navigation system, it was considered PIT [DoD07] and it did not

complete the CDS process. Currently, a CDS that fits criteria similar to PIT are being included in Draft DODI 8540.01aa, which is a UCDMO sponsored DoD policy.

1.3 Motivation

Between the C&A Transformation and the changes to the Common Criteria (CC), the DoD and IC assessment community are in flux for all levels of robustness for stationary systems, let alone other environments of operation. In addition, there is an argument that the previous and current assessment methodologies did/do not achieve the stated robustness because the assessors were/are delivered completed products. The argument is that assessing a completed product does not allow for sufficient insight into the product to determine its stated robustness. Therefore, to provide the confidence that the system operates as it should, sufficient insight and engagement must occur during the development lifecycle. This information, combined with the inconsistent backgrounds and education of ISSEs, assessors, and AOs, provides further evidence of a key gap in existing assessment methodologies.

As such, there is an opportunity to provide an assessment methodology, which includes mathematical models, for all environments of operation that can be combined with the current and future assessment methodologies. This new methodology improves the confidence in the system by integrating an assessor into the development process to achieve greater insight, improves cost savings by preventing duplicate assessments, and reduces the time it takes to conduct assessments by allowing future assessments to build on past assessments. These benefits occur as a direct result of the implementation of the models within the new methodology, as well as existing methodologies. To evaluate the new methodology and models, an assessment will be conducted by a team, which has mixed experience in assessing cyber systems, implementing the new methodology and models.

Although, one would assume that this area of study has been well researched, historically it has not. The fact that only the US Government has been formally requiring assessments of cyber systems, until recently, may have limited such research, as US Government personnel do not generally publish. Also, unless

someone has participated in an assessment, they are not likely to understand the processes, barriers, and concerns.

The goal of this dissertation is to provide a basis for an assessment methodology and models for those systems for use in the DoD and IC. The new methodology is not meant to subsume existing methodologies. In realizing this goal, this dissertation has achieved the following objectives:

- Determined that a gap exists in national policy and assessment techniques for providing objective assurance evidence to the appropriate risk decision authorities. A search of relevant literature found no work that:
    - Mathematically modeled flaws, vulnerabilities, countermeasures, threat sources, impact, probabilities (which are not mathematical), attack vectors, and risk from an OE and situational perspective.
    - Defined and mathematically modeled Threat Source, Threat Source Motivation, Threat Source Capability, and Attack Vector, and Attack Source as a function.
    - Defined and mathematically modeled Threat as a function of the Threat Source's Tactics, Techniques, and Procedures using an Attack Vector to exploit an Attack Surface.
    - Defined and modeled Risk as the probability, which is not mathematical of a result occurring from a Threat against a situational instance at an opportunity in time for a specific motivation.
    - Defined and modeled Impact of Risk impacting operations.
- Determined a lack of national policy and methodologies for providing sufficient assurance evidence to the appropriate risk decision authorities for medium and high robustness systems. A comparison of the existing assessment processes found the following:
    - DCID 6/3 process has the capability but does not explicitly require specific evidence but rather it is left to the assessor and authorizing official (aka accreditor).

  - UCDMO CDS process, though currently implementing the C&A Transformation 800-53 controls, is not able to create sufficiently acceptable profiles for CDS.
  - C&A Transformation has not materialized as expected and does not address high robustness systems.
  - CCEVS has no protection profiles approved by the U.S. Government for medium and high robustness systems allowing for evaluation by the U.S. Government agencies.
- Introduces an assessment methodology that complements existing assessment methodologies, separate technical and operational environment methodologies, and introduces situational perspective. This consists of the following tasks:
  - Decompose generalized existing methodologies.
  - Introduces a new assessment methodology.
  - Decompose technical and operational environment methodologies
  - Introduce situational perspective
- This dissertation introduces a technique that provides mathematical assessment models. The assessment models consist of the following tasks:
  - Introduce new assessment models for vulnerabilities, countermeasures, threat sources, attack vectors, attack surfaces, probabilities (which are not mathematical), impact, and risk from a technical, OE, and situational perspective.
  - Introduce a new model of Threat Source, Threat Source Motivation, Threat Source Capability, Probabilities (which are not mathematical), Attack Vector, and Attack Surface as a function.
  - Introduce new model of Risk as the probability, which is not mathematical, of a result occurring from a Threat against a situational instance at a opportunity in time for a specific motivation.
- Using the lessons learned from the decomposition and model tasks, this dissertation introduces new assessment methodology. This new methodology provides increased assurance evidence, as evidenced by the results discussed in Chapter 5, thereby decreasing the risk assumed by the approving authority. No

current methodology provides such a level of evidence as this new methodology. This objective consists of the following task:

  - Develop a new assessment methodology that complements multiple existing assessment methodologies.

- This dissertation evaluates the assessment models, as well as evaluates the subsequent methodology by conducting an assessment of a system using these models and methodology. The validation evidence is included within Appendix B and the results of the validation are discussed in Chapter 5. The metrics to measure the methodology are its usefulness, objectiveness, and it is a useful guidebook to assessors of varying experience levels.

1.4 Dissertation Overview

This dissertation discusses assessment techniques, and suggested modifications of those techniques to improve the assurance of systems, as well as reduce time required to conduct system assessment. Chapter 2 covers the evolution of assessment methodologies and techniques. Chapter 3 provides detailed models for assessment. Chapter 4 covers existing assessment methodologies and introduces new assessment methodology using the models from Chapter 3. The conclusions of this dissertation are presented in Chapter 5 with possible future work detailed in Chapter 6.

Chapter 2 Assessment Decomposition

The information provided in this chapter and in this dissertation has been researched using the sources available to the public. The nature of assessments to this point has been focused and formulated by the US Government and as a result very little research has been conducted into security assessment methodologies. There is a plethora of research into design of secure systems and techniques into compromising such systems, but not into actual security assessment models and methodologies.

This chapter presents some basic security goals of assessments within the DoD and IC. This is followed by some of the basic IA concepts, such as Defense in Depth (DiD) and Defense in Breadth (DiB). Then, the methodologies of assessments currently in use within the DoD and IC are presented. Finally, additional concepts are introduced based upon those current concepts and methodologies.

The DoD and IC missions must continue to operate, regardless of the presence of a security compromise. The DoD and IC conduct system assessments to determine the level of assurance of the instantiation of a system within a specific site or operational environment to do just that. This is done to provide an understanding of the risk that system represents to that site and that site in turn to the greater DoD and IC enterprises. Essentially, these assessments provide the DoD and IC with the combined measure of the breadth and depth of defenses resident in its systems. DoD systems are supposed to be designed to implement a combination of defensive methodologies, the two most common of which are DiD and DiB.

DiD, illustrated in Figure 1, is the layering of protection mechanisms, generally technical in nature, working from the outside boundaries into the smallest defensible layer, which are the software applications [CNS10] [Sma11] [Kew13]. The layers illustrated are from the Gateway to the software applications, but those layers are adjustable based upon the complexity of the system, with the smallest defensible layer always being software applications.

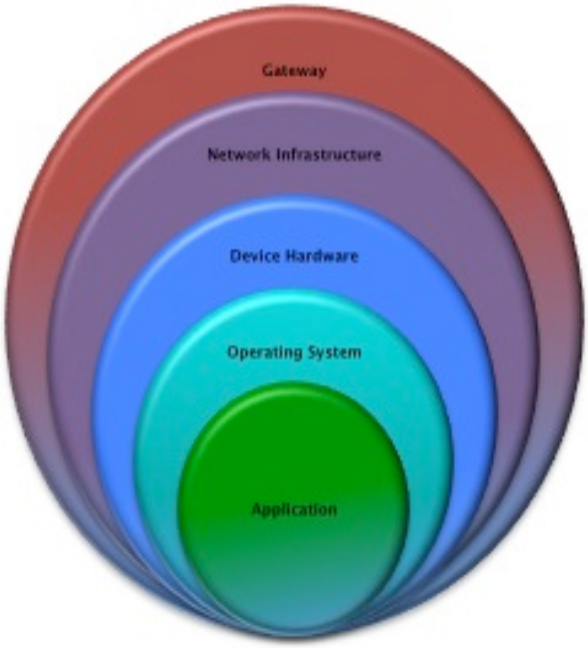

Figure 1 Defense in Depth

DiB, illustrated in Figure 2, is multiple mechanisms (the small circles), both technical and non-technical, within a single layer of defense (the large circle), increasing the robustness of that single layer [CNS10] [Sma11] [Kew13] [Cle13]. DiB provides mechanisms to mitigate the dependencies among layers, thereby reducing the attack surface for that layer [CNS10] [Cle13].

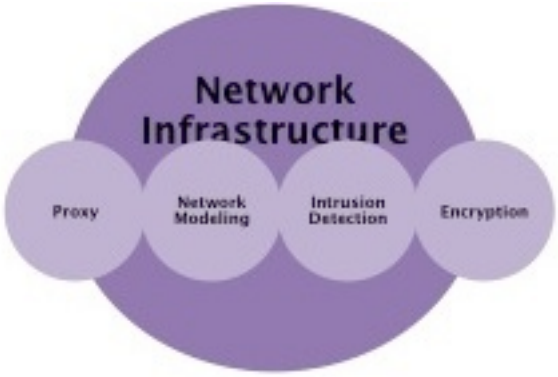

Figure 2 Defense in Breadth

Within the DoD and IC, assessments are conducted against a single instantiation of a system of the combined DiD and DiB protection methodologies, which are illustrated in Figure 3. This assessment combines two types of testing conducted during at least two, possibly more, test events. The first type of testing, referred to as a technical assessment[14], is conducted on an instantiation in the lab and is conducted prior to placement of the system in the operational site setting. Technical assessments focus on the assessment of the technical assurance aspects of the system, however, the technical testing is always from, and includes, the security aspects of the operational site. The second type of testing, referred to as an operational site assessment, is conducted once the system is instantiated at the operational site, and may include physical connection to live networks. The operational site assessment focuses on the assessment of the assurance of the instantiation site, as well as the technical aspects of the robustness system within the site's environment.

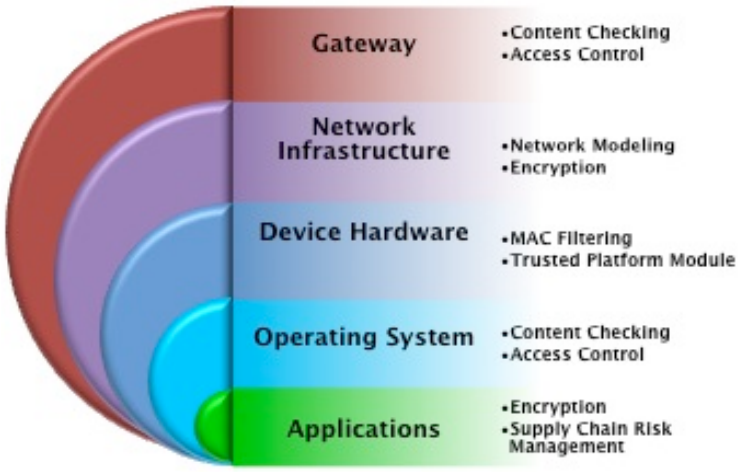

Figure 3 Defense in Depth and Breadth

Although these assessments do not delineate between technical testing and operational site testing, the IA controls, which are the security aspects verified by these assessments, can be separated into two groups. One group of controls is specific to the technical capability of the system, such as DCID 6/3 ResrcCtrl, which specifies

[14] A technical assessment evaluates only the device or system and its capabilities, not anything organizationally or environmentally based.

that an object does not contain any residual data from the former subject prior to being assigned, allocated, or reallocated by the Security Support Structure. The other group of controls is specific to the system's operating environment. An example of such a control is DCID 6/3 control Access 1 [Dir99], which specifies the physical access to the system. The delineation of DiD and DiB is currently expressed by DiD being applied to technical controls and DiB[15] being applied to operational environment controls [NIS13].

Unfortunately, NIST SP 800-53 [NIS13] security controls, which were supposed to provide this separation, regularly mix technical and operational assurance requirements in the same control. An example is Access Control 17 (AC-17). Its first enhancement for high robustness requires the operational site to monitor remote access, which is a control implemented by the operational environment or site, usually by reviewing audit logs of the system and site. The third enhancement of AC-17 requires the high robustness system to route all remote access through a limited number of access control points, which is a technical control implemented by the OS.

To allow better reuse of systems and assessments, there should be a distinct delineation between technical and operational assessments. This would separate not only the assessments, but also the countermeasures and risk, and allow for a correct implementation of DiD and DiB architectures. Currently, risk analysis combines technical and operational threats and countermeasures, which are based upon the first operational instantiation of the system. Even though a risk analysis is conducted for each instantiation of the system in a specific operational environment, the risk associated with medium and high robustness systems is usually that of the first operational instantiation's risk analysis.

An example that validates the separation of technical and operational assessment is the CDS product MultiLevel Web (MLWeb). This is a product from Naval Research Laboratory (NRL) that included formalisms. Unfortunately, the first instantiation was for SABI, which is considered, at best, a medium assurance

[15] Defense in Breadth is a planned, systematic set of multidisciplinary activities that seek to identify, manage, and reduce risk of exploitable vulnerabilities at every stage of the system, network, or subcomponent life cycle (system, network, or product design and development; manufacturing; packaging; assembly; system integration; distribution; operations; maintenance; and retirement). NIST Special Publication 800-39, at H-4, n.78.

assessment methodology. As a result of the SABI assessment[16], there was a misconception that MLWeb was only medium robustness, when in fact the evidence available and presented was sufficient for high robustness.

If a technical assessment had been conducted on MLWeb, then it would have been approved at high robustness using existing assessment methodologies (minus the operational influence) and the risk would have been technically based, not operationally based [NRL13]. Separating the technical assessment and risk from operational assessment and risk would have prevented the misconception that cost the government millions in recertification and instantiation costs.

The hypothesis of this dissertation is that assessors can separate the technical assessment from the operational site assessment. The work presented here defines the process for separating these assessments. This dissertation also provides a simple demonstration of the benefits of separating technical from operational assessment; a comparison of an example assessment of a system that was assessed using the existing methodology and assessed using the separated methodology. The comparison demonstrates that the separated approach provides a more accurate risk assessment.

The existing assessment methodology focus is on stationary systems, such as desktops and servers. The assessor, with the approval of and in coordination with the approving or accrediting authority, subjectively determines the approach to assess a vehicle. This approach is based entirely upon the assessor's education and experience. Whether the system is an aircraft or a desktop PC, however, certain aspects of each component must be considered in an assessment of the overall system.

The aspects of an assessment are: Artifacts (2.1), Threats (2.2), Flaws (2.3), Vulnerabilities (2.4), Countermeasures (2.5), Attack Vectors (2.6), Probabilities (2.7), Impact (2.9), and Risk (2.8). These aspects are detailed in the following subsections.

2.1 Artifacts

Artifacts are any documents, diagrams, mathematical proofs, testing, and ISSE's notes of each component of the system and the overall system that provide the

[16] This evaluation documentation is not available outside US Government control

assessor(s) and decision makers with the assurance of the robustness in the system. Currently, the majority of systems do not have an ISSE involved in the entire lifecycle of the system, if there is an ISSE involved at all. Thus, assessors and decision makers must rely on documents and designs provided by the vendor, which are biased to provide the assurance specified by the vendor. The vendor may provide mathematical proofs and those proofs can be verified by the assessors or an independent third party, however the assessor must then confirm that the vendor provided an accurate correspondence between those mathematical models and the actual instantiation. Testing of a system and its components, however, can only provide assurance regarding the exploits being exercised against known vulnerabilities.

2.2 Threats

The aspects of threats[17] and the threats discussed within this section are a small sampling of the plethora of articles, papers, etc authored by academia, industry, and government sources. Additional information regarding threats is detailed in Section 3.4.

Every government is concerned about hardware, firmware, or software products that were designed or developed, at least in part by citizens of countries other than their own, including those that may be resident in their country. These products are considered to have foreign involvement and are to be considered foreign sourced. Due to globalization, these products are rarely developed or built in the same country where they are designed. Globalization is a cause of multiple threats with regards to hardware, firmware, and software. Supply Chain Risk Management (SCRM) identifies, possibly mitigates, and monitors such threats [Geo12]. These threats are not limited to the US Government.

If the product design [Geo12] was modified prior to building the product, but after it left the design team's management, there are several concerns to be considered. Vulnerabilities to certain exploits could be designed into the product, thus allowing a "back door" not found in the original design. Any of the three main

[17] Per http://www.oxforddictionaries.com/us/definition/american_english/cyberthreat, a threat is the possibility of a malicious attempt to damage or disrupt a system.

components of a system (hardware, software, and firmware) could be designed to send information to a specified location, or "phone home". If the purchaser is known, a Denial of Service (DoS) attack could be built in, causing a system to shut down on a specific day or time for a specified length of time. This would deny the users access to that system, allowing a conventional attack to succeed, which might otherwise fail.

This type of espionage is quite malicious, because the exploits above could still occur even if the component was produced in the country of design. Espionage also allows the possibility of the design to be covertly exported. Not only does this allow the adversary to know possible vulnerabilities, but the ability to improve the design or incorporate specific defenses into existing systems.

There are countermeasures for the modification of the design when it is in development. One method is to implement the principle of least privilege and give only enough of the design for a component to be instantiated, but not the entire design. This would prevent the entire design from being known, as well as creating a black box development environment. In addition, formal assessments of the design could be used prior to putting the design into the formal production chain. For example, there is an effort to formally assess the RTL register transfer logic for the Centaur processor (an Intel clone) [Slo11]. The RTL is the description of how registers (could be a flip flop, memory bank, single port, etc.), are updated, i.e. state changes caused by the instructions.

There are few possible countermeasures for the compromise of the hardware foundation. Both DiD and DiB could mitigate some of the threats against hardware and firmware. It could prevent phoning home by defensive router access control lists, which could possibly prevent exploits if the expected ports are disabled. However, a DoS attack has limited countermeasures.

2.2.1 Espionage

Espionage is a constant threat. It could be corporate or state sponsored external espionage, or an authorized individual that intentionally releases data without approval (insider threat). All components suffer this threat.

Most people in the U.S. know that the hardware components of their laptops, tablets, etc are manufactured outside of the U.S. Some companies have processes to confirm that chips are manufactured to the specific design, as referenced above, provided to the manufacturer without any additional capabilities. Many people categorize such espionage as a supply chain threat [Fil12][i]. It wasn't until the late 1990's when the US Government shifted from a majority (80%) of government built systems or Government Off The Shelf (GOTS) to a majority of Commercial Off The Shelf (COTS) (80%) that supply chain became the pervasive, persistent threat it is today [Bar13]. There are many papers on the topic of supply chain threat [Cha12] [Jac12] [Geo12] [Ias13]. One of the more ironic papers, given subsequent articles by Edward Snowden for The Guardian, was by E. Iasiello, who listed cyber espionage and terrorism as the top cyber threat in an article for The Guardian [Gre13].

Everyone should be concerned about the source of software, even the general public user. Foreign involvement exists when any component is designed or developed, at least in part, by citizens of countries (regardless of residency) other than the one in which the component is utilized. Therefore, any software that has foreign involvement must be considered foreign source software.

Whether a system is an open architecture, an open system, open source software, or proprietary, foreign influence is a threat. *Open architectures* are those whose specifications, either as officially approved standards or privately designed, are made public by the designers. An *open system* typically employs consensus based standards and modular design. *Open source software* refers to any application developed as a public collaboration or whose source code is made available to be freely shared, used, modified, improved, or redistributed. Open source software may be sold or licensed as a commercial product or may be distributed at no cost. The antithesis of open source is proprietary. *Proprietary software* is owned by a commercial entity, and sold or licensed. Proprietary software may not be redistributed without permission of the creator of the software.

Red Hat Enterprise Linux v5 (RHEL5) is open source, but distributed by Red Hat for a license fee [Red12]. Occasionally, software that was previously produced only in proprietary form, becomes available as open source. For example, Solaris

went from proprietary source in Trusted Solaris 8 by Sun Microsystems to open source in Oracle Solaris 10 [Ora10].

Everyone is concerned about an adversary introducing malicious code into software or subverting its security protections. During open source software development, anyone can make changes, but changes to the source tree generally occur via a community vetting process. Commercial, proprietary software normally has no such "community" vetting process. Therefore, there is an effort within the US Government to verify security and functionality of COTS products [Cha12]. In the past, proprietary software was most likely developed using in-house development (i.e., no foreign involvement). However, this is no longer true, as most companies have at least part of their development off-shore. For example, Microsoft has development units in India [Wik16]. In addition, many developers use libraries or code obtained from unknown sources.

2.2.2 Malware

Malware, or malicious software, such as a Trojan Horse[18], is intentionally malicious and subverts the intended operation, possibly covertly, of the system. The malware can be maliciously embedded in a product or can be a result of subsequent modification of the product through an attack. Maliciously embedded source code is no less likely in open source software than in proprietary software. If good Configuration Management (CM) processes are followed for both software and systems, this can decrease the potential of the insertion of malware, regardless if it is proprietary or open source. Similar to the threat of espionage, the basis of the threat of malware could come from a corporation, an individual or be state sponsored.

Proactively managing systems by deploying, base-lining, and monitoring effective standardized, security configurations improves the potential of identifying malware. An example would be the use of integrity checking software to determine if key files have been modified from the baseline configuration. CM also has the

[18] A Trojan Horse is an application that provides normal, useful functionality, but hides code that operates in a malicious manner – modifying files, propagating viruses, opening up network communications or sending data to unauthorized computers.

potential for preventing the subversion of the supply chain of software components (e.g. Libraries), as well as the potential for preventing some flaws.

2.3 Flaws

A flaw is defined in CNSSI 4009[19] as an error of commission, omission, or oversight in an information system that may allow protection mechanisms to be bypassed. The distinction between a flaw and a vulnerability, which is discussed in the next section, is very subtle and differs from person to person, document to document. By using the CNSSI 4009 definitions for both, the subtleness is captured in a single concept, exploitability.

2.4 Vulnerabilities

There are two major categories of IA concerns: vulnerabilities and malware. A vulnerability is defined by CNSSI 4009 as a weakness in an information system, system security procedures, internal controls, or instantiation that could be exploited by a threat source.

The majority of source code vulnerabilities, such as buffer overflows, can be eliminated if proper software engineering/programming practices are followed [CER13]. CM reviews can identify configuration-based vulnerabilities. However, as the number of lines of source code increases, the ability of verification and validation testing to discover vulnerabilities decreases [And13]. There comes a point where testing becomes an intractable problem and latent vulnerabilities in the software become increasingly probable [Mee13]. Also, and equally as likely in a large system, vulnerabilities due to improper assumptions in design, development, or instantiation may not be mitigated by good practices, processes, or procedures and may not be discovered by testing[20] since tests are often written from design specifications [Gre13].

Reliably detecting code defects is difficult and detecting well-engineered, embedded, malicious content is even more difficult. Most significant software applications are at least 1 million lines of code in size. Windows XP has 45 million

[19] CNSSI 4009 is the National Information Assurance Glossary.
[20] Testing includes functional testing, security testing, static and dynamic source code analysis.

lines of code [Mic12] and RHEL7 has 30 million lines of code [McP08]. The NSA estimates a person can formally review 5 to 10 thousand lines of code per year. A semi-formal review only increases this by a single order of magnitude. Code reviews with the assistance of tools can review more lines, resulting in lower levels of assurance. However, this number is inversely related to the overall length of the source code. As the number of lines of source code increases, the complexity and the misuse of the principle of least privilege increase, and subsequently the number of lines of source code that can be reviewed per year decreases.

There are tools available that assert the ability to locate malicious and malformed code. However, a well-crafted, maliciously, embedded compromise will not be detected by these commercial tools. As such, it is not feasible to reveal all malicious intent within source code with commercial static and dynamic assessment tools. There is an excellent example from a US Naval Postgraduate School project to subvert a kernel in as few lines as possible. A student inserted 8 lines total (5 lines in one location, 3 in another) into the Linux kernel and successfully subverted the kernel [And13].

2.5 Countermeasures

In NIST SP 800-53 a countermeasure is defined as any actions, devices, procedures, techniques, or other measures that reduce the vulnerability of an information system. There are countermeasures for the risk in the use of both proprietary and open source systems and in the software development process. Verification and validation testing conducted on proprietary and open source systems during an assessment or certification should reveal known vulnerabilities, and countermeasures can be instituted prior to instantiation. Both can follow good CM processes and software engineering practices. Open source approaches do allow for the potential of discovering how data is being processed, the protocols being utilized, and communication channels. With this knowledge, countermeasures may be put in place prior to the software's implementation. In reality, it must be assumed that all software, including development tools, contain foreign source software, regardless of whether it is open source or proprietary development.

Although there is open source firmware and hardware, the majority of both are proprietary. There are countermeasures, but the countermeasures require vendors to share proprietary information with those that validate or vet these products. With the recent revelation of cooperation between the US Government and major product vendors, it is likely that vendors may find it difficult to justify to their shareholders such future cooperation [Gre13].

