2021

A GENERAL FRAMEWORK FOR CHARACTERIZING AND EVALUATING ATTACKER MODELS FOR CPS SECURITY ASSESSMENT

Christopher S. Deloglos
Virginia Commonwealth University

Christopher Deloglos
Virginia Commonwealth University

A GENERAL FRAMEWORK FOR CHARACTERIZING AND EVALUATING ATTACKER MODELS FOR CPS SECURITY ASSESSMENT

By

CHRISTOPHER STEPHEN-JAMES DELOGLOS

A dissertation submitted in partial fulfillment of
the requirements for the degree of

DOCTOR OF PHILOSOPHY

VIRGINIA COMMONWEALTH UNIVERSITY
COLLEGE OF ENGINEERING
Department of Electrical and Computer Engineering

DECEMBER 2021

To the Faculty of Virginia Commonwealth University:

The members of the Committee appointed to examine the dissertation of CHRISTOPHER STEPHEN-JAMES DELOGLOS find it satisfactory and recommend that it be accepted.

Carl Elks, Ph.D., Chair

Ashraf Tantawy, Ph.D.

Patrick Martin, Ph.D.

Nathan Lau, Ph.D.

Barry Johnson, Ph.D.

ACKNOWLEDGMENT

Funding for this research was provided by the Virginia Commonwealth Cyber Initiative (CCI). Preliminary work for this dissertation was performed in collaboration with Qatar University under the NPRP 9-005-1-002 grant from the Qatar National Research Fund (a member of The Qatar Foundation).

I start by thanking Dr. Carl Elks for not simply instructing, but also mentoring me in this research. Thank you for always having an open door. To Dr. Tantawy, thank you for inspiring me into attacker models and setting me on this journey. To my fellow partners in crime, Rick, Smitha, and Abhi, thank you for all the support and encouragement you gave me through this process and being a sounding board when I needed one. Asa, thank you for your creative interpretations of random and obscure ideas. Brandon, thank you for the hundreds of ten-second conversations that compose the net sum of our academic interaction. You taught me efficient conversation.

I also thank Matt Gibson and Dr. Georgios Bakirtzis, as well as my other committee members Dr. Patrick Martin, Dr. Barry Johnson, and Dr. Nathan Lau for the formative conversations that helped sculpt the foundations of this research.

I owe a lifetime of gratitude to my amazing wife, Serena. Thank you for sacrificing your days, your nights, and so many moments in between so that I could pursue this degree. You are the most amazing person I've ever known and I shall never deserve you. I love you!

To Dad and Mom, I am the man I am because of the example you laid forth. Thank you for loving me, caring for me, instructing me in the way of wisdom, and teaching me diligence. Mom, thank you for being my teacher all the way through high-school. To any who doubt, a mother who takes responsibility for her child's education is a woman deserving of honor. To do so and succeed is worthy of glory. Thank you for all you sacrificed to help me succeed and for how you loved me through it all. Dad, thank you for being my mentor, my role model, and my friend. Thank you for the example you set of how to be a Godly man.

To my siblings, thank you for pursuing me, investing in me, and having my back as I've disappeared for the months it's taken to write this dissertation. You all are such a gift. Non, thank you for being the voice of reason. It grounded me when I needed it most. Steph, thank you for being the voice of crazy. You are right that life would be boring without you. Stephen, thank you for being the best brother ever, and for getting all the jokes that nobody else does.

To Dr. Ted Williams and my other close friends, thank you for your constant support, encouragement, and prayers through this long and exciting process.

Finally, thanks be to my Lord and Savior, Jesus Christ, who sustained me through this journey.

A GENERAL FRAMEWORK FOR CHARACTERIZING AND EVALUATING ATTACKER MODELS FOR CPS SECURITY ASSESSMENT

Abstract

by Christopher Stephen-James Deloglos, B.S.
Virginia Commonwealth University
December 2021

Associate Professor: Carl Elks

Characterizing the attacker's perspective is essential to assessing the security posture and resilience of cyber-physical systems. The attacker's perspective is most often achieved by cyber-security experts (e.g., red teams) who critically challenge and analyze the system from an adversarial stance.

Unfortunately, the knowledge and experience of cyber-security experts can be inconsistent leading to situations where there are gaps in the security assessment of a given system. Structured security review processes (such as TAM [1], Mission Aware [2], STPA-SEC [3], and STPA-SafeSec [4]) attempt to standardize the review processes to impart consistency across an organization or application domain. However, with most security review processes, the attackers' perspectives are ad hoc and often lack structure. Attacker modeling is a potential solution but there is a lack of uniformity in published literature and a lack of structured methods to integrate the attacker perspective into established security review processes.

This dissertation proposes a generalized framework for characterizing and evaluating attacker models for CPS security assessment. We developed this framework from a structured literature survey on attacker model characteristics which we used to create an ontology of attacker models from a context of security assessment. This generalized framework facilitates the characterization and functional representation of attacker models, leveraged in a novel scalable integration workflow. This workflow leverages an intermediate functional representation module to integrate attacker models into a security review process. In conclusion, we demonstrate the efficacy of our attacker modeling framework through a use case in which we integrate an attacker model into an established security review process.

Keywords: Attacker Modeling, Security Review Process, Attacker Model Ontology, Integration Framework, Cyber-Physical Systems

Contents

Page

APPENDIX

List of Tables

List of Figures

Summary of Acronyms

AM Attacker Model

AMC Attacker Model Characterization

AMF Attacker Modeling Framework

AMFR Attacker Model Functional Relationship

CPS Cyber-Physical Systems

DDoS Distributed Denial of Service

EPRI Electric Power Research Institute

HAZCADS Hazard and Consequence Analysis for Digital Systems

ICS Industrial Control System

IT Information Technology

OT Operations Technology

PLC Programmable Logic Controller

SCADA Supervisory Control and Data Acquisition

SRP Security Review Process

SRPC Security Review Process Characterization

STAMP Systems Theoretic Accident Modeling and Processes

STPA System-Theoretic Process Analysis for Security

STPA-Sec System-Theoretic Process Analysis for Security

STRIDE Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege

TAM Technical Assessment Methodology

Summary of Publications

C. Deloglos, C. Elks, and A. Tantawy, "An Attacker Modeling Framework for the Assessment of Cyber-Physical Systems Security", in *Lecture Notes in Computer Science*, A. Casimiro, F. Ortmeier, F. Bitsch, *et al.*, Eds., vol. 12234 LNCS, Cham: Springer, 2020, pp. 150–163, ISBN: 9783030545482. DOI: 10.1007/978-3-030-54549-9{_}10

C. Deloglos, A. Tantawy, and C. Elks, "A Framework for Describing Attacker Models", in *12th Nuclear Plant Instrumentation, Control and Human-Machine Interface Technologies*, vol. 12, NPIC&HMIT 2021, 2021, pp. 1349–1360. DOI: 10.13182/T124-34535

C. R. Elks, A. Tantawy, R. Hite, *et al.*, "Realizing Verifiable I&C and Embedded Digital Devices for Nuclear Power Design, Verification and Demonstration of the SymPLe Architecture", U.S. Department of Energy, Tech. Rep., 2019. DOI: USDepartmentofEnergyDE-NE0008445

M. Gibson, C. Elks, A. Tantawy, *et al.*, "Achieving Verifiable and High Integrity Instrumentation and Control Systems through Complexity Awareness and Constrained Design", Idaho Operations Office, Idaho Falls, ID (United States), Tech. Rep., Jul. 2019. DOI: 10.2172/1547345. [Online]. Available: http://www.osti.gov/servlets/purl/1547345/

G. Bakirtzis, G. L. Ward, C. J. Deloglos, *et al.*, "Fundamental Challenges of Cyber-Physical Systems Security Modeling", Apr. 2020. DOI: 10.1109/DSN-S50200.2020.00021. [Online]. Available: http://arxiv.org/abs/2005.00043%20http://dx.doi.org/10.1109/DSN-S50200.2020.00021

C. Elks, C. Deloglos, A. Jayakumar, *et al.*, "Specification of a Bounded Exhaustive Testing Study for a Software-based Embedded Digital Device", Tech. Rep., 2018. [Online]. Available: http://www.inl.gov

C. R. Elks, A. Tantawy, M. Gibson, *et al.*, "Lessons and Experiences Learned Applying Model Based Engineering to Safety Critical FPGA Designs", 11th International Workshop on the Application of FPGAs in NPPs, Dallas, Tech. Rep., Aug. 2018. [Online]. Available: https://www.researchgate.net/publication/332370279

D. C. Elks, C. Deloglos, A. Jayakumar, *et al.*, "Realization of a Automated T-Way Combinatorial Testing Approach for a Software Based Embedded Digital Device", Idaho National Laboratory, Idaho Falls, ID (United States), Tech. Rep., Jun. 2019. DOI: 10.2172/1606019

S. Gautham, A. Varma Jayakumar, R. Hite, *et al.*, "MODEL-BASED DESIGN ASSURANCE AND VERIFICATION IN THE CONTEXT OF IEC-61508 SIL-4 STANDARD", in *Safety Critical Software Development, Qualification, and V&V*, 2021. DOI: 10.13182/T124-34548

P. Beling, B. Horowitz, C. Fleming, *et al.*, "Security Engineering – Decision Support Tool", Systems Engineering Research Center, Tech. Rep., Jun. 2020. DOI: ERC-2020-TR-008

Soli Deo Gloria

Chapter One

Introduction

1.1 Background

Cyber-Physical Systems (CPSs) is broadly recognized as the field of technology that integrates cyber operations with physical systems through the use of sensory equipment and controls technology.

Where the pre-CPS era was primarily defined by isolated embedded systems interacting with the physical world, the integration of cooperating embedded systems composed of control, networks, sensory, autonomy, and human interaction components gave rise to the realm of CPSs. The power of CPSs has been realized and embraced as they are deployed across the technological spectrum. From military and national defense interests to utility infrastructures such as energy, manufacturing, transportation, and communication. Even to manufacturing environments and consumer products, CPSs have become the rule where they used to be the exception.

Cyber-physical systems have evolved as the security challenges of networked and connected environments become more commonplace in the CPS domain. While cyber-physical systems receive a public view of being commonplace, the transition into an integrated world has been riddled with challenges for many industries where Information Technology (IT) and Operational Technology (OT) have historically been separate. The origins of cyber-physical systems precluding their rise to prevalence are in operational technology where blanket security schemes mitigated security requirements, now commonly referred to as perimeter-based security techniques [15]. Techniques such as air-gapping or otherwise separating operational technology networks from the connected and integrated information technology networks [16] allowed the benefits of cyber-physical systems to be realized without inheriting the security risks of information technology.

As technology continued to advance, the requirements for cyber-physical systems resulted in modifications and advancements to the technology to meet the needs of more complicated processes with more automation and less dependence on physical orientation and proximity. The benefits of integrating IT and OT pushed cyber-physical systems further and further into the networked realm, which introduced vulnerabilities and threats to the OT world that hitherto were irrelevant.

Cyber technologies evolved in an environment of hackers and security awareness, quickly adapting to the changing threat landscape. Cyber-physical technologies, however, were developed on the foundations of long-term reliability and safety and tended to adapt much slower [17]. Power grids, communication systems, and other critical public infrastructures, for example, are designed for long-term safety and reliability, and the rapid ecosystem of security management with regular software patches and hardware vulnerability remediation does not integrate well into this slow-moving environment [16].

Compounding the urgency of the cyber-physical systems security issue is that violations of security in a CPS can have physical consequences - resulting in dangerous or unsafe behavior. Where cyber-attacks can only cause physical damage indirectly, cyber-physical attacks have the potential to cause injury or loss of life by direct physical means. As a field, cyber-physical security inherits the challenges and traits of cyber-security and must expand those to meet the physical safety and security requirements. The interdependence between physical safety and software-based safety and security is at the center of the cyber-physical universe and drove CPS security research in the face of emerging threats.

1.2 Motivation

The dangers of cyber-physical security threats are significant and have been demonstrated across many industries. In 2000, the sewage system of the Maroochy Shire Council in Queensland, Australia, became the target of one of the first publicly reported SCADA (Supervisory Control and Data Acquisition) system attacks [18]. In the attack, a contractor used a personal computer to mimic a pumping station using radios and stolen SCADA software. The attacker reportedly released more than 750,000 gallons of sewage water into public spaces, resulting in \$176,000 of damages to the city council and \$500,000 of expenses to the contractor.

In 2005, the Athens affair [19] showed again how hacking computing resources could lead to unexpected and significant losses in the real world. While the Athens affair was, strictly speaking, a cyber-attack, it laid the foundations for understanding the consequences of integrating hardware and software systems. In the Athens Affair, Greece's largest cellular service provider (Vodafone Greece) was infiltrated when attackers reprogrammed their network switches to redirect and record phone calls. The provider, who was unauthorized to perform wire-tapping at the time, upgraded their infrastructure to handle their growing technological needs and installed fuller-featured equipment with unused wire-tapping capabilities. Hackers that remain unidentified to this day were able to exploit this unused feature to redirect phone calls from over 100 high-profile victims, including the prime minister of Greece and his wife, the ministers of justice, foreign affairs, and national defense, and a myriad of other politicians, embassy workers, and activists. The consequences of the attack were extensive, including a $76 million fine to the service provider and a suicide alleged to be a result of the wire-tapping. However, no audio recordings were ever discovered.

Examples of the dangers of cyber-attacks in cyber-physical industries did not take long to manifest and include [20]–[26]. In 2010, the Stuxnet worm [27] became a center of attention in CPS security. It demonstrated the susceptibility of a safety-critical CPS-based Industrial Control System (ICS) to cyber-attacks of a new category of complexity. The Stuxnet worm infected an air-gapped network in an Iranian nuclear fuel refinement plant, causing damage to refinement centrifuges and crippling the plant operation for several months before it was discovered. This worm succeeded by exploiting both known and zero-day vulnerabilities and taking advantage of operator process deficiencies to enter the ICS, then propagating laterally through the ICS to its target system. Stuxnet revealed inadequacies in blanket security approaches such as air-gapping or assuming complete operator process control, prompting a need for refined security processes during industrial control systems' design, development, and operation. In particular, Stuxnet revealed a need for security assessment methods that account for a broad set of threat actors, even as they continue to evolve and adapt.

As the need for process automation continues to grow, so does the scope and magnitude of CPS attacks, and so do the predictions of CPS attack severity and frequency. While financial projections are skeptical at best, Gartner in [28] predicts that the damages from fatality-related CPS incidents will reach over $50 billion by 2023. On December 23rd, 2015, hackers performed a coordinated assault on three Ukrainian power distribution centers, taking 30 substations offline and leaving over 230,000 individuals without power, including the power distribution centers whose backup generators were taken offline as a step in the attack. This attack became unique when the attackers reprogrammed the firmware on several of the serial-to-ethernet converters at the substations to inhibit engineers from remotely re-booting the substations. As an added measure, the attackers launched a DDoS (Distributed Denial of Service) attack that took out the power company's call centers to inhibit customer communication. The attack reportedly was in effect for six hours until engineers could manually bring all 30 substations back online.

In summary, the threats against cyber-physical systems are real, the consequences are severe, and cyber-physical systems are only getting more advanced, integrated, and complicated. From ransomware operators seeking to make an income to hacktivists, terrorists, and nation-states playing for cyber dominance, the threat actors against cyber-physical systems are dangerous and constantly adapting to subvert security measures.

1.3 A Problem and a Potential Solution

Despite a boom of security research targeting the CPS field, the evidence suggests a significant disparity between state of the art in literature and state of the art in practice. In a 2021 publication [16], Jamil performed a study of security practices being employed in various CPS-based industries and found that the methods and techniques applied to security assurance fell far short of the security research available for those technologies. At the core of this problem, Jamil identified threat modeling practices to be notably lacking. Jamil states, "most of the participants do not use quality assurance techniques for the threat models that they produce and depend on the experience and skills of the expert who performs the threat model " [16]. To put this statement in context, most participants in the study cited STRIDE [29] as their guiding threat model, which was born in the IT world of cyber security and has no concept of the physical environment, physical threats, or physical consequences. This study demonstrates a significant disparity between the tools and resources available for cyber-physical security experts and the means employed in the field. This disparity becomes even more apparent when considering the common practice of assigning IT professionals to OT security jobs without the training and experience to understand the complexities that physical components bring to the security challenge [16].

In both cyber and cyber-physical security, the challenges of comprehensive security assurance have facilitated the practice of red/blue teaming. This practice shifts the focus of the assessment by splitting the security review into two teams pursuing the same security goal by different means. The purpose of the blue team is to demonstrate system security through security assurance practices, often involving semi-formal security review processes. The goal of the red team is to ensure system security by proving the absence of system vulnerabilities by attacking the system and identifying existing vulnerabilities so that the blue team can remove them. This makes the operations of red teams critically dependent on their ability to identify and emulate threats as they try to adopt the attacker's perspective [30].

One of the critical factors that can limit the abilities of the red team is not having appropriate models to reason about the attacker consistently. Classical security assessment relies on red-team security experts to analyze a system and identify vulnerabilities and is limited by the extent of the experts' knowledge and experience. A lack of methods to guide and inform the security assessment results in ad hoc threat modeling practices with inconsistent results. Experts such as Adam Shostack strongly advocate against practices that require security experts to "think like an attacker" in favor of more structured and formal methods [31]. Progressive security practices challenge the expectation that a security professional will be able to effectively or efficiently postulate the behavioral characteristics of a largely unknown threat actor [32]–[34]. In the 2020 publication [34], Moeckel interviewed twelve senior-level security management officials and identified this type of 'thinking like the attacker' as a common practice.

Jamil asserts in [16] that in practice, the lack of quality assurance for threat models leads to a lack of confidence in these ad hoc threat models generated by security experts. He states, "The managers sometimes request threat models for their CPSs from more than one expert." While a diversity of expert opinion is a potential solution, it does not solve the bigger problem: the lack of confidence in expert opinion undermines security assurance. The threat models used in security review processes need to be bolstered with more formal threat modeling practices.

Similarly, security review processes used by blue teams share a common underlying principle: they require the security expert to inform the review process of the nature of an attacker. This may be observed explicitly, such as in a HAZCAD [35] analysis process where the security expert defines what the goals of the attacker would be at a high level, or implicitly such as in a STRAT analysis where the security expert associates risk levels with each vulnerability, implicitly quantifying metrics such as the likelihood of attack via a vulnerability or probability of attack success for an attacker exploiting that vulnerability.

In summary, any expert-driven threat analysis is limited by the expert and may fail where the expert's knowledge or experience falls short. Several promising solutions exist to enhance different aspects of security assessment. Security review processes add consistency in the methods, principles, and assumptions used to analyze and review a CPS [1], [4], [15], [36], [37]. Vulnerability databases and search engines provide consistency in assessing known, context-agnostic vulnerabilities for CPS components [38]–[42]. By contrast, despite the existence of attacker modeling literature, there is a significant lack of research in integrating attacker models into security review processes to aid security experts in the assessment of the attacker and in understanding how the attacker assessment impacts the security review process. Despite increasing formalism behind the security review process and vulnerability assessment techniques, the critical component of analyzing attacker behavior is often not addressed, undermining security assurance and limiting the quality of the attacker assessment to the knowledge and experience of red-team and blue-team security experts.

Attacker behavioral modeling has received much attention as its field of research apart from integration with security review processes. Attacker models (AMs) present solutions for various security challenges, including those prompted by security review processes of reasoning about attacker behavior and identifying how that behavior influences the security assessment. Despite the availability and utility of attacker models, in application, they are rarely integrated into red-team and blue-team security review processes due to the following challenges.