These risks, combined with information sharing that links more information infrastructure together and the fact that the majority of systems are low assurance, are increasing the need for stronger software, firmware, and hardware assurance. Remember, the security of a system or a network is only as good as its weakest link. A prime example of a weak link is the Heartbleed vulnerability, which was contained within an extensively examined and used open source cryptographic library [Gru14]. This library is used the world over by entities ranging from banks to hotels to governments to implement the Transport Layer Security (TLS) protocol, thus compromising financial, political, and commercial systems around the globe using a security mechanism that most viewed as trusted.

2.6 Attack Vectors

While a system may have vulnerabilities and a threat source may desire to exploit those vulnerabilities, a physical mechanism must exist by which that exploit may be conducted against those vulnerabilities. Attack vectors are those physical (analog or digital) mechanisms, whether persistent or one time, that allow vulnerabilities to be exploited by threats.

2.7 Probabilities

In this dissertation, probabilities are defined as in the Oxford Dictionary, which is something is probable or likelihood of something happening and not the mathematical definition. During an assessment, there are a multitude of probabilities to consider. However, this dissertation only focuses on those regarding exploitation of systems. Of those, the three probabilities to be discussed are:

- The probability that a threat source will attack the system

- The probability of the success of the attack against the system
- The probability of certainty of the knowledge of the threats, flaws, etc

The first two probabilities are quantified by the third probability, which is driven by the extent of the assessor's exposure to threats and their knowledge of the system under assessment. The greater the exposure to threats and their capabilities, and the greater the knowledge of the system, will allow the assessor to provide a more accurate assessment of the assurance of the system, thereby provide a more accurate assessment of the risk to and of the system.

2.8 Risk

In the US Government, risk drives the selection of required security controls for an information system, because it is viewed that the controls will protect an organization's operations and assets [NIS12]. The old model of Risk = Threats x Vulnerability is no longer sufficient to describe a system's risk to the DoD. The prior sections of this dissertation indicate some of the complexity in determining risk. There is no single adversary to defend against. Increasingly, threats are being categorized based upon the amount of funding available to the adversary to incorporate the well-funded individual or cell not affiliated with any corporation or government.

Assessments are conducted to determine the risk of instantiating a system. As previously indicated, there isn't a consistent methodology in use today within the DoD and IC for determining risk. As previously described, there is no single assessment methodology. More agencies and services, but not all, are starting to use the NIST SP 800-59 Risk Management Framework (RMF). The UCDMO, one of the most visible exceptions, uses the Risk Decision Authority Criteria to determine risk [Byr10]. Interestingly, one of the goals of the UCDMO is to improve reciprocity among agencies and services with regards to CDS. Considering that the UCDMO was established in 2006, the reciprocity has been slow to occur and now with all other agencies and services implementing RMF and not the UCDMO, it is even less likely to occur. However, it is rumored (no documentation to date) that for the first time ever,

as of 2014, any CDS that is approved in the TSABI process will automatically be granted a SABI approval. Also in 2014, the UCDMO became the Unified Cross Domain Services Management Office (UCDSMO).

2.9 Impact

The impact of an exploit has generally been included within the risk assessment. However, as systems which were previously isolated are interconnected to improve information sharing, the impact is no longer limited to a single system as it was in the past. As such, impacts must be considered separately from risk and at multiple levels ranging from the single system to the entire Internet.

2.10 Assessment

There is no one size fits all system, nor is there any single situation in which all systems can be instantiated. Each situation must be assessed to determine if the situation requires high, medium, or low robustness. As previously discussed, the CC EAL maps systems to levels of robustness. EAL 1 and 2 are low assurance and an example would be the Citrix Presentation Server 4.5 at EAL2+[21]. Medium assurance is EAL 3 and 4 with RHEL5 at EAL4+ as an example. Of note, it seems that more US companies are having labs in the United Kingdom conduct NIAP assessments because there is a misperception that those labs do not conduct as rigorous an assessment, and therefore certification is easier. High assurance is EAL5 and above, with the XTS-400 at EAL5+ as an example. The most current assessment for the XTS-400 was conducted in Canada, as the US is no longer evaluating systems above EAL2.

If there is a requirement for software to perform only what is specified without fail, then high robustness is required. This is known in the aviation community as safety critical, and known as security critical in the assurance community. As formal methods are normally conducted only on the security relevant aspects of software or hardware, formal methods do not necessarily mitigate foreign involvement or incorrect/invalid assumptions. In verifying a microprocessor, why model just

[21] The plus sign, +, indicates the EAL was *augmented* to include assurance requirements beyond the minimum required for a particular EAL.

microcode, why not the gates or the silicon that makes up the transistor? Does modeling that level of detail give you more assurance? There is a cost benefit trade-off for each layer modeled because each layer exponentially increases the costs. The recommended layer to stop modeling is where security decisions stop (layer n), which is also the layer where the "always invoked aspect" stops. In the microprocessor example, microcode is the lowest layer that enforces the security architecture. The micro-architecture (layer n-1), however, is security agnostic, so the belief is that it is not beneficial to formally assess this layer. However, the mechanisms at this layer implement the higher-level security mechanisms. Whichever layer is modeled, the formal proofs are assessment artifacts that can be reviewed by the formal methods and assessment communities.

As stated earlier, artifacts are the basis for any assessment, including certification and accreditation efforts. These artifacts range from formal proofs to penetration testing. For all levels of robustness, but specifically medium and high robustness, security concerns must be included starting at the design and cannot be realized by any amount of testing once the system is instantiated.

Artifacts can also be categorized as either technical or environmental, related to the environment in which the system operates, which this dissertation refers to as operational environment (O). An example of a technical artifact is a report from penetration testing conducted in the laboratory assessment of the system. Whereas, the network vulnerability assessment report is an example of a operational environment artifact.

A technical aspect is any aspect directly attributable to the system regardless of the operational environment. Whereas, operational environment aspects are those attributable to the physical space where the system is located. Both technical and operational environment aspects can be further subcategorized.

Security verification testing, including penetration testing, can only exploit what is known and cannot prove that a system does only what it is designed to accomplish. Only artifacts representing the security aspects of the design and instantiation can give insight that a system does only what it is designed to accomplish. These artifacts do not have to be formal proofs. Use of tools and peer reviews by an independent security professional can indicate if good security coding practices were followed

during development. However, mathematical proofs are not interpretive/subjective. So, regardless of the assessor's experience and opinion, the assurance provided by the proofs is consistent. Therefore, mathematical proofs are always beneficial for the assessment.

2.11 Conclusions

To allow better reuse of systems, there should be a distinct delineation between technical and operational assessments, controls/countermeasures, risk assessment, and artifacts. This would not only separate the assessments, but also the countermeasures, risk, and the impact. It would allow for greater reciprocation of system assessments because many times the operational environment, usually the classification/compartment of the data, is classified and so the test cases associated with that environment are also classified. However, it is uncommon that the system itself is classified and therefore technical assessment test cases could be shared more freely. Currently, the risk analysis combines technical and operational threats and countermeasures, which are based upon the first operational instantiation.

Chapter 3 Assessment Models

This chapter presents models for a number of aspects an assessor must consider when assessing a system, regardless of its complexity or connectivity. As an assessment is rarely a single, continuous event, these individual models are iteratively addressed so the assessor is able to represent each impression of the system's capabilities, correlate the models to the evidence, and provide a level of assessment detail previously achieved. As the assessor's knowledge of the system increases, the content of these models will go from generalized to specific as the assessment progresses.

Instead of an a priori risk determination, operational risk should be determined by the operational assessors of the system based primarily upon the technical risk derived from a technical assessment and further characterized by their operational assessment. Currently, a vulnerability assessment, as defined by CNSSI 4009, is the systematic examination of an information system or product to determine the adequacy of security measures, identify security deficiencies, provide data from which the effectiveness of proposed security measures can be predicted, and confirm the adequacy of such measures after instantiation [CNS10].

A formula expressing risk in this context would greatly assist assessors. A good start is the well-known formula:

Risk (R) = Threats (T) x Vulnerabilities (V) x Impact (I) [Cox08]

The impact is in terms of the value of the asset with the vulnerabilities and the associated threats [Nic02]. An asset's value is variable and can be based upon multiple aspects, including the perceived importance to the mission, the perceived importance to the adversary, our ability to replace the asset, and time.

The major components of this formula will be further specified to more accurately reflect the flaws, countermeasures, vulnerabilities, threats, probabilities, attack vectors, impacts, and risk on the operational environment. Flaws are modeled based upon whether they are technical or operational in nature, and if there is a

known exploit or not. Vulnerabilities are defined as those flaws with perceived, partial, or no countermeasures.

The operational environment is more than just a fixed, land base operational environment, such as a workstation or NOC. A specific operational environment is a situational instance or state, which reflects a physical characterization of the operational environments (such as an aircraft in flight vs an aircraft parked on the deck of a carrier). Each situational instance/state will be individually modeled for countermeasures, vulnerabilities, threats, probabilities, attack vectors, impacts, risk, and effects to greatly improve the conciseness and objectivity of evidence for assessment, to provide increased reciprocation among assessors, and justification of effort (cost, etc.) for design and assessment of systems. The proposed set of categories is as follows:

Vehicle

- Air (VAR)
- Land (VL)
- Submersible (VU)
- Afloat (VAF)
- Space (VS)

Stationary/Land Based

- Unsecured area computer (Server, Workstation, or Laptop) (LU)
- Secured area computer (Server, Workstation, or Laptop) (LS)
- Unsecured area NOC (LNU)
- Secured area NOC (LNS)

These categories are broad and generalized. A more specific detailing will define a situational instance or state, which reflects the physical situation of a particular operational environment. These individual models will comprise the overall assessment model. As described in Chapter 4, the individual models may be iteratively developed, fulfilling the needs of the assessor to represent their initial impression of the system's capabilities, represent the system's capabilities as it is

assessed, and finally, to representatively correlate or map the completed models to the empirical evidence of the assessment.

The following sections provide the aspects detailed for each of above characteristics. There are many possible ways to represent assessment aspects. It is not a stretch to imagine each assessor has their models that they utilize. The models provided are just one way to convey the information in a consistent, easily understood manner. The following sections contain examples that are included to provide realistic modeling, but are not actual representations of the systems modeled.

3.1 Flaw models

At the start of an assessment, the flaws of the system may or may not be known. As such, the assessor must use the following model to conduct something similar to a high-level flaw hypothesis methodology. As the assessment is conducted, and flaws are identified, the model will be updated.

There are many ways to categorize flaws. In this work, we define flaws in terms of three orthogonal taxonomies: origin, vulnerability to exploitation, and existence of countermeasures.

Flaw Origin: Flaws (F) exist in both the technical system and in all environments of operations and are defined as:

F represents the set of all flaws

$F_T \subseteq F$ is the set of technical flaws

$F_O \subseteq F$ is the set of operational environment flaws

An operational environment may already contain one or more systems, and as such, is expected to contain technical flaws that are separate, but not necessarily distinct, from a technical flaw of a system being integrated into it. The technical flaws are separate in that they exist on a system that is already part of the operational environment and not the system being integrated into the operational environment, but are not necessarily distinct because both systems could be identical layer 3

switches performing different functions within DiD and/or DiB. While the flaws may not necessarily be distinct, the identification and mitigations must be distinct. In this work, it is assumed that all flaws can be designated with an origin that is either technical, environmental (which may contain existing technical flaws considered within the operational environment), or may be designated uncategorized (which includes unknown technical or environmental flaws and will be represented as F_U), and therefore:

$$F = F_T \cup F_O \cup F_U$$

Flaw Exploit Classes: In addition to categorizing flaws by origins, we can further categorize them into exploit classes, specifically those with known exploits (E) and those without known exploits (!E). It is assumed that uncategorized flaws, specifically the unknown flaws, are a subset of those without known exploits.

$$F_U \subseteq F_{!E}$$
$$F = F_{!E} \cup F_E$$

Combining Categorizations: Due to the orthogonal nature of our taxonomies, we can now further categorize flaws in terms of more than one of the taxonomies:

The technical flaws can be subdivided into those with known exploits ($F_{T,E}$) and all other technical flaws ($F_{T,!E}$) where:

$$F_{T,E} = \{f \mid f \in F_T \,/\backslash\, f \in F_E\}$$
$$F_{T,!E} = \{f \mid f \in F_T \,/\backslash\, f \in F_{!E}\}$$

Therefore, technical flaws are defined as:

$$F_T = F_{T,!E} \cup F_{T,E}$$

Similarly and generically, operational environment flaws are those with known exploits ($F_{O,E}$) and all other operational environment flaws ($F_{O,!E}$)[22]:

$$F_{O,E} = \{f \mid f \in F_O \wedge f \in F_E\}$$
$$F_{O,!E} = \{f \mid f \in F_O \wedge f \in F_{!E}\}$$
$$F_O = F_{O,!E} \cup F_{O,E}$$

System Specific Flaw Categorization: The preceding gives us the categorization of all possible flaws. However, for a specific system, a subset of flaws will be examined that are relevant to that system. For example, assuming system s1, the following notation will be used:

F_{T1} = {f | f ∈ F_T ∧ f is a technical flaw for system s1}
F_{O1} = {f | f ∈ F_O ∧ f is an operational environment flaw for system s1}

The notations $F_{T1,E}$, $F_{T1,!E}$, $F_{O1,E}$, $F_{O1,!E}$ will be used similarly.

The following is an example of the possible groups of flaws in a laptop computer (s1) in a hotel lobby, which, by the naming convention, implies a certain level of functionality, connectivity, and exposure. The notation below identifies technical and operational aspects of the system as flaws. As an example, F_{OS} represents flaws in an OS. Some of the possible flaws, which are identified as either technical or operational:

Technical

- Operating System including unencrypted password files (F_{OS}) ⊆ F_{T1}
- Anti-Virus including not-up-to-date AV signatures (F_{AV}) ⊆ F_{T1}
- Bluetooth including enabling Bluetooth ($F_{Bluetooth}$) ⊆ F_{T1}
- Browser including not implementing latest version of browser ($F_{Browser}$) ⊆ F_{T1}
- Email including allowing message preview (F_{Email}) ⊆ F_{T1}
- System firewall including not enabling firewall ($F_{Firewall}$) ⊆ F_{T1}

[22] We do not have to do the same for uncategorized flaws since they are a subset of the flaws with no known exploits.

- Logging/Auditing including not protecting log files from unauthorized modifications ($F_{Logging}$) $\subseteq F_{T1}$
- Supply chain including counterfeit software (F_{Supply}) $\subseteq F_{T1}$
- Wireless including enabling wireless modem ($F_{802.11}$) $\subseteq F_{T1}$

Operational

- Configuration Management including not patching software (F_{CM}) $\subseteq F_{O1}$
- Hotel firewall including assuming that there is a firewall ($F_{HotelFW}$) $\subseteq F_{O1}$
- Wired connectivity including not having wired connectivity when the laptop doesn't have wireless (actually disabled on some US Navy laptops) ($F_{Internet}$) $\subseteq F_{O1}$
- Wireless Connectivity including not having wireless connectivity when a laptop doesn't ethernet connection capability ($F_{802.11}$) $\subseteq F_{O1}$

Flaws within the technical implementation of the Bluetooth, 802.11, and firewall include user controlled settings such as disabling the firewall (firewalls are software based on most laptops and desktops). In this case the preponderance of flaws is within the technical set. The hotel firewall may or may not mitigate the technical flaw of the user disabling the firewall application, and the potential exists for the hotel firewall to make the technical flaw of the user disabling their own firewall more severe. The connectivity flaws are availability concerns, therefore should neither counter nor exacerbate any technical flaws.

Operational Environment Flaw Categorization: Unlike technical flaws, operational environment flaws may also be specific to a category, c, such as VAR (Air Vehicle). There are many possible categories of flaws; therefore we will parameterize the operational environment flaw specifications with a category, when needed. The following notation will be used for these (and similar notation for the flaws with no known exploits).

$$F(c)_{O,E} = \{ f \mid f \in F_{O,E} \wedge f \text{ is in category "c"} \}$$

Where "c" represents any of the categories (such as VAR), or a special category "All" or "Unspecified", which means the flaw does not belong to a specific category. In addition, the special category "ALL" contains all flaws such that $F(ALL)_{O,E} = F_{O,E}$. When we define the system we are assessing, we will be able to define the categories for that system, therefore for system s1:

$$F(c)_{O1,E} = \{f \mid f \in F(c)_{O,E} \wedge c \text{ is a category of system s1}\}$$

The following is an example of the possible groups of flaws in an air vehicle (such as Joint Strike Fighter (JSF)), which by its name does not imply any level of functionality, software applications, or connectivity. Some of the possible flaws, which are identified as either technical or operational:

Technical

- System of Systems including system interactions (F_{SoS}) $\subseteq F_{T1}$
- Operating Systems including multiple OSes (F_{OSs}) $\subseteq F_{T1}$
- Blue Force Tracker including enabling Blue Force Tracker (F_{BFT}) $\subseteq F_{T1}$
- Controlled Interfaces[23] including allowing files between security domains based on file extension (F_{CIs}) $\subseteq F_{T1}$
- Damage Controller including engaging damage control software ($F_{DamageControl}$) $\subseteq F_{T1}$
- Ejection including connecting the ejection to aircraft network (F_{Eject}) $\subseteq F_{T1}$
- System encrypted transport including Heartbleed (F_{SSL}) $\subseteq (F_{T1})$
- Ethernet including connecting a not-hardened/unpatched laptop while not in-flight ($F_{Ethernet}$) $\subseteq F_{T1}$ (EX. Maintenance ports)
- System firewalls including conflicting firewall Access Control Lists (ACLs) ($F_{Firewalls}$) $\subseteq F_{T1}$
- Logging/Auditing including not protecting audit logs from unauthorized modifications ($F_{Logging}$) $\subseteq F_{T1}$
- RF communications including enabling software defined radios ($F_{RFComms}$) $\subseteq F_{T1}$

[23] A *Controlled Interface* is a mechanism that facilitates adjudicating the security policies of different interconnected ISs (e.g., controlling the flow of information into or out of an interconnected IS).

- Weapons Systems including connecting tactical weapons systems to aircraft network ($F_{Weapons}$) $\subseteq F_{T1}$
- Wireless including enabling wireless modem ($F_{802.11}$) $\subseteq F_{T1}$

Operational

- Connectivity including requiring aircraft communicating with its manufacturer ($F_{Internet}$) $\subseteq F_{O1}$

The JSF is a complex system of systems, which are only interconnected once implemented within the aircraft; as such there are multiple interconnected systems. Weapon systems represent those systems that provide kinetic and cyber, offensive and defensive capabilities. Connectivity is an operational flaw in JSF because the air vehicle requires connectivity to maintain airworthiness.

To be clear, during the assessment of a system, we must always address the technical flaws, because the technical flaws are always part of the system, including when it is instantiated into an environment. In addition, we will need to assess the specific operational environment flaws, categorized appropriately, at the time the system is instantiated. Such flaws may expose previously unknown technical flaws, as well as operational environment flaws. Therefore, the set of flaws examined for system s1 will simply be:

$$F_{s1} = F_{O1} \cup F_{T1}$$

However, the above formula indicates flaws are static or unchanging. Therefore, the model requires expansion, by allowing these sets of flaws to be defined dynamically.

System States: The first computer was aptly named the Turing State Machine. Computers alter states every time a binary decision is completed. So, computers and networks exist in fluidity, each constantly changing. However, the current common

practice is to assess a single state, the state at which the system exists at the time of assessment.

In the 1950's, the Rand Corporation estimated nuclear explosions by modeling numerous “states” of the explosion. Similarly, the U.S. Center for Disease Control (CDC) forecasts the infection rate of a contagious disease by modeling the state or spread of infection at specific time increments, such as 24 hours, 48 hours, 1 week, 1 month, 3 months, etc [Joh09] [Bel11]. By applying these similar concepts to a system, it is possible to model the fluidity of systems' states without modeling every single state.

To model the dynamic nature of the system we define the sets of flaws for a specific system when it is in a particular state. In other words, the operational states that affect the environment of the system. The flaws for a specific state will be those possible flaws given the value of the state variables. We will use the notation $(F_{T1})_n$ to represent the set of technical flaws for system s1 in state *n*. We will use similar notations for the other sets of flaws.

To continue with the example of the JSF, a sample of possible operational states will be modeled. All of the models will share the possible baseline flaws listed above, which will be represented by $(F_{JSFBaseline}) \subseteq (F_{T1})_{JSF}$.

JSF powered up, parked, US military base within the continental US $(F_{S1})_{JSFParkedUS}$

Technical Flaws

- Baseline $(F_{JSFBaseline}) \subseteq (F_{T1})_{JSFParkedUS}$
- RF communications including enabling to RF communications while the aircraft is in maintenance $(F_{RFComms}) \subseteq (F_{T1})_{JSFParkedUS}$

Operational Flaws

- Maintenance connectivity including connectivity to manufacturer every 30 days to maintain airworthiness $(F_{MaintCx}) \subseteq (F_{O1})_{JSFParkedUS}$
- RF communications including possible RF jamming while parked $(F_{RFComms}) \subseteq (F_{O1})_{JSFParkedUS}$

RF Communications has two aspects. The technical aspect is the actual communication systems themselves (hardware, firmware, and software). The operational environment RF Communications flaws are the actual voice communications to entities, such as aircraft controllers. The Maintenance Connectivity represents the Ethernet connection used by maintenance personnel to monitor multiple systems including propulsion and weapons systems.

JSF in flight within international air space (F_{S1})$_{JSFInFlightI}$

Technical

- Baseline ($F_{JSFBaseline}$) $\subseteq$ (F_{T1})$_{JSFInFlightI}$
- RF communications including enabling software defined radio possibly allowing access to aircraft network ($F_{RFComms}$) $\subseteq$ (F_{T1})$_{JSFInFlightI}$

Operational

- Blue Force Tracker including Blue Force Tracker signal emitting from aircraft (F_{BFT}) $\subseteq$ (F_{O1})$_{JSFInFlightI}$
- RF communications including possible geo-location based upon use of RF ($F_{RFComms}$) $\subseteq$ (F_{O1})$_{JSFInFlightI}$
- Satellite communications including possible geo-location based upon satellite communications ($F_{SatComms}$) $\subseteq$ (F_{O1})$_{JSFInFlightI}$

The Blue Force Tracker allows the aircraft to determine if another aircraft is friend or foe. In this case the RF Communications are for direct communications to other aircraft, ships, etc. The Satellite Communication flaws are those specific communications the aircraft has to satellites, not including the RF aspects.

JSF In flight over Crimean region of Ukraine in time of conflict (F_{S1})$_{JSFInFlightConflictArea}$

Technical

- Baseline ($F_{JSFBaseline}$) $\subseteq$ (F_{T1})$_{JSFInFlightConflictArea}$

- RF communications including enabling software defined radio possibly allowing access to aircraft network ($F_{RFComms}$) $\subseteq$ $(F_{T1})_{JSFInFlightConflictArea}$

Operational

- Blue Force Tracker including Blue Force Tracker signal emitting from aircraft (F_{BFT}) $\subseteq$ $(F_{O1})_{JSFInFlightConflictArea}$
- Electronic Warfare (EW) Signature including RF emanations from aircraft (F_{EWS}) $\subseteq$ $(F_{O1})_{JSFInFlightConflictArea}$
- RF communications including possible geo-location based upon use of RF ($F_{RFComms}$) $\subseteq$ $(F_{O1})_{JSFInFlightConflictArea}$
- Satellite communications including possible geo-location based upon satellite communications ($F_{SatComms}$) $\subseteq$ $(F_{O1})_{JSFInFlightConflictArea}$
- Weapons Systems including connecting tactical system to aircraft network ($F_{Weapons}$) $\subseteq$ $(F_{O1})_{JSFInFlightConflictArea}$

The EW signature is comprised of those aspects of the aircraft that provide the adversary with an electronic identification of the aircraft. This is distinctly different from the cyber signature of the aircraft. The weapons systems flaws are categorized as operational flaws in a conflict environment because of the combination of flaws in other systems and their connections to the weapons system providing an adversary an attack mechanism to exploit such flaws to inappropriately launch weapons, detonate in place, or use as a cyber relay launch point to access ships, etc through a trusted channel.

The three preceding examples of the JSF provide a glimpse into three possible states, both technically and operationally, of the aircraft. If $(F_{T1})_n$ and $(F_{O1})_n$ represents the set of technical and operational environment flaws for system s1 in state *n,* then: the set of all flaws for system s1 in state n is:

$$(F_{S1})_n = (F_{T1})_n \cup (F_{O1})_n$$

An assessment of system s1 will then consist of an assessment of the system with

respect to flaws in all possible states of the system. If we let $Ev_F(F_{s1})$ represent the assessment of the flaws in system s1 and i represent one of the possible k states, then:

$$Ev_F(F_{s1}) = \cup_{i=1...k} Ev_F((F_{s1})_i)$$

When flaws are identified, the ISSE may also quickly identify other associated aspects, such as countermeasures. As such, it is suggested that the ISSE document these items together,

3.2 Countermeasure models

Just as flaws exist in a fluid state, so do the countermeasures. This is made more so by the fact that countermeasures are not just on the same system, but exist in more than one layer (DiD) and in more than one component (DiB). Countermeasures are implemented to reduce the vulnerability of an information system.

The following is detailed further in Chapter 4. At the start of an assessment, one hopes that the countermeasures of the system should be well identified. However, rarely are all of the countermeasures identified at the start of the assessment, primarily due to the complexity of most systems masking some countermeasures and the system owner not always understanding what countermeasures are actually associated with which flaws or vulnerabilities. Therefore, the assessor will not associate the countermeasure model to the flaw model during the initial assessment. Only after flaws and their associated countermeasures are identified, will the two models be correlated.

During the initial assessment, countermeasures will be generically mapped to a flaw area (such as $F_{802.11}$). This allows the assessor to create a high-level representation of the system. This representation will be updated throughout the assessment with the final countermeasure model mapped directly to a flaw model.

Countermeasures (M) may be partial (M_P), complete[24] (M_C), perceived (the countermeasure is in place for a flaw but does not actually mitigate that flaw) (M_{NT}), or not known (M_{NK}). A countermeasure is not known if the countermeasure is

[24] A complete countermeasure is one that fully mitigates the associated flaw.

possible and applied for this flaw, but not realized as an applied countermeasure. As the assessor's knowledge of the system's flaws and countermeasures increases, the association between the flaws and countermeasures and their associated completeness will solidify.

The set of all possible countermeasures are defined as:

$$M = M_{NK} \cup M_{NT} \cup M_P \cup M_C$$

Countermeasures can be grouped into two categories, those that are complete and all others (M_{NC}). It is assumed that unknown countermeasures are not a subset of complete countermeasures because it is not known if they are complete or not.

$$M_{NC} = M_{NK} \cup M_{NT} \cup M_P$$

The technical mitigations are complete ($M_{T,C}$) or not known technical countermeasures ($M_{T,NC}$) where:

$$M_{T,C} = \{m \mid m \in M_T \,/\backslash\, m \in M_C\}$$
$$M_{T,NC} = \{m \mid m \in M_T \,/\backslash\, m \in M_{NC}\}$$

Therefore, technical countermeasures are defined as:

$$M_T = M_{T,C} \cup M_{T,NC}$$

Similarly and generically, operational environment countermeasures are those that are complete ($M_{O,C}$) and all other operational environment countermeasures ($M_{O,NC}$):

$$M_{O,C} = \{m \mid m \in M_O \wedge m \in M_C\}$$
$$M_{O,NC} = \{m \mid m \in M_O \wedge m \in M_{NC}\}$$

Therefore, operational countermeasures are defined as:

$M_O = M_{O,C} \cup M_{O,NC}$

System Specific Countermeasure Categorization: The preceding gives us the categorization of all possible countermeasures. However, for a specific system, an assessor examines the subset of countermeasures that are appropriate to that system. For example, assuming system s1, the following notation will be used:

$M_{T1} = \{m \mid m \in M_T \wedge m \text{ is a technical countermeasure for system s1}\}$

$M_{O1} = \{m \mid m \in M_O \wedge m \text{ is an operational environment countermeasure for system s1}\}$

The notations $M_{T1,C}$, $M_{T1,NC}$, $M_{O1,C}$, $M_{O1,NC}$ will be used similarly.

Continuing the previous example of a laptop computer (s1) in a hotel lobby, the list below contains some of the groups of countermeasures possible in such a system. The notation below identifies technical and operational aspects of the system as countermeasures. As an example, M_{OS} represents countermeasures in an OS; one such countermeasure would be Role Based Access Control. Possible countermeasures are identified as either technical or operational:

<u>Technical</u>

- Operating System including Discretionary Access Control (DAC) (M_{OS}) $\subseteq M_{T1}$
- Anti-Virus (AV) including enabling AV (M_{AV}) $\subseteq M_{T1}$
- Bluetooth including disabling Bluetooth ($M_{Bluetooth}$) $\subseteq M_{T1}$
- Browser including not allowing 3rd party cookies ($M_{Browser}$) $\subseteq M_{T1}$
- System firewall including not allowing inbound connections ($M_{Firewall}$) $\subseteq M_{T1}$
- Mobile Code including disabling Javascript and ActiveX ($M_{MobileCode}$) $\subseteq M_{T1}$
- Wireless including wireless modem not set to automatically connect ($M_{802.11}$) $\subseteq M_{T1}$

Operational

- Wireless connectivity including using a wireless hotspot device so as to have no need to rely on hotel managed network ($M_{802.11}$) $\subseteq M_{O1}$

While the above groupings are somewhat vague, the idea is to identify groups of countermeasures that would apply to groups of flaws and then to correlate one or more countermeasures to one or more flaws as the evidence is acquired.

Operational Environment Countermeasure Categorization: Unlike technical countermeasures, operational environment countermeasures may be specific to a category, c. The following notation will be used for these (and similar notation for countermeasures that are not complete).

$$M(c)_{O,C} = \{m \mid m \in M_{O,C} \wedge m \text{ is in category “c”}\}$$

Countermeasures that do not belong to a specific category will be “Unspecified”. In addition, there is the special category “ALL” which contains all flaws such that $F(ALL)_{O,E} = F_{O,E}$. When the system being assessed is defined, then categories for that system will be defined, and therefore for system s1:

$$M(c)_{O1,C} = \{m \mid m \in M(c)_{O,C} \wedge c \text{ is a category of system s1}\}$$

During the assessment of a system, the technical countermeasures must be assessed to determine their effect on the technical flaws, which are always part of the system, including when implemented in an environment. In addition, the specific operational environment countermeasures will need to be assessed and categorized appropriately when the system is instantiated to determine their effect on technical and operational environment flaws. The set of countermeasures examined for system s1 will simply be:

$$M_{s1} = M_{O1} \cup M_{T1}$$

Just as flaws are not static or unchanging, neither are countermeasures. Similarly, sets of countermeasures must be defined dynamically. Just as with flaws, to model the dynamic nature of the system we are going to define the sets of countermeasures for a specific system when it is in a particular state.

To continue the example of laptop computer (s1), this time in the state of residing in a secure space $(M_{s1})_{SecureSpace}$. Some possible countermeasures are identified as either technical or operational:

Technical

- Operating System including Discretionary Access Control $(M_{OS}) \subseteq (M_{T1})_{SecureSpace}$
- Email including disabling preview $(M_{Email}) \subseteq (M_{T1})_{SecureSpace}$ [25]
- Anti-Virus including enabling AV $(M_{AV}) \subseteq (M_{T1})_{SecureSpace}$
- Bluetooth including disabling Bluetooth $(M_{Bluetooth}) \subseteq (M_{T1})_{SecureSpace}$
- Browser including not allowing 3rd party cookies $(M_{Browser}) \subseteq (M_{T1})_{SecureSpace}$
- System encrypted transport including Secure Socket Layer $(M_{SSL}) \subseteq (M_{T1})_{SecureSpace}$
- System firewall including enabling FW $(M_{Firewall}) \subseteq (M_{T1})_{SecureSpace}$
- Mobile code including disabling Javascript $(M_{MobileCode}) \subseteq (M_{T1})_{SecureSpace}$
- Supply Chain including anonymous buys $(M_{Supply}) \subseteq (M_{T1})_{SecureSpace}$
- Wireless including disabling wireless modem $(M_{802.11}) \subseteq (M_{T1})_{SecureSpace}$

Operational

- Configuration Management including patch management plan $(M_{CM}) \subseteq (M_{O1})_{SecureSpace}$
- External encrypted transport including Type 1 device $(M_{Type1}) \subseteq (M_{O1})_{SecureSpace}$
- Gateway firewall including outbound Access Control List (ACL) black listing IPs $(M_{GatewayFW}) \subseteq (M_{O1})_{SecureSpace}$
- Physical security including limiting physical access to system $(M_{PS}) \subseteq (M_{O1})_{SecureSpace}$

[25] It should be noted that email flaws and countermeasures can be generically stated, but an evaluator would require knowledge of the specific email server and client implemented on the workstation. The same goes for the browser, OS, AV, and firewall

The countermeasures for a specific state will be those countermeasures that are possible given the value of the state variables. The notation $(M_{T1})_n$ and $(M_{O1})_n$ will be used to represent the set of countermeasures for system s1 in state *n*, then a single state of countermeasures for system s1, including the null set, is:

$$(M_{S1})_n = (M_{T1})_n \cup (M_{O1})_n$$

An assessment of system s1 will then consist of an assessment of the system with respect to flaws and countermeasures in all possible states of the system. If we let Ev(s1) represent the assessment the flaws and countermeasures of the system s1 and i represent one of the possible k states, then:

$$Ev(s1) = U_{i=1...k}\, Ev_F((F_{s1})_i) \cup U_{i=1...k}\, Ev_M((M_{s1})_i)$$

It is not possible to map all of the permutations of flaws and countermeasures [Ste13]. Adversaries are constantly attempting to avoid or overcome countermeasures[26], adding to the fluidity of countermeasures and their associated flaws. The next section provides a mathematical model to be used as a basis for an assessor to document a mapping of a system's flaws and countermeasures.

3.3 Vulnerability models

Vulnerabilities (V) are those flaws that are not completely mitigated by countermeasures. The very high level, deceptive, conceptual model of vulnerabilities has been defined as:

$$V = F \times M$$

[26] NIST SP800-30 refers to this as threat shifting.

More accurately, vulnerabilities are defined as the following to indicate that vulnerabilities are based on a subset of flaws that are not completely mitigated by countermeasures:

$$V = \{f \mid f \in F \text{ and } M_C(f) = \varnothing\}$$

Of note, though while possible, it is extraordinarily rare (in fact this author has never witnessed), for a flaw to be completely mitigated in an operationally instantiated system. As most assessors do not have a mathematical background, a more visually simplistic representation of the above model would be that vulnerabilities are flaws with partial (V_P), perceived (V_{NT}), not known (V_{NK}), or no (V_{NoM}) countermeasures:

$$V = V_{NoM} \cup V_{NK} \cup V_{NT} \cup V_P$$

Since vulnerabilities are based upon flaws, in this work we will define vulnerabilities the same as flaws, in terms of three orthogonal taxonomies: origin, vulnerability to exploitation, and levels of existence of countermeasures.

Vulnerability Origin: Since vulnerabilities are a subset of flaws, vulnerabilities exist in both the technical system and in all environments of operation and are defined as:

V represents the set of all vulnerabilities

$V_T = \{f \mid f \in F_T \text{ and } M_C(f) = \varnothing\}$ is the set of technical vulnerabilities

$V_O = \{f \mid f \in F_O \text{ and } M_C(f) = \varnothing\}$ is the set of operational environment vulnerabilities

Since vulnerabilities are a subset of flaws, the taxonomy discussions from the flaw discussions will be applied to vulnerabilities, and so the notations V_U, $V_{!E}$, V_E, $V_{T,E}$, $V_{T,!E}$, $V_{T1,E}$, $V_{T1,!E}$, $V_{O,E}$, $V_{O,E}$, $V_{O1,E}$, $V_{O1,!E}$ will be used similarly.

As with countermeasures, vulnerabilities can be grouped into two categories, those that have complete countermeasures, hence not a vulnerability, and all others (V_{NC}). It is also assumed that vulnerabilities with countermeasures that are not known are not a subset of the vulnerabilities with complete countermeasures because it is not known if they are complete or not. Therefore, any vulnerability that is not completely mitigated is represented by:

$$V_{NC} = V_{NoM} \cup V_{NK} \cup V_{NT} \cup V_{P}$$

Since both flaws and countermeasures have fluidity, vulnerabilities have fluidity from the constant changing states of both flaws and countermeasures. Just as with flaws and countermeasures, to model the dynamic nature of the system we are going to define the sets of vulnerabilities for a specific system when it is in a particular state.

To continue the example of an unclassified laptop computer (s1), this time in the state of residing in a secure space $(V_{s1})_{SecureSpace}$. The example vulnerabilities identified are those flaws that have well-known countermeasures but those countermeasures are not implemented. As with flaws, vulnerabilities are identified as either technical or operational:

Technical

- Operating System including implementing end of life OS such as Windows XP $(V_{OS}) \subseteq (V_{T1})_{SecureSpace}$
- Email including not enabling spam filters $(V_{Email}) \subseteq (V_{T1})_{SecureSpace}$
- Anti-Virus [Luc14] including AV provider not digitally signing updates $(V_{AV}) \subseteq (V_{T1})_{SecureSpace}$
- Browser including using end of life browser $(V_{Browser}) \subseteq (V_{T1})_{SecureSpace}$
- System encrypted eransport including Heartbleed not mitigated $(V_{SSL}) \subseteq (V_{T1})_{SecureSpace}$
- System firewall including allowing remote connections $(V_{Firewall}) \subseteq (V_{T1})_{SecureSpace}$
- Mobile Code including allowing third party cookies $(V_{MobileCode}) \subseteq (V_{T1})_{SecureSpace}$

- Logging/Auditing including not implementing DAC on audit files ($V_{Logging}$) ⊆ $(V_{T1})_{SecureSpace}$
- Supply Chain including updating software without verifying checksums (V_{Supply}) ⊆ $(V_{T1})_{SecureSpace}$

Operational

- Configuration Management including not patching software and firmware (V_{CM}) ⊆ $(V_{O1})_{SecureSpace}$
- External encrypted transport including not using Type 1 devices as required by policy (V_{Type1}) ⊆ $(V_{O1})_{SecureSpace}$
- Gateway firewall including allowing all outbound traffic ($V_{GatewayFW}$) ⊆ $(V_{O1})_{SecureSpace}$
- Physical security including intentionally leaving doors open to restricted spaces (V_{PS}) ⊆ $(V_{O1})_{SecureSpace}$ (physical security vulnerabilities are assessed as part of cyber assessments)

The Bluetooth and Wireless vulnerabilities are not listed because in a secure space, those capabilities are physically disabled. The vulnerabilities for a specific state will be those flaws and countermeasures that are possible given the value of the state variables. The notation $(V_{T1})_n$ and $(V_{O1})_n$ will be used to represent the set of vulnerabilities for system s1 in state *n*. Similar notations will be used for the other sets of vulnerabilities. A single state of vulnerabilities for system s1, including the null set, is:

$$(V_{S1})_n = (V_{T1})_n \cup (V_{O1})_n$$

An assessment of system s1 will then consist of an assessment of the system with respect to flaws and countermeasures in all possible states of the system. Just as in the prior section, we let Ev(s1) represent the assessment the flaws and countermeasures of the system s1 and i represent the possible k number of states, then:

$$Ev(s1) = (\cup_{i=1...k} Ev_F((F_{s1})_i)) \cup (\cup_{i=1...k} Ev_M((M_{s1})_i))$$

The environments of operations have unique sets of flaws and countermeasures, some technical and some specific to the environment. A vast majority of the DoD EOs are connected in some form or another, at some level of constant connectivity, to the network infrastructure that supports the DoD, which is known as the Global Infrastructure Grid (GIG). The land based GIG is the Internet. A DoD research project, ARPANET, conducted by the Advanced Research Projects Agency, was the progenitor to the Internet [Wal13] [Opp01]. Unfortunately, the designers of that original packet switched network, circa 1968, did not account for the threat of someone intentionally behaving maliciously [Opp01].

3.4 Threat models

There are many threat models and no one model will be effective or efficient for everyone. Some, such as NIST Special Publication 800-30 R1 consider threat models at multiple levels, such as organization/agency, mission, and system [NIS12]. The threat models proposed in this dissertation are for a detailed system level threat assessment.

A threat assessment, per CNSSI 4009, is a process of formally evaluating the degree of threat to an information system or enterprise and describing the nature of the threat. Threats are from a variety of sources, such as adversaries, disgruntled insiders, and natural disasters and there are many definitions of cyber threats[27]. A threat, as defined by the NIST Special Publication 800-30 R1, is the potential for a threat-source to exercise (accidentally trigger or intentionally exploit) a specific vulnerability. However, the NIST use of vulnerability refers to what this dissertation equates to a flaw and there is no consideration of countermeasures in the NIST definition, unlike this dissertation that considers a vulnerability a union of the set of flaws and those corresponding, unique countermeasures. In this dissertation, a threat

[27] CNSSI 4009 defines a threat as any circumstance or event with the potential to adversely impact organizational operations (including mission, functions, image, or reputation), organizational assets, individuals, other organizations, or the Nation through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service.

(TR) is a tuple consisting of some combination of one or more threat source (TS), that threat source's one or more capabilities (TC), with one or more threat source's motivation(s) (TSM) for exploiting a vulnerability and is defined as:

$$TR = TS^{+} \times TC^{+} \times TSM^{+}$$

Unlike flaws and vulnerabilities, threat sources are defined based upon the intent to exploit:

TS represents the set of all Threat Sources

$TS_{Intentional} \subseteq TS$ is the set of threats intentionally exploiting a vulnerability[28]

$TS_{Accidental} \subseteq TS$ is the set of threats where a situation or method[29] may accidentally trigger a vulnerability [NIS12]

The accidental threat source includes such things as nature (TS_{Nature})[30] such as an earthquake or hurricane, as well as the accidental human ($TS_{AccidentalHuman}$), which is when a person performing a task triggers a vulnerability unintentionally. The accidental threat source is then:

$$TS_{Accidental} = TS_{AccidentalHuman} \cup TS_{Nature}$$

The major categories of intentional human[31] threat sources could be expressed as state sponsored ($TS_{StateSpon}$), insider ($TS_{Insider}$), terrorist ($TS_{Terrorist}$), hacker (TS_{Hacker}), and organized crime ($TS_{OrgCrime}$). This is not meant as an all-encompassing list, but enough to provide a basis for an assessor. A threat source, such as state sponsor, is a set because state sponsored not only indicates adversaries in the direct employ of a nation state, but also those that act on behalf of the nation state out of some motivation. The set of a threat source's intentional exploitation of a

[28] NIST SP800-30 defines this as the intent and method targeted at the intentional exploitation of a vulnerability.
[29] The procedure or process for accomplishing a task, such as an exploit.
[30] Other examples of such threats are floods, tsunami, tornados, sunspots, and fires.
[31] NIST SP800-30 refers to these as adversarial.

vulnerability contains the set of accidental threat sources, which is the null set for intention, and is represented by:

$$TS_{Intentional} = TS_{StateSpon} \cup TS_{Insider} \cup TS_{Terrorist} \cup TS_{Hacker} \cup TS_{OrgCrime} \cup TS_{Accidental}$$

Some would consider that state sponsored, terrorist, and organized crime categories to be in the single category of well-funded adversaries, which is the greatest capability. However, the word capability is not sufficiently concise. The NIST Special Publication 800-30 R1 considers level of expertise, number of resources, and the ability to generate opportunities to support continuous, coordinated, successful attacks in calculating a threat-source's capabilities [NIS12]. These considerations will be used to provide an initial definition of threat source capabilities:

TC represents the set of all Threat Source Capabilities

$TC_{LevelOfExpertise} \subseteq TC$ is the set of a threat source's levels of expertise

$TC_{Resources} \subseteq TC$ is the set of the number of resources of a threat source

$TC_{Success} \subseteq TC$ is the set of a threat source's levels of success

A threat source has one or more capabilities. A single capability will have one or more resources with each resource having varying levels of expertise and subsequently that capability will have varying degrees of success. A high level example of a threat source of a nation state having multiple capabilities is the US Government having the cyber capabilities of the DoD (the individual services) and IC (NSA, etc). Within this example, each capability has multiple resources (many individual people) each of which will have varying levels of expertise and subsequently the resources, and therefore the capability, will have varying levels of success in attacks.

$$TC = (TC_{LevelOfExpertise})^{+} \times (TC_{Resources})^{+} \times (TC_{Success})^{+}$$

A more concise breakdown of capabilities will be categorized by the following sets [NIS12]:

$TC_{LevelOfExpertise}$ = {VerySophisticated, Sophisticated, Moderate, Limited, VeryLimited}

$TC_{Resources}$ = {Unlimited, Significant, Moderate, Limited, VeryLimited}

$TC_{Success}$ = {MultipleContinuousCoordinated, MultipleCoordinated, Mulitple, Limited, VeryLimited}

Therefore, a state sponsored threat source may have differing levels of expertise, resources, and success based upon that nation's focus on cyber warfare. Most people assume the greatest level of expertise, resources, and success when considering state sponsored threat sources, but depending upon the situational instance that may not hold true. An example would be the comparison a US aircraft in foreign airspace. If the airspace is claimed by Mexico, it would imply one level of cyber threat source capabilities, but if it was claimed by Russia, it would imply a significantly different capability. NIST Special Publication 800-30 R1 also considers the adversaries' ability to analyze information obtained by differing levels of information reconnaissance or lack thereof, which it refers to as targeting. However, that level of depth will be included within the threat source's capabilities for this dissertation.

Threat Source Motivation: The above sets indicate the intentional exploitation or accidental triggering of a vulnerability, as well as the capability of a threat source, but not the motivation. It is often not modeled because there may be no motivation (nature), it may change for each category of threat, and within each threat source based upon situations. Some examples of motivations include financial ($TSM_{Financial}$), national security ($TSM_{NationalSecurity}$), power (TSM_{Power}), information gathering (TSM_{Intel}), and forcible change (TSM_{Change}).

$$TSM = TSM_{NationalSecurity} \times TSM_{Power} \times TSM_{Intel} \times TSM_{Financial} \times TSM_{Change}$$

The motivations listed above are just examples, there are many more possible motivations that are not listed.

Threat Source Motivation Covertness: NIST Special Publication 800-30 R1 considers the level of intrusion (undermine, severely impede, or destroy a core mission or business function, program, or enterprise) and the level of covertness ($TSM_{Covertness}$) (attack detection/disclosure of tradecraft).[32] The level of intrusion a threat source is capable of achieving will not be considered in this dissertation, as such delineation is not necessary for NSS. Determining the actual level of covertness is extremely difficult. In the case of this dissertation, it is not the determination of the covertness, instead it is the perceived level of motivation to be covert that will be modeled. We can categorize threat source motivation for covertness into two classes, those threat sources that are motivated to be covert and those not motivated to be covert.

$$TSM_{Covertness} = (TSM_{Covertness})_{MotivatedToBeCovert} \text{ x } (TSM_{Covertness})_{NotMotivatedToBeCovert}$$

Obviously, a state sponsored threat source may include all of the above motivations:

[32] Threat motivations from NIST SP800-30 are the following. 1. The adversary seeks to undermine, severely impede, or destroy a core mission or business function, program, or enterprise by exploiting a presence in the organization's information systems or infrastructure. The adversary is concerned about disclosure of tradecraft only to the extent that it would impede its ability to complete stated goals. 2. The adversary seeks to undermine/impede critical aspects of a core mission or business function, program, or enterprise, or places itself in a position to do so in the future, by maintaining a presence in the organization's information systems or infrastructure. The adversary is very concerned about minimizing attack detection/disclosure of tradecraft, particularly while preparing for future attacks. 3. The adversary seeks to obtain or modify specific critical or sensitive information or usurp/disrupt the organization's cyber resources by establishing a foothold in the organization's information systems or infrastructure. The adversary is concerned about minimizing attack detection/disclosure of tradecraft, particularly when carrying out attacks over long time periods. The adversary is willing to impede aspects of the organization's missions/business functions to achieve these ends. 4. The adversary actively seeks to obtain critical or sensitive information or to usurp/disrupt the organization's cyber resources, and does so without concern about attack detection/disclosure of tradecraft. 5, The adversary seeks to usurp, disrupt, or deface the organization's cyber resources, and does so without concern about attack detection/disclosure of tradecraft.

$TSM_{StateSpon} = TSM_{NationalSecurity} \cup TSM_{Power} \cup TSM_{Intel} \cup TSM_{Financial} \cup TSM_{Change} \cup TSM_{Covertness}$

Whereas a terrorist threat source motivations may include only:

$TSM_{Terrorist} = TSM_{Financial} \cup TSM_{Change} \cup (TSM_{Covertness})_{NotAtAllMotivatedToBeCovert}$

Threat Focus: Threats are specific to operational environments. Usually, threats are more generic to technical capabilities, but there are cases where the threats are specific to a technical capability (such as a CDS). Threats are defined as:

TR represents the set of all threats
$TR_T \subseteq TR$ is the set of threats against a specific technical system
$TR_O \subseteq TR$ is the set of threats against operational environments

Combining Categorizations: Threats can be further categorized in terms of more than one of the taxonomies:

The threats to technical capabilities can be subdivided into those with covert motivations ($TR_{T\ MotivatedToBeCovert}$) and all others ($TR_{T\ !MotivatedToBeCovert}$) where:

$TR_{T\ MotivatedToBeCovert} = \{tr \mid tr \in TR_T \ /\backslash\ tr \in TR_{MotivatedToBeCovert}\}$
$TR_{T,!MotivatedToBeCovert} = \{tr \mid tr \in TR_T \ /\backslash\ tr \in TR_{!MotivatedToBeCovert}\}$

Therefore, technical threats are defined as:

$TR_T = TR_{T,!MotivatedToBeCovert} \cup TR_{T\ MotivatedToBeCovert}$

Similarly and generically, operational environment threats are those with motivation to be covert ($TR_{O\ MotivatedToBeCovert}$) and all other operational environment threats ($TR_{O,\ !MotivatedToBeCovert}$):

$TR_{O,\ MotivatedToBeCovert} = \{tr \mid tr \in TR_O \wedge tr \in TR_{MotivatedToBeCovert}\}$

$(TR_{O,\ !MotivatedToBeCovert}) = \{tr \mid tr \in TR_O \wedge tr \in TR_{!MotivatedToBeCovert}\}$

$TR_O = (TR_{O,\ !MotivatedToBeCovert}) \cup TR_{O,\ MotivatedToBeCovert}$

System Specific Threat Categorization: The preceding gives us categorizations of possible threats. However, for a specific system, a subset of threats appropriate to that system will be examined. For example, assume system s1, then the following notation will be used:

$TR_{T1} = \{tr \mid tr \in TR_T \wedge tr$ is a threat against the technical capability for system s1$\}$

$TR_{O1} = \{tr \mid tr \in TR_O \wedge tr$ is a threat against the operational environment for system s1$\}$

The notations $TR_{T1,\ MotivatedToBeCovert}$, $TR_{T1,!\ MotivatedToBeCovert}$, $TR_{O1,\ MotivatedToBeCovert}$, $TR_{O1,!MotivatedToBeCovert}$ will be used similarly.