- Attacker models are challenging to understand due to inconsistency in definition and structure.
- The assumptions made by attacker models are challenging to identify when not explicitly documented.

- Misalignment of critical assumptions between attacker models and security review processes makes integration infeasible.
- Functionally integrating attacker models into security review processes requires significant manual effort.

For the benefits of attacker modeling research to be fully realized and transitioned into practice, methods need to be devised to understand and integrate attacker models into security review processes performed in real-world applications. We propose a generalized attacker modeling framework that is structured to comprehend all attacker models and a workflow to guide attacker model integration into a security review process.

1.4 Research Objectives and Value Propositions

This research aims to aid in the realization of the benefits of attacker models by developing a generalized framework for understanding attacker models and introducing attacker modeling methods and practices into security review processes.

To achieve this, we performed a broad literature review of attacker modeling research and security assessment research to develop a modular Attacker Modeling Framework (AMF) that aids in defining, understanding, and applying attacker models. This framework is composed of two primary modules. The first module is called the Attacker Model Characterization (AMC). It is a generalized semi-formal method for describing attacker models and characterizing the attributes that make them unique for application in security review processes. The second module is called the Attacker Model Functional Representation (AMFR) and is a generalized semi-formal method for capturing the functional implementation of an attacker model. This research was developed using a systems approach to fully understand the domain of attacker modeling to create a more general attacker modeling framework that can be used for quantifying attacker model assessment for security review processes.

While we demonstrate the utility and value of each of the AMF modules on their own, the novelty of the attacker modeling framework truly lies in how we leverage these modules together to facilitate selecting and integrating attacker models into security review processes. In developing our attacker modeling framework, we recognized the gap between the attacker modeling community and the security assessment community and the need to bridge this gap by making a framework that is accessible to both. From the perspective of the attacker modeling community, our framework recognizes the need for attacker modeling experts to understand the characteristics of their attacker models that are uniquely valuable to security assessment research and know how their attacker models functionally integrate with security review processes. To do this, our AMF aids the attacker model researcher in characterizing and describing their attacker model in a manner that is accessible and valuable to the security assessment community. Our AMF breaks down the attacker model into an ontology composed of the subset of attacker model characteristics critical to integration with security review processes. Our framework also identifies the relationship of information exchanged between attacker models and security review processes and how the attributes of the attacker model influence their compatibility with and their value to those security review processes.

From the perspective of the security assessment community, our framework recognizes the challenges involved with achieving a quantitative assessment of attacker models and identifying the characteristics and attributes that influence the compatibility and value provided by attacker models. To do this, we develop a systematic and semi-formal process that can be used by security experts to document, evaluate, and integrate attacker models into security review processes. This framework aids the expert in predicting the utility and effort for integration of different attacker models. We also recognize that the security community is heavily influenced by the cost and effort associated with developing security practices and that the effort required for using our attacker modeling framework must be reasonable and scalable for it to be a valuable solution. As such, we developed our attacker modeling framework with a concept of hierarchical abstraction, where we leverage the information available at higher levels of attacker model abstraction first to minimize the manual effort required throughout the integration process. Then we progress to lower levels of abstraction as the results and scope of attacker model assessment narrow toward the perspective of the security review process.

Another valuable aspect of this AMF is that we develop and demonstrate it in the context of a systems-based workflow where we use tabular documentation schemes to capture data in a manner conducive to data utilization. This was influenced by the documentation schemes commonly employed by semi-formal security review processes familiar with the requirements of large-scale analyses and optimizing the organization and accessibility of information.

In conclusion to this work, we demonstrate the application of the attacker modeling framework in an integration case study using an established security review process developed by the Electric Power Research Institute (EPRI) called the Technical Assessment Methodology (TAM) [1]. In this case study, we identify a component of the security assessment process that the inclusion of an attacker model may aid. We then compile a diverse set of attacker models with broadly different characteristics and purposes, use the attacker modeling framework to identify valuable attacker models, and integrate a single selected attacker model into the TAM.

1.5 Contributions

The primary contribution of this work is that it bridges the gap between attacker modeling research and actual practices of CPS attacker assessment. Different attacker models boast a wide range of functionality and capabilities and can be of utility to security review processes in many different ways. The novel work of this dissertation in making a bridge between attacker models and security review processes realizes the value of existing attacker models, making them accessible for real-world application. Tangibly, the contribution of this dissertation includes:

1. A structured review on CPS attacker models and the core principles that contribute to integration with security review processes
2. The development of a deep and broad body of knowledge on attacker models that is generalized and can describe a wide diversity of attacker models
3. A deconstruction of attacker models into an ontology that captures the characteristics of attacker models that are critical to integration with security review processes
4. The design of a modular attacker modeling framework for objectively characterizing cyber-physical systems attacker models
5. The creation of a workflow for integrating the attacker model into any compatible security review process
6. A case study demonstrating the utility of the attacker modeling frameworks

1.6 Research Road Map

In Chapter 1, we introduce the critical issues surrounding CPS security. We identify the need for including well-formed attacker models in security review processes and identify the challenges that have inhibited their utilization thus far. This chapter proposes our attacker modeling framework and describes its value to attacker modeling research and integrating attacker models into security review processes.

Chapter 2 provides a literature review of attacker modeling, CPS security review processes, and the core composing topics. It identifies the research performed thus far as well as gaps in the existing literature.

Chapter 3 provides an overview of our AMF, describes the process used to develop it, and lays the foundations for how the various components of our AMF work together to achieve its collective goals.

Chapter 4 dives into the first of the two components of the attacker modeling framework, the attacker model characterization (AMC). It describes the composition of the AMC, the various attributes defined in it, how to use it, and finally demonstrates and documents the application of the AMC to several attacker models.

Chapter 5 dives into the second of the two components of the attacker modeling framework, the attacker model functional representation (AMFR). In this chapter, we describe the various components of the AMFR, present and discuss the semi-formal notation used to represent the

AMFR, and finally, demonstrate the creation of AMFRs for several different attacker models. In the early stages of this dissertation research, we developed and published our own attacker model to address a specific gap in attacker modeling research. At the end of this chapter, we describe that attacker model in detail and discuss in parallel how the various characteristics of the attacker model influence the creation of the AMFR for that attacker model.

Chapter 6 describes our semi-formal workflow for evaluating, selecting, and integrating attacker models into security review processes. We discuss our developed scoring methods, structured evaluation methods, and documentation schemes and identify how scalability and abstraction contribute to a valuable integration process with reasonable effort.

In Chapter 7, we demonstrate the utility of our attacker modeling framework in a use case where we identify a potential use for attacker models in EPRI's Technical Assessment Methodology [1], evaluate several attacker models, and integrate one into the TAM.

Finally, in Chapter 8, we provide our conclusions. We summarize the research performed, our discovered results, and we identify future research directions.

Chapter Two

Literature Survey

Attacker modeling is a composite topic, integrating research from several aspects of cyber and cyber-physical systems modeling and security. This dissertation is only possible due to work completed in establishing a background in topics such as formal and semi-formal methods of security assurance, attacker modeling, attack modeling, CPS modeling, vulnerability analysis, and attacker-CPS interaction modeling. An overview of related contributing literature is discussed in this chapter, as well as our structured literature survey of attacker models.

2.1 The State of Cyber-Physical Systems Security

Information technology (IT) and operational technology (OT) are well-established in academic research and industrial application. In composition, cyber-security application to physical process automation represents a relatively young field and requires more than the sum of the parts to understand. Information technology can be defined as "the technology involving the development, maintenance, and use of computer systems, software, and networks for the processing and distribution of data " [43]. IT infrastructure's networked and integrated environment results in a complicated threat space, which has been the driving force behind the development of cyber-security practices since the invention of the internet. A review of the evolution of threats and cyber-security practices in IT systems is deserving of several dissertations and books on its own. Well-received in the cyber-security field, a representation of the evolution of cyber attacks and threats can be found in Bruce Middleton's 2017 publication, "A History of Cyber Security Attacks: 1980 to Present" [44].

The Merriam-Webster dictionary describes operational technology as "Hardware and software that detects or causes a change, through the direct monitoring and/or control of industrial equipment, assets, processes and events." [45]. The origins of OT can be traced as far back as the industrial revolution in the 1800s, but truly exploded as a field in the 1960s [46]. However, in the 1990s, the fundamentals of OT began to change with the evolution of digital technology, and the 21st-century ushered in a new era of integration between IT and OT systems with industrial control systems (ICS) integrating programmable logic controllers (PLC) and supervisory control and data acquisition (SCADA) systems. The demands of connectivity and integration in these environments necessitated integrating IT and OT, introducing the threat actors of the IT world to the OT environment.

2.1.1 Perimeter-Based Security in IT and OT

For many years, the gold standard for OT security assurance was perimeter-based security practices [15]. Perimeter-based security approaches attempt to establish secure boundaries around systems, assuring a safe operating environment within those prescribed boundaries. Early perimeter-based techniques were realized by physically separating and air-gapping OT and IT networks, eliminating attack paths into the OT system. While perimeter-based approaches are generally seen as necessary, they are not sufficient. Notably, several attacks such as the iconic Stuxnet attack [27] demonstrated that even physical partitioning does not inhibit all attack paths, demonstrating the inadequacy of such security measures and the hazards of assuming a secure operating environment. Moreover, technological requirements for process automation have pushed toward connectivity and the integration of networked controls, which has made it harder and harder to defend the boundaries between OT and IT. As the complexity of OT integration increases, the sophistication and diversity of attacks have increased in kind. To quote the Cyolo Team, "The Modern Network Perimeter is Full of Holes" [47]. This has prompted a need for more robust security assurance practices that delve past the boundaries into analyzing how the critical system components and processes relate to mission objectives and how system vulnerabilities lead to mission failure [15].

2.1.2 Influence of CPS Safety Practices in CPS Security

The most apparent difference between the influences of cyber-security practices and cyber-physical security practices is (1) the level of formalism often employed and (2) the notion of consequences. Attack trees have early origins in the history of cyber security, first being published by Salter in the 1990's [48]. Attack trees have been a standard go-to for cyber-security experts and define the security state of a system in terms of the various attacks that may be performed against it. A thorough review of the evolution and different methodologies of attack trees can be read in [49]. At the simplest, attack trees take on the form of trees, where the topmost node is the goal of an attacker and the branching nodes below are steps in the attack process, as seen in Figure 2.1. Attack trees have been expanded to include many useful features, notably formal classifications for attack steps, decision criteria, and likelihood and probability associations with attack steps.

State of the art in CPS security assurance is the product of cyber-security and safety assurance practices. The origins of CPS safety in preventing loss and assessing risk have evolved into a

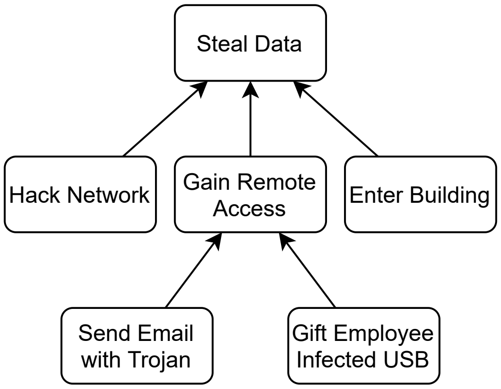


Figure 2.1 A sample attack tree with the goal of stealing data.

practice structured by semi-formal and formal semantics, such as in safety standards such as IEC 61508 [50], ISO 26262 [51], and DO-178C [52]. For example, verification and validation in a CPS observe strict quantifiable metrics to ensure the level of safety required of a particular device for a particular application. On the other hand, cyber-security processes have historically observed far less structure and uniformity in semantics, metrics, and evaluation standards. Safety assurance practices established the foundations of formal methods for relating the system mission to component behavior.

STPA (Systems Theoretic Process Analysis) [36], a hazard analysis model founded in STAMP (Systems Theoretic Accident Modeling and Processes) [53] is one such practice that evolved from this space designed for safety assurance of cyber-physical systems. Many CPS security methods evolved as transpositions of the principles of STPA to the security domain. Examples include security analysis methods such as STPA-SEC (System-Theoretic Process Analysis for Security) [3], STPA-SafeSec [4], HAZCADS (Hazard and Consequence Analysis for Digital Systems) [35], and Mission Aware [15].

2.1.3 Research Gap: Attacker Modeling in Security Assurance

Formal and semi-formal security assessment methods continuously evolve as the nature of threats and state-of-the-art technology progresses. Studies of state of the art in practice [16], [34] identify that a gap exists between state-of-the-art attacker modeling research and the threat modeling methods being employed in practice by security review processes. We posit that functionally integrating an attacker model into an established security process is non-trivial due to the challenge of understanding the attacker model and the security review process well enough to facilitate the integration.

A common conclusion of attacker behavioral models is to deduce security metrics by describing the likelihood or probability of an attacker performing one or more actions such as in [5], [54]–[59].

The vast majority of these works claim that the value of the attacker model will be realized as the model and results are integrated into a security review process but never carry the research out to the step of integration. Of all the attacker models reviewed, only LeMay's research group in [54] went on to build the attacker model into a functional tool [60] and demonstrate its application in security analyses [61]. We posit that a primary reason that attacker modeling research terminates prior to the step of integration with security review processes is because of the significant manual effort required by the attacker modeling expert to understand an SRP well enough to integrate an attacker model.

2.2 Attacker Modeling

In Chapter 1, we identified the following challenges that are inhibiting the adoption of attacker models in security review processes:

- Attacker models are challenging to understand due to inconsistency in definition and structure.
- The assumptions made by attacker models are challenging to identify when not explicitly documented.
- Misalignment of critical assumptions between attacker models and security review processes makes integration infeasible.
- Functionally integrating attacker models into security review processes requires significant manual effort.

The first reason for this gap is that it is difficult for an expert of a different domain to comprehend the diverse range of attacker modeling literature. Works that seek to develop underlying foundations for attacker modeling practices are promising but have yet to gain traction in the field [57]. Attacker models are limited by a lack of unity in the structural description of attacker-system interactions and in the application of attacker decision theory. Fundamentally, attacker models vary in form, from structured ways of thinking [56] to rigorously formal system-theoretic behavioral models [62], [63]. This diversity makes it challenging for an expert in applied security assessment to distinguish valuable and relevant attacker modeling literature. In order to establish a fundamental basis for discussing attacker models, it is necessary first to understand and be able to represent their core and common attributes at a level of detail conducive to general conversation. This generalized characterization of attacker models does not exist in the attacker modeling literature to date. Our attacker modeling framework contributes to the state-of-the-art by implementing an attacker model characterization workflow which serves as a standard basis for describing attacker models in the context of integration with a security review process.

It is commonly accepted in attacker modeling that modeling complex human behavior requires the assertion of bounding assumptions that reduce the complexity of the problem to a manageable size. This can readily be observed in attacker profiling techniques where different threat models bound the profile properties of the threat actor to include differing characteristics [64]. As a rule, attacker models seek to abstract away irrelevant detail, allowing a higher-level analysis of attacker behavior. While it is more common to document assumptions made in security review processes [1], [15], identification of assumptions is strongly lacking as a practice in attacker modeling. In this dissertation's structured literature survey, no single attacker model was identified that explicitly identified the simplifying assumptions made about the attacker and attack procedures. Identifying the assumptions made in attacker models is often critical when considering integrating with the theoretic foundations of security review processes. While some assumptions may be benign, others are critical, and it is difficult to distinguish which attributes of an attacker model may cause incompatibility with a security review process without performing a deep dive into the literature. In our attacker modeling framework, we create a workflow for objectively identifying and comparing the assumptions made by attacker models to evaluate attacker model compatibility with security review processes.

The effort required for integrating an attacker model into a security review process is significant [16], [34]. It can be compounded when the attacker model is designed from a perspective of progressive theoretical research rather than for the goal of real-world application. We posit that integrating an attacker model fundamentally requires a complete understanding of the flow of information in and out of the attacker model and the information available from and required by the security review process. Currently, there is no generalized guiding literature for integrating attacker models into security review processes. In order to fill this research gap, our attacker modeling framework implements a module that creates a functional representation of the attacker model, capturing the flow of data and describing at a high level how the input data of the attacker model is leveraged to produce modeling results.

2.2.1 Literature Survey Overview

One of the contributions of this dissertation is a structured literature survey of attacker models. Chapter 3 presents the literature survey workflow, which is used to guide the development of the attacker model characterization attributes. In Chapter 4, we discuss these attributes in-depth and identify and discuss the associated attacker modeling literature. Therefore, in expectation of the literature survey, the remainder of this chapter is dedicated to discussing literature that more generically sets the stage for the dive into the individual attacker models.

In reviewing the literature on attacker modeling, four categories of questions naturally arise. The first and most apparent is, "What does this attacker model do?" Attacker models vary in their architecture and fundamental perspectives on how the attacker is associated with the system through attacks. The second and third categories of questions involve what information the model needs to execute and what information the model provides during/upon completion. The final category involves assumptions about the attack process relevant to the context of a security review process.

2.2.2 Describing the Attacker

In attacker modeling, a common approach is to create a correlation model where the designer selects a series of attacker properties such as skill level, resources, intent, and motivation and attempts to develop cumulative correlation functions that effectively predict attacker behavior when applied to real-world attackers [65]. Adepu et al. apply a general description of an attacker as a super-set of attacker intents and the associated CPS-Domain [57]. Monteuuis, by contrast, defines an attacker threatening an automated vehicle as a set of properties that includes membership, motivation, scope, methods, and goals [56]. These two attacker descriptions imply different domains of the attacker profile. Adepu's generalization of the nature of the attacker as the product of the intents of the attacker and the CPS interaction implies that the attacker's behavior will always be intent and goal-driven and adapts to the CPS the attacker confronts. Monteuuis's attacker description is more context-specific and implies that only five unique attacker properties influence the attacker's behavior, one of those being goals. These attacker descriptions may be appropriate in the given context. However, they capture different aspects of the attacker's characteristics, and their underlying frameworks are incompatible, Adepu's being developed using system-theoretic syntax while Monteuuis' is developed without a semantic or formal foundation. While attacker profiling is a common, if not casual, practice in attacker modeling, there tends to be little resolution on the efficacy of various profiles. In order to establish a common basis, Rocchetto et al. performed a literature search and created a six-profile model able to effectively describe attacker profiles from the majority of cited literature [64].

2.2.3 CPS Topology Description

In a review paper of CPS security methodologies, Dibaji et al. assert that the quality of description of a CPS network topology influences the viability of various defense mechanisms and must be a point of consideration for cyber and cyber-physical applications [66]. Various methods have been proposed to describe CPS topologies. Cheh et al. developed a security-focused topology capable of maintaining information about the security state of the CPS [67]. Choley et al. apply a basic SysML topology to aid in an attack-surface driven security review process [68].

Beling et al. [69] describes a systems engineering ontological metamodel for CPSs uniquely tailored for safety, security, and resilience applications. This model can capture many aspects of a

CPS, including requirement, physical, functional, interface, safety, security, and resilience elements. The security-aware design methodology of the metamodel makes it a viable platform for integrating the security analyst perspective and review process into the CPS design and development process.

While many CPS topology descriptions are available, few readily lend to system security perspectives. The methods of capturing CPS topology, architecture, and interactions have historically lacked formality. Recent efforts in model-based security have expanded state of the art by formally describing the relationships between the models used to describe cyber-physical system operations [70].