Threat sources, capabilities, and motivations are fluid and change with the situational instance, and at times the technical system. Therefore, the threat states are a combination of all possible threat sources, threat source capabilities, and threat source motivations:

$TR_{States} = TS \times TC \times TSM$

The threat sources, capabilities, and motivations will also change based upon the operational environment category. Just as with flaws, countermeasures, and vulnerabilities, to model the dynamic nature of the system we are going to define the sets of threats for a specific system when it is in a particular state. Organized crime may have financial motivation to exploit a vulnerability on an unmanned air vehicle

(V_{UAV}) in the situational instance of the unmanned VAR in Afghanistan airspace ($V_{UAVinAfghan}$) and would not be concerned about covertness:

$$TR_{State\ n} = (TS_{OrgCrime} \times (TSM_{Financial} \cup (TSM_{Covertness})_{NotMotivatedToBeCovert}) \times (TC_{LevelOfExpertise})_{Sophisticated}) \cup V_{UAVinAfghan,\ State\ n}$$

However, the state sponsored entity of Russia ($TS_{StateSponR}$) may have financial, national security, information gathering, power, and forcible change motivations to exploit a vulnerability within a U.S. UAV in Ukraine ($V_{UAVinUkraine}$) very covertly:

$$TR_{State\ n} = TS_{StateSponR} \times (TSM_{NationalSecurity} \cup TSM_{Power} \cup TSM_{Intel} \cup TSM_{Financial} \cup TSM_{Change} \cup (TSM_{Covertness})_{MotivatedToBeCovert}) \times (TC_{LevelOfExpertise})_{VerySophisticated}) \cup V_{UAVinUkraine,\ State\ n}$$

The threats for a specific state will be those that are possible given the value of the state variables. The notation $(TR_{S1})_n$ will be used to represent the set of threats for system s1 in state *n*, then: A single state of threats for system s1, including the null set, is:

$$(TR_{S1})_n = ((TS_{T1})_n \times (TC_{T1})_n \times (TSM_{T1})_n) \cup ((TS_{O1})_n \times (TC_{O1})_n \times (TSM_{O1})_n)$$

An assessment of system s1 will then consist of an assessment of the system with respect to vulnerabilities, and threats in all possible states of the system. Just as in the prior sections, we let Ev(s1) represent the assessment of the vulnerabilities and threats of the system s1 and i represent the possible k number of states, then:

$$Ev(s1) = (\cup_{i=1\ldots k} Ev_V((V_{s1})_i)) \cup (\cup_{i=1\ldots k} Ev_{TR}((TR_{s1})_i))$$

Motivation implies some amount of probability[33] of attack occurrence, as does a situational instance, such as a U.S. aircraft in an adversary's airspace during a time of

[33] NIST Special Publication 800-30 refers to this as likelihood.

kinetic war. The threat source itself may imply a probability of occurrence, such as a tornado if the system is located in the U.S. state of Kansas, which is highly prone to tornadoes. For every threat, there will be associated probabilities. By including the probabilities, which are extremely subjective, it allows the ISSE to express any correlations between threats, vulnerabilities, attack vectors based upon their experience and knowledge; including how certain they are of the information.

3.5 Probability models

There are three probabilities regarding exploitations to consider during an assessment. There is the probability that a threat source will attack (PA). Then there is the probability of the success of the attack (PS). Finally, there is the probability of certainty (PC) of the knowledge. The overall probability (P) that an attack will occur with some levels of covertness and success would then be defined as a tuple:

$$P = PA \times PS \times PC$$

To provide the greatest repeatability and reciprocation of the assessment, all three probabilities must be included in calculations. For conciseness, each probability will be individually identified. To be consistent with the NIST Special Publication 800-30 R1, the two of the probabilities in this case will not be a value between 0 and 1, but based upon those in the NIST Special Publication 800-30 R1 for likelihoods (Almost Certain, Highly Likely, Somewhat Likely, Unlikely, and HighlyUnlikely).

PA = {AlmostCertain, HighlyLikely, SomewhatLikely, Unlikely, HighlyUnlikely}
PS = {AlmostCertain, HighlyLikely, SomewhatLikely, Unlikely, HighlyUnlikely}

However, those categories of likelihood do not correlate to the probability of certainty of knowledge. By substituting the word “certain” for the word “likelihood”, the vocabulary is correct and the values are maintained.

PC = {AlmostCertain, HighlyCertain, SomewhatCertain, Uncertain, HighlyUncertain}

These probabilities will be defined in terms of their three orthogonal taxonomies: vulnerability countermeasure completeness, origin, and threats knowledge.

Probability: As with everything else, the probabilities exist for both the technical system, as well as in all operational environments.

P represents the set of all probabilities

$P_T \subseteq P$ is the set of probabilities for technical capabilities of a system

$P_O \subseteq P$ is the set of probabilities for an operational environment

The probabilities can be grouped into two categories, those systems that have vulnerabilities with complete countermeasures (P_{VC}) and all others (P_{VNC}).

$P = P_{VNC} \cup P_{VC}$

Threat Classes: As well as categorizing probabilities by vulnerability countermeasure completeness, we can categorize probabilities into threat classes, specifically those with known threats and those without known threats. It is assumed that uncategorized threats are a subset of those without known threats.

$P_U \subseteq P_{!TR}$

$P = P_{!TR} \cup P_{TR}$

Combining Categorizations: Due to the orthogonal nature of the taxonomies, probabilities can be further categorized in terms of more than one of the taxonomies:

The technical probabilities can be subdivided into those systems that have vulnerabilities with known exploits ($P_{VT,E}$) and all other technical vulnerabilities ($P_{VT,!E}$) where:

$$P_{VT,E} = \{ p \mid p \in P_T \,/\backslash\, p \in P_E \}$$
$$P_{VT,!E} = \{ p \mid p \in P_T \,/\backslash\, p \in P_{!E} \}$$

Therefore, technical vulnerabilities are defined as:

$$P_{VT} = P_{VT,!E} \cup P_{VT,E}$$

Similarly and generically, operational environment probabilities are those systems that have known exploits ($P_{VO,E}$) and all other operational environment vulnerabilities ($P_{VO,!E}$):

$$P_{VO,E} = \{p \mid p \in P_O \,/\backslash\, p \in P_E\}$$
$$P_{VO,!E} = \{p \mid p \in P_O \,/\backslash\, p \in P_{!E}\}$$
$$P_{VO} = P_{VO,!E} \cup P_{VO,E}$$

System Specific Probabilities Categorization: The preceding gives us the categorization of all possible probabilities. However, for a specific system, a subset of probabilities that is appropriate to that system will be examined. For example, assume system s1, then the following notation will be used:

P_{T1} = {p | p ∈ P_T ∧ p is a probability for the technical capabilities of system s1}

P_{O1} = {p | p ∈ P_O ∧ p is a probability for the operational environment for system s1}

The notations P_{VT1E}, $P_{VT1!E}$, $P_{VO1,E}$, $P_{VO1,!E}$ will be used similarly.

Probability of attack, probability of attack success, and probability of knowledge certainty are fluid and change with the situational instance. Therefore, the probability

states are combination of all probabilities of attack, probabilities of attack success, and probabilities of knowledge certainty:

$$P_{States} = PA \times PS \times PC$$

The probabilities of attack, probabilities of attack success, and probabilities of knowledge certainty also change based upon the operational environment category. Just as with flaws, countermeasures, vulnerabilities, and threats, to model the dynamic nature of the system we are going to define the sets of probabilities for a specific system when it is in a particular state. To continue the example from the prior section, the state sponsored entity of Russia has motivations to exploit a vulnerability within a U.S. UAV, which is in Ukraine, very covertly, and there is an almost certain probability that Russia will attack, with a highly likely probability of success, based upon highly certain probability of knowledge:

$$P_{State\ n} = TR_{Russia,State\ n} \cup V_{UAVinUkraine,State\ n} \cup (PA_{AlmostCertain} \times PS_{HighlyLikely} \times PC_{HighlyCertain})$$

The probabilities for a specific state will be those that are possible given the value of the state variables. The notation $(P_{S1})_n$ will be used to represent the set of probabilities for system s1 in state *n*, then:

$$(P_{S1})_n = ((PA_{T1})_n \times (PS_{T1})_n \times (PC_{T1})_n) \cup ((PA_{O1})_n \times (PS_{O1})_n \times (PC_{O1})_n)$$

An assessment of system s1 will then consist of an assessment of the system with respect to vulnerabilities, threats, and probabilities in all possible states of the system. Just as in the prior sections, we let Ev(s1) represent the assessment the vulnerabilities, threats, and probabilities of the system s1 and i represent the possible k number of states, then:

$$Ev(s1) = \cup_{i=1...k} Ev_V((V_{s1})_i) \cup Ev_{TR}((TR_{s1})_i) \cup Ev_P((P_{s1})_i)$$

Probabilities must be considered for both technical and operational environment assessments. However, during the Technical Assessment the probabilities are far from certain. During the operational environment assessment, because the operational environment will have specified threats, there is a greater certainty of knowledge of the probably the threat attack and greater certainty of knowledge of the probably the attack will be successful. No matter the probabilities, without a physical mechanism/vector through or by which the exploit may be conducted against a vulnerability, the exploit won't succeed.

3.6 Attack Vector models

An attack vector (AV) is a physical (analog or digital) mechanism or vector through an exploit by a threat source may be conducted against a vulnerability. There are five categories of attack vectors; cyber (AV_{Cyber}), kinetic ($AV_{Kinetic}$), radio frequency (AV_{RF}), supply chain ($AV_{SupplyChain}$), and unknown ($AV_{!Known}$).

$$AV = AV_{Cyber} \cup AV_{Kinetic} \cup AV_{RF} \cup AV_{SupplyChain} \cup AV_{!Known}$$

The obvious attack vector for most systems is cyber (AV_{Cyber}), more specifically network connectivity, which in the case of the DoD is the GIG. But the GIG is nothing more than some combination of Internet trunk lines [Sne15] ($AV_{TrunkLines}$) and commercial leased lines ($AV_{LeasedLines}$):

$$AV_{Network} = AV_{TrunkLines} \cup AV_{LeasedLines}$$

The DOD now depends upon commercial industry to supply products, including networks, more than ever before in US history. As such, supply chain ($AV_{SupplyChain}$) attack vectors are increasing and increasingly successful. The supply chain attack vector is an attack where the physical connection between the threat source and vulnerability may exist only once. These attacks include more than just inserting malware into software ($AV_{Software}$) or selling thumb drives ($AV_{Hardware}$) containing malware. There is the intentional embedding of malicious code within firmware

during manufacturing ($AV_{Manufacturing}$), such as discussed in section 2.2.2. The DoD relies on critical infrastructure ($AV_{CriticalInfra}$) to provide not just power, water, and sewage removal, but also for satellite and Internet communications. So, supply chain attack vectors include the following:

$$AV_{SupplyChain} = AV_{Hardware} \cup AV_{Software} \cup AV_{Manufacturing} \cup AV_{CriticalInfra}$$

Something that commercial industry within the US does not normally contend with are kinetic attack vectors ($AV_{Kinetic}$). These include such things as missile strikes ($AV_{Missile}$), homicide bombers (AV_{HB}), and improvised explosive devices (AV_{IED}):

$$AV_{Kinetic} = AV_{Missile} \cup AV_{HB} \cup AV_{IED}$$

Commercial industry is more adept when contending with some radio frequency (AV_{RF}) attack vectors. Obviously, wireless communications such as 802.1x ($AV_{802.1x}$) and cellular broadband (AV_{Cell}) are well known to commercial industry. However, attack vectors such as the international frequency for all unmanned aerial vehicles, KU band, (AV_{KU}) and satellite communications ($AV_{CommercialSats}$ & $AV_{GovtSats}$) are more familiar to military applications:

$$AV_{RF} = AV_{802.1x} \cup AV_{Cell} \cup AV_{KU} \cup AV_{CommercialSats} \cup AV_{GovtSats}$$

Attack vectors will be defined in terms of their four orthogonal taxonomies: attack vector origin, attack vector knowledge, multi-vector, and persistence.

Attack Vector Origin: As with everything else, attack vectors exist for both the technical system and in all operational environments and are defined as:

AV represents the set of all attack vectors

$AV_T \subseteq AV$ is the set of attack vectors for a technical system

$AV_O \subseteq AV$ is the set of attack vectors for an operational environment

The attack vectors can be further grouped into two categories, those attack vectors that are known (AV_{Known}) and all others ($AV_{!Known}$).

$$AV = AV_{Known} \cup AV_{!Known}$$

Multiplicity Classes: Threat sources must have at least one, but may have multiple, attack vectors to facilitate their exploits. As well as categorizing attack vectors by whether an attack vector is known, we can categorize attack vectors into multiplicity classes, specifically those threat sources with multiple attack vectors (AV_{Multi}), those with at least one attack vector ($AV_{AtLeastOne}$), and all others (AV_{Others}).

$$AV = AV_{Multi} \cup AV_{AtLeastOne} \cup AV_{Others}$$

Persistence Classes: It is important to note that the physical connection between the threat source and the vulnerability must exist only once, and not necessarily be persistent, for the compromise to occur. As well as categorizing attack vectors by multiplicity, we can categorize attack vectors into persistence classes, specifically those threat sources with persistent connection ($AV_{Persistent}$) and all others ($AV_{!Persistent}$).

$$AV = AV_{Persistent} \cup AV_{!Persistent}$$

Combining Categorizations: Due to the orthogonal nature of the taxonomies, attack vectors can be further categorized in terms of more than one of the taxonomies:

Operational environment attack vectors can be subdivided into those systems that have at least one persistent known attack vector ($AV_{EO,AtLeastOne,Persistent,Known}$) and those systems that do not have a known persistent attack vector ($AV_{EO,!Known!Persistent}$) where:

$AV_{O,AtLeastOne,Persistent,Known} = \{av \mid av \in AV_O \,/\backslash\, av \in AV_{Known}\}$

$AV_{O,!Known!Persistent} = \{av \mid av \in AV_O \,/\backslash\, av \in AV_{!Known}\}$

Therefore, operational environment attack vectors are defined as:

$$AV_O = AV_{O,AtLeastOne,Persistent,Known} \cup AV_{O,!Known!Persistent}$$

Though not as intuitive as in an operational environment, attack vectors do exist for technical systems and can be subdivided into those systems that have a known attack vector ($AV_{T,Known,}$) and those systems that do not have a known attack vector ($AV_{T,!Known}$) where:

$AV_{T,Known} = \{av \mid av \in AV_T \,/\backslash\, av \in AV_{Known}\}$

$AV_{T,!Known} = \{av \mid av \in AV_T \,/\backslash\, av \in AV_{!Known}\}$

Therefore, attack vectors for technical systems are defined as:

$$AV_T = AV_{T,Known} \cup AV_{T,!Known}$$

System Specific Probabilities Categorization: The preceding gives us the categorization of all possible attack vectors. For a specific system, a subset of attack vectors appropriate to that system will be examined. For example, assume system s1, then the following notation will be used:

AV_{T1} = {av | av $\in AV_T \wedge$ av is an attack vector for the technical capabilities of system s1}

AV_{O1} = {av | av $\in AV_O \wedge$ av is an attack vector for the operational environment for system s1}

The notations $AV_{O,AtLeastOne,Persistent,Known}$, $AV_{O,!Known!Persistent}$, $AV_{T,Known}$, $AV_{T,!Known}$ will be used similarly.

Just as with everything else, attack vectors are not static or unchanging, but are fluid and change with the situational instance, and at times, the technical system. Therefore, the attack vector states are a combination of all possible attack vectors:

$$AV_{States} = AV_{Cyber} \cup AV_{Kinetic} \cup AV_{RF} \cup AV_{SupplyChain} \cup AV_{!Known}$$

The knowledge of an attack vector, its origin, its multiplicity, and its persistence change based upon the operational environment category. Just as with flaws, countermeasures, vulnerabilities, threats, and probabilities, to model the dynamic nature of the system we are going to define the sets of attack vectors for a specific system when it is in a particular state.

To continue the example from the prior section, the state sponsored entity of Russia has motivations to exploit a vulnerability within a U.S. UAV in Ukraine very covertly using RF communications or supply chain attack vectors, and there is an almost certain probability that Russia will attack with a highly likely probability of success based upon highly certain probability of knowledge:

$$AV_{State\ n} = TR_{Russia,State\ n} \cup V_{UAVinUkraine,State\ n} \cup (PA_{AlmostCertain} \times PS_{HighlyLikely} \times PC_{HighlyCertain}) \cup AV_{RF} \cup AV_{SupplyChain}$$

In this situational instance, there is a possibility of a kinetic attack vector being engaged. However, with this attack vector there is attribution that does not exist with the other attack vectors and subsequently the probability that this attack vector would be used is highly unlikely.

$$AV_{State\ m} = TR_{Russia,State\ m} \cup V_{UAVinUkraine,State\ m} \cup (PA_{HighlyUnlikely} \times PS_{AlmostCertain} \times PC_{AlmostCertain}) \cup AV_{Kinetic}$$

The preceding examples provide the two possible states for the attack vectors for that situational instance of a specific system in an operational environment. If $(AV_{T1})_n$ and $(AV_{O1})_n$ represent the set of attack vectors for system s1, in state *n,* then the set of all attack vectors for system s1 in state n is:

$$(AV_{S1})_n = (AV_{O1})_n \cup (AV_{T1})_n$$

An assessment of system s1 will then consist of an assessment of the system with respect to vulnerabilities, threats, probabilities, and attack vectors in all possible states of the system. Just as in the prior sections, we let Ev(s1) represent the assessment of the vulnerabilities, threats, probabilities, and attack vectors of the system s1 and i represent the possible k number of states, then:

$$Ev(s1) = \cup_{i=1...k} Ev_V((V_{s1})_i) \cup Ev_{TR}((TR_{s1})_i) \cup Ev_P((P_{s1})_i) \cup Ev_{AV}((AV_{s1})_i)$$

As shown above, not all threat sources would employ every attack vector. In the U.S., it is highly unlikely that organized crime would employ a kinetic attack vector. Just as not every attack vector would be available to every threat source. An attack vector employed to exploit a satellite would be limited to those threat sources with access to satellites as an example.

Attack Surface: While the term attack surface is more commonly associated with software [Wik13], it is being used more often in reference to environments of operation. The attack surface (AS) is all vulnerabilities, both in technical and operational environments that are accessible by attack vectors [Ste13].

$$AS = AV_T \cup V_T \cup V_O \cup AV_O$$

An assessment of system s1 will now consist of an assessment of the system with respect to threats, probabilities, and attack surfaces in all possible states of the system. Just as in the prior sections, we let Ev(s1) represent the assessment of the threats, probabilities, and attack surfaces of the system s1 and i represent the possible k number of states, then:

$$Ev(s1) = \cup_{i=1...k} Ev_{AS}((AS_{s1})_i) \cup Ev_{TR}((TR_{s1})_i) \cup Ev_P((P_{s1})_i)$$

None of the existing assessment methodologies consider attack surface, though Special Publication 800-53A R1 does define attack surface in regards to penetration testing of an operational environment. The term is becoming more interchangeable with vulnerabilities, it does convey more than vulnerabilities. As such, both terms are appropriate and will be used within this document.

3.7 Impact models

An impact (I) is the variable result of a threat exercising an attack vector on an attack surface. Normally, impacts are defined in terms of the magnitude of harm, such as in the NIST Special Publication 800-30 R1, using words such as catastrophic, limited, etc. to describe the magnitude. NIST Special Publication 800-30 R1 categorizes harm as damage to operations[34] (I_{OPS}), assets[35] (I_{Assets}), organizations[36] (I_{Org}), and the nation[37] (I_{Nation}). Interestingly enough, loss of human life[38] (I_{Life}) is not included in any of these categories. Nor is there a category that includes allies[39] (I_{Allies}) or global[40] (I_{Global}) impacts in the NIST Special Publication 800-30 R1 categories. Therefore, impacts will be expanded to include those not considered by NIST Special Publication 800-30 R1.

$$I = I_{OPS} \cup I_{Assets} \cup I_{Org} \cup I_{Nation} \cup I_{Life} \cup I_{Allies} \cup I_{Global}$$

Impact Type: The NIST categories of impact contain a mix of both operational and technical impacts. This, again, makes it difficult to separate the technical aspects of a current assessment from an operational environment aspect. To provide clarity, each

[34] NIST SP800-30 operational impacts include the inability to perform current and future missions/business functions and damage to image or reputation.
[35] NIST SP800-30 asset impacts include damage to or loss of physical facilities, systems, networks, IT equipment, component supplies and intellectual property.
[36] NIST SP800-30 organizational impacts include harms due to noncompliance, direct financial cost, and damage to reputation.
[37] NIST SP800-30 national impacts include damage to critical infrastructure sector, loss of government continuity of operations, damage to reputation, damage to ability to achieve national objectives, and harm to national security.
[38] Loss of human life is a very real impact not normally indicated in cyber assessments, but when assessing vehicles it would be an operational impact.
[39] Allied impacts include loss of coalition operations, damage to reputation (such as NATO), and damage to ability to achieve coalition objectives.
[40] Global impacts would include the complete failure of the Internet, global-wide virus infection, and global-wide critical infrastructure failure.

of the above categories will be further identified to include whether the impact is technical (I_T) or operational (I_O), thus separating technical and operational impacts.

I represents the set of all impacts

$I_T \subseteq I$ is the set of impacts for a technical system

$I_O \subseteq I$ is the set of impacts for an operational environment

To provide clarity, impacts will be further subcategorized within the technical (I_T) and operational environment (I_O).

$$I_T = (I_T)_{Assets} \cup (I_T)_{Org} \cup (I_T)_{Nation} \cup (I_T)_{Allies} \cup (I_T)_{Global}$$

$$I_O = (I_O)_{OPS} \cup (I_O)_{Assets} \cup (I_O)_{Org} \cup (I_O)_{Nation} \cup (I_O)_{Life} \cup (I_O)_{Allies} \cup (I_O)_{Global}$$

Impacts will be defined in terms of their two orthogonal taxonomies of technical impacts (fail-safe and fail-secure), as well as their two orthogonal taxonomies of operational impacts (loss of life and mission completion).

Fail Safe: This is a term, which has long been associated with aircraft, and only occasionally with computer systems. It references the capabilities of a system to not adversely affect human life or other devices in the event of the system's failure. Technical impacts can be categorized into those systems that fail safe ($(I_T)_{FailSafe}$) and all others ($(I_T)_{!FailSafe}$).

$$I_T = (I_T)_{FailSafe} \cup (I_T)_{!FailSafe}$$

Fail Secure: References the capabilities of a system to not allow unauthorized access to data in the event of the system's failure [DHS16]. Technical impacts can be further categorized into those systems that fail secure ($(I_T)_{FailSecure}$) and all others ($(I_T)_{!FailSecure}$).

$$I_T = (I_T)_{FailSecure} \cup (I_T)_{!FailSecure}$$

Operational Loss of Life: While fail-safe is a technical taxonomy that involves adverse effect on human life, which may or may not result in loss of life, directly from the system, the operational loss of life impact categorizations are those impacts which result in loss of life in the operational environment ($(I_O)_{LossOfLife}$) and all others ($(I_O)_{!LossOfLife}$).

$$I_O = (I_O)_{LossOfLife} \cup ((I_O)_{!LossOfLife}$$

Mission Completion: Regardless of the importance of the mission of a system within an operational environment, impacts are categorized by the ability to complete the intended mission ($(I_O)_{Completion}$) and all others ($(I_O)_{!Completion}$).

$$I_O = (I_O)_{Completion} \cup ((I_O)_{!Completion}$$

System Specific Impact Categorization: The preceding gives us the categorization of all possible impacts. For a specific system, a subset of impacts appropriate to that system will be examined. For example, assume system s1, then the following notation will be used:

$$I_{T1} = \{i \mid i \in I_T \wedge i \text{ is an impact on the technical capabilities of system s1}\}$$
$$I_{O1} = \{i \mid i \in I_O \wedge i \text{ is an impact on the operational environment for system s1}\}$$

As impacts are a direct consequence of a vulnerability being exploited through an attack vector with a certain level of probability of attack and success by a threat source with specific capabilities and motivation, impacts are just fluid and changing at those aspects on which they depend. Just as in other sections, the dynamic nature of the system will be modeled by defining the impacts for a specific system when it is in a particular state.

A recent attack will be detailed. Using the Russian attacks on the Georgian websites [Mar08], which would be the highly likely threat ($PA_{HighlyLikely}$) of a state

sponsored ($TS_{StateSponR}$) (($TC_{LevelOfExpertise}$)$_{VerySophisticated}$) successful attack ($PS_{HighlyLikely}$) on the attack surface (AS_{LU}) via the network ($AV_{Network}$) attack vector to an unsecured server (V_{LU}) motivated to forcibly change (TSM_{Change}) the behavior of the Georgian government, the following impacts are possibilities:

$$AS_{LU} = V_{LU} \cup AV_{Network}$$

$$I_{State\ r} = (TS_{StateSponR} \times TSM_{Change} \times (TC_{LevelOfExpertise})_{VerySophisticated}) \cup (PA_{HighlyLikely} \times PS_{HighlyLikely} \times PC_{HighlyCertain}) \cup AS_{LU}$$

The preceding example provides a possible state for impacts for that situational instance of a specific system in an operational environment. If $(I_{T1})_n$ and $(I_{O1})_n$ represent the set of impacts for system s1, in state *n,* then the set of all impacts for system s1 in state n is:

$$(I_{S1})_n = (I_{O1})_n \cup (I_{T1})_n$$

An assessment of system s1 will then consist of an assessment of the system with respect to threats, probabilities, attack surfaces, and impacts in all possible states of the system. Just as in the prior sections, we let Ev(s1) represent the assessment of the threats, probabilities, attack surfaces, and impacts of the system s1 and i represent the possible k number of states, then:

$$Ev(s1) = \cup_{i=1\ldots k} Ev_{AS}((AS_{s1})_i) \cup Ev_{TR}((TR_{s1})_i) \cup Ev_{P}((P_{s1})_i) \cup Ev_{I}((I_{s1})_i)$$

Impact models are the last consideration in determining risk. Some would define impact as the probability of risk impacting operations. In some ways, risk is difficult to define because a global impact would imply greater risk. However, a greater risk may be the greater risk of a negative impact to a system. NIST has the well-accepted definition that impact is a component of risk, and as such it will be followed.

3.8 Risk models

NIST Special Publication 800-30, Guide for Conducting Risk Assessments[41], defines risk as a function of the likelihood of a given threat-source's exercising a particular potential vulnerability (unmitigated flaw in this dissertation), and the resulting impact[42] of that adverse event on the organization. This definition does not imply any consideration of a situational instance other than a land-based system. Nor does it seem to consider vectors of attack or success of the attack. NIST explicitly indicates adverse impact in describing risk, which is interesting because any impact would be adverse just by virtue of identifying impact.

To address these considerations, risk (R) is further refined to be the probability of threat source(s) with the capabilities of exercising attack vector(s) to exploit vulnerability for specific motivation(s), the probabilities of success of the attack(s), the certainty of the knowledge, and the resulting impact(s).[43] To provide the greatest clarity for this complex equation, the equation will be built up from its constituent pieces.

First, the set of threats, which consists of a threat source, its associated capabilities, and its associated motivations, is represented.

$$TR = (\{TS\} \times TC^{+} \times TSM^{+})^{+}$$

For every threat, there is a probability that the threat source will attack, which is represented by Threat Attack set (TA):

$$TA = (TR^{+} \times \{PA\})^{+}$$

---

[41] Interestingly, it is not the Risk Management Framework (NIST SP800-39) but the Guide for Conducting Risk Assessments which provides the basis for threats, threat sources, and the risk model for the NIST publications, and hence, the US Government.

[42] NIST Special Publication 800-39 describes types of adverse impacts at all tiers in the risk management hierarchy.

[43] CNSSI 4009 Defines risk as a measure of the extent to which an entity is threatened by a potential circumstance or event, and typically a function of 1) the adverse impacts that would arise if the circumstance or event occurs; and 2) the likelihood of occurrence.

For every vulnerability, there are one or more attack vectors, whose pairing creates the set Attack Vector Vulnerability (AVV), as represented by:

$$AVV = (AV^{+} \times \{V\})^{+}$$

The probability of a successful attack can only occur if the threat has attack vector(s) to a vulnerability that it can exploit, which is represented by the set Attack Possibilities (AP):

$$AP = (TA^{+} \times (\{PS\} \times (AVV)^{+})^{+})^{+}$$

The information just modeled all depends upon the assessor's certainty of their knowledge of this information. However, that certainty level may vary depending upon whether the information is in regards to a threat, the probability of attack, the attack vector, or the vulnerability. As such, the above models become:

$$TR = (\{TS\} \times TC^{+} \times TSM^{+})^{+} \times \{PC\})^{+}$$
$$TA = (TR^{+} \times (\{PA\} \times \{PC\}))^{+}$$
$$AVV = ((AV^{+} \times \{PC\})^{+} \times (\{V\} \times \{PC\}))^{+}$$

Again, risk is the probability of threat source(s) with the capabilities of exercising attack vector(s) to exploit vulnerability for specific motivation(s), the probabilities of success of the attack(s), the certainty of the knowledge, and the resulting impact(s).

$$R = ((TA^{+} \times (\{PS\} \times AP^{+})^{+})^{+} \times I^{+})^{+}$$

Risk Origin: In NIST documentation, risk is a mix of both operational and technical risks. This, again, makes it difficult to separate the technical aspects of a current assessment from an operational environment aspect. As risk can be quantified separately in the technical system and in all operational environments, it is defined as:

R represents the set of all risk

$R_T \subseteq R$ is the set of risk for a technical system

$R_O \subseteq R$ is the set of risk for an operational environment

System level risk is never a single value because there is never just one flaw and countermeasure equating to a single vulnerability, and no one threat or impact to consider. Just as with the aspects previously discussed, risk is not static but fluid. As risk is a set of states, risk is refined to be the probability of a threat source with the capability of exercising an attack vector to exploit a vulnerability of a situational instance at an opportunity in time for a specific motivation, the probability of success of that attack, the certainty of the knowledge, and the resulting impact.[44] Representatively:

$$R = ((TA^{+} \times (\{PS\} \times AVV^{+})^{+})_{n} \times I_{n}$$

If $(R_{T1})_{n}$ and $(R_{O1})_{n}$ represent the set of risk for system s1, in state *n,* then the set of all risk for system s1 in state n is:

$$(R_{S1})_{n} = (R_{O1})_{n} \cup (R_{T1})_{n}$$

An assessment of system s1 will provide the risk for system s1. Just as in the prior sections, we let R(s1) represent the risk the threats, probabilities, attack surfaces, and impacts of the system s1 and i represent the possible k number of states, then:

$$R(s1) = \cup_{i=1\ldots k} Ev_{AS}((AS_{s1})_{i}) \cup Ev_{TR}((TR_{s1})_{i}) \cup Ev_{P}((P_{s1})_{i}) \cup Ev_{I}((I_{s1})_{i})$$

[44] CNSSI 4009 Defines risk as a measure of the extent to which an entity is threatened by a potential circumstance or event, and typically a function of 1) the adverse impacts that would arise if the circumstance or event occurs; and 2) the likelihood of occurrence.

The dynamic interaction between threats exploiting flaws and defensive entities implementing countermeasures for flaws is constant battle.[45] This fluidity requires risk not to be a single instance in time decision, but regularly re-assessed. One key decision is how often should risk be reconsidered to provide sufficient and accurate determinations.[46] The other key decision in the implementation of any risk decision methodology, including the RMF[47], is determining which security controls are to be assessed during assessments.

In NIST Special Publication 800-30 R1, the Guide to Conducting Risk Assessments, risk is generally assessed and grouped by types of impacts.[48] Based upon the models provided in this dissertation, risk can be assessed and grouped in multiple ways, thus providing increased situational awareness of risk to which one is exposed. NIST Special Publication 800-30 R1 also considers risk for time frames when impacts are likely to be experienced, which is equivalent to situational instances in this dissertation. This re-enforces the importance that situational instances be considered during operational assessments (especially for vehicles).

3.9 Technical and Operational Assessment Decomposition

In the 8500.2[49], it is stated that to achieve an acceptable level of IA requires interdependency among people, operations, and technology. This concept is demonstrated in the security controls, which are the required or prescribed countermeasures for a system, within 8500.2, DCID 6/3, and the NIST 800-53A. All three policies detail prescribed or required (dependent upon the policy) management, operational, and technical countermeasures to be implemented by systems.

The results of an assessment of an instantiation, using one of the above three methodologies, provides a body of evidence that is a mixture of operational and

---

[45] NIST Special Publication 800-30 R1 Defines this as threat shifting, which is the response of adversaries to perceived safeguards and/or countermeasures (i.e., security controls), in which adversaries change some characteristic of their intent/targeting in order to avoid and/or overcome those countermeasures.

[46] NIST Special Publication 800-30 R1 leaves the length of effectiveness of the results of a risk assessment to the organization to determine

[47] NIST Special Publication NIST 800-59

[48] NIST Special Publication 800-30 considers risk up through the organization level, which may roll up system levels risks into a single risk at the organization level

[49] 8500.2 E3.2.4.4. Integrated technical and non-technical defenses. Achieving an acceptable level of information assurance is dependent upon a synergy among people, operations and technology

technical arguments demonstrating that the assurance claim about the system is true. In NIST 800-53A, objects to be assessed include specifications, mechanisms, activities, and individuals, which again reinforces the mix of technical (specifications and mechanisms) and operational (activities and individuals).

In the 8500.2, enacted in 2003, there are 8 categories of security controls with a total of 157 security controls.[50] Whereas in the NIST 800-53A, which was enacted in 2010, there are 18 categories of security controls with a total of 634 security controls.[51] The growth in the number of categories and security controls is directly related to the introduction of lessons learned from the 8500.2 and DCID 6/3, as well as new technology. Currently, the categorization of controls is based upon a mixture of security functionality, environment, configuration management, personnel, and management in all of the existing methodologies.

The models presented in this chapter can guide an assessor in identifying and tailoring the security controls that a system must meet, as well as providing a risk model for the assessment of the security controls from any of the existing methodologies. These same models can assist the assessor is determining which security controls are technical and which are operational based, thus allowing the assessor to determine the Range of Warfare (RoW)[52] that is applicable to the operational environment.

3.10 Conclusion

Individual models for flaws, countermeasures, vulnerabilities, threats, probabilities, attack vectors, attack surfaces, impact, and risk have been detailed in this chapter. The use of these models should greatly improve the conciseness and

---

[50] 8500.2 Security categories and number of associated controls: Security Design & Configuration 31, Identification and Authentication 9, Enclave and Computing Environment 48, Enclave Boundary Defense 8, Physical and Environmental 27, Personnel 7, Continuity 24, Vulnerability and Incident Management 3

[51] 800-53A Security categories and number of associated controls: Technical 4, 262 Access Control 90, Audit and Accountability 146, Identification and Authentication 32, System and Communications Protection 94; Operational 9, 267 Awareness and Training 8, Configuration Management 41, Contingency Planning 45, Incident Response 20, Maintenance 22, Media Protection 18, Physical and Environmental Protection 48, Personnel Security 12, System and Information Integrity 53; Management 5, 85 Security Assessment and Authorization 13, Planning 8, Program Management 11, Risk Management 13, System and Services Acquisition 40

[52] Range of Warfare is the span from kinetic warfare to cyber warfare and everything in between.

objectivity of evidence for assessment by decoupling the technical assessments from operational environment assessments. Thus, allowing the assessor to more easily identify and tailor the security controls for that system, regardless of the assessment methodology. Further, the suggested concept of instances of environments of operation or states, allows for greater understanding by the assessor and risk authority of the scope and level of risk for the variety of systems within the DoD.

Chapter 4 Assessment Methodology

The reason robustness and assurance assessments are conducted on systems is to measure the confidence that the security of the system is implemented correctly and determine the risk that the system poses to the environments of operation. In such an assessment, evidence must be collected and assessed against a model. In existing assessment methodologies, that model is a live, single instantiation of a system in a specific operational environment. The examples provided for the methodology and models in this dissertation are based on a system. However, the methodology and models may also be applied to a system of systems.

It is very important to understand that the methodology presented in this dissertation is not meant to be a checklist or a formula to grade systems. Instead, it is meant provide an objective characterization of the system. As such, the examples are intentionally varied to require the reader to consider the abstraction of the methodology.

4.1 Model Methodology

The models presented support existing methodologies as guides for assessors to use for system design and development, security controls tailoring, and risk determinations. These same models provide a basis for a new assessment methodology, which will be referred to as the Model Methodology (MM). The MM is complementary to existing methodologies and can be used for either or both technical and operational assessments. For a technical assessment, the MM is conducted by an ISSE functioning as an assessor as the system is designed and developed, thereby completing that assessment when the system has completed development and prior to instantiation at any operational environment. The MM is also conducted by an ISSE functioning as an assessor for an operational environment assessment of a system as it is instantiated within that site.

4.2 Key Aspects

An ISSE is key to the entire methodology as it removes any possible bias from a vendor, design team, program manager, command, etc. It also forces the government

entity funding the system to allot and fund a separate entity, thereby increasing the visibility and difficulty of not funding the security engineer at the necessary levels to continue the assessment.

Applying the same methodology allows for ease of comparison of results of the technical and operational assessments, i.e. comparing oranges to oranges. The MM provides the ISSEs mechanisms to map the evidence to mathematical models to represent ISSE's assessment findings, thereby providing consistency of ISSE's findings. Currently, assessors and decision makers must rely on documents and designs provided by the vendor, which are biased to provide the assurance specified by the vendor. Testing of a system and its components, however, can only provide assurance regarding the exploits being exercised against known vulnerabilities.

Even with the use of models, there will always be some amount of human bias in an assessment. The MM includes models that allow the ISSE to quantify their subjective views, but its overall purpose is meant to limit the bias by presenting what is known, verifiable as well as the unknown and unverified. The models and their correlation to the evidence will help the risk acceptor visually identify the aspects of the system that were considered and the data correlation will indicate what was not confirmed with evidence.

The use of the models increases objectiveness/explicitness, repeatability, and knowledge of system robustness from assessor to risk acceptor, as well as assessor to assessor. The MM will increase system reuse, reciprocity, and risk acceptance while decreasing the amount of time for subsequent assessments. In both applications of the methodology, the individual models will be iteratively developed, fulfilling the needs of the assessor to represent their initial impression of the system's capabilities, represent the system's capabilities as it is assessed, and finally, to representatively correlate or map the completed models to the empirical evidence of the assessment.

The models, and the results they yield, must be simple enough for non-computer scientists or non-mathematicians to utilize and understand because the models provide a level playing field of understanding for those that implement them, as well as those that interpret their results. That will increase their use among all assessors and the lack of complexity will increase the consistency of their implementation. Although the models may be implemented in any methodology, the MM provides the

greatest benefit because of the separation of technical and operational assessments and risks.

The MM can be implemented at any time within the development lifecycle of a system. The earlier in the lifecycle the MM is implemented, the greater the evidence, such as documents, diagrams, mathematical proofs, testing, and ISSE's notes of each component of the system, that is available to the assessor. In addition, MM strongly integrates assessor with system's developers and engineers, which provides expertise and critical information to assessor as the system is developed.

4.3 Organizational ideas

The MM can produce a large amount of data. Organization of the models is vital. Otherwise key data could be lost, literally. As all of the other models are in direct relation to the flaw models, it is suggested that the models be grouped together with the flaw to which they relate. Many times when a flaw is identified, the ISSE will probably also identify attack vectors, threats, probabilities and possible countermeasures. Therefore, organizing by flaw is quite intuitive. This organizational structure will also allow for ease of mapping evidence to the models.

There will be times, such as during document reviews, where there will not be clear mappings between related models and flaws. In fact, document reviews can produce attack vectors, countermeasures, and threats without relating to a flaw already identified. In these cases, tables or lists of the models by document may provide organizational structure until other models are correlated to these document models.

Anytime threats are modeled, probabilities will also be modeled since the two are directly related. Threat motivation models describe what motivates the threat source. The probability of attack models describes how badly the threat source wants to attack. The level of confidence of an ISSE regarding a threat source, its capabilities, and its motivation is modeled by the probability of certainty of knowledge.

4.4 Stages

Within the MM, there are multiple stages with each stage correlating to the

progression of the assessor's exposure to the system. Key stages include:

- Initial Exposure
- System Familiarization
- Continuous Review
- Assessment
- Data Correlation

At each stage, the assessor iterates the individual models to represent their impression of the system's capabilities. As the ISSE 's knowledge of the system increases, the content of these models will go from generalized to specific as the assessment progresses. At each stage, the ISSE's correlate the models to the evidence available to them at that stage. The MM will provide a level of assessment detail not previously provided.

It is important to note that each assessment is individualistic and therefore, the number of stages and the stage at which a model is created will vary wildly based upon the system functionality, and the point in the lifecycle in which the system enters the MM, and the information available at that time. As such, any consistency or standardization of the number stages, the timing of the stage, or the time each stage encompasses is not to be expected. Nor is there any expected consistency or standardization at which stage a model is created and completed.

4.4.1 Initial Exposure

The initial stage encompasses the three sub-stages of initial exposure to the system. The ISSE must identify function of the system, its complexity, and possible states. In this early stage, the ISSE must gain an understanding of the entire system, including items such as peripherals (e.g. printers, scanners, removable media, etc) and sensors (e.g. antennae, cameras, etc.). Initial models may include flaws, countermeasures, attack vectors, threats, and probabilities.

4.4.1.1 Initial Contact

The initial contact is the very first exposure to the system that provides a very

basic overview of the system and its requirements. A rough concept of the system's architecture should be noted. In this stage the system's basic function is identified and possibly its complexity. The ISSE starts forming the models and may consider possible, obvious states.

Immediately upon being contacted there will be initial thoughts about the system that must be represented. Not all of the models may be considered for each initial contact, but the ISSE will work through the progression of the models. The following are some of the possible of flaws to initially consider for any system:

Technical

- Operating System (F_{OS}) $\subseteq F_{T1}$
- Applications (F_{Apps}) $\subseteq F_{T1}$
- Bluetooth ($F_{Bluetooth}$) $\subseteq F_{T1}$
- Wireless ($F_{802.11}$) $\subseteq F_{T1}$
- RF communications ($F_{RFComms}$) $\subseteq F_{T1}$

Operational

- RF communications ($F_{RFComms}$) $\subseteq F_{O1}$

Therefore, the initial flaw models would be something similar to:

$F_T = F_{OS} \cup F_{Apps} \cup F_{Bluetooth} \cup F_{802.11} \cup F_{RFComms}$

$F_O = F_{RFComms}$ (where $F_{RFComms}$ may be a singleton)

$F_{s1} = F_T \cup F_O$

The following are some of the possible mitigations to initially consider for any system:

Technial

- Operating System implements Discretionary Access Control (M_{OS}) $\subseteq M_T$
- Applications that aren't required are disabled (M_{Apps}) $\subseteq M_T$
- Bluetooth disabled ($M_{Bluetooth}$) $\subseteq M_T$

Operational

- RF communications are encrypted ($M_{RFComms}$) $\subseteq M_O$

Therefore, the initial countermeasure models would be something similar to:

$M_T = M_{OS} \cup M_{Apps} \cup M_{Bluetooth}$

$M_O = M_{RFComms}$ (where $M_{RFComms}$ may be a singleton)

$M_{s1} = M_T \cup M_O$

The above are high level, possible flaws that are not specific to a system. In this stage, most likely, the ISSE will not have had access to any documentation, so the above representation may be the entire detail to be noted. At this sub-stage, the many of the models will be those that are glaringly obvious, because there is just too much that is unknown.

4.4.1.2 Initial Review

The initial review represents something along the lines of a review of documentation provided to the ISSE or the ISSE is briefed on the system in order to provide more insight to the system. The ISSE updates the models and, if not already in existence, possibly creates the initial vulnerability, threats, attack vector, and attack surface models.

In this sub-stage, the ISSE does the initial mapping of the actual system instantiation to the models. This is done by identifying the actual hardware, software, and firmware as it is associated to the flaw, countermeasures, etc. Finally, the ISSE correlates the evidence used to map the system instantiation to the models.

There are many options available to the assessor to map the system as the assessment progresses. The following is a possible mapping of the system to the flaws:

Technical

- Operating System is unpatched Windows 7 as documented in section 4.1 of High Level Design Document (HLDD) ($F_{OSisUnpattchedWin7}$) $\subseteq F_{T1}$

- Unnecessary applications (Apache web server on UAV) not removed as documented in section 4.2 of HLDD ($F_{AppsNotRemoved}$) $\subseteq F_T$
- Wireless not using WPA as documented in section 4.4 of HLDD ($F_{802.11NoWPA}$) $\subseteq F_T$

Therefore, the flaw models would be updated to:

$F_T = F_{OSisUnpattchedWin7} \cup F_{AppsNotRemoved} \cup F_{802.11NoWPA}$

$F_{s1} = F_T \cup F_O$

The following are some of the possible mitigations to initially consider for any system:

Technical

- Operating System implements Discretionary Access Control as documented in section 4.1 of HLDD (M_{OS_DAC}) $\subseteq M_T$
- Bluetooth disabled as documented in section 4.4 of HLDD ($M_{BluetoothDisabled}$) $\subseteq M_T$

Operational

- RF communications encrypted as documented in section 4.4 of HLDD ($M_{RFCommsEncrypted}$) $\subseteq M_O$

Therefore, the countermeasure models would be updated to:

$M_T = M_{OS\_DAC} \cup M_{BluetoothDisabled} \cup M_{RFCommsEncrypted}$

$M_O = M_{RFComms}$ (where $M_{RFComms}$ may be a singleton)

$M_{s1} = M_T \cup M_O$

At this very initial stage, vulnerabilities, which are not completely mitigated flaws, should be identified, but the veracity of the vulnerabilities must be considered:

Technical

- Operating System is not patched as documented in section 4.1 of HLDD ($V_{OSNotPatched}$) $\subseteq V_T$

- Unnecessary applications (Apache web server on UAV) not removed as documented in section 4.2 of HLDD ($V_{AppsNotRemoved}$) $\subseteq V_T$
- Wireless not using WPA as documented in section 4.4 of HLDD ($V_{802.11NoWPA}$) $\subseteq V_T$

Therefore, the initial vulnerability models would be something similar to:

$V_T = V_{OSNotPatched} \cup V_{AppsNotRemoved} \cup V_{802.11NoWPA}$

$V_O = V_{802.11}$ (where $V_{802.11}$ may be a singleton)

$V_{s1} = V_T \cup V_O$

At this sub-stage, the key is to model the system as the ISSE reviews the documents. An ISSE should model everything that they note. At this point, the modeling of the ISSE's perceptions of the system is what is important. Certain points may jump out, such as a possible attack vector or probability of success of an attack, which though not founded in evidence, should be modeled because as the ISSE gains knowledge of the system their perceptions change and these initial models will allow the assessors to refer back to those ideas when the system was new to them.

While these initial models may not be accurate, the process will provide a mechanism for the ISSE to learn the accuracy of their models, thus allowing the ISSE to refine and improve their assessment techniques. This is especially true for mapping flaws, countermeasures, threats, and probabilities. The ISSEs, within a single assessment, will be able to see the accuracy of their initial models, because as the assessment progresses the models are refined, not by deleting the earlier models, but by appending the more refined models below the prior models.

4.4.1.3 Initial Architectural Review

The initial architectural review is the foundation for the ISSE's assessment of the system. In this sub-stage, many of the aspects of the system will be modeled, critical aspects of the system mapped to the model, architectural evidence correlated to the models, and initial risk of the assessment documented. Flaws, countermeasures, vulnerabilities, and attack vectors are frequently modeled in this sub-stage.

The critical aspects of a system are those very key concepts or details of a system, which are individualistic to each system. Critical aspects can include such items as:

- Security mechanisms
- Mission specific functionality
- Complexity

There is no general list of critical aspects that applies to every system. Obviously, security is a very general key concept that applies to all, but it is important to approach each assessment with prior experience without bias of similar systems.

Architectural evidence is also specific to a system, but there may be more similarity of evidence based upon the functionality of the system. This evidence is usually some combination of diagrams, documents, and scans. Assessors shouldn't automatically assume networking capability. As assessments may be for any combination of software, firmware, and hardware, the content of the evidence will vary widely. For example, most CDS systems must have evidence of how the security mechanism provides transfer capability, labeled separation, or isolation, depending upon the functionality of the CDS.

This is the first stage where the ISSE should identify the possible states. It is important to consider the functionality of the system when identifying possible states. As an example, the following are some of the possible states of an aircraft (UAV or manned) to consider:

- Aircraft powered up, parked, US military facility within the US airspace
- Aircraft in flight in US airspace
- Aircraft in flight within International air space
- Aircraft in flight in adversary's space covertly
- Aircraft in flight over conflict region in time of conflict

Whereas, in contrast, possible states of a CDS:

- The facility cleared for highest classification/compartment processed by the CDS

The initial risk determination will be based upon minimal evidence. This will not just be assessment evidence but also evidence of the implementation approach, as well as the risk attitude or culture of the designers, developers, and implementers. This initial risk determination may be heavily based upon experience because of the lack of evidence.