2.2.4 Research Gap: Attacker Model Validation

Despite the plethora of attacker models developed, the question of which is correct remains inconclusive primarily due to the inability to validate attacker profiles against real-world data [54]. Data for validating attacker behavior is challenging to capture in the wild. The most common technique is to use attack simulations where actors are given tasks and emulate the behavior of threats. These studies are expensive to perform and difficult to execute without biased results.

This dissertation recognizes that validating a proposed attacker model may be a necessary step toward security assurance. In light of the challenges of acquiring real-world validation data, the goal of this dissertation in integrating AMs into SRPs may contribute to AM validation in two ways. First, on its own, the value of an AM is critically dependent on its validity. When considered in the context of an SRP, the driving factor is no longer attacker model validity, but rather if the AM improves the quality of the security review, which requires the development of quality metrics and depends on the form and function of the AM being considered and the process outlined by the SRP. For specific attacker models, integration into an SRP may establish the value of the attacker model where direct validation would have been unfeasible. A goal of the AMF developed in this dissertation and the formalization it provides for AM integration into SRPs is to significantly reduce the burden of labor required for this integration, making this form of AM value proving more accessible.

Second, a critical aspect of AMs and SRPs, highlighted and explored throughout this dissertation, is that they are both critically dependent on well-formed assumptions that reduce the complexity of the attacker and the system to a level that can realistically be modeled. Justifying attacker model assumptions is complicated to prove outside of an application context. While several AMs lay preliminary groundwork for application scenarios, the broader scope of SRPs provides a much more thorough description of the systems, processes, and other application details that may aid in assumption justification.

Therefore, while the challenge of validating a particular attacker model against a real-world scenario falls outside the scope of this dissertation, the framework proposed in this dissertation is intended to serve as a vehicle to facilitate the design, development, and SRP integration of a wide variety of attacker models, which would then serve as a foundation in future research of attacker-model validation.

2.2.5 Vulnerability Database Integration

In application to a whole system, researching, compiling, and organizing all vulnerability information related to a CPS is a monumental task. Databases such as the Common Attack Pattern Enumeration and Classification (CAPEC) [39], the Common Weakness Enumeration (CWE) [40], the Common Vulnerabilities and Exposures (CVE) [41], and the Common Platform Enumeration (CPE) [42] have been applied to attack modeling to aid in vulnerability research [59], [71]. Attacker modeling literature has demonstrated the utility of these databases in composing vulnerability information into tailored vulnerability data sets [5]. Furthermore, tools have been proposed that combine these databases into hybrid search engines such as the CYBOK tool [38], which manages architectural system models and can perform automated data querying on various fields of information in component descriptions.

2.2.6 Range of Attacker Model Survey

The form and function of attacker models chosen for the literature survey range from high-level abstractions of attacker behavior to algebraically intense probabilistic models. At the high level, Monteuuis et al. [56] applies methodical reasoning to model the specific use case of attackers against connected and automated vehicles, resulting in categorical predictions of behavior from four types of attackers. Monteuuis' model goes in-depth into the nuance of security for automated vehicles, giving particular attention to how the relationships between various components influence the progression of various attacks. Mo et al. [72] similarly developed a system-theoretic security approach to modeling the attack process derived in application to a power grid in order to determine the attackers' actions and goals. While Monteuuis' and Mo's models are use case-specific, more generic high-level attacker models have been developed, as demonstrated by Vigo et al. [73] who presents an attacker model focusing on the effect of an attacker's physical interference on network nodes.

Basin et al. [74] presents an attacker model for formal reasoning on security protocols, observing the effects of attacks on physical properties of the CPS. Basin applies formal methods to model how the spatial orientation of nodes in a communication system influences the attacker's ability to succeed. In a similar application, McEvoy et al. [62] proposes an alternative to the commonly applied Dolev-Yao model in order to model communications systems. While still being founded at a high level in formal reasoning, these attacker models establish a more robust foundation through symbolic methods of system property derivation.

Where formal reasoning provides high-level insights, more rigorous attacker models seek to provide specific and conclusive analysis. Teixeira et al. [63] symbolically models attackers performing various attack methods against a CPS based on an attack space composed of system knowledge, disruption resources, and disclosure resources. Orojlloo et al. [55] applies a Markovian analysis method to model the dynamic behavior of countermeasure-equipped systems under attack for particular attackers.

Ekelhart et al. [59] applies an attack simulation engine to abstracted attack vectors to evaluate the number and prominence of various attack paths through a CPS. While Ekelhart's model does not go in-depth into the behavior of the CPS, it provides a computationally robust framework that can provide quantitative analysis on the various attack vectors through the CPS. Extending on this concept, Le May et al. [75] presents the ADVISE method, which is an attacker model that describes the relationship between the attacker and the system as an abstract attack execution graph, defined as the set of attack steps, access, knowledge, skill, and goals. The ADVISE method is integrated into modeling tools to reduce the effort in evaluating various CPSs.

Even more rigorous probabilistic functions are applied by Adepu et al. [57] who describes a generalized approach for modeling attackers and attacks in a CPS, demonstrating the model against a water treatment plant. Where Adepu limits the attacker's influencing factors to intents, Deloglos et al. [5] developed a generic attacker model that allows for a wide variety of attacker profiles to be applied. This model applies probabilistic correlation functions to model the behavior of a non-deterministic attacker against a CPS.

Chapter Three

Attacker Modeling Framework Overview

This chapter gives an overview of our attacker modeling framework, starting with the workflow we used to develop it. We describe the composition of the attacker modeling framework and give a brief overview of its components. We also discuss the application of the AMF and summarize how the components are leveraged to integrate attacker models into security review processes. The purpose of this chapter is to briefly and concisely facilitate the big picture perspective of our attacker modeling framework. In Chapters 4, 5, and 6 we dive into the AMF components in-depth.

3.1 What are Attacker Models

The literature review in Chapter 2 paints a broad picture of what attacker models are and several ways they can be used. Due to this breadth and diversity, we begin by specifying our definition of an attacker model for this research. The field of cyber-security as a whole is no stranger to threat modeling. Threat models are commonly seen as algorithms or processes that model or simulate characteristics of a specific attacker or attack against a system. In security fields, the terms threat model and attacker model are sometimes used interchangeably [64]. However, with regard to attacker behavior, there are critical distinctions between the two. For this research, we define attacker models how they are most commonly defined in the field of attacker modeling: as attacker behavioral models. The critical distinction between a threat model and an attacker model is that where a threat model explicitly defines the nature of a threat, an attacker model qualitatively or quantitatively defines the threat's behavior as a function of the nature of the threat. By this definition, it could be said that attacker models are the subset of threat models that explore how the attacker's nature influences the attacker's behavior.

The critical implication behind this definition is that for an attacker model, if the nature of the attacker changes, then the modeled behavior of the attacker may change as a result. While it is common to see attackers discussed in SRPs, it is a rare exception to see a relationship defined between the nature and the behavior of the attacker.

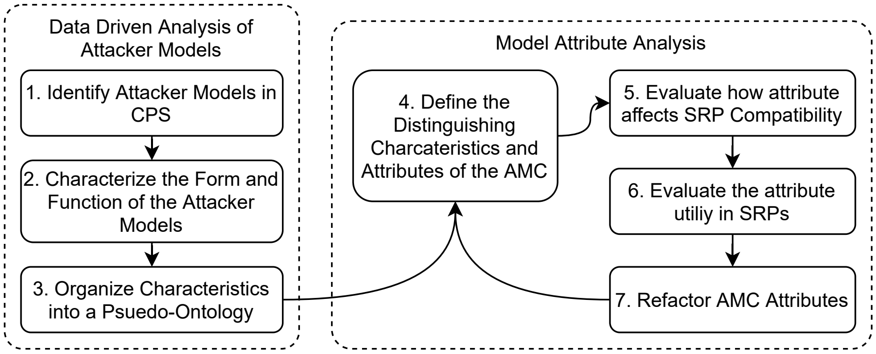

Data Driven Analysis of Attacker Models
1. Identify Attacker Models in CPS
2. Characterize the Form and Function of the Attacker Models
3. Organize Characteristics into a Psuedo-Ontology
Model Attribute Analysis
4. Define the Distinguishing Charcateristics and Attributes of the AMC
5. Evaluate how attribute affects SRP Compatibility
6. Evaluate the attribute utiliy in SRPs
7. Refactor AMC Attributes

Figure 3.1 The workflow used to create the attacker model characterization module.

3.2 Development Workflow

The creation of our attacker modeling framework required the intermediate formulation of several key concepts such as, "How do you describe the relevance of an AM to an SRP?", "How do you distinguish between an AM's inherent value and its value to an SRP?", "What is the relationship between the attributes of an attacker model and the assumptions it makes?", and, "What is the level of abstraction necessary for generalization of the AMF?" Throughout this dissertation, we identify and explain our conclusions for these concepts and several others and identify their contribution to the attacker modeling framework on the whole.

Finding solutions to these questions and synthesizing our generalized attacker modeling framework was done through a data driven literature survey and analysis. This development workflow can be seen in Figure 3.1 and consists of two primary efforts. The first is the collection of attacker model attributes. This started with a broad literature review of existing attacker models where we explored attacker models from their historical origins to state-of-the-art. For each attacker model, we characterized the form and function using attribute tagging and plain-text descriptions to identify each attacker model's underlying structure and operative mechanisms. To say attacker models are diverse in form and function is an understatement. This literature survey was a prime example of achieving order through the chaos as we worked through dozens of attacker models with hundreds of characteristics and eventually used the broad set to formulate our attacker modeling framework. As we developed this list of attributes, we formulated a pseudo-ontology of attacker models. There were several challenges with this process, not the least of which is that the field of attacker modeling is young enough that there is rarely consensus on how to represent common terms, ideas, and techniques. A good example of this can be seen in [64] where Rocchetto performs a literature review of attacker profiling techniques in several attacker models and provides a table of terminology mapping for the various terms used across different publications.

The second effort of the AMF creation was the refinement of the various attributes using the security review process perspective. The final set of attributes we use to describe attacker models is captured in this dissertation as what we call the Attacker Model Characterization (AMC), and is presented in depth in Chapter 4. The number and scope of attribute results can explode to an unusable size if left unbounded and may reduce the practical utility of the AMF. Therefore, rather than capturing attributes to describe all characteristics of attacker models, we are specifically concerned with those characteristics that make attacker models uniquely useful or unuseful to integration with security review processes. Our workflow was centered mainly around identifying and justifying which attributes those are.

To boil down our generalized pseudo-ontology of attacker model attributes in step 3 to a final set of AMC attributes we developed an attribute analysis process captured in Figure 3.1 as steps 4, 5, 6, and 7. In step 4, we compile the set of attributes into an intermediate AMC. These steps were primarily concerned with identifying categories of attributes and relationships between attributes. Our AMC employs a concept of attribute dependency, and we often had to refine our notions of what attributes are genuinely dependent on others as we identified research that employed the dependent without the dependency. In step 5, we evaluate how attributes affect compatibility with security review processes. This involved an evaluation of several different SRPs, including EPRI's TAM [1], STPA-SEC [3], STPA-SafeSec [4], EPRI's HAZCADS [35], and MissionAware [15] in order to determine if and how different attacker model characteristics could inhibit integration.

In step 6, we evaluated the value that different attacker model characteristics contributed to security review processes. This inherently required the development of the concept of value for attacker models. We discuss this notion in depth in Section 6.1. An important finding of our research is that an objective valuation of an attacker model is difficult to defend outside of the context of an application. Part of the novelty of our integration framework is the creation of a scoring-aided AM evaluation workflow capable of providing objective valuations of attacker models relative to the characteristics of a security review process. In application, this means that evaluating a set of attacker models for integration with two different security review processes will result in two different valuations, one for each SRP. In step 7 we apply the results from steps 5 and 6 to refactor the set of AMC attributes. This step included the removal of attributes that proved inconsequential to SRP integration, the consolidation of similar attributes, and the division of complex attributes that are better characterized as multiple attributes.

Steps 4, 5, 6, and 7 were repeated as a refinement feedback loop until the AMC reached a steady state of attribute consistency where sequential cycles resulted in the same set of attributes. It should be noted that between the time of the AMC creation and the publication of this dissertation, additional attacker modeling literature was published or discovered and integrated into the basis of the AMC. When discovered, new attacker models would be introduced into the refinement process by performing steps 1, 2, and 3 for the specific attacker model, then integrating the results into the existing AMC in step 4 and repeating the refinement cycle until the AMC again reached a steady state. While this process was performed manually, the development of support tools and the application of natural language processing could serve as an aid to maintaining and scaling the development workflow.

3.3 A High-Level Perspective

The AMF developed in this dissertation is an over-arching framework for describing attacker models and their functional implementations in a way that uniquely facilitates the integration of those attacker models into security review processes. The potential applications of the AMF are extensive.

Previous literature on attacker modeling frameworks has identified many attacker behaviors that may be modeled and a multitude of methods to model them. The lack of formalism surrounding the form and function of attacker models can make it difficult to readily understand how the attacker models work and what assumptions they make. The proposed attacker modeling framework defines a security process for describing the diverse range of behaviors captured by attacker models. Our generalized attacker model characterization allows AM developers to better describe how their AMs work and the core principles by which they operate. It also provides greater insight into the utility of the AM by identifying its operative requirements and the valuable insights and results it provides.

Also, contextual assumptions are perhaps the most critical aspect of cyber- and cyber-physical security and are discussed in-depth in Chapter 4. The widespread disagreement surrounding the efficacy of various cyber- and cyber-physical security research practices can largely be attributed to contextual assumptions made by those researchers but not identified or defended adequately. The attacker model characterization process in the proposed AMF identifies critical assumptions made in attacker modeling research and defines a process for documenting those assumptions.

Finally, a common fault in attacker modeling literature is the inadequate clarification of what the attacker model is doing without first requiring an understanding of how it is doing it. Goals, objectives, and methods are often described in parallel with the functional description of the attacker model. This reduces the usability of the attacker model by requiring a higher effort for comprehension and an even higher effort for interpreting assumptions made by the attacker model. In order to reduce this effort, we divide the AMF into two primary components, the Attacker Model Characterization (AMC) and the Attacker Model Functional Representation (AMFR), as seen in Figure 3.2.

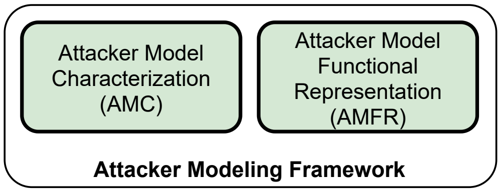

Attacker Model Characterization (AMC)
Attacker Model Functional Representation (AMFR)
Attacker Modeling Framework

Figure 3.2 The two primary components of our attacker modeling framework.

3.3.1 Attacker Model Characterization (AMC) Overview

The purpose of the AMC is to describe the distinguishing characteristics of an attacker model. A well-formed description of an attacker model that captures distinguishing characteristics is challenging to create objectively without defining a context of application. The scope of this dissertation in integrating attacker models into security review processes is the context we used to develop our AMC. While our AMC may have application outside of this context, we expect it would need to be expanded to provide an adequate description for applying AMs in different research fields.

One of the most critical functions of the AMC is that it aids in identifying assumptions for both AMs and SRPs. When developing an AM or an SRP it becomes evident that modeling the attacker in complete detail is as impossible as comprehensively assessing the security state of a CPS. Neither can be developed without making assumptions about the attacker and the system. A well-formed attacker model does not need to be a comprehensive attacker model but rather is one that makes well-reasoned assumptions and documents them explicitly. Likewise, a well-formed SRP is one that intelligently develops and documents simplifying assumptions that make the cost and effort to perform a review realistic. Far too often in both research fields, these assumptions are not explicitly stated and can be challenging to identify without both a broad understanding of the research field and an exhaustive understanding of the particular AM or SRP. When considering the objective of integrating an AM into an SRP, assumption mismatch can cause a well-formed AM and a well-formed SRP to be incompatible, and thus the AMC must be able to identify and distinguish these assumptions.

Another critical function of the AMC is to identify the goals and objectives of attacker models. The broad diversity of attacker models comes with a broad diversity of goals and objectives, and the value an AM provides to an SRP depends on the original goals and objectives of the AM. The AMC identifies the generic categories and objectives of attacker models and aids in identifying the value an AM may provide to different SRPs.

3.3.2 Attacker Model Functional Representation (AMFR) Overview

The attacker model functional representation seeks to capture at a high level how the attacker model functionally works. The different implementations of attacker models vary in complexity, ranging from structured methods of thinking to formal implementations of discrete-time functions and system-theoretic models. The goal of the AMFR is to be a standardized functional representation methodology that captures how an AM is functionally realized in enough detail to understand how the flow of information is leveraged to model attacker behavior. This includes breaking down the attacker model into functional subsystems with rules-based descriptions, all related by data flow. The partitioning of functionality and the definition of well-formed boundaries between partitions aids in reducing the effort to comprehend the attacker model.

In addition, the AMFR contributes to the attacker model integration process by characterizing the functional boundaries of the attacker model, which reduces the effort to identify the hand-off between the attacker model and the security review process.

3.4 Applying the Attacker Modeling Framework

While our AMF has broad application in attacker model development and research, its true novelty is realized when applied to integrate attacker models into security review processes. In Chapter 6, we develop the attacker model integration workflow, which is a semi-formal method for evaluating attacker models and integrating them into security review processes. The workflow divides the process accordingly. First the AMC is leveraged to evaluate attacker models. In Chapter 4, we also introduce the Security Review Process Characterization (SRPC), which is similar in form to the attacker model characterization, except that it is developed from the perspective of the security review process. Our workflow guides the security expert to evaluate the various attacker models using a combination of scoring methods and attribute analysis. This workflow guides the evaluation of attacker model compatibility, attacker model incompatibility remediation, attacker model valuation, and attacker model selection.

Next, the workflow leverages the AMFR to identify the bounds and data hand-off between the selected attacker model and the security review process. This includes the characterization of all input and output data from the attacker model and all provided and expected data from the security review process. It then uses this characterization to map the integration of the attacker model to the security review process. The attacker model integration workflow significantly reduces the burden of evaluating how well an AM fulfills the needs and functional requirements of an SRP.

The AMC and AMFR are described in full detail in Chapters 4 and 5 respectively, and the integration process is presented in Chapter 6.

Chapter Four

Attacker Model Characterization

In Chapter 3 we gave an overview of our Attacker Modeling Framework and identified that it is composed of two primary modules, which we call the Attacker Model Characterization (AMC), and the Attacker Model Functional Representation (AMFR). In this Chapter, we discuss the first of those two modules.

4.1 Overview of the Attacker Model Characterization Module

The purpose of the AMC is to describe an attacker model, capturing characterizing information including that information which is relevant to a red-team or blue-team security expert considering the attacker model for integration with a security review process.

4.1.1 Interface Characterization

In order to capture this information, the AMC identifies characteristics about the attacker model which we call *attributes*, which are categorized into three groups of information types we call *interfaces*. The proposed AMC can be observed in Figure 4.1. The AMC breaks down the characterization of the attacker model into three categorical interfaces which capture 1) the input information required by the attacker model, 2) the output information provided by the attacker model, and 3) contextual information about the attacker model. The design of the AMC was influenced by both the diverse nature of existing attacker models and by the expectations and requirements of security review processes that may integrate them such as STPA-SEC [3], HAZCAD [35], and Bakirtzis' ontological metamodel [76].