4.4.2 System Familiarization

System familiarization is the first stage where the ISSE is integrated with the system engineers. It is the basis for the complete mapping of the system to the models. In this stage, the possible threats to the system within its possible states should be identified.

4.4.2.1 Complete document review

The complete document review is the brunt of the mapping of the system, based upon documentation, to the models. The models should be well formed by the completion of this sub-stage. If no threat assessment has been conducted, research into possible threats must be requested or conducted by the ISSE. By the end of this sub-stage, the states to consider for the system should be identified.

4.4.2.2 Hands on

The hands-on sub-stage is dependent upon the point in the system lifecycle in which the ISSE encounters the system. If the system is in the development stage or beyond, this stage will allow the ISSE to map the actual system aspects (i.e. physical testing of the system) to the models, and therefore the documented system. In this sub-stage, previously unidentified flaws, countermeasures, and attack vectors may be identified and modeled. The risk of the system is revisited, and the AO is provided an initial risk recommendation.

This is the first sub-stage where the mapped models are integrated into the states. Using the example of a small (less than 30” in diameter) UAV, the states to consider could be:

- Aircraft in flight testing within a US Military base confines within the US airspace (UAV_{Base})
- Aircraft in flight in US civilian airspace (UAV_{US})
- Aircraft in flight conducting covert operations (UAV_{Covert})
- Aircraft in flight over conflict region in time of conflict ($UAV_{Conflict}$)

Using the mappings above and providing additional mappings, an example integration of the flaws to the possible states:

$F_{T,Base} = F_{OSisUnpattchedWin7} \cup F_{AppsNotRemoved} \cup F_{802.11NoWPA}$

$F_{O,Base} = F_{WeatherLimitingVisibility}$ (where $F_{WeatherLimitingVisibility}$ may be a singleton)

$F_{Base} = F_{T,Base} \cup F_{O,Base}$

$F_{T,US} = F_{OSisUnpattchedWin7} \cup F_{AppsNotRemoved} \cup F_{802.11NoWPA}$

$F_{O,US} = F_{WeatherLimitingVisibility} \cup F_{802.11Interference} \cup F_{CivilianAircraftInterference} \cup F_{AccidentalJamming}$

$F_{US} = F_{T,US} \cup F_{O,US}$

$F_{T,Covert} = F_{OSisUnpattchedWin7} \cup F_{AppsNotRemoved} \cup F_{802.11NoWPA}$

$F_{O,Covert} = F_{WeatherLimitingVisibility} \cup F_{802.11Interference} \cup F_{CivilianAircraftInterference} \cup F_{AccidentalJamming} \cup F_{Detection}$

$F_{Covert} = F_{T,Covert} \cup F_{O,Covert}$

$F_{T,Conflict} = F_{OSisUnpattchedWin7} \cup F_{AppsNotRemoved} \cup F_{802.11NoWPA}$

$F_{O,Conflict} = F_{WeatherLimitingVisibility} \cup F_{802.11Interference} \cup F_{CivilianAircraftInterference} \cup F_{KineticWeapons} \cup F_{IntentionalJamming}$

$F_{Conflict} = F_{T,Conflict} \cup F_{O,Conflict}$

4.4.3 Continuous Review

As the ISSE may encounter the system at any point in its lifecycle, the continuous review stage is focused on a technical assessment of a developing system.

During this stage, the ISSE is continually reviewing aspects of the system and updating the mapping of the system to the models, correlating new evidence to the models, updating possible states, updating mapped models to states, and updating the AO as required by the AO.

4.4.4 Assessment

The assessment stage is the official and final assessment of system in this implementation of the MM, which is usually conducted by a team of ISSEs. In this stage, many tasks must be completed.

Tasks that were completed in previous stages should be reviewed as the first task in the final assessment to make sure the entire team is on the same page prior to starting the actual assessment. The next task is to verify the system's architecture with the system's ISSEs, followed by physical verification.

In another task, the ISSEs must revisit any previous testing conducted, especially any tests during the Hands-On stage. At a minimum, spot-checking should be conducted on any of the previous testing. Any testing conducted in this stage, must be correlated to the models at this point, as well as the models of the previous stage where testing was conducted.

During this stage, the team will organize the models into their agreed structure, whether it is a tree or tables, etc. It is a significant task to organize the previously developed models and integrate into the models for this assessment. This task should be done in conjunction with the mapping of the system and its evidence.

The possible states to consider are finalized. The mapping of evidence to the system will include the models of these states. It is important to note that it may not be possible to actually assess the system in all of its possible states. However, the ISSE's must still map as much evidence to the possible states as available.

This is the stage in which the ISSE attempts to exploit modeled flaws and vulnerabilities using modeled attack vectors, by-pass modeled countermeasures, and exfiltrate data from an insider perspective. This stage should identify additional flaws, vulnerabilities, and attack vectors, as well as possibly identifying additional countermeasures. In simple terms, the vast majority of evidence and the reality of

system's behavior will be identified.

There are multiple outcomes to the final assessment. The completeness of system documentation is assessed and documented. The system's architecture is verified. Finally, the AO is provided an updated risk recommendation.

4.4.5 Data Correlation

The data correlation stage is the official correlation of evidence to the models. This will be the most complete correlation of system evidence to the models to this point. The AO is provided the final risk recommendation, the format of which is dependent upon the AO. This translation of evidence to the models is a key point. Evidence is anything that provides verification of a compromise, as well as anything that provides verification of the inability to compromise.

Evidence that confirms vulnerabilities and attack vectors, at a minimum, will be restricted and in most cases, classified, even for an unclassified system. That will also cause the correlation of the evidence to the models to also be restricted or classified. As with the examples provided in the models sections, the following is realistic data but not an actual representation of any known system.

As mentioned in earlier sections, evidence can be many things, the most common being documents and actual testing. An important item to remember is that documentation is rarely as accurate a representation of the system as actual testing. This shortfall is because the documentation isn't constantly updated for those systems in development and is not updated once the system is instantiated. The exception to this shortfall are those systems designed and developed using formal modeling.

In many cases the ISSE's must rely on documentation, as some aspects (vulnerabilities, attack mechanisms, etc.) of the system may not be able to be tested due to lack of capability, time, or funding. That is why ISSEs must be very adept at analyzing documentation, as it may form the basis for their assessment arguments. An excellent instance of documentation used as evidence is the reliance on OS design documentation to "verify" internal aspects of the OS.

As an example, on the JSF there is a controlled interface[53] that implements a formally modeled OS, called SecureOS. The SecureOS's security mechanism labels all data, storage, network connections, and processes. In this case, the ISSE can review the design documents from the Low Level Design Document down to the formal assumptions of the system. The ISSE would correlate the evidence presented in those documents to the models.

In this limited example, the JSF would have the following flaws, countermeasures, and attack vectors:

- Pilot's helmet is from foreign supplier and connects to the aircraft network via 802.11 ($F_{SupplyChain}$)
- Operating System providing separated security domains ($M_{SecureOS}$)
- Wireless including enabling wireless modem (AV_{RF})

In this following instance, the adversary could use the 802.11 communications link as the attack vector to activate an exploit stored in Pilot's helmet to attempt to gain access to the aircraft network.

$$JSF_{Exploit1} = F_{SupplyChain} \cup AV_{RF}$$

However, the SecureOS contains the 802.11 connection point for the helmet, thus requiring all communications to and from the helmet through SecureOS. Thus, the above exploit would be mitigated, as represented below:

$$JSF_{S1} = F_{SupplyChain} \cup M_{SecureOS} \cup AV_{RF}$$

To correlate the countermeasure model to evidence, the ISSE must confirm as evidence that SecureOS has the ability to assuredly label. Very few ISSEs have the capability to assess the robustness of a labeled OS. Therefore, the ISSE must rely on

[53] Per DCID 6/3 a controlled interface is a mechanism that facilitates adjudicating the security policies of different interconnected information systems (e.g., controlling the flow of information into or out of an interconnected information system.

documentation, preferably assured through another source (such as a Common Criteria Evaluation). The data correlation may look similar to:

> SecureOS implements labeled security mechanism as documented in Section 4.4, pages 54-80 of the Common Criteria Target Of Evaluation
>
> $M_{LabelMechanism} \subseteq M_{SecureOS}$
>
> $M_{SecureOS} \subseteq M_{T1}$

It is obvious that as the assessment progresses the volume of data representing the models and the data correlation will rapidly increase. The organization of the models is a key point, because, there may not necessarily be evidence to correlate to all models. There may be no documented evidence to correlate threat sources, their capabilities, and probably not their motivations. As previously noted, the probability models provide the ISSE with the capability to model their subjective perspectives with regard to probability of and success of an attack because there will probably be very little evidence to map to these models. The probability of the certainty of knowledge provides the ISSE with the capability to model their confidence in their knowledge, for which there may or may not be evidence to map.

4.5 Conclusion

The MM provides an objective representation or characterization of the system by providing the mechanisms to map the mathematical models to the evidence from the assessment findings. The models, and the results they yield, provide an equal level of understanding for those that implement them, as well as those that interpret their results. There are multiple stages within the MM, with each stage representing and refining the ISSE's impression of the system.

As with all assessment methodologies, the MM is not meant to be a checklist and unlike the NIST SP 800-30, it is not meant to provide a formula to grade a system. Table 1 is a simple summary of the stages, their primary outputs, and the expected models for that stage. It is important to note that while Table 1 does contain the expected outputs, it does not contain expected evidence or the method that was used

to obtain the evidence, such as penetration testing, as those are unique to each assessment.

Table 1 Methodology Overview

Stage/Substage	Output	Possible Models
Initial Exposure	Identify the function of the system, its complexity, possible states	Flaws, Countermeasures
Initial Contact	Basic overview of the system and its requirements	Flaws, Countermeasures
Initial Architectural Review	Foundation of the assessment, modeling many system aspects	Flaws, Countermeasures, Vulnerabilities, Attack Vectors
System Familiarization	Basis for the complete mapping of the system including identifying threats within possible states	Threats, Probabilities
Complete document review	Brunt of the mapping of the system, acquire or conduct threat assessment, identify possible states	Threats, Probabilities
Hands on	Map actual system to models, map models integrated into states	Flaws, Countermeasures, Attack Vectors, Probabilities
Continuous Review	Continuous assessment of system and updating of models, correlating evidence to models, updating states, updating models to states	Flaws, Countermeasures, Vulnerabilities, Threats, Attack Vectors, Attack Surface, Probabilities
Assessment	Review of previous stages, verify system's architecture, revisit previous testing, correlate testing to models, finalize states, conduct testing including exploiting the system	Flaws, Countermeasures, Vulnerabilities, Threats, Attack Vectors, Attack Surface, Probabilities, Impact
Data Correlation	Final correlation of the evidence to the models	Flaws, Countermeasures, Vulnerabilities, Threats, Attack Vectors, Attack Surface, Probabilities, Impact, Risk

Chapter 5 Validation

The MM is a methodology and a collection of models detailed by an ISSE to provide a consistent, repeatable, objective way to pass on knowledge to a wide audience. However, the methodology and models must be evaluated to provide an indication of its tangible benefits. Thus, an assessment was conducted implementing the MM to to empirical measure the usefulness and objectiveness of the MM. Essentially, how well did the models and methodology actually work? Did these provide a manner of transferring the subjective assessment into a model that actually helps other assessors? Was it a useful reference guide to someone that had not previously conducted or participated in an assessment? The following paragraphs discuss the answers to these questions.

5.1 Validation Approach

The approach to validate the MM was multi-faceted. One part of the validation approach was having other assessment experts review this dissertation and provide their expert opinions. Another aspect was presentations, but publicly available and USG specific presentations. The final part of the validation approach was to conduct assessment implementing the MM.

Only one assessment was conducted because to the expense of an assessment, the time it takes to conduct an assessment, and ability to release the actual models and findings of the assessment. Two assessors conducted the assessment with other technical experts participating as necessary to conduct certain technical aspects of the assessment. Only two assessors were selected to provide the maximum perceptibility of progress with the least amount of influence.

The assessment validation approach was to have two assessors conduct an assessment of a Department of Defense (DoD) project implementing the methodology and models. The first assessor had no prior assessment experience and initially conducted the assessment as the lone assessor. Then, after a time, the second assessor that has extensive assessment experience joined the assessment team. At the end of the assessment, both assessors provided their individual assessments, as well as a combined assessment of the MM.

5.2 Measures

To evaluate the methodology, some measure is required. There are five methodology measures. Two of which are the assessor's individual perspectives of the methodology and models. The experienced assessor provides two more measures with regards to the inexperienced assessor. The final measure is the combined perspective of the two experience extremes conducting an assessment.

While the inexperienced assessor had an educational background in forensics, there was no evaluation or assessment background. Nor was there any offensive cyber background or offensive perspective. This is important because the assessor had no experience whatsoever to draw on to know where to start an assessment as the lone assessor. Therefore, the inexperienced assessor was expected to provide the most valuable measure with regards to the MM being a useful reference guide. The inexperienced assessor's report is contained in Appendix D.

The experienced assessor was a subject matter expert with experience in conducting assessments using multiple methodologies, as well as having offensive cyber experience. This experience provides an offensive perspective on the modeling of vulnerabilities, attack vectors, threats, and probabilities. As such, it was expected that this assessor would provide a good measure on those models in addition to the methodology. Also, the experienced assessor was expected to analyze the MM in comparison to the prior assessment methodologies implemented by that assessor.

The experienced assessor also provided two more individual measures. After the assessor joined the in-progress assessment, they determined the inexperienced assessor's approach and the extent accomplished of the in-progress assessment. Once the experienced assessor determines extent of assessment progress, an analysis was conducted. The extent the inexperienced assessor actually accomplished with the models and methodology was determined, and compared to the extent they were expected to accomplish without methodology and models. The direct contrast of the experienced assessor's offensive background to the inexperienced assessor's lack of offensive background provided an excellent measure of the inexperienced assessor's models to capture flaws, attack vectors, threats, and probabilities. By having both of the assessors involved in the same assessment of a system allowed the more

experienced assessor to provide a metric on the inexperienced assessors implementation of the MM. The experienced assessor's report is contained in Appendix E.

The final measure was the combined perspective of the two assessors. The two assessors conducted this analysis at the end of the assessment. This measure, though influenced by their vastly differing experiences, was to determine if the models conveyed objective information and if the methodology provided a useful reference guide regardless of an assessor's experience level. Their report is contained in Appendix C.

5.3 Validation Assessment

The Defense Advanced Research Projects Agency (DARPA) works with academia and industry to solve the hard problems facing the US Government. The High Assurance Cyber Military System (HACMS) project is a partner effort among academia, industry, and the US Government, whose goal is to demonstrate the cyber security and financial benefits of implementing high assurance aspects within a system. An assessment will be conducted of HACMS instantiated on a COTS quad copter (UAV) using the MM.

The assessment was conducted using Flaw Hypothesis Methodology[54] (FHM) with the verification of a limited number of possible flaws. A paper detailing the implementation of the MM, including the models generated, was released to the HACMS Program Manager (PM) and the author upon its completion. The assessment report was released after the HACMS PM had reviewed the report.

All systems aboard the UAV were to be assessed, though only tests where there was no chance of damage/destruction to the UAV were conducted. The assessment focused on those aspects of the UAV that an adversary would target. The following paragraphs will discuss the models created during the assessment. The models themselves are included in Appendix B.

The eight system states of the UAV that the assessment team considered are

[54] Flaw hypothesis methodology is a systems analysis and penetration prediction technique where a list of hypothesized flaws in a system are compiled through analysis of the specifications and documentation for the system.

listed in a table in Appendix B. As the focus of the assessment was the UAV itself, and did not include the controller, there are no states representing the controller, its location, or its operator.

Appendix B contains an overview table of the models for flaws, countermeasures, and vulnerabilities. The table also includes the associated states for the models. The flaw models are further delineated into technical and operational states in a table in Appendix B.

Appendix B also contains a table of the vulnerability models. This table includes flaws that have mitigations. Interestingly, the team added 3 models not considered before for vulnerabilities: severity, predisposing conditions, and pervasiveness. These additional models, provides verification that the MM represents ISSE's assessment findings, thereby providing consistency in assessments. It also provides verification that human bias continues to influence assessments because of the addition of a severity and pervasiveness ratings, which are based purely on human experience.

The threat sources modeled appear in a table within Appendix B. The team included some of the threat sources listed within in this dissertation, but added a number of their own threat sources. The team also broke out the sources into types, with the types of individual, group, organization, nation-state, and nature. These types could indicate funding, political, technical, and/or like-mindedness, which could indicate motivations or capabilities influences. Another model included was adversary's intent, which is a rating from very low to very high. This is another example of the human influence biasing assessments. There is no evidence provided by the team to justify their rating of threat sources intent.

Appendix B contains the table of threat capabilities associated with the threat sources. Also included in the table is an associated the probability of successful attack for each threat source. The team chose to use rating based, descriptive wording for the threat source's expertise, resource, success, or capability, again ranging from very low to very high.

The threat source's motivations, including covertness, are itemized in a table in Appendix B. The team added a new model for targeting, which ranges from very low to very high.

Appendix B contains the table of the probabilities correlated to the threat source. Interestingly, this is the only table that the team uses descriptive language versus rating language.

5.4 Summary of Assessors' Findings

The inexperienced assessor reported that the MM significantly advanced their ability and confidence to conduct assessments. The same assessor identified that assessors could keep lists of the models that will grow with time. While it is common to reuse models for functionally similar systems, but no previous consideration had been given to a list of models.

The experienced assessor reported that the inexperienced assessor was further along in the assessment than would have been expected for having no prior assessment experience. The report also indicated that the MM increased thoroughness, especially for flaws and threat models, than was expected for a first assessment. The experienced assessor reported that the models provide standardized form for communicating findings. Such confirmation from an experienced assessor is validation of the usefulness of the MM. The experienced assessor confirmed that each assessor conducts assessments on an individual basis, and there is no consistent assessment methodology. That the MM provides the ability for any assessor to have a reference to another assessor's system characterization and considerations is another indication of the usefulness of the MM with which the assessors agree.

Both of the assessors indicated that the models provided a good reference guide and allowed them to characterize the system. The two assessors came to the same conclusion as the author, in that assessors could reference the models for future assessments. It was also the assessors' conclusion that the models are methodology independent, as well as scalable, both in system complexity and model detail. While methodology independence and scalability were both goals of the MM, the level of detail included within a model was assumed to vary based upon the information available. So, scalability was an assumption that has been confirmed.

In the assessors' report, the assessors were focused on the models early in the assessment. There could be many reasons why they indicated the models were only

used early in the assessment, though there are two distinct probabilities. First, the team was against a deadline for conducting the assessment, and therefore they prioritized the verification of the possible flaws over the process. Or, since the assessment was limited to FHM, there was no need to iterate the models.

The experienced assessor confirmed that each assessor conducts assessments on an individual basis, and there is no consistent assessment methodology. Both assessors concluded on the need for a standardized form for communicating findings, and that the models provide such a means. Such confirmation from an experienced assessor is validation of the usefulness of the MM. That the MM provides the ability for any assessor to have a reference to another assessor's system characterization and considerations is another indication of the usefulness of the MM with which the assessors agree.

Both of the assessors strongly indicated the need to provide a number to represent the risk factor of a system. The experience assessor had previously assessed systems using the NIST SP 800-30 methodology, therefore a number was to be expected, since that methodology does provide a numeric risk rating. The junior assessor more strongly associated the probabilities to numeric values. Both assessors want an equation to objectively state the risk value of the system.

There are several reasons why numeric risk ratings were not included or encouraged in the MM. Primarily, past history has shown that numeric rating systems have proven to be misleading and can provide an illusion of assurance. That said, just because the methodologies in the past weren't accurate doesn't mean one can't be developed which is accurate. Such a methodology would need to be able to capture unknowns, whether they are vulnerabilities, threats, etc., numerically. That would be no small feat. The development of a numeric risk rating is an excellent candidate for future work.

5.5 Expert Opinions

In addition to the expert, experienced assessor that conducted the validation assessment, other experts reviewed this dissertation and proffered their opinions. Primarily, cyber assessors and risk acceptors or AOs, conducted these informal,

expert validations. These reviews were conducted by the request of the experts, two of which are included in this dissertation to be representative of the assessment community. The opinions provided are the opinions of the experts and not official comments.

A Designated Authorizing Official (DAO) is a representative of an AO authorizes systems on behalf of the AO. One of the experts that reviewed this dissertation and provided an opinion is a DAO, who is an expert in Cybersecurity and an experienced DAO, but has not ever assessed a system. This expert indicated that they will use the models to guide their assessors to the level of characterization provided by models, as well as use the methodology as guide in requesting information from assessors

The other expert to review this dissertation is an experienced assessor, who has conducted assessments using multiple methodologies. After reviewing this dissertation, this assessment expert indicated they had never considered level of thoroughness provided by models, and this dissertation was a great reference guide. The expert also indicated they will use methodology and models on future assessments in which they are involved.

5.6 Presentations

As this dissertation has been years in writing, there have been many presentations, some of which are publicly available. Papers were submitted to multiple conferences. The accepted conferences included the High Confidence Software and Systems Conference, the Layer Assurance Workshop, and the Cyber Security Symposium. As these conferences are attended by a wide variety of cyber personnel, both in experience and area of expertise, the feedback represented a cross section of the target audience for this dissertation.

5.7 Conclusions

There are several key aspects of the MM presented in this dissertation. The MM provides mechanisms to map assessment evidence to mathematical models to represent assessment findings, thereby providing consistency of ISSE's findings, as well as a mechanism to communicate those findings. The individual models are

iteratively developed, fulfilling the needs of the assessor to represent their initial impression of the system's capabilities, represent the system's capabilities as it is assessed, and finally, to representatively correlate or map the completed models to the empirical evidence of the assessment. The models and their correlation to the evidence will help the risk acceptor visually identify the aspects of the system that were considered and the data correlation will indicate what was not confirmed with evidence.

In conclusion, this dissertation presents an assessment methodology, the MM, that complements existing assessment methodologies, mathematical models to provide objective characterization of a system, situational state perspectives into assessments, and separated technical and operational assessments. The methodology and models were evaluated against a set of measures by a team of assessors consisting of personnel with mixed levels of assessment experience, as well as by experts in the field of system assessment.

Chapter 6 Future Work

There are several easily identified aspects of future work. These include further validation, mapping the NIST RMF security controls to the MM, abstracting the NIST 800-53 R4 security controls, creating a mapping of abstracted security controls, and creating additional models. Two less obvious future work projects would be to map the MM to DO-178C and map the fault tree for security critical aspects to the fault tree for safety critical aspects.

6.1 Further Validation

The evaluation presented in this dissertation was based on a small sample. Additional assessments should be conducted implementing the MM, by a variety of assessors. Following each assessment, an analysis should be conducted to provide refinement and further validation the methodology and models.

6.2 Map MM to NIST 800.53 R4 Security Controls

As previously discussed in this dissertation, there are differing policies and implementations of security controls. Currently, the most commonly implemented security controls are those from the NIST 800-53 process. The Joint Chiefs of Staff (JCS) network was compromised even though their security posture complied with all security controls of the NIST 800-53, as well as implementing perimeter-based security, patch management, and mitigating or removing known vulnerabilities.

If a JCS ISSE were to implement the MM, they must individually map the models to the NIST 800-53 security controls they are required to meet. This would be true of any ISSE implementing the MM and its models. The single most useful future work project would be to map the NIST 800-53 R4 security controls to the models, which would provide consistency and greatly ease the workload of all ISSEs.

6.3 Abstract NIST 800.53 R4 Security Controls

If the security controls were abstracted, it would ease any mapping of future security control frameworks or implementations. There is no doubt that there will be follow on to the NIST 800-53 security control framework. There always is a follow

on. Abstracting the security controls provided in the NIST 800-53 R4 would provide the greatest benefit of the existing security controls. It is not known if such an abstraction is possible.

6.4 Map MM to Abstracted Security Controls

If the abstracted security controls were mapped into the MM and the models, it would increase the consistency among all assessment communities currently in use. It would also provide a mechanism to more easily map the MM to future security controls. Even if the abstraction is possible, it is not known if it would be possible to map those abstractions to the MM and the models.

6.5 Additional Models

Creating additional models to further define risk in the terms of criticality origin, criticality, and data sensitivity would allow the ISSE to provide the AO with a more accurate representation of the ISSE's perception of risk. The original NIST Special Publication 800-30 considered, but did not define, both data sensitivity and data criticality in the Risk Assessment Methodology [NIS02]. The risk model provided in NIST Special Publication 800-30 R1 does not consider mission criticality[55], system criticality, data sensitivity[56], or data criticality.[57] Nor do the policies set forth within NIST Special Publication 800-53A, DCID 6/3 and DoDI 8500.2. Interestingly, these concepts aren't included in any of the new risk methodologies. These risk considerations would provide further detail on the value of what is being protected, and further justification for separating technical and operational environment assessments, as the mission and system criticality are specific to the operational environment.

---

[55] NIST Special Publication 800-30 R1 per NIST Special Publication 800-60 Criticality is a measure of the degree to which an organization depends on the information or information system for the success of a mission or a business function

[56] NIST Special Publication 800-30 R1 per NIST Special Publication 800-60 Sensitivity is a measure of the importance assigned to information by its owner, for the purpose of denoting its need for protection

[57] The NIST Special Publication 800-60[56] considers data criticality and sensitivity but only with respect to security categorization.

The criticality of the mission, system, and data are crucial in determining the operational risk associated with a system. Mission criticality is the importance of completing the intended mission. Similarly, system criticality is the importance of the system continuing to operate as intended. Whereas, data criticality is the perceived importance of the data residing, transported, or acquired on the system during the mission. As previously discussed in this proposal, there are multiple methodologies to quantify data sensitivity currently. Classification levels for the DoD are an example of one such methodology. However, the information, such as classification, may not be known precisely during development.

The extent of mission, system, and data criticality may not be known during a technical assessment. Therefore, it is important to note that while this data may be considered during a technical assessment, such models are more for future operational ISSEs and all other risks.

6.6 Determine the optimal approach to document the models

It quickly becomes apparent that the MM creates an extraordinary number of models. The author uses tables, which allow for easy searching and sorting. This method does not lend itself to easy visualization of the correlation of the models to each other. Others use trees to correlate the models to each other and to the evidence. Research needs to be conducted to determine the optimal method of documenting the models.

6.7 Implement Mathematical Probabilities

The probabilities provided in this dissertation use English notation to allow the ISSE to capture their characterization of the probabilities. Research will need to be conducted to determine if the English notation can be put into mathematical terms and still provide the same level of characterization. The danger of this approach is that it could quickly turn the objectiveness into a grading scale whose implementation is inconsistent, thus defeating the purpose of the modeling of the probabilities in English.

6.8 Map MM to DO-178C

A less obvious course of future work is to map the MM to the DO-178C. There is no mapping, currently, between the security critical tree and the safety critical tree. That mapping may need to be completed prior to mapping the MM to the DO-178C. There have been multiple attempts over the last several years to create this one to one mapping, but so far all efforts have met with failure. It may be possible to map the MM to the DO-178C without mapping between the trees. The mapping of MM to DO-178C may also yield the mapping of the security critical tree to the safety critical tree as one of its effects.

The mapping to DO-178C would allow the reuse of all those systems assessed using DO-178C to be reused with confidence within the security community. Because DO-178C requires formal modeling of its systems, the mapping would provide high robustness systems to the security community that are not currently in use by the security community. Both of these mappings, the trees and the models, may be arduous tasks.

Bibliography

[And13] Anderson, E. A., Irvine, C. E. and Schell, R. R. Subversion as a Threat. in *Journal of Information Warfare*. N.D. Retrieved 19 July 2013 from Journal of Information Warfare: https://www.jinfowar.com/journal/volume-3-issue-2/subversion-threat-information-warfare.

[Bai08] Bailey, M. The Unified Cross Domain Management Office: Bridging security domains and cultures. Defense Technical Information Center, July 2008. Retrieved July 3, 2016 from Defense Technical Information Center (DTIC®): http://www.dtic.mil/get-tr-doc/pdf?AD=ADA487191.

[Bar13] Barnett RDML (Ret), Jamie. Supply Chain Threat. Potomac Institue for Policy Studies, 2012. Retrieved 2013 from Potomac Institute: http://www.potomacinstitute.org/attachments/article/1276/Cyber%20paper%20Barnett.pdf.

[Bel11] Belik V., G. T. B. D. Recurrent hostmobility in spatial epidemics: beyond reaction-diffusion. in *European Physical Journal* B (EPJ B), vol. 84, no. DOI: 10.1140/epjb/e2011-20485-2, p. 579–587, 2011.

[Byr10] Byrd, P. R., Naegle, B. and Boudreau, M. Test Plan Framework for Cross Domain Solution (CDS) Devices. US Naval Postgraduate School, June 2010. Retrieved 19 July 2013 from US Naval Postgraduate School: http://edocs.nps.edu/npspubs/scholarly/JAP/2010/Jun/10Jun_Byrd_JAP.pdf.

[CER13] CERT: Computer Emergency Response Team Secure Coding, 2013. Retrieved 15 July 2013 from Carnegie Mellon University Software Engineering Institute Computer Emergency Response Team: http://www.cert.org/secure-coding/.

[Cha12] Chabrow, E. DoD Takes Aim at Supply Chain Threat Defending against Off-the-Shelf Tech Reprogrammed to Spy, Steal. in *Government Information Security*, 4 December 2012. Retrieved 15 July 2013 from Information Security Media Group, Corp: http://www.govinfosecurity.com/dod-takes-aim-at-supply-chain-threat-a-5333/op-1.

[Chi12] Chirgwin, R. Software bug flattens NYSE trader, Plunged into $US440 MILLION loss. in *The Register*, 2 August 2012. Retrieved September 2013 from The Register: http://www.theregister.co.uk/2012/08/02/knight_capital_trading_bug/.

[Cle13] Cleghorn, L. Network Defense Methodology: A Comparison of Defense in Depth and Defense in Breadth. in *Journal of Information Security,* vol. 4, no. doi:10.4236/jis.2013.43017, pp. 144-149, 2013.

[Cli95] Clinton, William J. Executive Order 12958. Office of the President, 17 April 1995. Retrieved 25 September 2013: http://www.fas.org/sgp/clinton/eo12958.html.

[CNS10] Committee on National Security Systems. Committee on National Security Systems Instruction 4009: Committee on National Security Systems Glossary. 26 April 2010. Retrieved 24 September 2013 from US Department of Defense, Committee on National Security Systems: https://www.cnss.gov/CNSS/openDoc.cfm?shh5A9gWEjxsV+001OxD3g==.

[Coh86] Cohen, B., Harwood, W. T. and M.I., J. The Specification of Complex Systems. Wokingham: Addison-Wesley, 1986

[Cox08] Cox Jr, L. A. Some limitations of "Risk = Threat x Vulnerability x Consequence" for risk analysis of terrorist attacks. National Institutes of Health, 2008 Dec;28(6) :1749-61. doi: 10.1111/j.1539-6924.2008.01142.x. Epub 2008 Oct 16. Retrieved 4 July 2016 from National Institutes of Health US National Library of Medicine: http://www.ncbi.nlm.nih.gov/pubmed/19000071.

[DHS16] Department of Homeland Security, Industrial Control System Cyber Emergency Response Team: Common Cyber Security Language. US Department of Homeland Security, N.D. Retrieved 4 July 2016 US Department of Homeland Security, Industrial Control System Cyber Emergency Response Team: https://ics-cert.us-cert.gov/sites/default/files/documents/Common%20Cyber%20Language_S508C.pdf.

[Dir99] Director of. Central Intelligence. Director of Central Intelligence Directive 6/3 PROTECTING SENSITIVE COMPARTMENTED INFORMATION WITHIN INFORMATION SYSTEMS. Central Intelligence Agency, 5 June 1999. Retrieved http://www.fas.org/irp/offdocs/dcid.htm.

[DoD07] UD Department of Defense. DoD Directive 8500.01E. US Department of Defense, 34 April 2007. Retrieved 4 June 2012: http://www.dtic.mil/whs/directives/corres/pdf/850001p.pdf.

[DOD85] US Department of Defense. DEPARTMENT OF DEFENSE STANDARD DEPARTMENT OF DEFENSE TRUSTED COMPUTER SYSTEM EVALUATION CRITERIA, DoD 5200.28-STD. December 1985. Retrieved 4 July 2016 from National Institute of Standards and Technology Computer Security Resource Center: http://csrc.nist.gov/publications/history/dod85.pdf

[Exp16] Experienced Assessor. An Assessment Methodology and Models for Cyber Systems, Appendices B, C, E. University of Idaho, June 2016.

[Fil12] Filsinger, J., Fast, B., Wolf, D., Payne, J. F. X., and Anderson, M. Supply Chain Risk Management Awareness. Armed Forces Communication and Electronics Association (AFCEA) Cyber Committee, February 2012. Retrieved 24 September 2013 from Armed Forces Communication and Electronics Association (AFCEA) Cyber Committee: http://www.afcea.org/committees/cyber/documents/Supplychain.pdf.

[Geo12] Georgia Institute of Technology, Emerging Cyber Threats Report 2013: Insecurity of the Supply Chain: Hard to Detect, Expensive to Fix, and a Policy Nightmare. in *Georgia Tech Cyber Security Summit 2012*, (2012), Georgia Institute of Technology, 4. Retrieved 24 September 2013, from Georgia Institute of Technology: http://www.gtcybersecuritysummit.com/pdf/2013ThreatsReport.pdf.

[Gig12] Gigante, G. and Pascarella, D. Formal Methods in Avionic Software

Certification: The DO-178C Perspective. Springer-Verlag, Berlin Heidelberg, 2012, (1}. Retrieved 4 July 2016 from Springer.com: http://link.springer.com/chapter/10.1007%2F978-3-642-34032-1_21#page-2.

[Gre13] Greenwald, G. and MacAskill, E. NSA Prism program taps in to user data of Apple, Google and others. in The Guardian, 6 June 2013. Retrieved15 July 2013 from The Guardian: http://www.theguardian .com/world/2013/jun/06/us-tech-giants-nsa-data.

[Gru14] Grubb, B. Man who introduced serious 'Heartbleed' security flaw denies he inserted it deliberately. *The Sydney Morning Herald*, 11 April 2014. Retrieved 12 July 2016 from The Sydney Morning Herald: http://www.smh.com.au/it-pro/security-it/man-who-introduced-serious-heartbleed-security-flaw-denies-he-inserted-it-deliberately-20140410-zqta1.html

[HRC11] HRCDSTT. UCDMO High Robustness Tiger Team Presentation. in *Unified Cross Domain Management Office Conference*, 2011.

[Ias13] Iasiello, E. Stuffing the Genie Back in the Bottle: Can Threats to the IT Supply Chain Be Mitigated? in *Foreign Policy Journal*, 2 April 2013. Retrieved15 July 2013 from Foreign Policy Journal: http://www. foreignpolicyjournal.com/2013/04/03/stuffing-the-genie-back-in-the-bottle-can-threats-to-the-it-supply-chain-be-mitigated/.

[Ine16] Inexperienced Assessor. An Assessment Methodology and Models for Cyber Systems, Appendices B, C, D. University of Idaho, June 2016.

[Inf07] Information Assurance Directorate. US Government Directory Protection Profile For Medium Robustness Environments. National Security Agency, 1 September 2004. Retrieved24 September 2013, from Common Criteria Portal: http://www.commoncriteriaportal.org/files /ppfiles/PP_DIR_MR_V1.0.pdf.

[Inf13] InfoSecurity: 2013 Information Security Threat Predictions: Cyber War, Cloud and BYOD, 2013. Retrieved 15 July 2013 from The

Guardian: http://www.guardian.co.uk/media-network/partner-zone-infosecurity/2013-information-security-threat-predictions.

[Jac12] Jackson, W. Supply Chain Threats 'hard to detect, expensive to fix'. in *Government Computer News*, 15 November 2012. Retrieved 24 September 2013 from Government Computer News: http://gcn.com/Articles/2012/11/15/Supply-chain-threats-hard-to-detect-expensive-to-fix.aspx?p=1.

[Joh09] Johnson, T. Mathematical Modeling of Diseases: Susceptible-Infected-Recovered (SIR) Model. 2009. Retrieved 23 July 2013 from University of Minnesota Morris: www.morris.umn.edu/academic/math/Ma4901/.../Teri-Johnson-Final.pdf or https://www.researchgate.net/publication/242272678_Mathematical_Modeling_of_Diseases_Susceptible-Infected-Recovered_SIR_Model.

[Kew13] Kewley, D. L. and Lowry, J. Observations on the effects of defense in depth on adversary behavior in cyber warfare. in *USMA_IEEE02*, 18, 1-8 Retrieved 5 June 2013: http://craigchamberlain.com/library/insider/Observations%20on%20the%20effects%20of%20defense%20in%20depth%20on%20adversary%20behavior%20in%20cyber%20warfare.pdf

[Mar08] Markoff, J. Georgia Takes a Beating in the Cyberwar With Russia. in *The New York Times*, 12 August 2008. Retrieved 24 September 2013 from The New York Times: http://www.nytimes.com/2008/08/13/technology/13cyber.html?_r=0.

[McP08] McPherson, A., Proffitt, B., and Hale-Evans, R. Estimating the Total Development Cost of a Linux Distribution. The Linux Foundation, October 2008. Retrieved July 2016, from The Linux Foundation: http://www.linuxfoundation.org/sites/main/files/publications/estimatinglinux.html

[Mee13] Meeks, M., Anderson, R., Khoo, W. M. and Aloteibi, S. Hunting for vulnerabilities in large software : the OpenOffice suite. N.D. Retrieved 19 July 2013 from The Computer Laboratory University of Cambridge:

http://gonullyourself.org/library/openoffice9.pdf.

[Mic12] Microsoft: History, A history of Windows:Highlights from the first 25 years. Microsoft Corp, N.D. Retrieved 5 February 2012 from Microsoft: http://windows.microsoft.com/en-US/windows/history.

[Nav12] NavSource Online: PCU Gerald R. Ford. NavSource, N.D. Retrieved 4 June 2012 from NavSource: http://www.navsource.org/archives/02/78.htm.

[NIA09] National Information Assurance Partnership –Common Criteria Evaluation Validation Scheme: FAQs. 2009. Retrieved 5 February 2013 from National Information Assurance Partnership: http://www.niap-ccevs.org/faqs/niap_evolution/FAQs28Mar_v6.pdf.

[NIA11] National Information Assurance Partnership - Common Criteria Evaluation Validation Scheme: Home page. 2011. Retrieved 25 September 2013 from National Information Assurance Partnership: http://www.niap-ccevs.org/.

[NIA111] National Information Assurance Partnership - Common Criteria Evaluation Validation Scheme: Defining the CCEVS. 2011. Retrieved 25 September 2013 from National Information Assurance Partnership: http://www.niap-ccevs.org/about/defined/.

[NIA12] National Information Assurance Partnership- Common Criteria Evaluation Validation Scheme: Frequently Asked Questions for NIAP/CCEVS and the Use of Common Criteria in the US, 28 March 2012. Retrieved 24 September 2013, from National Information Assurance Partnership: http://www.niap-ccevs.org/NIAP_Evolution/faqs/niap_evolution/FAQs28Mar_v6.pdf.

[Nic02] Nichols, Arthur. A Perspective on Threats in the Risk Analysis Process. SANS Institute, 2002. Retrieved 4 July 2016 from SANS: https://www.sans.org/reading-room/whitepapers/auditing/perspective-threats-risk-analysis-process-63

[NIS12] National Institue of Standards and Technology. Special Publication

800-30 R1: Guide for Conducting Risk Assessments. National Institue of Standards and Technology, September 2012. Retrieved 11 August 2013 from National Institute of Standards and Technology Computer Security Resource Center: http://csrc.nist.gov/publications/nistpubs/800-30-rev1/sp800_30_r1.pdf.

[NIS13] National Institute of Standards and Technology. NIST Special Publication 800-53 R4 Security and Privacy Controls for Federal Information Systems and Organizations. National Institute of Standards and Technology, April 2013. Retrieved 25 September 2013: http://dx.doi.org/10.6028/NIST.SP.800-53r4.

[NRL13] US Naval Research Laboratory Center for High Assurance Computer Systems: Products, 2013. Retrieved 19 July 2013 from US Naval Research Laboratory: http://www.nrl.navy.mil/itd/chacs/5544/products.

[NSA12] National Security Agency: Commercial Solutions for Classified Program. National Security Agency, March 2012. Retrieved 25 September 2013 from National Security Agency: http://www.nsa.gov/ia/programs/csfc_program/index.shtml.

[NST00] National Security Telecommunications and Information Systems Security Committee (NSTISSC). National Security Telecommunications and Information Systems Security Policy (NSTISSP) No. 11, Subject: National Policy Governing the Acquisition of Information Assurance (IA) and IA-Enabled Information Technology (IT) Products Fact Sheet. National Security Telecommunications and Information Systems Security Committee, January 2000. Retrieved http://www.niap-ccevs.org/cc-scheme/nstissp11_factsheet.pdf.

[Opp01] Oppliger, R. Internet and Intranet Security. Artech House, 2001, 12. Retrieved 12 July 2016 from Artech House computer security series, ISBN 1580531660.

[Ora10] Oracle: SOLARIS WITH TRUSTED EXTENSIONS. Oracle, N.D.

Retrieved 5 February 2012 from Oracle: http://www.oracle.com/us/products/servers-storage/solaris/solaris-trusted-ext-ds-075583.pdf.

[Red12] Red Hat Linux: About. Red Hat, N.D. Retrieved 5 February 2012 from Red Hat: http://www.redhat.com/about/.

[SAI07] Science Applications International Corp. Boeing Secure Network Server Security Target. Science Applications International Corp, 6 April 2007. Retrieved June 2012 from Common Criteria Portal: http://www.commoncriteriaportal.org/files/epfiles/st_vid10127-st.pdf.

[Slo11] Slobodova, A., Davis, J., Swords, S., and Hunt Jr, W. A Flexible Formal Verification Framework for Industrial Scale Validation University of Texas Austin Computer Science Department, 2011. Retrieved 24 September 2013 from University of Texas Austin Computer Science Department: http://www.cs.utexas.edu/~jared/publications/2011-memocode-centaur.pdf.

[Sma11] Small, P. E. Defense in Depth: An Impractical Strategy for a Cyber World. SANS Institute, 14 November 2011. Retrieved 25 September 2013 from SANS Institute: http://www.sans.org/reading_room/whitepapers/assurance/defense-depth-impractical-strategy-cyber-world_33896.

[Sne15] Sneps-Sneppe, M. On telecom services and the DISN evolution. in *International Journal of Open Information Technologies* ISSN: 2307-8162 vol. 3, no. 8, 2015. Retrieved 4 July 2016 from International Journal of Open Information Technologies: http://injoit.org/index.php/j1/article/download/224/178.

[Ste13] Stephenson, P. R. and Prueitt, P. S. Towards a Theory of Cyber Attack Mechanics. N.D. Retrieved 3 August 2013 from OntologyStream: http://www.ontologystream.com/gFT/Towards%20a%20Theory%20of%20Cyber%20Attack%20Mechanics.PDF.

[Sto89] Stoneburner, G. and Snow, D. The Boeing MLS LAN Headed Towards an INFOSEC Security Solution. in *National Institute of Standards and*

Technology, National Computer Security Center 12th Annual Computer Security Conference, (Baltimore, MD, 1989), National Institute of Standards and Technology, 254-266. Retrieved July 2016: http://csrc.nist.gov/publications/history/nissc/1989-12th-NCSC-proceedings.pdf.

[Wal13] Waldrop, M. DARPA and the Internet Revolution. 50 Years of Bridging the Gap, (78-85). N.D. Retrieved 26 July 2013 from US Defense Advanced Research Projects Agency: www.darpa.mil/WorkArea/DownloadAsset.aspx?id=2554.

[Win12] Wingfield, B. 1 Power-Grid Cyber Attack Seen Leaving Millions in Dark for Months. *Bloomberg Business Week Bloomberg Technology,* February 2012. Retrieved 24 September 2013 from Bloomberg Business Week Bloomberg Technology: http://www.bloomberg.com/news/2012-02-01/cyber-attack-on-u-s-power-grid-seen-leaving-millions-in-dark-for-months.html.

Appendix A Acronyms

AO	Authorizing Official
C	Confidential (DoD Classification Level)
C&A	Certification & Accreditation
CCEVS	Common Criteria Evaluation and Validate Scheme (CCEVS)
CDS	Cross Domain Solution
CI	Controlled Interface
CSFC	Commercial Solutions for Classified program
CVN	Nuclear-powered Aircraft Carrier (Aircraft Carrier, Nuclear)
DAA	Designated Approving Authority
DCID	Director Central Intelligence Directive
DIACAP	DoD Information Assurance Certification and Accreditation Process
DITSCAP	DoD Information Technology Security Certification and Accreditation Process
DoD	Department of Defense
EAL	Evaluation Assurance Level
FAA	Federal Aviation Authority
FIPS	Federal Information Processing Standard
FOUO	For Official Use Only
GIG	Global Information Grid
GPS	Global Positioning Satellite System
IA	Information Assurance
IC	Intelligence Community
ICCITSE	International Common Criteria for Information Technology Security Evaluation
IO	Information Operations
IT	Information Technology
ISSE	Information Security System Engineer
JWICS	Joint Worldwide Intelligence Communication System
LAN	Local Area Network
LSPP	Labeled Security Protection Profile

MLS	Multi-Level Security
MM	Model Methodology
NIACAP	National Information Assurance Certification and Accreditation Process
NIAP	National Information Assurance Acquisition Policy
NIPRNet	Non-classified Internet Protocol Router Network
NIST	National Institutes of Standards and Technologies
NOC	Network Operating Center
NSA	National Security Agency
NSS	National Security System
NSTISSP	National Security Telecommunications and Information Systems Security Policy
OS	Operating System
PIT	Platform Information Technology
PL	Protection Level
PP	Protection Profile
S	Secret (DoD Classification Level)
SABI	Secret And Below Interoperability
SCI	Sensitive Compartmented Information
SIPRNet	Secret Internet Protocol Router Network
TABI	Top Secret And Below Interoperability
TCSEC	Trusted Computer System Evaluation Criteria
TOE	Target of Evaluation
TS	Top Secret (DoD Classification Level)
TSABI	Top Secret SCI And Below Interoperability
U	Unclassified (DoD Classification Level)
UAV	Unmanned Aerial Vehicle
UCDMO	Unified Cross Domain Management Office
UCDSMO	Unified Cross Domain Services Management Office
US	United States
USG	United States Government
USN	United States Navy

Appendix B Validation Assessment Models

The validation assessment team authored all of the tables contained within this appendix [Ine16] [Exp16].