4.1.2 Attributes

An attribute defines a characteristic of an attacker model that specifies behavior, form, or structure. At its core, the AMC is a finite collection of attributes that together capture and describe all the

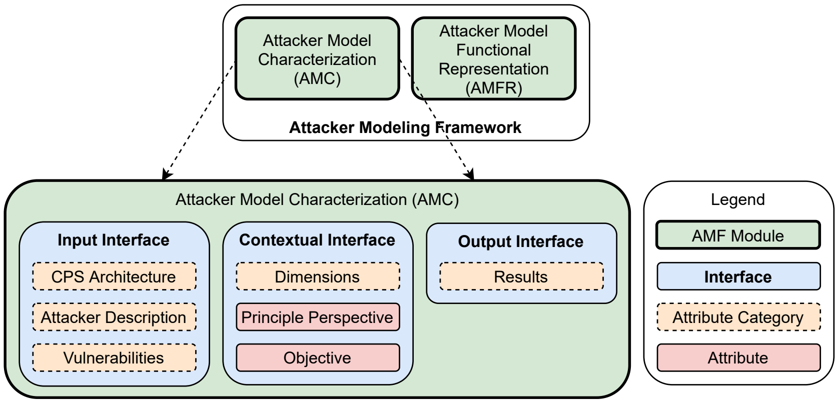


Figure 4.1 The high-level definition of the attacker model characterization module, including its three interfaces.

characteristics of an attacker model which are relevant to integration with a security review process. It captures these attributes in enough detail to evaluate the AM's compatibility with an SRP and to identify the assumptions it makes that are relevant to integration with an SRP. We consider the AMC to be defined as the set A of k attributes such that $A = \{a_1, a_2, ..., a_k\}$. Each categorical interface is defined as a hierarchical subset of these attributes which are often grouped into larger sets of attribute categories. Given the input interface, the contextual interface, and the output interface, each having l, m, and n attributes respectively where $k = l + m + n$, the set of attributes in each interface is defined as $PI = \{a_1, ..., a_l\}$, $PC = \{a_1, ..., a_m\}$, and $PO = \{a_1, ..., a_n\}$ where the attributes in each set are exclusive such that $A = PI \cup PC \cup PO$.

Each attribute is captured as a tuple of a descriptive statement D and an indicator I such that $a = \{D, I\}$. Indicators can be either in the form of a boolean or as a selection of a finite set of options. The indicator asserts whether or not the attacker model implements the particular attribute. For the boolean indicator, this is achieved by a value of *true* indicating that the AM does implement that attribute, whereas a value of *false* indicates that it does not. The selection indicator not only indicates if an AM implements an attribute, but also categorizes how it implements that attribute using a finite set of options. The descriptive statement is a short summary of how the attacker model implements that attribute.

Attributes may be associated via a parent/child relationship where the existence of the child attribute in an AM necessitates the existence of the parent attribute. This is useful for attributes that can be realized in one or more ways but when certain realization methods critically affect SRP compatibility or carry strong assumptions.

A significant contribution of this AMC is the determination of which attributes should be included in the interfaces, which should not, and for the attributes with selection options, which options should be available. Attributes were identified through an extensive review of existing attacker modeling and security review process literature, which is defined in Section 3.2. Not only did this process aid in identifying which attributes to include in the AMC, but it also aided in understating the affect that including different attributes has on the AM, including but not limited to the following:

- How the presence or absence of the attribute changes the compatibility of the AM with a security assessment process.
- How the attribute distinguishes between attacker models.
- If the attribute creates or implies assumptions that are commonly relevant to security review processes.

The documentation of these attributes in creating an AMC for a specific attacker model allows a high-level abstraction of the composition of the attacker model. This leads to a framework that readily facilitates the evaluation of many attacker models with a fraction of the effort that would be required from an unguided attacker model evaluation.

Example Attribute

Consider evaluating the attribute Time (further discussed in Section 4.2.3) for different attacker models. The attacker model defined by Teixeira et al. [63] models the CPS as a discrete-time system. For this AM, the attribute Time may be described as {True, Describes the CPS as a discrete-time system}. Contrast this to the AM defined by Adepu et al. [57], which considers as one of the criteria for success for an attack whether or not the attacker succeeds within a certain amount of time. This may be described as {True, Criteria for attack includes successful realization of all attack intents within a certain period of time}. While both AMs include concepts of Time, the short description provides context how the Time attribute is realized. Alternatively, Time could be defined for the AMs using the same descriptions but with a selection indicator instead of the boolean indicator as the set {Discrete, Continuous}. For the Time attribute in the given examples, both AMs would be defined with the same description, but Teixeira's AM would have the Discrete selection while Adepu's would have the Continuous selection. The attribute format allows the description of attributes that are realized very differently between attacker models.

4.2 Contextual Interface

The purpose of the contextual interface is to summarize and make accessible information describing how the attacker model works and what assumptions it makes about the nature of the attacker, the nature of the CPS, and the nature of how the attacker relates to the CPS. The attributes of the contextual interface can be seen in Figure 4.2 and consist of the principle perspective of the

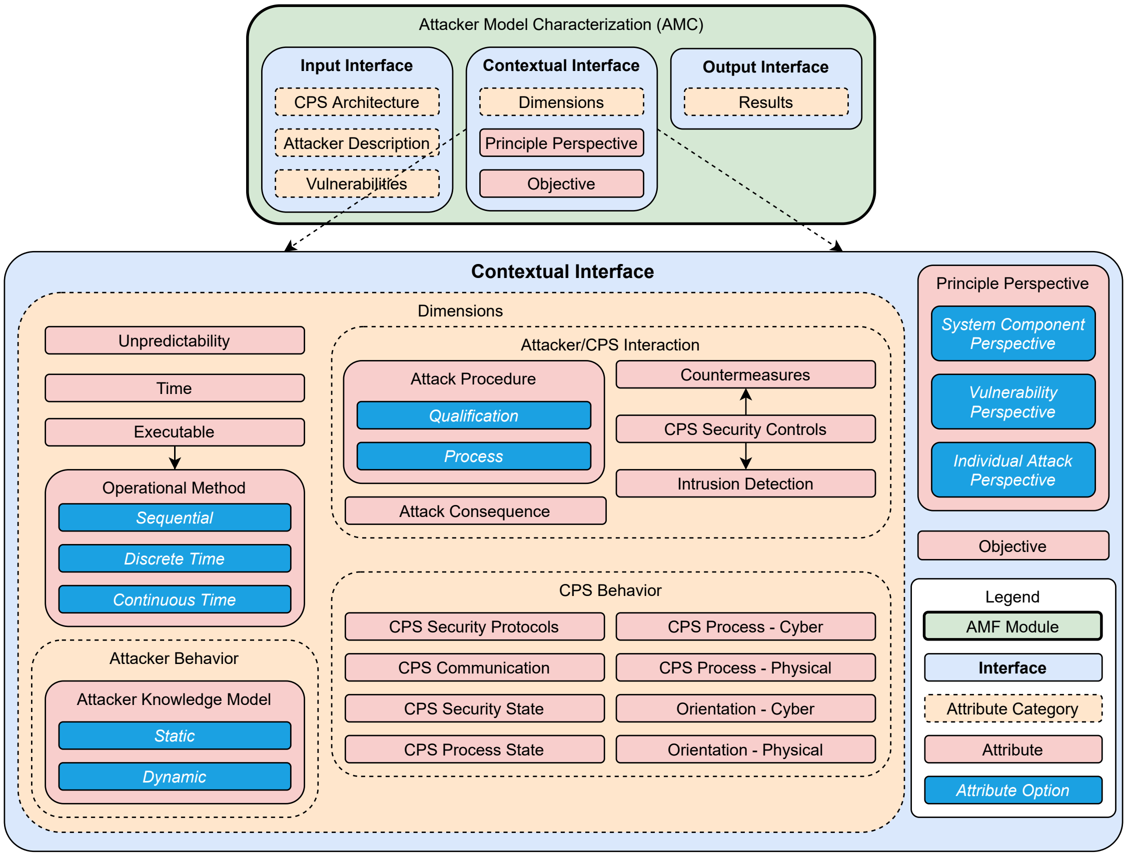


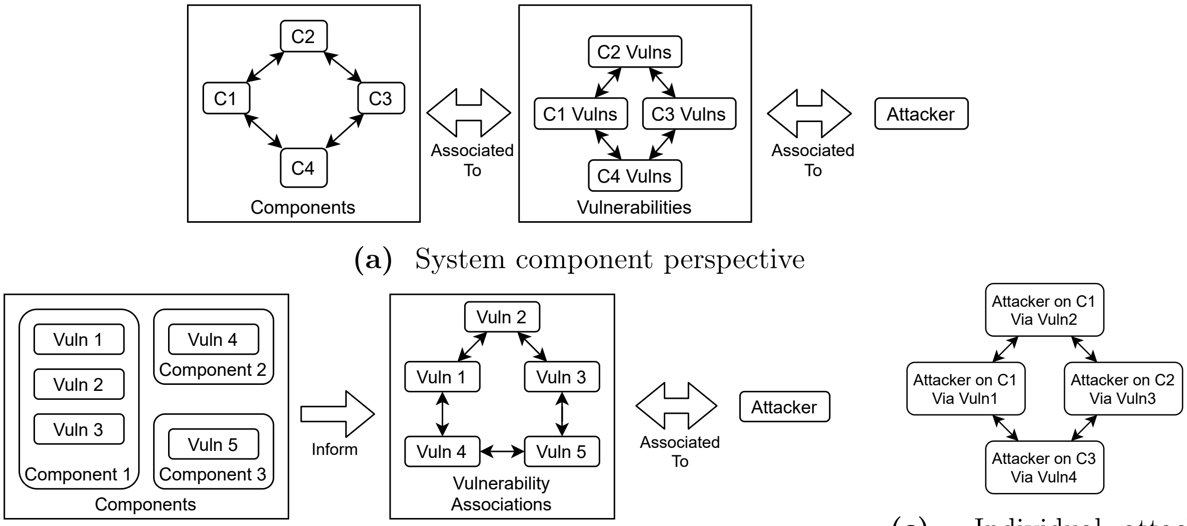
Figure 4.2 The contextual interface of the attacker model characterization with its hierarchically organized set of attributes.

attacker model, the objective, and the dimensions of analysis of the attacker model.

4.2.1 Principle Perspective of the Attacker Model

The question of how this attacker model relates the attacker to attacks and to the CPS is deceptively difficult to answer without performing an in-depth dive into each attacker model. The progression and value of attacker modeling research is in large part realized as researchers find new, unique, and verifiable ways to evaluate this relationship between the attacker and the system. The proposed AMC identifies three principle perspectives that at a high level capture the nature of the relationship between the attacker, attacks, and the CPS. These are the *system component* perspective, the *vulnerability* perspective, and the *individual attack* perspective. These perspectives are not intended as a formal or rigorous formulation such as in [70], but rather to identify the fundamental orientation of the how the attacker is related to the CPS.

The system component perspective first defines the system as an association of components, then associates vulnerabilities to those components, finally associating the attacker to the vulnerabilities.



(a) System component perspective

(b) Vulnerability perspective

(c) Individual attack perspective

Figure 4.3 Associative representations of the three contextual principle perspectives an attacker model can take on the relationship between the attacker, system components, and vulnerabilities. Image taken from our earlier publication on attacker modeling frameworks in [6].

This perspective orients the attacker model around the system design. It also tends to orient around interactions between components and better capture how attackers navigate through systems. A graphical representation of how the system, the attacker, and vulnerabilities relate for the system perspective can be seen in Figure 4.3a.

The vulnerability perspective begins by identifying vulnerabilities associated with components. It then develops an association of vulnerabilities, then associates the vulnerabilities to the attacker. The vulnerability perspective tends to orient the attacker model around attacks and attack procedures. A graphical representation of how the attacker, vulnerabilities, and the system components relate for the vulnerability perspective can be seen in Figure 4.3b.

The individual attack perspective is normally observed in research when an attacker model is designed to model a specific situation, and is normally derived in the context of a particular attack, attacker, and/or CPS. As such, application of the individual attack perspective in an SRP usually requires either 1) the SRP conforms to the attacker model's specific assumptions of the attack, attacker, and the CPS or 2) the SRP experts perform some amount of work to adapt the attacker model to the SRP context. While attacker models that use the individual attack perspective may require the greatest amount of effort to integrate, the constraints of a very specific context tend to allow them to produce more robust and insightful results. A graphical representation of how the attacker, vulnerabilities, and the system relate for the vulnerability perspective can be seen in Figure 4.3c.

While attacker models tend to stick to a single principle perspective, it is possible for an attacker model to feature additional perspectives as a step in the modeling process. For example, Deloglos et al. [5] develops an attacker model which utilizes the system perspective initially, but then develops individual attack perspectives in the form of an attack-decision tree to better describe how certain vulnerabilities influence the paths an attacker can take through a system. Orojloo et al. [55] develops a model based on the component perspective which quantitatively evaluates the relationships between components to associate vulnerability dependency chains, developing a vulnerability perspective. While there is research available from all three principle perspectives, it is arguable that the CPS security community is moving toward the system perspective.

While the principle perspective employed by an attacker model does not in and of itself enable or inhibit the functionality within the attacker model, it is noted that certain perspectives may be preferred for particular applications. Attacker models employing systems-theoretic methods tend to use the system component perspective because systems-theoretical methods have a fundamental dependency on the operative nature of the system and the relationships between system components. Attacker models that attempt to partition the analysis of the system architecture from the analysis of system vulnerabilities often employ the vulnerability perspective. This can be seen in the application of attack trees where the fundamental analysis is of vulnerability relationships [49].

4.2.2 Objective

Attacker models tend to be goal-oriented where the correlation of an attacker to a system is used to deduce an unknown about the security state of the CPS. Due to the diversity of attacker models, the AMF does not define a finite list of objectives. The objective is unique from all other attributes in that it does not contain a boolean or selection indicator and is simply composed of a short description. The goal of the *objective* attribute is to capture a high-level description of the purpose the attacker model was developed for by the original author.

4.2.3 Dimensions of Analysis

Capturing how an attacker interacts with a CPS is a considerable challenge in attacker modeling. We adopt the common term in the field of attacker modeling of a *dimension* of analysis to categorize an attribute that describes an aspect of the conceptual universe the attacker and the system reside in. These dimensions can be seen in Figure 4.2.

The goal of the dimensions of analysis attributes is to call out and identify assumptions made by the AM, SRP, or both that are critical for consideration for integration, but may be difficult to recognize if not explicitly identified. The dimensions of analysis aid in understanding why certain AMs may be more suited to contexts and contextual assumptions than others. Teixeira et al. [63]

describes cyber-physical attacks as residing in a three-dimensional space consisting of system knowledge, disruption resources, and disclosure resources. According to Teixeira, various attacker models are more readily suited to address different quadrants of this space based on the dimensions they do or do not include in their attacker modeling process. While the dimensions of the attacker models reviewed in the development of this work are summarized in Figure 4.2, additional dimensions may be necessary to describe attacker models as the landscape for attacker modeling research continues to evolve.

Unpredictability

Unpredictability is the notion of including behavior in an attacker model that intentionally makes the attacker behavior unexpected or hard to predict. Unpredictability requires that a system provided with the same input cannot have an absolutely predictable output. This is employed in attacker models when modeling uncertainty in various processes.

Deloglos et al. [5] demonstrates how a lack of knowledge of the attacker may be modeled by introducing an unpredictable attacker. The attacker in Deloglos' model is defined as a property-based attacker with a range of values for each property. Each property range is randomly sampled at the beginning of the attack process, resulting in an unknown attacker.

Unpredictability is often used in attacker models as an alternative to exhaustive modeling where the state-space required to model all possible attack procedures and outcomes explodes to an unmanageable solution. In such instances, unpredictability can be used with repeated execution to evaluate the attack variance that occurs from the changing characteristics of the attacker model.

Time

An attacker model that includes the *time* attribute attempts to capture the concept of physical time as a part of the modeling methodology. Time-dependency is observed in several of the attacker models and can serve several different functions. One example is when the CPS, the attacker, and/or their process of interaction are modeled in discrete or continuous time, as was done in [55], [57], [63], [72], [74]. Models that utilize time can leverage it to determine metrics such as success/failure criteria for attack outcomes as in [57], or to calculate physical processes such as real-world communication delays as in [74]. Attacker models that do not utilize time-dependent attacker or system models can still apply time principles in different ways. LeMay in [54] assigns time values to various attacker actions, estimating the time for a complete attack as the sum of times for all attacker actions.

In the field of attacker modeling, making the assumption of time-invariance allows for drastic simplification of attacker models. As is the case for most attributes, the inclusion or exclusion of time in an AM does not in itself imply a higher or lower quality attacker model. However, certain features and functions in attacker models can be considered more or less effective based on the inclusion of *time*. For example, modeling intruder detection or countermeasures in AMs without the inclusion of time makes the assumption that detection or countermeasure deployment process is time-agnostic. Research has demonstrated contrarily that the effectiveness of intrusion detection and countermeasure deployment systems is critically dependent on how much time it takes to detect and counter the attacker [54], [72]. Therefore, it is reasonable to assess that an AM that claims intrusion detection or countermeasures modeling capabilities without a concept of time and without justifying the assumption of time-agnosticism is of concern.

When considering the effect of time-dependency in an AM for application in an SRP, it is important to recognize that, while the application of time is bounded inside the AM, how time is leveraged may influence what information must be provided to the AM. For example, AMs that leverage intrusion detection or countermeasures often depend on attacker actions being provided with associated time for completion. Similarly, AMs that output security metric violations often require as input time-dependent CPS process properties.

Executable

While an attacker model, by definition, must be able to at some level model the behavior of an attacker and the interactions between the attacker and the system, it does not necessarily provide an algorithm for simulating the execution of an attack. An attacker model is executable if it leverages the modeled behavior of the attacker to simulate an attack process, producing attack results. Deloglos' AM in [5] is executable as an algorithm with a feedback loop that sequentially executes attack steps against the CPS. The series of sequential attack steps constitutes an entire attack. Orojloo in [55] implements an executable attacker model as a set of continuous-time equations that model the change of the system state as the attack progresses. Ekelhart in [59] applies an executable attack simulation engine to abstracted attack vectors to evaluate the number and prominence of various attack paths through a CPS.

While the quality of an attacker model may be significant, its utility in application depends on whether or not it is executable. When considering integration into an SRP or any application that requires execution of the attacker model, the ability to execute the simulation of an attack process is a requirement.

Operational Method

The *operational method* is a child attribute of the executable attribute and indicates if the attacker model is fundamentally executed as a discrete time system, a continuous time system, or as a series of sequential steps. Deloglos in [5] uses a sequential operation where the attack is modeled as a sequential execution of a feedback loop of attacker-system interactions. Le May in [54] models the attack execution as a series of sequential steps composing a modified attack tree. Adepu's model in [57] captures the state of the CPS as a function of continuous time. Similarly, Basin's AM in [74], which is particularly concerned with modeling physical aspects of wireless communication systems, models the communication process as a function of continuous time.

4.2.4 Attacker/CPS Interaction Attribute Category

The dimensions of analysis category is further divided into several sub-categories for clarity. The first of these is the Attacker/CPS Interaction category. Attributes in this category define how the attacker model relates the attacker to the CPS.

CPS Security Controls

CPS Security Controls are CPS-implemented controls used for detecting attacks and/or intervening in the case of an ongoing attack with the intent of inhibiting attack progression or mitigating attack consequences. These are most commonly realized via Intrusion Detection and Countermeasures which are both child attributes of the CPS Security Controls attribute.

Countermeasures

Ekelhart's et al. attacker model [59] includes system controls as preconditions to attacker actions, modeling the effects of preventative and detective controls such as an antivirus on a computer. In using this, Ekelehart's AM predicts the difference in attack outcomes against a system without controls versus against a system with controls. Le May's attacker model [54] similarly includes system countermeasures as a variable in the probabilistic functions that predict attack success. Where both of these AMs model controls at a low-level, an example of high-level control modeling is in Moteuis' AM [56] where he reasons about the effects of countermeasures on various attack scenarios and how the existence of countermeasures influences attacker behavior. The design and use of countermeasures requires some amount of preliminary knowledge of expected attacks. When considering a formal method or algorithm for modeling countermeasures in an attacker model, the relationship between the countermeasure and the attack must be explicitly defined in order to formulate a mechanism by which the countermeasure can inhibit the attack process. In AMs with a component or vulnerability perspective, this relationship can be defined either by the model or assumed as input to the model via attack vector associations to countermeasure actions or via attacker vector associations to security properties. In AMs that use the individual attack perspective, the relationship between countermeasures is most commonly defined as a part of the AM. Also, the ability to deploy countermeasures implies the ability to detect attacks. The applications of countermeasures without employing attack detection identifies an assumption of an alternative source of information that is able to indicate the occurrence of an attack to trigger countermeasure deployment.

Intrusion Detection

Intrusion detection is the process of monitoring system behavior in order to identify behaviors indicative of the presence of an unauthorized intruder. It is often associated with CPS control actions, being included as a prerequisite to the deployment of countermeasures as a part of attack detection. Ekelhart et al. [59] includes intrusion detection as a control method, modeling its effects on the attack outcome. Le May et al. [54] includes the probability of intrusion detection as a variable in the function used to calculate the attractiveness of different attack steps. McEvoy et al. [62] created an attacker model base on a *pi*-calculus variant with the primary purpose of intrusion detection. This model used *pi*-calculus to formally define both the system infrastructure and the adversary capabilities, then developed a formulation to detect anomalous control readings resulting from attacker interaction with the system.

While both similar, intrusion detection differs from attack detection in its fundamental objective. The primary goal of attack detection is to monitor for and identify attack actions performed by the attacker against a system. Intrusion detection, on the other hand, seeks to identify the presence of an attacker in a system, either as a result of or even apart from hostile action. Like countermeasures, intrusion detection requires an explicit formulation characterizing the relationship between the attacker (or the attacks) and the system. This formulation is either defined as an inherent characteristic of the attacker model or is required as input to the attacker model.

Attack Procedures

Attack procedures are core to attacker models and are realized using either a qualification scheme or a process scheme. The *attacker procedure* is the component of an attacker model that relates the attacker, the attack, and the CPS, modeling the execution of the attack. In a qualification scheme, attacks are treated as succeed/fail depending on whether some property-based criteria have been met by the attacker and/or the CPS state. In a process scheme, attacks succeed or fail based on the resultant value of an algebraic or system-theoretic process.

In the attacker model developed by Deloglos in [5], attacker actions are selected based on probabilistic equations, but the execution of the attack is performed through a simple qualification scheme where if the attacker has the ability to perform the attack it succeeds and if the attacker does not have the ability it fails. Similarly, Ekelhart in [59] uses a qualification scheme that assigns preconditions to attacker actions and if an action meets all preconditions for a particular target, it succeeds. Adepu in [57] uses a process-based attack procedure where the AM models the interaction of the attacker with the system and success of an attack is marked by the system entering an invalid state. System-theoretic approaches are common such as those in [55], [62], [63], [72], as well as Basin's in [74] which models the interaction of honest and dishonest agents in the system as a continuous-time interaction process and leverages security protocols to identify violations of allowed system behavior. LeMay in [54] employs a Markovian decision process to calculate the likelihood of all attack actions being selected, then selects the action with the highest likelihood.

The mechanism for attack procedure has a profound impact on the efficacy of an attacker model. The attack procedure is the component of the AM that models the decision process of the attacker and how different characteristics of the target system, the attacks, and the attacker interact to result in attacker behavior. One of the greatest variables in cyber-security is the human factor. We as humans are very complex creatures and modeling or predicting the behavior of attackers is a monumental challenge. The fact that many security review processes don't utilize security attacker behavioral modeling, however, does not mean that it's not happening somewhere in the process. Often it just means that assumptions or subjective reasoning is being applied that the security experts may or may not be aware of, effectively implementing a qualification-based attack procedure reasoned out by the security expert.

As described in Section 3.3, in order to be viable, attacker models must make simplifying assumptions that reduce the complexity of the attack procedure enough to arrive at a reasonable solution. While the burden of effort to implement an attacker model is a product of many different characteristics, it generally holds true that the higher the complexity of an attacker model, the more effort it takes to comprehend and implement the attacker model. While this burden of effort is captured in part by the description of the attack procedure attribute, it is also represented in the input interface of the AMC which captures the information required by the AM to execute, which is often a direct result of the complexity of the attack procedure of the AM.

Attack Consequences

Attack consequences capture how the effects of the attack inhibit the system from realization of its operational function and goals. Attack consequence descriptions vary depending on the CPS modeled and the AM, but typically associate the success or failure of attacks to CPS services, processes, and security properties. Adepu et al. in [57] associates attack result to system process properties and performance metrics to identify how the system performance is affected by various attack actions. Orojloo et al. in [55] demonstrates how system process properties can be used to calculate the risk associated with various attack actions.

Evaluation of attack consequences require an understanding of not only proper system behavior and process, but improper system behavior and process. While proper system behavior can be proven or disproven using security protocols, understanding improper system behavior requires a more thorough understanding of the system architecture and can lead into complex analyses of unexpected system behaviors and unexpected system states. This problem is commonly addressed in security review processes by separating the attack analysis process and the consequence analysis process into separate well-formed steps of the security reviews. This can be observed in EPRI's toolbox of security tools where attack consequence analysis is performed in the HAZCAD tool and the results are fed into the attack analysis performed in the TAM tool toward the end of the TAM workflow.

4.2.5 CPS Behavior Attribute Category

The second sub-category of the dimensions attribute category is CPS Behavior attribute category. Attributes in this category capture how the CPS operates. This includes structural aspects such as the cyber and physical orientation of the CPS to process-related aspects such as safety and security protocols.

CPS Security Protocols

CPS security protocols are properties and procedures that define acceptable CPS behavior with regard to security requirements. CPS security protocols are often used in AMs for identifying how attacker interaction violates the system by identifying when and where system behavior deviates from the proper protocols. Application of CPS security protocols is often analogous to how property proving is applied in CPS assessment for safety assurance.

These security protocols are often received as input to the attacker model as a part of the CPS description and violation of these protocols is described in conclusion to the simulation of the attacker model. Basin et al. in [74] develops security protocols for four system behaviors which are authenticated ranging, ultrasound distance bounding, delayed key disclosure, and secure time synchronization. Basin extends research on security protocols to include physical aspects of security protocols and leverages attacker models for security protocol verification. Vigo in [73] developed a higher-level attacker model which explored several ways that security protocol violation can affect a CPS based on different types of attackers. Vigo particularly focused on bridging security protocols between the cyber and the physical worlds and how physical interactions change procedures for automated security protocol verification. Security protocols are most commonly applied in one of two ways. The first is for attack detection where security properties are leveraged as bounds for proper system operation and violation of those bounds implies an attack. The second is for proving system security and stability in spite of attacks, in which case security protocols are employed by the system and the absence of process deviation from those protocols during attacks us used to prove security protocols. For both cases, the application of security protocols in attacker models assumes the availability of detailed knowledge of the system design, the system processes, and the system controls, as well as detailed knowledge of how the attacker can relate to the system.

In considering integrating AMs that employ security protocols into SRPs, it is of note that the detailed knowledge of the system design, system processes, and system controls is often expected as input to the attacker model from the SRP, and not formulated by the AM itself. The burden of producing security protocols for a complex CPS is significant and in an SRP that does not already provide the protocols, this burden may inhibit the usage of such an AM.

CPS Communication

The CPS Communication attribute identifies if an attacker model defines communication interactions between components of the CPS. In [74], Basin models physical communication between wireless components, specifically evaluating the effects of distance-based communication delays on intruder detection by monitoring violations of time-based security protocols. In [5], Deloglos identifies all communication channels between system components and the types of communication employed, which are leveraged to evaluate attack propagation using the communications. McEvoy's AM in [62] is specially designed for SCADA systems to use *pi*-calculus to represent communications between components in the SCADA network.

While the vast majority of attacker models employ some form of communication modeling, the form and function of it varies drastically. The Component Relationships attribute of the input interface complements this attribute by identifying what input information is required by the AM from the SRP in order to model the communication-based interactions between the CPS components.

CPS Process State

The CPS Process State attribute is common to nearly all attacker models with an operational method and identifies how the AM captures the process state of the CPS as the attack progresses. AMs that have a discrete-time operational method tend toward representing the process state as a snapshot of all process values at any given point in discrete time, as is observed in [55], [63]. Similarly, AMs that employ sequential operational methods tend to capture the process state for each step of the sequence as the set of all internal variables in the AM. AMs that employ a continuous-time operational methods can capture the state of the system at periodic or a-periodic intervals such as in [57], [74] or can leverage system-theoretic models that can formulaically resolve the state of the system at any given point in time.

CPS Security State

The CPS Security State attribute identifies how the AM captures the security state of the CPS as the attack progresses. AMs such as [5], [54], [59] maintain a record of system components that have been compromised as the attack progresses, which can be combined with a record of the CPS Process State to evaluate the progression of the attack. AMs such as [62], [72], [74] record the CPS Process State throughout the attack and apply security protocols to evaluate when, where, and how the system behavior deviated from the expected behavior.

Cyber Processes

The Cyber Processes attribute indicates if the AM models cyber processes of any form. This can be observed in attacker models such as in [57] where Adepu models the plant control process which uses sensor information to evaluate appropriate control actions, sending commands to various actuators. Basin in [74] models authentication processes that occur in nodes on a wireless network. McEvoy's attacker model in [62] models in depth the SCADA communication process between several nodes on a SCADA network. Modeling cyber processes is difficult without well-defined relationships between components. As such, Cyber processes are most commonly observed in AMs developed from the system component perspective and the individual attack perspective, AMs developed from the vulnerability perspective tend to be process-agnostic.

Physical Processes

The physical side of cyber-physical systems adds a layer of complexity in requiring consideration of not only the Cyber Processes but also the Physical Processes involved in modeling the CPS. Certain attacker models simplify this by presenting a CPS model that considers the physical outcome as an instantaneous result of the cyber-process rather than a process in itself. Other attacker models separate the cyber and physical processes and use a unique method for representing the physical process. For example, Adepu's attacker model in [57] models all physical processes in the plant and defines performance metrics to measure physical properties such as the pH of water in a water-treatment process, the water level in the tank, and the flow rate of water through a pipe.

Orientation - Cyber and Physical

CPS Orientation is often partitioned into cyber orientation and physical orientation. While modeling physical processes are commonly observed in attacker modeling literature, the impact of spatial orientation receives far less attention than does cyber orientation. Cyber orientation is a fundamental requirement of all attacker models and is the method that represents the relationship between various components in the system. Cyber orientation is often captured using communication methods such as those captured by the CPS Communication attribute such as in [5], [56], [62], [63], [72]–[74], but can also be represented in more abstract form when simpler component associations are necessary such as in [54], [55], [57], [59]. Physical orientation, however, is much less commonly employed and can be observed in AMs such as Basin in [74] where the distance between wireless nodes is used to measure the propagation delay of messages for range-based authentication. It can also be observed in AMs such as [55], [63] where the physical orientation of various components is used to model the relationships of process-based interactions between components and between the attacker and components.

4.2.6 Attacker Behavior Attribute Category

The final sub-category of the Dimensions attribute category is the Attacker Behavior attribute category. Attributes in this category identify how the attacker model implements the behavior of the attacker. In the literature survey we only identified one attribute in the Attacker Behavior category that is critical to SRP integration. The decision to include the Attacker Behavior as its own category was made to clearly distinguish attributes related to the CPS Behavior from those related to the Attacker Behavior, both being clearly distinguished from attributes related to the interactions between the attacker and the CPS.

Attacker Knowledge Model

The attacker knowledge model identifies whether the attacker knowledge of the target system can change over time. The options for the attacker knowledge model are static and dynamic. In the instance of a static model, the attacker begins the attack with a full knowledge of all information that will be used in the attack process. In the dynamic model, an attacker has the ability to learn various types of information as the attack progresses. In [5] Deloglos models the attacker knowledge as static with respect to resources and vulnerabilities, but dynamic with respect to known system components. The attacker begins the attack with a limited initial knowledge of the system and only has awareness of the systems that the attacker has control of, or that have direct communication channels to systems the attacker has control of. From there the attacker learns the system as the attack progresses. Attacker models such as Adepu's in [57] claim the capability of employing both forms of attacker knowledge models, although in Adepu's case only the static model is demonstrated. Mo in [72] demonstrates an attacker model that uses the individual component perspective in which the attacker knowledge is defined as a static part of the attack procedure.

Dynamic attacker knowledge modules are valuable when trying to model the concept of an intelligent and learning attacker. It should be noted, however, that exhaustive modeling of a dynamic

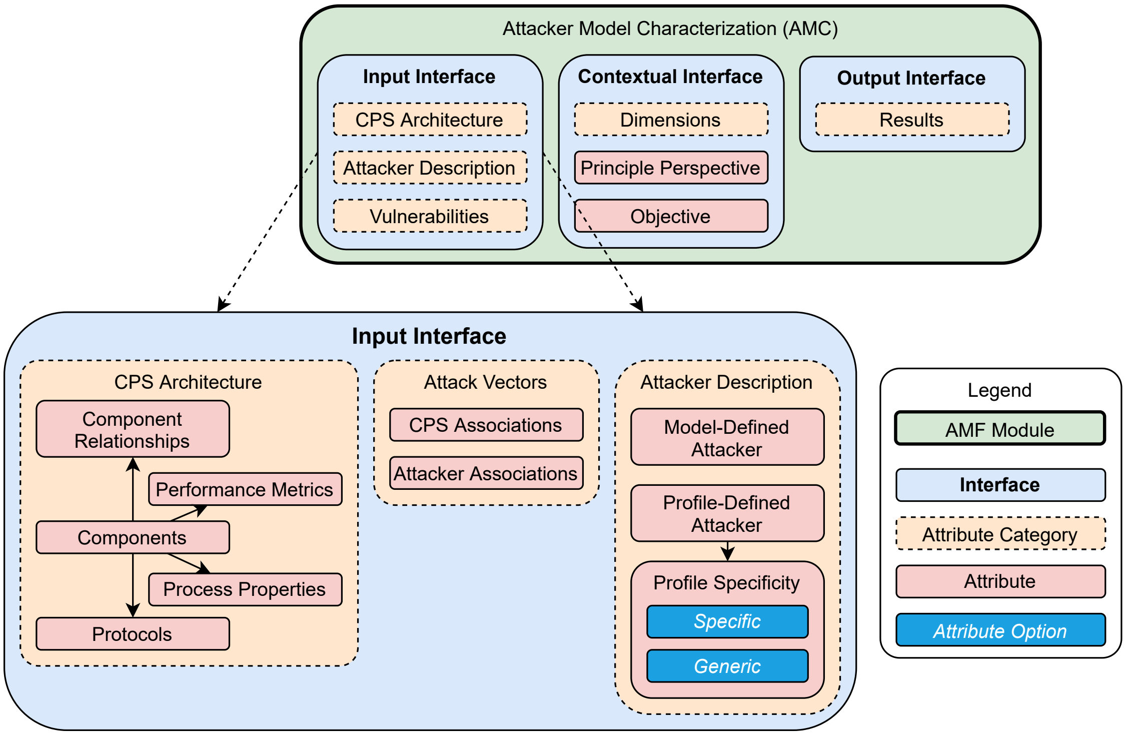


Figure 4.4 Hierarchical input interface attributes for the attacker model characterization.

attacker can quick result in a state-space explosion.

4.3 Input Interface

The information required for an attacker model to execute is often not stated up front, but rather requires effort and a holistic understanding of the attacker model to determine. When moving from the theoretical evaluation of attackers to practical application in a security review process in the context of a real and complex CPS, the effort of translating data from the attacker model to the SRP without a consistent data taxonomy is considerable. At a base level, all attacker models require as input some information about the attacker and the system to function. Categorically, attacker models tend to take as input information about the CPS, attack vectors, and a description of the attacker. The attributes used to describe each of these for the proposed AMC can be seen in Figure 4.4 and discussed in the sections below.

4.3.1 CPS Architecture

Despite the availability of frameworks for describing CPS topologies [67], [69], [77], attacker models tend to utilize ad hoc methods, describing only the aspects of the CPS relevant to the attacker model. In reviewing attacker models, five categories of input information were identified as being pertinent to the description of the CPS architecture as summarized in the CPS architecture attribute category in Figure 4.4. It should be noted that attributes of the input interface are only intended to capture a high-level description of the input information required by the AM. The functional form of input information is provided by the attacker model functional representation and described in Chapter 5.

Components

Beginning from an abstract view, a CPS is defined as a composition of system components, each being described by various properties. These properties are unique to each attacker model but usually include device identifiers, device descriptions, cyber and/or physical orientation, and component characteristics related to attack success and security compromise. The different dimensions of various attacker models require different component properties. Both Adepu's AM in [57] and Deloglos' AM in [5] represents the CPS as an abstract domain model and requires as input to the system a complete list of all components. Monteuuis' AM in [56] uses the individual attack perspective and models a static system architecture with all components predefined, thereby requiring no components as input. Basin's AM in [74], by contrast, requires as input descriptions of all components, each defined with security protocols for authentication behavior.

Component Relationships

While all of the attacker models in some capacity describe the CPS components, not all models take as input the relationships between them. Where the Cyber-Orientation attribute is present for nearly all attacker models, many attacker models define the component orientations in the model itself rather than taking the orientation as input to the model. Component relationships are a child attribute of components and exist in all observed attacker models, but are implemented in a wide variety of ways. This is observed in [55] and [54] which take as input variants of attack trees that assume that the context of CPS component relationships has already been integrated to produce attack chains. It is also observed in [56] and [72] which define the nature of the relationships between CPS components as a part of the attacker modeling process rather than as input to the attacker model. Relationships between components are often defined in terms of communication protocols such as those characterized by the CPS Communications attribute of the contextual interface.

When considering the utility of different AMs in an SRP, the description of the component relationships attribute is useful to quickly identify if the component relationship information defined by the CPS architecture in the SRP is adequate to meet the requirements of the AM input interface.

Process Properties

Several types of CPS process properties are identified in attributes of the contextual interface. The process properties that are required as input to the AM are identified by Process Properties attribute. The ability to take as input these system process properties and model property violation in the context of the attacker model is a promising feature. Adepu's AM in [57] defines a scope of relevant system process properties and performance metrics which are required as input to the attacker model. McEvoy's AM is unique in that it models in great detail the process and properties of a SCADA system. Process properties are provided to the model as input which define the operations of sensor input, supervisor processes, and control loops. Orojloo in [55] demonstrates a method for describing system process properties using fuzzy logic, which are expected as input to the AM.