Table 2 System States

State	Abbreviation
Parked, powered off, safe zone	PFS
Parked, powered on, safe zone	PN
Parked, powered on, armed	PNA
In flight, safe space (SFL)	SFL
In flight, Conflict space	CFL
Parked, powered off, conflict zone	PFSC
Parked, Powered on, conflict zone	PNC
Parked, powered on, armed, conflict zone	PNAC

Table 3 Flaw, Countermeasure, and Vulnerability Models

Type	Technical/ Operational	State	Mitigation	Vulnerability
F-OS-VMCPU Overload	Technical	PN, PNA, SFL, CFL, PNC, PNAC	OSVMisolation	
F-OS-CryptoASide ChannelAttack-GlobalState	Technical	PN, PNA, SFL, CFL, PNAC, PNC		V-OS-CryptoASide ChannelAttack-GlobalState
F-OS-peripheral drivers	Technical	PN, PNA, SFL, CFL, PNC, PNAC		V-OS-peripheral drivers

Type	Technical/ Operational	State	Mitigation	Vulnerability
F-OS-hypervisor rootkit	Technical	PN, PNA, SFL, CFL, PNAC, PNC		V-OS-hypervisor rootkit
F-OS-Bidirectional HTMLComms	Technical	PNA, SFL, CFL, PNAC		V-OS-Bidirectional HTMLComms
F-OS-hypervisor drivers/ HWcomms	Technical	PN, PNA, SFL, CFL, PNAC, PNC		V-OS-hypervisor drivers/ HWcomms
F-OS-CANBus Packet Reassembly	Technical	PN, PNA, SFL, CFL, PNAC, PNC		V-OS-CANBus Packet Reassembly
F-SOS-Arming State-broken	Technical	PN, PNA, SFL, CFL, PNAC		V-SOS-Arming State-broken
F-SOS-MC/FC independently DecryptPackets	Technical	PN, PNA, SFL, CFL, PNAC		V-SOS-MC/FC independently DecryptPackets
F-SOS-CANBus Overload	Technical	PN, PNA, SFL, CFL. PNAC		V-SOS-CANBus Overload
F-SOS-peripheral StoreMalicous Material	Technical	PN, PNA, SFL, CFL, PNC, PNAC		V-SOS-peripheral StoreMalicous Material
F-SOS-GC/ ReceiverPacket Interception	Technical	PN, PNA, SFL, CFL, PNAC		V-SOS-GC/ ReceiverPacket Interception

Type	Technical/ Operational	State	Mitigation	Vulnerability
F-SOS-Hyperthreading	Technical	PN, PNA, SFL, CFL, PNC, PNAC		V-SOS-Hyperthreading
F-SOS-Odroid Reboot	Technical	PN, PNA, SFL, CFL, PNC, PNAC		V-SOS-Odroid Reboot
F-SOS-EmulateOdroid KillSwitch	Technical	PN, PNA, SFL, CFL, PNC, PNAC		V-SOS-EmulateOdroid KillSwitch
F-GPS-Spoofing	Technical	PNA, SFL, CFL, PNAC	M-GPS-Anti-spoofing algorithm	
F-GPS-Timing	Technical	PNA, SFL, CFL, PNAC		V-GPS-Timing
F-Calibration-PostCal Orientation	Technical	PNA, SFL, CFL, PNAC	M-Single CalibrationBlue ForceControl	
F-Calibration-Barometer	Technical	PNA, SFL, CFL, PNAC		V-Calibration-Barometer
F-Calibration-Compass Jamming	Technical	PNA, SFL, CFL, PNAC		V-Calibration-Compass Jamming
F-Camera	Technical	PN, PNA	M-Camera-ExternalNetwork	

Type	Technical/ Operational	State	Mitigation	Vulnerability
F-Encryption-MC/FC Jamming	Technical	PN, PNA, SFL, CFL, PNC, PNAC		V-Encryption-MC/FC Jamming
F-Encryption-DisablesFor Landing	Technical	SFL, CFL, PNA, PNAC		V-Encryption-DisablesFor Landing
F-Encryption-Receiver Mismatch	Technical			V-Encryption-Receiver Mismatch
F-Encryption-InputChecking	Technical			V-Encryption-InputChecking
F-Encryption-SignalEmissions	Technical	PN, PNA, SFL, CFL, PNC, PNAC		V-Encryption-SignalEmissions
F-VoltageSpoof	Technical	PN, PNA, PNC, PNAC, SFL, CFL		V-VoltageSpoof
F-Internet	Technical & Operational	PN, PNA, SFL, CFL, PNC, PNAC		V-Internet
F-EthernetPort	Technical	PN, PNA, PNC, PNAC	M-EthernetPort-IsolatedVM	
F-RFComms	Technical & Operational	PN, PNA, SFL, CFL, PNC, PNAC	M-RFComms-Higher BandwidthLink Components	
F-Logging	Technical & Operational			V-LoggingNot Tracked

Type	Technical/ Operational	State	Mitigation	Vulnerability
F-SupplyChain	Technical & Operational	PFS, PFSC	M-SupplyChain-PedigreeAnalysis	
F-CM-Peripheral Drivers	Technical	PN, PNA, SFL, CFL, PNC, PNAC		V-CM-Peripheral Drivers
F-CM-StateMismatch	Technical	PN, PNA, IF	M-CM-DisabledReboot PostArming	
F-CM-Emulate KillSwitch	Technical	PN, PNA, SFL, CFL, PNC, PNAC		V-CM-Emulate KillSwitch
F-Physical-Theft	Operational	PFS, PN, PNA, SFL, CFL, PNC, PNAC, PFSC		V-Physical-Theft
F-Physical-Kinetic	Operational	PFS, PN, PNA, SFL, CFL, PNC, PNAC, PFSC		V-Physical-Kinetic
F-USBPort	Operational	PFS, PN, PNA, PNC, PNAC, PFSC	M-USBPort-Anti-Tamper Housing	
F-MSDPort	Operational	PFS, PN, PNA, PNC, PNAC, PFSC	M-MSDPort-Anti-Tamper Housing	

Table 4 Flaw State - Parked, powered off, safe zone (PFS)

F(Technical) PFS	F(Operational) PFS
F-SupplyChain	F-Physical-Kinetic
	F-USBPort
	F-MSDPort
	F-Physical-Theft
	F-SupplyChain

Table 5 Validation Assessment Flaw State - Parked, powered on, safe zone (PN)

F(Technical) PN	F(Operational) PN
F-OS-VMCPUOverload	F-Physical-Theft
F-OS-CryptoASideChannelAttack-GlobalState	F-Physical-Kinetic
F-OS-peripheraldrivers	F-USBPort
F-OS-hypervisorrootkit	F-MSDPort
F-OS-BidirectionalHTMLComms	F-Internet
F-OS-hypervisordrivers/HWcomms	
F-OS-CANBusPacketReassembly	
F-SOS-ArmingState-broken	
F-SOS-MC/FC independentlyDecryptPackets	
F-SOS-CANBusOverload	
F-SOS-peripheralStoreMalicousMaterial	
F-SOS-GC/ReceiverPacketInterception	
F-SOS-OdroidReboot	
F-SOS-EmulateOdroidKillSwitch	
F-Camera	
F-Encryption-MC/FCJamming	
F-Encryption-SignalEmissions	
F-VoltageSpoof	
F-Internet	
F-EthernetPort	
F-RFComms	
F-CM-PeripheralDrivers	
F-CM-StateMismatch	
F-CM-EmulateKillSwitch	
F-RFComms	
F-SOS-Hyperthreading	

Table 6 Flaw State - Parked, powered on, armed (PNA)

F(Technical) PNA	F(Operational) PNA
F-OS-VMCPUOverload	F-Physical-Theft
F-OS-CryptoASideChannelAttack-GlobalState	F-Physical-Kinetic
F-OS-peripheraldrivers	F-USBPort
F-OS-hypervisorrootkit	F-MSDPort
F-OS-BidirectionalHTMLComms	F-RFComms
F-OS-hypervisordrivers/HWcomms	F-Internet
F-OS-CANBusPacketReassembly	
F-SOS-ArmingState-broken	
F-SOS-MC/FC independentlyDecryptPackets	
F-SOS-CANBusOverload	
F-SOS-peripheralStoreMalicousMaterial	
F-SOS-GC/ReceiverPacketInterception	
F-SOS-Hyperthreading	
F-SOS-OdroidReboot	
F-SOS-EmulateOdroidKillSwitch	
F-GPS-Spoofing	
F-GPS-Timing	
F-Calibration-PostCalOrientation	
F-Calibration-Barometer	
F-Calibration-CompassJamming	
F-Camera	
F-Encryption-MC/FCJamming	
F-Encryption-DisablesForLanding	
F-Encryption-SignalEmissions	
F-VoltageSpoof	
F-Internet	
F-EthernetPort	
F-RFComms	
F-CM-StateMismatch	
F-CM-EmulateKillSwitch	

Table 7 Flaw State – In flight, safe space (SFL)

F(Technical) SFL	F(Operational) SFL
F-OS-VMCPUOverload	F-Physical-Theft
F-OS-CryptoASideChannelAttack-GlobalState	F-Physical-Kinetic
F-OS-peripheraldrivers	F-RFComms
F-OS-hypervisorrootkit	F-Internet
F-OS-BidirectionalHTMLComms	
F-OS-CANBusPacketReassembly	
F-OS-hypervisordrivers/HWcomms	
F-SOS-ArmingState-broken	
F-SOS-MC/FC independentlyDecryptPackets	
F-SOS-CANBusOverload	
F-SOS-peripheralStoreMalicousMaterial	
F-SOS-GC/ReceiverPacketInterception	
F-SOS-Hyperthreading	
F-SOS-OdroidReboot	
F-SOS-EmulateOdroidKillSwitch	
F-GPS-Spoofing	
F-GPS-Timing	
F-Calibration-PostCalOrientation	
F-Calibration-Barometer	
F-Calibration-CompassJamming	
F-Encryption-MC/FCJamming	
F-Encryption-DisablesForLanding	
F-Encryption-SignalEmissions	
F-VoltageSpoof	
F-Internet	
F-RFComms	
F-CM-PeripheralDrivers	
F-CM-EmulateKillSwitch	

Table 8 Flaw State – In flight, conflict space (CFL)

F(Technical) CFL	F(Operational) CFL
F-OS-VMCPUOverload	F-Physical-Theft
F-OS-CryptoASideChannelAttack-GlobalState	F-Physical-Kinetic
F-OS-peripheraldrivers	F-Internet
F-OS-hypervisorrootkit	F-RFComms
F-OS-BidirectionalHTMLComms	
F-OS-hypervisordrivers/HWcomms	
F-OS-CANBusPacketReassembly	
F-SOS-ArmingState-broken	
F-SOS-MC/FC independentlyDecryptPackets	
F-SOS-CANBusOverload	
F-SOS-peripheralStoreMalicousMaterial	
F-SOS-GC/ReceiverPacketInterception	
F-SOS-Hyperthreading	
F-SOS-OdroidReboot	
F-SOS-EmulateOdroidKillSwitch	
F-GPS-Spoofing	
F-GPS-Timing	
F-Calibration-PostCalOrientation	
F-Calibration-Barometer	
F-Calibration-CompassJamming	
F-Encryption-MC/FCJamming	
F-Encryption-DisablesForLanding	
F-Encryption-SignalEmissions	
F-VoltageSpoof	
F-Internet	
F-RFComms	
F-CM-PeripheralDrivers	
F-CM-StateMismatch	
F-CM-EmulateKillSwitch	

Table 9 Flaw State – Parked, powered off, conflict zone (PFSC)

F(Technical) PFSC	F(Operational) PFSC
F-SupplyChain	F-SupplyChain
	F-Physical-Theft
	F-Physical-Kinetic
	F-USBPort
	F-MSDPort

Table 10 Flaw State – Parked, powered on, conflict zone (PNC)

F(Technical) PNC	F(Operational) PNC
F-OS-VMCPUOverload	F-Physical-Theft
F-OS-CryptoASideChannelAttack-GlobalState	F-Physical-Kinetic
F-OS-peripheraldrivers	F-USBPort
F-OS-hypervisorrootkit	F-MSDPort
F-OS-hypervisordrivers/HWcomms	F-RFComms
F-OS-CANBusPacketReassembly	F-Internet
F-SOS-peripheralStoreMalicousMaterial	
F-SOS-Hyperthreading	
F-SOS-OdroidReboot	
F-SOS-EmulateOdroidKillSwitch	
F-Encryption-MC/FCJamming	
F-Encryption-SignalEmissions	
F-Internet	
F-EthernetPort	
F-RFComms	
F-CM-PeripheralDrivers	
F-CM-EmulateKillSwitch	

Table 11 Flaw State – Parked, powered on, armed, conflict zone (PNAC)

F(Technical) PNAC	F(Operational) PNAC
F-OS-VMCPUOverload	F-Physical-Theft
F-OS-CryptoASideChannelAttack-GlobalState	F-Physical-Kinetic
F-OS-peripheraldrivers	F-USBPort
F-OS-hypervisorrootkit	F-MSDPort
F-OS-BidirectionalHTMLComms	F-RFComms
F-OS-hypervisordrivers/HWcomms	F-Internet
F-OS-CANBusPacketReassembly	
F-SOS-ArmingState-broken	
F-SOS-MC/FC independentlyDecryptPackets	
F-SOS-CANBusOverload	
F-SOS-peripheralStoreMalicousMaterial	
F-SOS-GC/ReceiverPacketInterception	
F-SOS-Hyperthreading	
F-SOS-OdroidReboot	
F-SOS-EmulateOdroidKillSwitch	
F-GPS-Spoofing	
F-GPS-Timing	
F-Calibration-PostCalOrientation	
F-Calibration-Barometer	
F-Calibration-CompassJamming	
F-Encryption-MC/FCJamming	
F-Encryption-DisablesForLanding	
F-Encryption-SignalEmissions	
F-VoltageSpoof	
F-Internet	
F-EthernetPort	
F-RFComms	
F-CM-PeripheralDrivers	
F-CM-EmulateKillSwitch	

Table 12 Validation Assessment Vulnerability Models

Vulnerability	Vuln. w/ Mitigation	Vuln. Severity	Predisposing Condition	Pervasiveness
	F-OS-VMCPU Overload	Low	Information related	Low
V-OS-CryptoASideChannel Attack-GlobalState		Very Low	Technical	Low
V-OS-peripheral drivers		Very Low	Operational	Low
V-OS-hypervisor rootkit		Very Low		
V-OS-BidirectionalHTML Comms		Low		
V-OS-hypervisordrivers/ HWcomms		Very Low		
V-OS-CANBusPacket Reassembly		Low		
V-SOS-ArmingState-broken		Low		
V-SOS-MC/FC independently DecryptPackets		Very Low		
V-SOS-CANBusOverload		Low		
V-SOS-peripheralStore MalicousMaterial		Very Low		
V-SOS-GC/ ReceiverPacket Interception		Low		

Vulnerability	Vuln. w/ Mitigation	Vuln. Severity	Predisposing Condition	Pervasiveness
V-SOS-Hyperthreading		Very Low		
V-SOS-OdroidReboot		Low		
V-SOS-Emulate OdroidKillSwitch		Very Low		
	F-GPS-Spoofing	Moderate		
V-GPS-Timing		Moderate		
	F-Calibration-PostCal Orientation	Low		
V-Calibration-Barometer		Low		
V-Calibration-CompassJamming		Low		
	F-Camera	Moderate		
V-Encryption-MC/FCJamming		Moderate		
V-Encryption-DisablesForLanding		Very Low		
V-Encryption-ReceiverMismatch		Low		
V-Encryption-InputChecking		Very Low		
V-Encryption-SignalEmissions		Very Low		
V-VoltageSpoof		Very Low		
V-Internet		Very Low		
	F-EthernetPort	Very Low		
	F-RFComms	Moderate		
V-LoggingNot Tracked		Low		
	F-SupplyChain	Low		

Vulnerability	Vuln. w/ Mitigation	Vuln. Severity	Predisposing Condition	Pervasiveness
V-CM-Peripheral Drivers		Low		
	F-CM-State Mismatch	Low		
V-CM-EmulateKill Switch		Low		
V-Physical-Theft		Moderate		
V-Physical-Kinetic		Moderate		
	F-USBPort	Low		
	F-MSDPort	Very Low		

Table 13 Validation Assessment Threat Source Models

Source	Type	Intent	Adversary's Intent
Outsider	Individual	Intentional	Very Low
Insider	Individual	Intentional	Moderate
Trusted Insider	Individual	Intentional	Moderate
Privileged Insider	Individual	Intentional	Moderate
Terrorist	Individual	Intentional	Very High
Hacker	Individual	Intentional	Low
Criminal Organization	Group	Intentional	Moderate
Competitor	Organization	Intentional	High
Supplier	Organization	Intentional	High
Partner	Organization	Intentional	High
Customer	Organization	Intentional	Very Low
Nation-state	Nation-State	Intentional	High
User	Individual	Accidental	
Privileged User/ Admin	Individual	Accidental	
Natural Disaster	Natural	Accidental	
Manmade Disaster	Organization	Accidental	
Infrastructure Failure/Outage	Organization	Accidental	
Unusual Natural Event	Nature	Accidental	

Table 14 Validation Assessment Threat Capability Models

Source	Expertise	Resource	Success	Capability
Outsider	Very Low	Low	Very Low	Very Low
Insider	Low	Medium	Medium	Medium
Trusted Insider	Medium	High	High	Medium
Privileged Insider	Very High	Very High	Very High	Very High
Terrorist	High	High	High	Medium
Hacker	Very High	High	Very High	High
Criminal Organization	High	High	High	Medium
Competitor	Very High	Medium	Medium	Medium
Supplier	High	High	High	Medium
Partner	High	Very High	High	Medium
Customer	High	Medium	Low	Low
Nation-state	High	Very High	Very High	High
User	Medium	Medium	Medium	Low
Privileged User/ Admin	Very High	Very High	Very High	Very High
Natural Disaster	Very Low	Very High	Medium	Very Low
Manmade Disaster	Very Low	Medium	Medium	Very Low
Infrastructure Failure/Outage	Very Low	High	High	Very Low
Unusual Natural Event	Very Low	Low	Low	Very Low

Table 15 Validation Assessment Threat Motivation Models

Source	Motivation	Targeting	Covertness
Outsider	Intelligence	Very Low	Yes
Insider	Power	Low	Yes
Trusted Insider	Power	Moderate	Yes
Privileged Insider	Financial	High	Yes
Terrorist	Change	High	No
Hacker	Financial	Low	Yes
Criminal Organization	Financial	Low	Yes
Competitor	Intelligence	High	Yes
Supplier	Intelligence	Moderate	Yes
Partner	Financial	Moderate	Yes
Customer	Change	Low	No
Nation-state	National Security	Very High	Yes
User	N/A	Very Low	No
Privileged User/ Admin	N/A	Very Low	No
Natural Disaster	N/A	Very Low	No
Manmade Disaster	N/A	Very Low	No
Infrastructure Failure/Outage	N/A	Very Low	No
Unusual Natural Event	N/A	Very Low	No

Table 16 Validation Assessment Probability Models

Source	Probability of Attack	Probability of Success of Attack	Probability of Certainty of Knowledge
Outsider	Highly Unlikely	Unlikely	Uncertain
Insider	SomewhatLikely	SomewhatLikely	Uncertain
Trusted Insider	SomewhatLikely	SomewhatLikely	Uncertain
Privileged Insider	SomewhatLikely	SomewhatLikely	Uncertain
Terrorist	SomewhatLikely	SomewhatLikely	Uncertain
Hacker	SomewhatLikely	SomewhatLikely	Uncertain
Criminal Organization	SomewhatLikely	SomewhatLikely	Uncertain
Competitor	SomewhatLikely	SomewhatLikely	Uncertain
Supplier	SomewhatLikely	SomewhatLikely	Uncertain
Partner	SomewhatLikely	SomewhatLikely	Uncertain
Customer	Unlikely	Unlikely	Uncertain
Nation-state	HighlyLikely	SomewhatLikely	Uncertain
User			
Privileged User/ Admin			
Natural Disaster			
Manmade Disaster			
Infrastructure Failure/ Outage			
Unusual Natural Event			

Appendix C Combined Assessor's Comments

The validation assessment team authored the text of this appendix [Ine16] [Exp16].

The Assessment Methodology, Models for Cyber Systems, by Jennifer Guild, provides a mathematical model to assess systems, as well as an assessment methodology that builds upon that model to assess those systems. The intentions of the paper were to enable an assessor with little to no experience, and no mathematical background, to thoroughly examine and assess a National Security System. The very nature of these systems are dynamic, as are the personnel that defend them, therefore the models proposed are also intended to be dynamic. Ms. Guild's dissertation provides potential solutions for an assessor to lay the groundwork for future assessments, so that a new or revisiting assessor can continue the assessment by updating the models to match the system's changing state and the knowledge of the assessor.

A team of security assessors, who conduct vulnerability assessments, was given the task of using the presented models to assess High Assurance Cyber Military Systems (HACMS) Unmanned Aerial Vehicle (UAV), including those systems implemented on the test craft, to determine their security posture using the Flaw Hypothesis Methodology. The purpose of the assessment was to use the Model Methodology to identify potential weaknesses in the UAV's infrastructure that could allow an attacker to gain unauthorized access to organizational data or affect mission capabilities. The assessment was carried out from a hypothetical perspective and was limited to the security of the control systems. The team referred to this as a "field test" of the models, because there were a number of improvements and exchanges between the author and the assessment team where points of obscurity were identified. The author used the team's feedback to address those elements of the dissertation.

Very early in the assessment, the team determined that the models provided a strong template, or a guidebook, for determining the various characteristics of a system as an early step in the assessment process. With the act of recording possible flaws, vulnerabilities, mitigations, and potential patterns and connections, an

assessor could reference the models for future assessments to be sure that possibilities were not missed. This list could grow and become more thorough with time and wisdom.

The assessment models can also be tailored to meet the needs of any type of system, and any type of assessment and level of detail. The models are scalable, and can be modified to meet the requirements of various types of systems, focusing on a single aspect, or the system as whole.

The current state of the industry is such that no one assessor will view a system the same way as the next. Methodologies are ad hoc and developed by the assessor on an individual basis. Bringing different perspectives to the process is very beneficial, and the author clearly highlights the need for a standardized method of communication of findings. Assessors are merely human, and their access to certain information, knowledge and experience are purely individualistic; thus, the methodology brings an element of cohesion, and allows for the next assessor to pick up a previous assessment, be it their own or another person's, and have written record of which characteristics and possibilities were considered, and which were not.

While the presented methodology provides a strong backbone for an inexperienced and non-technical assessor to objectively determine the characteristics of a system, the assessors found themselves struggling with the urge to use numbers to calculate the models in order to arrive at an average risk factor. The author contends that the models cannot be mathematically calculated and are only mathematical in form. She goes on to advise that mathematics has little to no value in this scenario and uses sets of qualitative values, such as High, Moderate, and Low. However, the team feels that using mathematical values to truly calculate the models would result in a clear and more precise form of communication because mathematics is a universally understood and accepted language. For example, to say that there is a moderate risk of an adversary conducting an attack could be interpreted differently by a given individual. Someone with little experience may interpret "moderate" as simply that, but one who has extensive knowledge of the adversary might weigh the term high slightly heavier, and view it as somewhat more serious than the first. Thus, by assigning numerical values to the original qualitative value, the assessor can express, with great specificity, exactly how much value they

are assigning to the attribute and prevent the interpretation from becoming skewed.

For example, on the scale provided in the dissertation's cited NIST 800-30, 0 - 100 ranges from "Very Low" to "Very High". If the assessor is assigning a "High" value, they can use 80 – 95, for their calculation. An assessor can express more accurately express a value of how high their concern might be, as there is enough difference between 80 and 95 to change the final outcome of the calculation.

When using quantitative values only, future and returning assessors could look at a previously built model, and might scratch their head and say, "Now how did they come to that conclusion? Why do they think that capability is so high?" If the models use math, they can be recalculated for understanding and accuracy as adversaries, systems, and assessors develop. Their logic is explained and cannot be refuted without changing a defined value, where as there are instances where logically calculating a qualitative term could be subjective and called into question.

Also, using qualitative measurements creates an extraordinary amount of data, and the models can become cumbersome and difficult to manage. Using mathematics to calculate values would be easier to represent information, possibly even converting the final calculation back to a final, qualitative term as the author recommends. Research should be conducted to determine the best method of model management.

The author's argument against using numbers is primarily a bureaucratic one. Her experience has been that Authorizing Officials often have little to no experience conducting assessments, and often set their own values. This is a very good argument, and an issue worth researching, but the field testing assessors continue to feel that an assessor could adjust their scale and calculations accordingly, as long as the scale is defined within the assessment for the next assessor that comes along.

One of the hindrances that the author has brought to light is the subjectivity and "messiness" of the field of vulnerability assessments. In its current state, it seems that an assessor can assess a system with one outcome one day, and another one the next day. This ad hoc approach is very dangerous and wasteful, and her dissertation provides a way for one assessor to communicate with the next in an attempt to bring cohesion to the methodology, if not the standards, of the assessment process. Her models are a tool that can be used to create a snapshot of the assessor's current knowledge and logic at the moment of assessment. The field testing assessors feel

that this body of work adds value to the corpora of knowledge regarding assessments for National Security Assessments, and hope that it will be used by future generations to help keep our systems robust and secure.

Appendix D Inexperience Assessor's Comments

The inexperienced assessor authored the text of this appendix [Ine16].

The Assessment Methodology Models for Cyber System, by Jennifer Guild, provides a mathematical model to assess systems, as well as an assessment methodology that builds upon that model to assess those systems. The intentions of the paper were to enable an assessor with little to no experience, and no mathematical background, to thoroughly examine and assess a Cyber System. The very nature of these systems are dynamic, as are the personnel that defend them, therefore the models proposed are also intended to be dynamic. Guild's dissertation provides potential solutions for an assessor to lay the groundwork for future assessments, so that a new or revisiting assessor can continue the assessment by updating the models to match the system's changing state and the knowledge of the assessor.

The inexperienced assessor determined that the models provided a strong reference for a first assessment. The models give an indicator of potential flaws and other characteristics that might not otherwise be considered until the assessor gained significant experience, and provides the ability to reference other assessors' system characterization and considerations and draw on their knowledge and wisdom.

With the act of recording possible flaws, vulnerabilities, mitigations, potential patterns and connections, an assessor could reference the models for future assessments to be sure that possibilities were not missed. The list of models could be collected over time to significantly advance the assessor's ability and confidence while conducting future assessments.

The inexperienced assessor was able to accumulate a thorough list of characteristics and assign possible mitigations to the varying flaws and possible vulnerabilities to unmitigated flaws. The assessor began did a preliminary threat model, but struggled with the lack of insight regarding factors such as motivation and capabilities. Without this insight, it was difficult to move forward, so the experienced assessor was brought on to make a team of two.

The models were extremely useful for setting up the foundation of the assessment. Without it, the inexperienced assessor would have had great difficulty in sorting the

information about the system in a cohesive manner. The models allowed for an easy transfer of detailed knowledge and perception of the system to the experienced assessor.

Appendix E Experienced Assessor's Comments

The experienced assessor authored the text of this appendix [Exp16].

When the experienced assessor joined the first assessor to make a team of two, he noticed that the first assessor had a well-rounded list of flaws connected to mitigations and flaws mapped to vulnerabilities. The threat model was nearly complete, but where experience became helpful was during the stages assigning severity to vulnerabilities, and the probabilities found in the threat model. Without the experience and access to knowledge of adversaries, it was difficult to assign value to the characteristics in the threat model that are needed to evaluate the remaining models. This is a subjective process based on the assessors prior experiences however; no more subjective than other frameworks designed for risk assessment.

Once the remaining values of the vulnerability severity and probability were set, it was much easier to continue with the models and view how the characteristics are related and ultimately assigning risk. There are other alternatives to this methodology model for the risk assessment process including NIST 800-30 and the ISO 27000 series. The mathematically modeled formulas provided here offer a more efficient and easily utilized methodology for all assessors regardless of their experience level.