The amount of input information required by an AM has a significant influence on AM compatibility. Depending on the scope of the SRP, architectural information such as process properties, security protocols, and performance metrics may be unavailable and the effort required to develop such properties or metrics increases exponentially as the CPS size and complexity increase.

Security Protocols

While security protocols and their effects on the attack process is an important aspect of CPS security, they are not well explored in attacker modeling literature. Basin et al. [74] explores protocol manipulation but also demonstrates that protocol descriptions are complex and require a detailed security process model.

Performance Metrics

While most attacker models evaluate attack outcomes as the attacker succeeding or failing to attack, more complex models may take as input performance metrics which are used to capture the system operating in an altered or non ideal states.

4.3.2 Attack Vectors, CPS Associations, and Attacker Associations

Attack vectors are a standard component of all reviewed attacker models, although the level of abstraction of the attack vector varies significantly between models. Like CPS components, attack vectors each tend to be assigned various properties that describe the nature of the attack vector and the vulnerabilities it exploits. The most significant architectural distinction between attacker models is if and how they associate attack vectors to the CPS and/or to the attacker.

CPS Associations

The three principle perspectives described in the contextual interface represent three different approaches that are used to model the relationship between the attacker, the attacks, and the CPS. For all three perspectives, it is necessary to formulate a relationship between the various vulnerabilities that the attacker exploits in an attack. As seen in Figure 4.3a the system component perspective relates vulnerabilities to components in the CPS, then relates components to each-other through the component relationships. For the vulnerability perspective in Figure 4.3b, vulnerabilities are still associated to CPS components, but are directly related to each other. The critical implication of this is that the vulnerability perspective assumes that the SRP can provide associations between vulnerabilities in the target system. In [5], [57], [63], [73], the vulnerabilities are provided by the SRP with associations to the CPS components they are related to. In [73], vulnerabilities are associated to components using various actions, including remove, read/write, reveal, reprogram, and starve. In [5], each component is defined with a list of associated vulnerabilities. In [59], attacker actions and exploits are defined with a set of preconditions. Components that satisfy all preconditions are then associated to attack actions.

Attacker Associations

Attacker associations are used to model how the attacker relates to the vulnerabilities and attacks. This is most commonly realized in the form of attack selection. In [5], Deloglos' attacker model applies the same properties to attack vectors that are used to characterize attacker profiles. Vulnerability properties are associated to attacker properties using a probabilistic function to predict vulnerability selection. Similarly, in [55], both attackers and attack steps are assigned four-property profile values consisting of knowledge, access, user interaction, and skill, which are formulaically associated. Because AMs developed using the individual attack perspective are designed for application in a specific scenario, then can derive much more rigorous algorithms for relating the attacker to vulnerabilities.

4.3.3 Attacker Description

Two primary techniques for describing the attacker are observed in the reviewed attacker models.

Profile-Defined Attacker

The first utilizes attacker profiling, a topic well explored in CPS security literature, in which various properties of the attacker are captured as attributes and correlated to the attacker profile as in [5], [54], [55], [57], [59]. An important distinction for the utility of a profile-based attacker model is whether the attacker profile is specific or generic, which is captured using the profile specificity child attribute. Generic attacker models tend to offer more utility by allowing the experts driving the security review process to define the attacker profile properties of interest. Rocchetto et al. provides a summary of property-based attacker profiling techniques in attacker modeling literature [64].

Model-Defined Attacker

A second technique is observed when the nature of the attacker is not taken as input but rather is defined by an objective comparison between the attacker and the system based on shared domains of information in which they interact as in [56], [62], [63], [72]–[74]. Basin in [74] demonstrates this by using the domains of time and agent/intruder location and applies them to physical properties of the network.

4.4 Output Interface

Despite the purpose of attacker modeling being so closely linked with security assessment, a critical question that is often not addressed in attacker modeling literature is, "How will the information provided by the attacker model be useful to a security expert?". In order to integrate the AMF into security process, we must first define what information from the AMF is of value and how it will be useful in the context of a security analysis. The output of an attacker model is critical in determining its compatibility to a security review process, and is often considered first in determining the value provided by the attacker model. The output interface for the AMC is described by the set of five categories of results we identified in our literature review. These can be seen in Figure 4.5.

4.4.1 Procedure

Attacker models that model *procedures* attempt to describe the process of how an attacker performs an attack. These varied in level of abstraction from a high level of summarizing attacks as single actions against the system as a whole, to low level evaluations of chains of vulnerability exploits and the propagation of the attacker through the system. In [56], Monteuuis reasons about how different attackers may go about performing attacks, resulting in a list of procedures that may be exploited. This high-level analysis can be contrasted to the low-level attack simulation engine developed by Ekelhart in [59] which implements an executable program that generates thousands of detailed attack chains, explicitly identifying which steps in different attack procedures lead to attack success or failure and, in the case of attack failure, documenting the failure mechanism.

Capturing the results of an attacker model is essential for developing a value proposition. We note that while a plethora of attacker models explore various characteristics of attacker behavior,

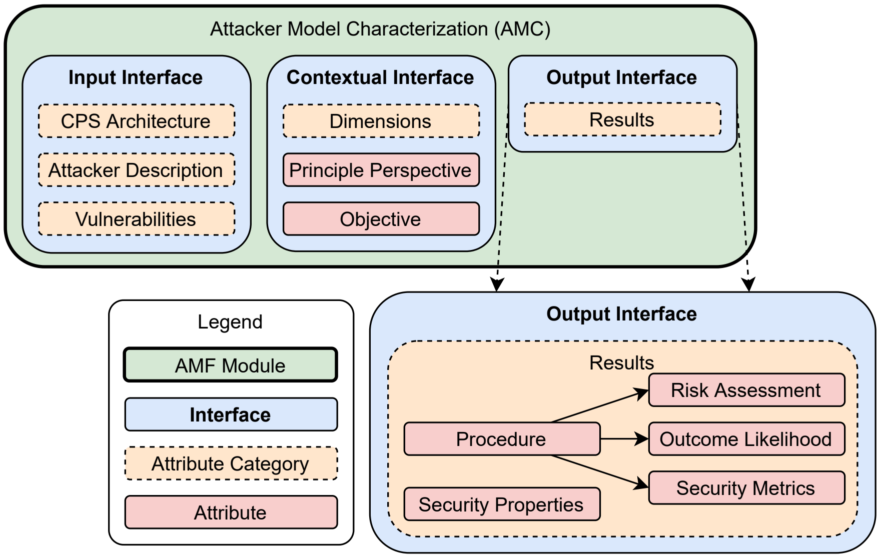


Figure 4.5 Hierarchical output interface attributes for the attacker model characterization module.

many do so with the research intent of developing attacker modeling mechanisms rather then developing results that can be used for security assurance. To fill this deficit, we created a generic *procedure* attribute capable of capturing the value proposition of the differing behaviors and procedures of attacker models.

4.4.2 Security Metrics Violation

Security metrics is a child of the procedures result. Attacker models that modeled *security metrics* take as input security properties about the CPS and provide as output instances where the attacker violated those security properties. These metrics often include provision of service, violation of safe behavior, and violation of time constraints. Basin in [74] developed an attacker model that modeled the effects of interactions between the attacker and the system. Basin's model bounded acceptable system behavior using security protocols and identified attacker intrusion by monitoring deviations from acceptable system behavior. Monteuuis in [56], by contrast, reasoned about the effects of different types of attacks on various models of security goals.

Security metrics are a powerful tool that philosophically fall inline with much of the STPA [36] style of thinking where security assessment is focused around the violation of bounded system behaviors, which readily leads to discussions of safe and unsafe control actions. It may be expected that the contextual attributes that contribute to the realization of this behavior in the attacker model will match closely to those of similar styles of security review processes.

4.4.3 Outcome Likelihood

While many attacker models identify possible attacker actions, attacker models with the *outcome likelihood* attribute advance a step further to identifying the likelihood of various outcomes. In [5], Deloglos' attacker model calculates the likelihood of target selection and action selection at each decision step made by the attacker. Similarly, Orojloo in [55] identifies the likelihood of different decisions at different attack steps. In [59], Ekelhart's AM used detailed attack characteristics to execute large numbers of attack permutations which were evaluated to calculate likelihood of success of different countermeasures.

One of the greatest challenges of providing outcome likelihoods is that likelihood values and probability values are inherently subjective metrics and must be defended as such. This often leads to a conversation of attacker model validation, which is one of the more challenging issues facing attacker models. In Section 2.2.4 we identify that the progression of AM verification research requires integration with security review processes to facilitate real-world validation. The groundwork laid by our attacker modeling framework will help attacker models achieve validation goals by reducing the burden of integrating AMs into SRPs.

4.4.4 Risk Assessment

Risk assessment is a common step in the security review process that is well explored in literature, but is rarely observed in attacker models. Risk analysis normally combines the likelihood of attack outcome with some cost and/or consequence evaluation in order to determine an objective measure of risk. This can be seen in [55] where Orojloo develops a technique to calculate the risk of various attacks from the attacker model.

The influence of the nature of an attacker on the risk to a CPS is difficult to objectively predict from an attacker modeling perspective. Formal methods of analyzing, calculating, and predicting risk lie on the security review process side of the gap between security review process literature and attacker modeling literature. The lack of integration of attacker models into security process results in a significant lacking of risk assessment practices in attacker modeling. However, the integration of attacker models into security review processes provides significant opportunity for applying the formal risk assessment methods of security research to integrated attacker models.

4.4.5 Security Properties

While several attacker models use *security properties* to some other end such as verifying system behavior or detecting and mitigating attacks, few produce security properties as a result of the attack. In [72], Mo demonstrates an attacker model that formally derives security properties and countermeasures in conclusion to evaluation of the attacker behavior. Similarly, Basin in [74] demonstrates a formal methodology for deriving security properties through a rigorous attacker model.

4.5 Attacker Model Characterization Examples and Findings

In order to demonstrate the capability of the AMC, a diverse set of attacker models was selected and an AMC was created for each using the attributes described in this chapter. The tabular documentation scheme we use can be seen in Table 4.1 for Adepu's attacker model in [57]. This table captures the entire set of attribute information and effectively represents the AMC for Adepu's AM. In total, we develop the AMCs for eleven diverse attacker models which can be seen in Appendix A. For each attacker model, the AMC was demonstrated to capture the information necessary to define the attributes of the input interface, contextual interface, and output interface.

The attribute documentation format of the set of an indicator and a description lends itself well to evaluation at differing levels of abstraction. For more detail on a specific AM, the attribute descriptions provide a useful summary for how the various attributes are realized in the attacker model. For higher levels of abstraction, the indicator can allow rapid visualization of large amounts of data. For example, Table 4.2 effectively presents the attribute indicator values for the contextual, input, and output interfaces for all eleven AMCs. This provides a high-level analysis of what attributes are realized across the spectrum of the attacker models documented.

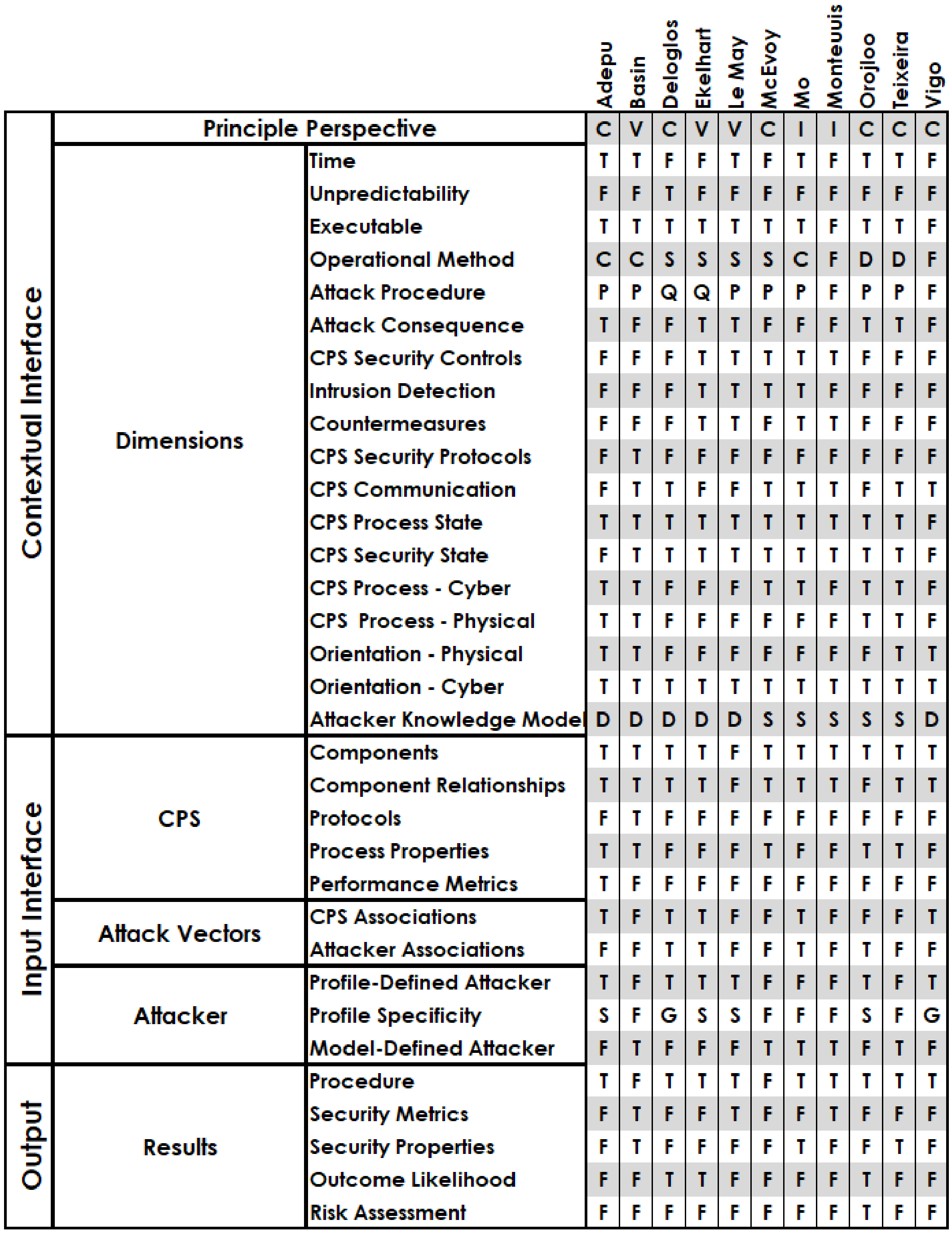
4.6 Security Review Process Characterization

The value in having an AMC or a database of AMCs is realized when it can be associated with a security review process. In order to associate an SRP with an AMC, the SRP must be described in the context of the various attributes in the AMC. This is done by creating a security review process characterization (SRPC), which has the same attributes as the AMC but is evaluated from a different perspective. Where the AMC is describing the attributes included in the attacker model, the SRPC describes how the various attacker model attributes influence their compatibility and/or value to the SRP. This can fundamentally be observed by the different questions asked when completing an AMC versus when completing an SRPC. Where the AMC answers the same question for each attribute in each interface, namely, "Does the Attacker Model consider this attribute?", the question for each interface of the SRPC is different. For attributes in the input interface, the SRPC answers the question, "Is the SRP able to provide the information required for this attribute to an attacker model?". For attributes in the contextual interface, the SRPC answers the question, "Does the SRP recognize the concept of this attribute as being valid?". For attributes in the output interface, the SRPC answers the question, "What resulting information does the SRP want from the attacker

Table 4.1 The attacker model characterization for the attacker model developed by Adepu et al. in [57]

	Category	Attribute	Ind	Adepu Description
Contextual Interface	Objective			Creates a generalized attacker and attack model for Cyber Physical Systems.
	Principle Perspective		C	Begins with the CPS architecture, associates attack vectors to components, then associates attack procedures to the attacker model.
	Dimensions	Time	T	Models the CPS in continuous time. Characterizes an attack as successful based on the requirement that all intents are realized in a certain amount of time.
		Unpredictability	F	
		Executable	T	Demonstrates the execution of the attacker model performing an attack on a secure water treatment testbed.
		Operational Method	C	Models the CPS, the attacker, and the attacker-CPS relationship in continuous time.
		Attack Procedure	P	Attacks are modeled by the changes they affect in the system state. For all cases except attacker reconnaissance, realization of attacker intent is marked by the occurrence of an invalid system state.
		Attack Consequence	T	Leverages defined system properties to deduce attack consequences in the form of identifying invalid system states.
		CPS Security Controls	F	
		Intrusion Detection	F	
		Countermeasures	F	
		CPS Security Protocols	F	
		CPS Communication	F	
		CPS Process State	T	While the CPS and attacker are modeled in continuous time, the system state is captured as a sequence of system states generated by periodically sampling the continuous domain.
		CPS Security State	F	
		CPS Process - Cyber	T	Leverages system properties (Ex. Water pH, Water Conductivity, etc….) and system performance metrics to model system state progression capturing cyber system processes.
		CPS Process - Physical	T	Leverages system properties (Ex. Water pH, Water Conductivity, etc….) and system performance metrics to model system state progression capturing physical system processes.
		Orientation - Physical	T	Considers the physical orientations of components by observing the physics of relationships in order to capture the effects of physical interactions (Ex. Water flowing between devices through a pipe).
		Orientation - Cyber	T	Considers the cyber-relationships between components with cyber attributes, capturing communication-based relationships.
		Attacker Knowledge Model	D	Model claims compatibility with both static and dynamic attacker knowledge models, but only demonstrates a static attacker knowledge model.
Input Interface	CPS	Components	T	Represents CPS as an abstract domain model consisting of components, process properties, and performance metrics.
		Component Relationships	T	Implied but not explicitly defined. Assumes two devices may be connected or not connected, establishing a definite attack propagation medium. Does not consider connection type.
		Protocols	F	
		Process Properties	T	Represents CPS process as an abstract domain model consisting of components, process properties, and performance metrics.
		Performance Metrics	T	Represents CPS as an abstract domain model consisting of components, process properties, and performance metrics.
	Attack Vectors	CPS Associations	T	Attack vectors are associated to CPS components through attack procedures, which are fed into the attacker model as input.
		Attacker Associations	F	
	Attacker	Profile-Defined Attacker	T	Uses a profiling scheme that captures the intents of the attacker as profile properties (ex. Damage, learn, alter).
		Profile Specificity	G	Profiles attacker as a set of intents.
		Model-Defined Attacker	F	
Output	Results	Procedure	T	Attacker model models the attack procedure performed by an attacker against system components producing a list of successful attack outcomes with attack descriptions including attack timing diagrams.
		Security Metrics	F	
		Security Properties	F	
		Outcome Likelihood	F	
		Risk Assessment	F	

Table 4.2 The attribute indicators summarized for the eleven attacker models in Appendix A



			Adepu	Basin	Deloglos	Ekelhart	Le May	McEvoy	Mo	Monteuuis	Orojloo	Teixeira	Vigo
Contextual Interface	Principle Perspective		C	V	C	V	V	C	I	I	C	C	C
	Dimensions	Time	T	T	F	F	T	F	T	F	T	T	F
		Unpredictability	F	F	T	F	F	F	F	F	F	F	F
		Executable	T	T	T	T	T	T	T	F	T	T	F
		Operational Method	C	C	S	S	S	S	C	F	D	D	F
		Attack Procedure	P	P	Q	Q	P	P	P	F	P	P	F
		Attack Consequence	T	F	F	T	T	F	F	F	T	T	F
		CPS Security Controls	F	F	F	T	T	T	T	T	F	F	F
		Intrusion Detection	F	F	F	T	T	T	T	F	F	F	F
		Countermeasures	F	F	F	T	T	F	T	T	F	F	F
		CPS Security Protocols	F	T	F	F	F	F	F	F	F	F	F
		CPS Communication	F	T	T	F	F	T	T	T	F	T	T
		CPS Process State	T	T	T	T	T	T	T	T	T	T	F
		CPS Security State	F	T	T	T	T	T	T	T	T	T	F
		CPS Process - Cyber	T	T	F	F	F	T	T	F	T	T	F
		CPS Process - Physical	T	T	F	F	F	F	F	F	T	T	F
		Orientation - Physical	T	T	F	F	F	F	F	F	F	T	T
		Orientation - Cyber	T	T	T	T	T	T	T	T	T	T	T
		Attacker Knowledge Model	D	D	D	D	D	S	S	S	S	S	D
Input Interface	CPS	Components	T	T	T	T	F	T	T	T	T	T	T
		Component Relationships	T	T	T	T	F	T	T	T	F	T	T
		Protocols	F	T	F	F	F	F	F	F	F	F	F
		Process Properties	T	T	F	F	F	T	F	F	T	T	F
		Performance Metrics	T	F	F	F	F	F	F	F	F	F	F
	Attack Vectors	CPS Associations	T	F	T	T	F	F	T	F	F	F	T
		Attacker Associations	F	F	T	T	F	F	T	F	T	F	F
	Attacker	Profile-Defined Attacker	T	F	T	T	T	F	F	F	T	F	T
		Profile Specificity	S	F	G	S	S	F	F	F	S	F	G
		Model-Defined Attacker	F	T	F	F	F	T	T	T	F	T	F
Output	Results	Procedure	T	F	T	T	T	F	T	T	T	T	T
		Security Metrics	F	T	F	F	T	F	F	T	F	F	F
		Security Properties	F	T	F	F	F	F	T	F	F	T	F
		Outcome Likelihood	F	F	T	T	F	F	F	F	T	F	F
		Risk Assessment	F	F	F	F	F	F	F	F	T	F	F

General:{T=True, F=False}, Principle Perspective:{C=System Component Perspective, V=Vulnerability Perspective, I=Individual Attack Perspective}, Operational Method:{S=Sequential, D=Discrete Time, C=Continuous Time}, Attack Procedure:{P=Process, Q=Qualification}, Attacker Knowledge Model:{S=Static, D=Dynamic} model?". The process of evaluating the compatibility and utility of a particular AM to an SRP then is done by comparing the SRPC to the AMC. The structured integration workflow for this process is described in Chapter 6.

Chapter Five

Attacker Model Functional Representation (AMFR)

In Chapter 3 we gave an overview of our Attacker Modeling Framework and identified that it is composed of two primary modules, which we call the Attacker Model Characterization (AMC), and the Attacker Model Functional Representation (AMFR). In Chapter 4 we discussed in depth the AMC. In this chapter, we describe the attacker model functional representation (AMFR). The purpose of the AMFR is to capture the functional basis of how the attacker model works. Where the AMC describes the various attributes of the attacker model, the AMFR captures at a high level how those attributes are realized.

Attacker models vary drastically in form and function, which makes developing a single functional representation framework to describe this diverse set a challenge. In Chapter 4 we reviewed several attacker models which demonstrate a diverse range of functional definition. These models ranged in complexity from high-level methods of reasoning about attacker behavior [56] to rigorously formalized AMs composed from *Pi*-calculus [62] or described by system-theoretic models [63]. These models also range in form from AMs that model the attacker behavior through a single sequence of steps [57] to AMs that model attacker behavior as a feedback loop [5]. Describing this range of attacker models requires the AMFR be defined at a certain level of abstraction that can capture the functional representation of AMs without manifesting structure or syntax that precludes the description of other types of AMs. As such, the AMFR we develop is intended to serve as an intermediary between the abstract description provided by the AMC and the full level of detail that would be required for complete implementation of the AM.

5.1 Composition of the Attacker Model Functional Representation

Traditional attacker behavior is captured as a series of observations regarding the attacker's motivations and the attacker's decision process. This is captured in the proposed AMFR as a meta-model

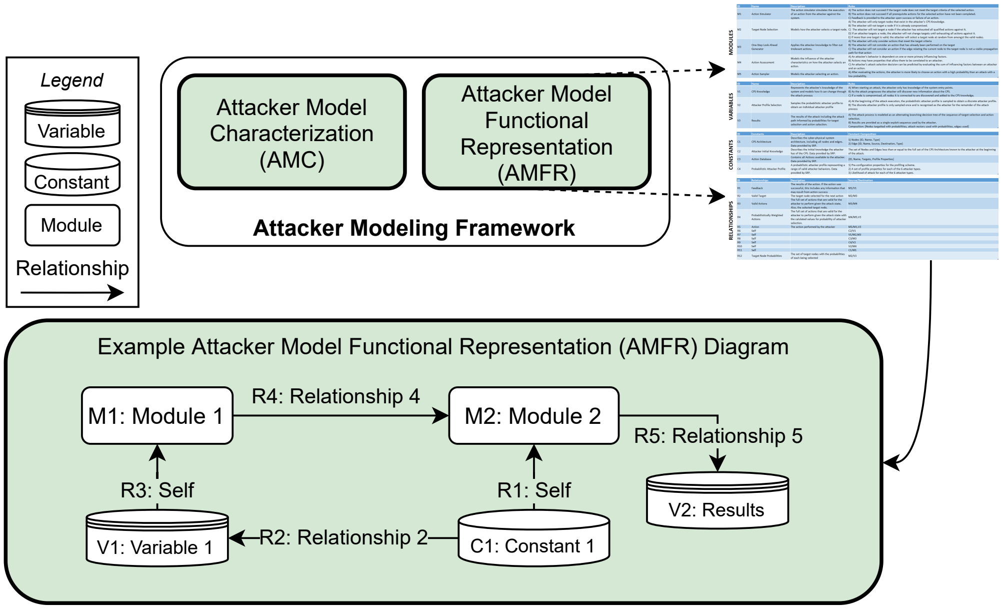


Figure 5.1 An example attacker model functional representation and its diagram with relationships between hypothetical modules "Module 1" and "Module 2" that utilize a variable "Variable 1" and a constant "Constant 1" and have a resulting variable "Results".

composed of modules, constants, and variables, all connected by relationships that characterize the flow of information where components are defined as being producers and/or consumers of data. These components together describe the functional behavior of the attacker model. As seen in Figure 5.1, the AMFR is developed as a tabular document describing the various AMFR components for the attacker model. This document may be used to generate a graphical representation of the AMFR which is seen as an example AMFR diagram.

5.1.1 A Word on Scope

We adopt the concept of scope to describe the level of decomposition at which the components of the AMFR (modules, constants, variables, and relationships) capture the functional behavior of the attacker model. The diversity of attacker models makes it difficult to define an appropriate scope for modules, constants, variables, and relationships that does not inhibit the description of other AMs created at a higher or lower level of abstraction. As such, for each component we define objective-driven principles that provide guidance for deciding the scope of the various AMFR components.

5.1.2 Modules

The decision process of the AM is captured in the proposed AMFR in several modules, each being composed of one or more rules. A module captures a bounded component of the functionality of the AM. A rule may be defined as a facet of an attacker's behavior which observes a cause/effect relationship between an influencing parameter and the attacker's actions. The proposed AMFR accepts a series of rules in each module, which together compose the functional representation of the decision process of the unique attacker model.

A module represents a functional component of the attacker model that takes input data, performs one or more operations, and then produces a result. Decomposing the functional implementation of an attacker model into the components of an AMFR necessitates the identification of boundaries around functionality that follow this input/operation/output module structure. The motivation for module boundary selection is to partition an attacker model into the fewest number of functional steps possible that effectively captures how the primary functional mechanisms in the attacker model leverage the input data initially provided to the attacker model to produce the output results produced by the attacker model. It is a common practice in attacker modeling to capture the functionality of the attacker model as an algorithm, flowchart, or other type of process diagram. These representations often serve as a good starting point for defining modules where operative steps in flowcharts or operations in an algorithm identify bounded functionality which can be partitioned into a single module. Each attacker model studied in the literature survey could be clearly represented with four to twelve modules per attacker model. Examples of modules are provided in the AMFR case studies in Sections 5.2 and 5.3.

This module-based methodology allows the representation of a wide variety of attacker behaviors from a diverse set of AM structures. In the case of AM development, this provides a test-bed to explore the role and influence of individual modules on the composite attacker decision process. This also provides a flexible framework that allows validation of the AMFR against a particular data-set where the rules that the AMFR implements may be refined and calibrated to achieve a model that accurately reflects the behavior of a known attacker or set of attackers. Modules are documented in the AMFR as the tuple of an ID, a title, a short description of the purpose of the module, and a set of rules defining the behavior of the module.

Example Module and Rule

When considering an attacker model that simulates an attacker performing an attack on a system, a critical behavior of the attacker will be the process by which the attacker selects the next target. In an AMFR, this may be represented as the module, "Target Selection Process". This module may be composed of several rules including, but not limited to, qualification rules such as "Attacker will not select a target it has not discovered", capability rules such as "Attacker will not select a target if the attacker has no exploits/resources for attacking that target", and preferential rules such as "Attacker will prefer targets of higher value". This would be captured in the AMFR as, {M1, "Target Selection Process", "Rule 1:...", "Rule 2:...", "Rule 3:..."}.

5.1.3 Constants

The first data type in the AMFR is the constant. A constant is any static datum or set of data that is not subject to change through the duration of the attack. The data in a constant is created before the execution of the attacker model. A constant is not capable of receiving data from variables or modules, but may provide data to either. A constant is intended to capture a collection of data with consistent or mostly consistent structure at the highest level of abstraction possible that does not include operative functionality.

Constants are documented in the AMFR as the tuple of an ID, a title, a short description of the data held in the constant, and a description of the expected origin of the data. The data for constants can be produced in one of two ways. First, it can be developed prior to the execution of the attacker model as the result of a preparatory step defined in the AM. Second, it can be provided from the SRP to the AM as input information which is provided prior to the execution of the attacker model. In the case of data provided by the SRP, the AMFR constant should be documented with as much detail as possible regarding the structure of the data. This is applied during the integration where the data provided from the SRP is mapped to the data accepted by the AM.

Example Constants

Consider a module that captures the attacker's process of selecting a target node as a Target Selection Process. In order to execute, this module may require several pieces of information such as the architecture of the CPS or a list of available actions. Many attacker models take as input a description of the CPS architecture and do not modify it through the attack process. Such a CPS architecture would be considered a constant as it is not subject to change. Similarly, if the AM treats the set of actions available to the attacker as a predefined list, it is considered a constant. This would be captured as, {C1, "Attack Actions", "Contains all actions available to the attacker. Data provided by SRP.", {ID, Name, Targets, Profile Properties}} where the ID, Name, Targets, and Profile Properties fields represent the structure of data expected for each action in the database. However, if the attacker is modeled as having the ability to learn new actions as the attack progresses, then the list of available actions would not be a constant, but rather is considered a variable.

5.1.4 Variables

Variables are the second type data in the AMFR and represent any data that is subject to change throughout the course of the attack. The distinction between static constants and dynamic variables in attacker models has significant implications on how the attacker model must use that data. For a constant, an AMFR must only define the structure of the data. For a variable the AMFR must define the structure of the data, as well as the process by which that data may change. We define variables as being able to implement a set of rules that describe how input data modifies the internal data of the variable. Variables are capable of taking input that is used to modify the internal data and are capable of providing output to other variables or modules. As such, variables are documented in the AMFR as the tuple of an ID, a title, a description of the data, and one or more rules defining how the data in the variable is manipulated.

Example Variables

Consider the example module "Target Selection Process". This module may require as input data such as a list of nodes that have already been targeted. The list of nodes that have been targeted, however, changes as the attack progresses and is therefore classified as a variable. This may be represented as the set, {V1, "Targeted Node List", "A list of nodes already targeted by the attacker", "Rule 1: If the attacker targets a node, the attacker becomes aware of the node, and it is added to this list."}.

5.1.5 Relationship

Relationships in the AMFR follow a node-edge format and serve to capture the flow of data between constants, variables, and modules in the AMFR. Relationships are described in the AMFR as the tuple of an ID, a title, a description of the data that is passed, the data source, and the data destination(s). In the case of relationships that originate from constants or variables, we define the relationship description of *self*, which may be used to represent the relationship as the provision of the full set of data in the originating constant or variable.

Example Relationship

Consider the example module "Target Selection Process" which takes as input a list of nodes in the target system. The architecture would likely be provided from a "CPS Architecture" constant as a subset of the constant's data. The relationship would be represented as that from the constant to the Target Selection Process module. This may be represented as, {R1, "CPS Node List", "A list of all nodes the CPS", C1/M1 } where C1 is the CPS Architecture constant's ID and M1 is the Target Selection Process Module's ID.

5.1.6 Results

The results of the attacker model are captured by the AMFR using the same semantics as a variable. The Results variable receives data via relationships and produces the output data from the AM, which is provided to the SRP. Similar to other variables, the Results variable is documented as the tuple of an ID, the title "Results", a description of the results captured, and a set of rules implemented to produce the output data. In addition, the results variable should capture in as much detail as possible the structure of the resulting data. This is used during the integration when mapping the result data from the AM to the results data expected by the SRP.

Example Result

Consider an attacker model that provides as output the expected attack path of the attacker. This could be captured as, {V3, "Results", "The results of the attack including the attack path", "Rule 1:...", "Rule 2:..."}.

5.1.7 AMFR Diagrams

The AMFR can be visually represented as a diagram where the symbols for modules, constants, variables, and relationships can be seen in Figure 5.1. The purpose of the diagram is to clearly provide a visual representation of the flow of data and process in the attacker model. In the instance of a module, variable, or constant with a number of relationships that inhibits clarity in the diagram, we adopt the relationships syntax of {<Source ID>, <Relationship ID>, <Destination ID>} which can be added to a component in the diagram in place of an arrow.

Relationships between modules are characterized by connected arrows, where upstream modules occur earlier in the attacker decision process. The flow of data between constants, variables, and modules is captured using relationships as seen in the example AMFR diagram in Figure 5.1. This diagram shows the relationships between modules M_1 and M_2, constant C_1, and variable V_1 and V_2, as R_1, R_2, R_3, R_4, and R_5. P_1 is a static constant which is passed entirely to M_2 and partially to V_1, where the subset of P_1 passed to V_1 is captured in the description of R_2. Variable V_1 captures the data from P_1 and applies via an internal rule or set of rules, then passes the entirety of its data to module M_1. Module M_1 performs operations on the received data in accordance with the rules it implements, then passes the data specified in R_4 to module M_2. Module M_2 performs an operation and passed the resulting data to V_2 via R_5 which is the final step in the process. R_5 may provide the data as output, or else implement a rule or set of rules that processes the data into its final form. While the structure of the AMFR is simple in form, it is powerful in being able to capture the functional representation of a wide diversity of attacker models.

5.1.8 How the AMC Informs the AMFR

While the AMC and the AMFR are two functionally separate modules and can be applied independently to the attacker model to achieve their respective goals, a completed AMC can be used to inform the AMFR in the definition of constants, variables, and modules. Because the attributes in the AMC only represent the subset of attributes which are relevant to integration with an SRP, an AMC alone does not provide enough information to create the full AMFR. The attributes in the various interfaces can, however, be useful in identifying several constants, variables, and modules.

Attributes in the input interface of the AMC identify the information expected by the attacker model as input, which may indicate constants in the AMFR. Attributes in the contextual interface of the AMC, particularly those of the Attacker/CPS Interaction attribute category, may indicate mechanisms in the attacker model that can be directly captured as modules or decomposed into modules. Attributes in the AMC output interface may identify the output variable of the AMFR. Therefore, while the AMFR can be developed through direct analysis of the attacker model, the characterization of the attacker model in the AMC aids by capturing a subset of those modules.

5.2 AMFR Case Study: Adepu's Attacker Model [57]

Adepu et al. developed a generalized attacker and attack model targeting cyber-physical systems. Adepu's model is designed to model a diverse set of system architectures with varying performance metrics and system properties under attack by a unique attacker with varying intents, start and end goals, and capabilities. Rather than providing a generalized description of Adepu's AM which may or may not be clear and may or may not capture the important information, we created the AMC for Adepu's attacker model which can be seen in Table 5.1 and effectively describes all the attributes about Adepu's model that are critical to our goal of integration with a security review process.

From Adepu's model we generated the AMFR. The most useful approach to reading attacker modeling literature and generating an AMFR will likely change from model to model. We hypothesize that the information to compose the AMFR can most easily be gathered by identifying which AMFR component type is most clearly defined in the literature, describing all instances of that component type, and then progressing to the next most clearly defined component type.

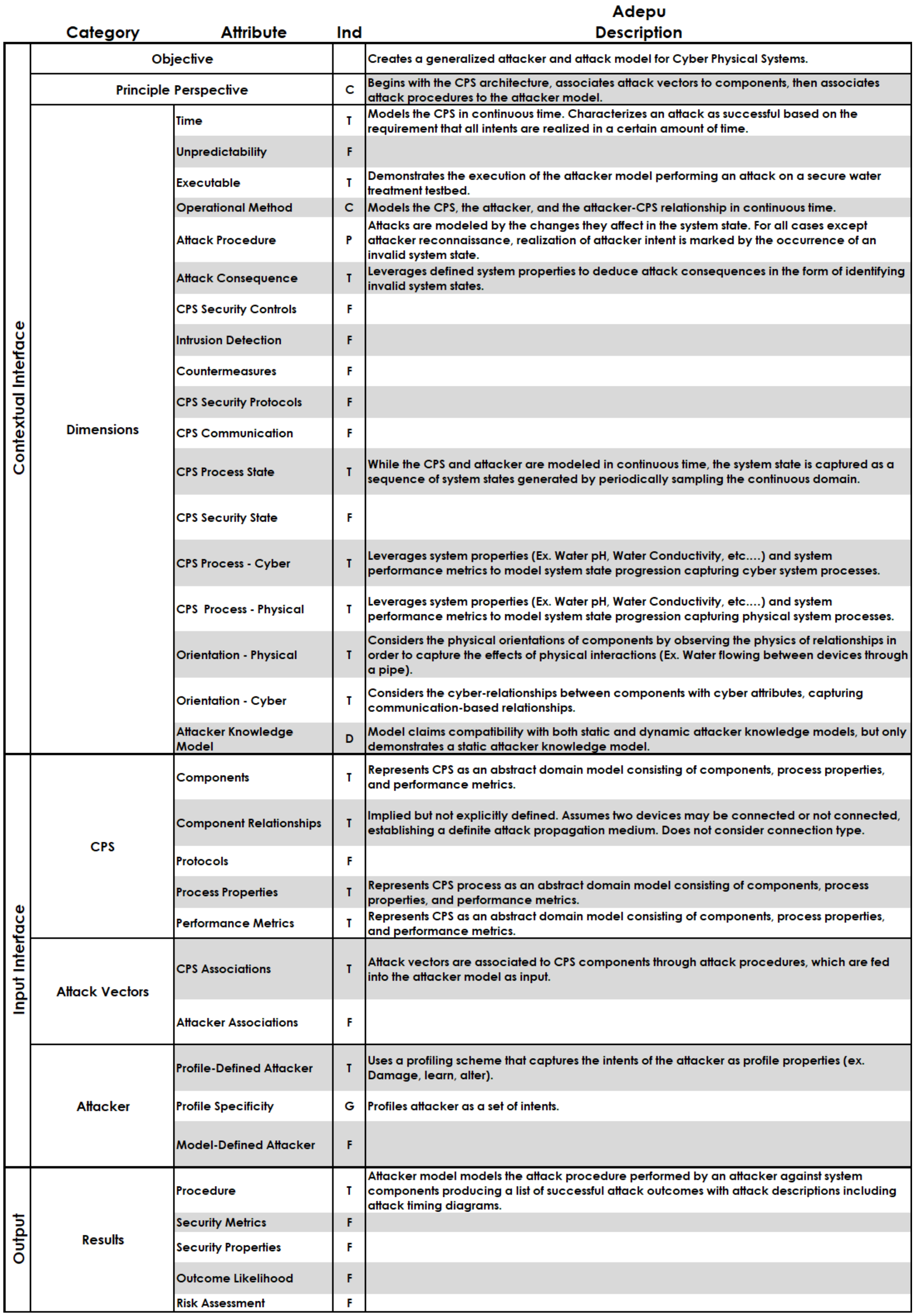
The workflow we found most helpful in generating the AMFR for Adepu's AM was the following:

1. Define all modules (Name and Description)
2. Define all variables (Name and Description)
3. Define all constants (Name and Description)
4. Define all relationships (Name, Description, and Source/Destination)
5. Update all relationships (ID)
6. Update all constants (ID and Constant Composition)
7. Update all variables (ID and Rules)
8. Update all modules (ID and Rules)

In this workflow, returning to the IDs once all components and relationships were defined was simply used to allow us to easily assign ID numbers in the order of progression of the AMFR relationships. While we do not formally define any method for choosing the ID values, we recognize the human tendency to put significance on sequential numbers. Defining our module, property, constant, and relationship ID values as the first letter prefix (M, P, C, or R respectively) followed by the number representing the order of relationship in the AMFR adds clarity to the AMFR diagram.

The AMFR for Adepu's AM can be seen in Table 5.2. Using the AMFR, we were able to effectively decompose Adepu's AM into module-sized components. We were able to identify the input data required by the model as constants, and we were able to capture the results of the model as a variable. All modules, constants, and variables were then associated using relationships. The AMFR diagram was created from the AMFR and can be seen in Figure 5.2.

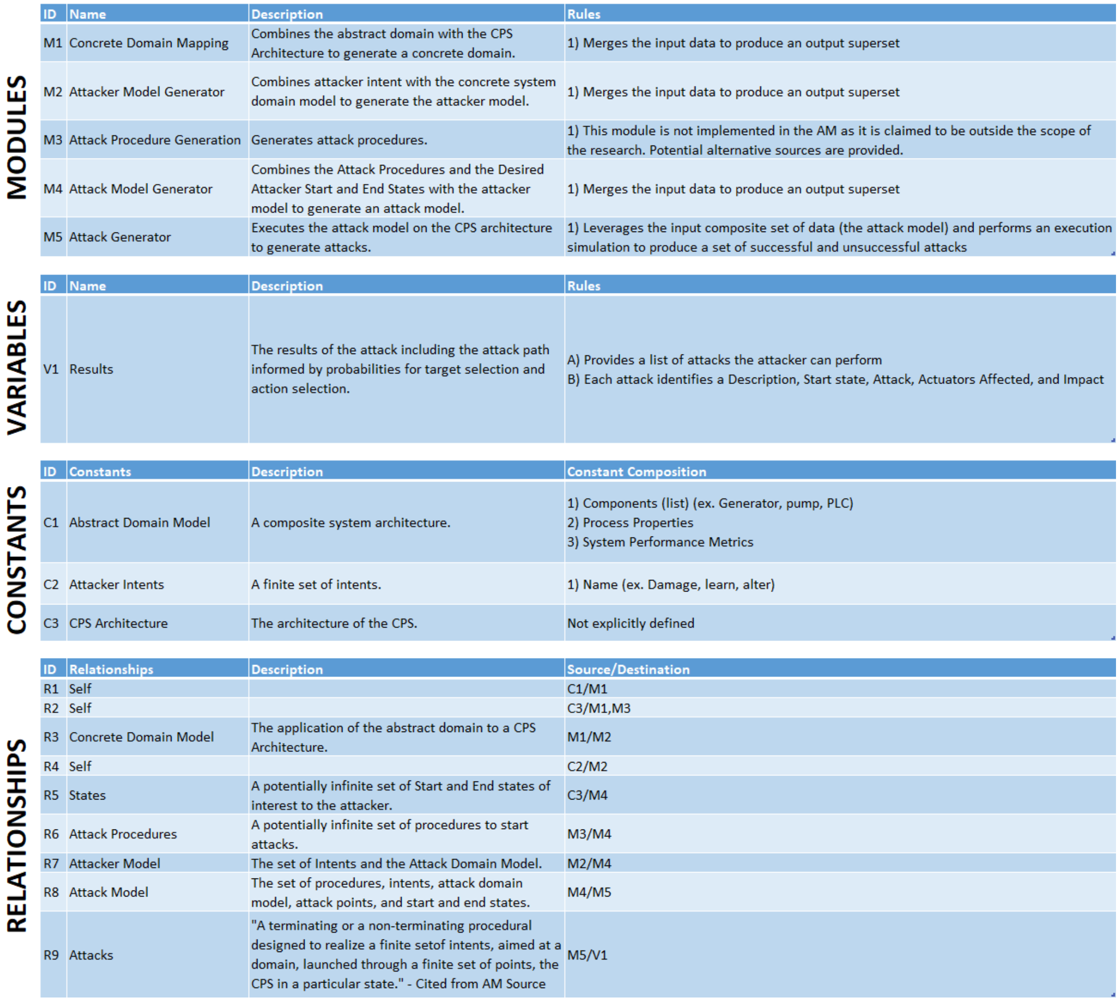
Table 5.1 Tabulated attacker model characterization data for Adepu's attacker model [57]



	Category	Attribute	Ind	Adepu Description
Contextual Interface	Objective			Creates a generalized attacker and attack model for Cyber Physical Systems.
	Principle Perspective		C	Begins with the CPS architecture, associates attack vectors to components, then associates attack procedures to the attacker model.
	Dimensions	Time	T	Models the CPS in continuous time. Characterizes an attack as successful based on the requirement that all intents are realized in a certain amount of time.
		Unpredictability	F	
		Executable	T	Demonstrates the execution of the attacker model performing an attack on a secure water treatment testbed.
		Operational Method	C	Models the CPS, the attacker, and the attacker-CPS relationship in continuous time.
		Attack Procedure	P	Attacks are modeled by the changes they affect in the system state. For all cases except attacker reconnaissance, realization of attacker intent is marked by the occurrence of an invalid system state.
		Attack Consequence	T	Leverages defined system properties to deduce attack consequences in the form of identifying invalid system states.
		CPS Security Controls	F	
		Intrusion Detection	F	
		Countermeasures	F	
		CPS Security Protocols	F	
		CPS Communication	F	
		CPS Process State	T	While the CPS and attacker are modeled in continuous time, the system state is captured as a sequence of system states generated by periodically sampling the continuous domain.
		CPS Security State	F	
		CPS Process - Cyber	T	Leverages system properties (Ex. Water pH, Water Conductivity, etc….) and system performance metrics to model system state progression capturing cyber system processes.
		CPS Process - Physical	T	Leverages system properties (Ex. Water pH, Water Conductivity, etc….) and system performance metrics to model system state progression capturing physical system processes.
		Orientation - Physical	T	Considers the physical orientations of components by observing the physics of relationships in order to capture the effects of physical interactions (Ex. Water flowing between devices through a pipe).
		Orientation - Cyber	T	Considers the cyber-relationships between components with cyber attributes, capturing communication-based relationships.
		Attacker Knowledge Model	D	Model claims compatibility with both static and dynamic attacker knowledge models, but only demonstrates a static attacker knowledge model.
Input Interface	CPS	Components	T	Represents CPS as an abstract domain model consisting of components, process properties, and performance metrics.
		Component Relationships	T	Implied but not explicitly defined. Assumes two devices may be connected or not connected, establishing a definite attack propagation medium. Does not consider connection type.
		Protocols	F	
		Process Properties	T	Represents CPS process as an abstract domain model consisting of components, process properties, and performance metrics.
		Performance Metrics	T	Represents CPS as an abstract domain model consisting of components, process properties, and performance metrics.
	Attack Vectors	CPS Associations	T	Attack vectors are associated to CPS components through attack procedures, which are fed into the attacker model as input.
		Attacker Associations	F	
	Attacker	Profile-Defined Attacker	T	Uses a profiling scheme that captures the intents of the attacker as profile properties (ex. Damage, learn, alter).
		Profile Specificity	G	Profiles attacker as a set of intents.
		Model-Defined Attacker	F	
Output	Results	Procedure	T	Attacker model models the attack procedure performed by an attacker against system components producing a list of successful attack outcomes with attack descriptions including attack timing diagrams.
		Security Metrics	F	
		Security Properties	F	
		Outcome Likelihood	F	
		Risk Assessment	F	

T=True, F=False, System Component Perspective, V=Vulnerability Perspective, I=Individual Attack Perspective, C=Continuous Time, P=Process, D=Dynamic

Table 5.2 Tabulated attacker model functional representation data for Adepu's attacker model [57]



MODULES

ID	Name	Description	Rules
M1	Concrete Domain Mapping	Combines the abstract domain with the CPS Architecture to generate a concrete domain.	1) Merges the input data to produce an output superset
M2	Attacker Model Generator	Combines attacker intent with the concrete system domain model to generate the attacker model.	1) Merges the input data to produce an output superset
M3	Attack Procedure Generation	Generates attack procedures.	1) This module is not implemented in the AM as it is claimed to be outside the scope of the research. Potential alternative sources are provided.
M4	Attack Model Generator	Combines the Attack Procedures and the Desired Attacker Start and End States with the attacker model to generate an attack model.	1) Merges the input data to produce an output superset
M5	Attack Generator	Executes the attack model on the CPS architecture to generate attacks.	1) Leverages the input composite set of data (the attack model) and performs an execution simulation to produce a set of successful and unsuccessful attacks

VARIABLES

ID	Name	Description	Rules
V1	Results	The results of the attack including the attack path informed by probabilities for target selection and action selection.	A) Provides a list of attacks the attacker can perform B) Each attack identifies a Description, Start state, Attack, Actuators Affected, and Impact

CONSTANTS

ID	Constants	Description	Constant Composition
C1	Abstract Domain Model	A composite system architecture.	1) Components (list) (ex. Generator, pump, PLC) 2) Process Properties 3) System Performance Metrics
C2	Attacker Intents	A finite set of intents.	1) Name (ex. Damage, learn, alter)
C3	CPS Architecture	The architecture of the CPS.	Not explicitly defined

RELATIONSHIPS

ID	Relationships	Description	Source/Destination
R1	Self		C1/M1
R2	Self		C3/M1,M3
R3	Concrete Domain Model	The application of the abstract domain to a CPS Architecture.	M1/M2
R4	Self		C2/M2
R5	States	A potentially infinite set of Start and End states of interest to the attacker.	C3/M4
R6	Attack Procedures	A potentially infinite set of procedures to start attacks.	M3/M4
R7	Attacker Model	The set of Intents and the Attack Domain Model.	M2/M4
R8	Attack Model	The set of procedures, intents, attack domain model, attack points, and start and end states.	M4/M5
R9	Attacks	"A terminating or a non-terminating procedural designed to realize a finite setof intents, aimed at a domain, launched through a finite set of points, the CPS in a particular state." - Cited from AM Source	M5/V1

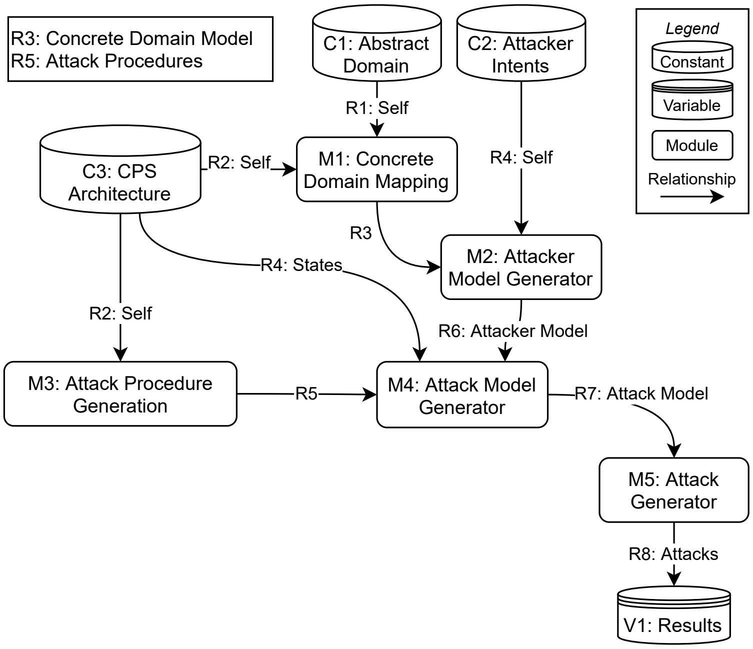


Figure 5.2 The attacker model from Adepu et al. [57] implemented using the attacker model functional representation.

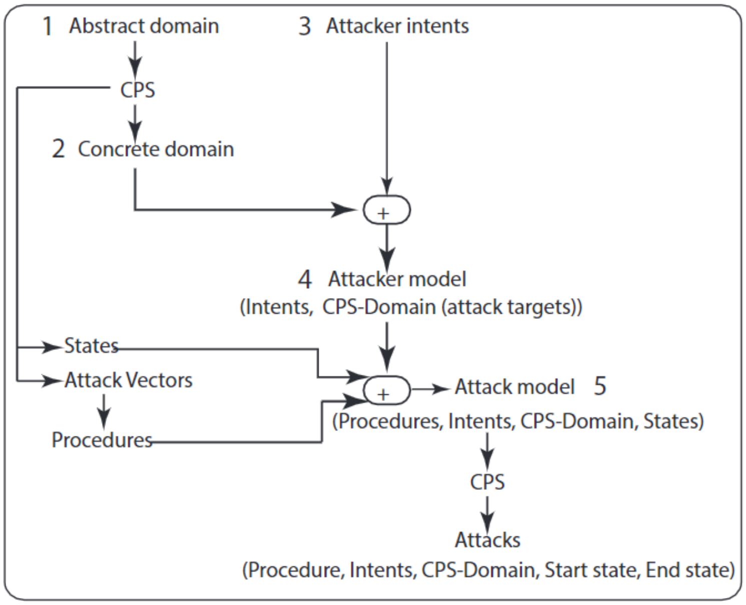

1 Abstract domain
3 Attacker intents
CPS
2 Concrete domain
+
4 Attacker model
(Intents, CPS-Domain (attack targets))
States
Attack Vectors
Procedures
+
Attack model 5
(Procedures, Intents, CPS-Domain, States)
CPS
Attacks
(Procedure, Intents, CPS-Domain, Start state, End state)

Figure 5.3 The 5-stage process for deriving attacks for a CPS using Adepu's attacker model (Image taken from [57]).

5.2.1 Conclusions of Adepu's AMFR

Adepu's attacker model was created as a 5-stage process and in the publication [57] Adepu created Figure 5.3 which captures this process as a diagram. We notice a key fundamental difference between Adepu's process diagram and the AMFR diagram that would make an AMFR diagram a valuable addition in Adepu's publication. First, Adepu's diagram, as is common in attacker modeling literature, was developed around a theoretical perspective. As such, the abstraction of the diagram may be conducive to interpreting the theory being applied, but it is not particularly useful for functional interpretation. In order to collect the necessary information to trace the data flow through the process, we had to read through the entire document, taking notes of the set notation used to characterize the various data elements, then compose that data into the constants, variables, and relationships in the AMFR. While Adepu labeled several of the data elements in the diagram, the functional components that relate them are not clear from the diagram and must be interpreted from set notation. To an expert in the same domain, this documentation scheme is expected and appropriate. When considering an expert in applied cyber-security who may not be familiar with the theoretical semantics and notation in Adepu's publication, the effort to functionally interpret this attacker model could be inhibiting. We propose that in a publication such as Adepu's, the functional implementation of the attacker model could be readily captured by including our tabulated AMFR and AMFR diagram in an appendix.

5.3 AMFR Case Study: Our Own Attacker Model

The early research of this dissertation was formative in understanding attacker models and how to design, create, and apply them. In the course of this discovery, we created our own attacker model to explore certain aspects of probabilistic attacker behavioral prediction. While the attacker model was not comprehensive, as most are not, the probabilistic behavior was considered novel and published in [5]. Our AM serves as a good case study to explore the more meticulous aspects documenting the AMFR modules, variables, constants, and relationships. As such, in this section we walk through the development of our attacker model in parallel with the process of creating the AMFR for the attacker model. In order to clearly distinguish the AMFR from the attacker model itself, we note that the AMFR in its entirety can be seen in Table B.2 in Appendix B and the AMFR diagram created from the table can be seen in Figure 5.7 at the conclusion of this section. Throughout the case study, in order to clarify the distinction between the AMFR and the attacker model, the attacker model is referred to in the third-person as Deloglos' attacker model. We also note that the variables and syntax used for the formulation of the attacker model are not included in the table of variables for this dissertation. The full documentation of the attacker model variables can be found in the original publication in [5].

The attacker model created by Deloglos is a probabilistic attacker model that leverages attacker profiling to predict the behavior of a non-deterministic attacker. The attacker model functional representation in Figure 5.7 depicts the relationship between the attacker and the CPS as a feedback loop which cycles through an attacker target selection and action selection process, iterating step by step through the decision process used to perform the attack. Rules defining attacker behavior are implemented in modules such as the CPS Knowledge and the Target Node Selection modules.

In Deloglos' cyclical action/feedback scheme, an action is some step the attacker performs in the attack process and feedback is any information the attacker receives as a result of the action. The results of the attack on the CPS is captured as the progression of the attack state. The attack state represents all data in a single cycle of the AM which is a tuple of the selected action as well as all static constants and dynamic variables in the AMFR.

5.3.1 Cyber-Physical System Architecture

In Deloglos' attacker model, the CPS is modeled and described as a simplified composition of nodes, edges, attack vectors, and entry points. A node represents a machine or other potentially vulnerable device that has functional purpose within the CPS. An edge represents a communication link between two nodes that may be used to transmit information, while an attack vector is any edge that may be used for an attack. An entry point is an edge directed into the CPS from outside the CPS that may be used by an attacker to gain access to the system. Establishing well-formed boundaries between nodes of the CPS and relationships between them allows the composite description of a Cyber-Physical System. Many programming tools and languages such as SysML [78] utilize the node/edge system description scheme and may be readily integrated with the CPS design process for attacker model automation.

5.3.2 Action Simulator

The action simulator is implemented using a qualification-based process. It takes as input the action being performed by the attacker as R_5 and the CPS architecture and produces as feedback the result of the action as R_1. The action simulator implements the following rules for action success evaluation.

- The action does not succeed if the target node does not meet the target criteria of the selected action.
- The action does not succeed if all prerequisite actions for the selected action have not been completed.
- Feedback is provided to the attacker upon success or failure of an action.

The feedback from the action simulator includes action success or failure, as well as any descriptive information associated with action success.

5.3.3 CPS Knowledge

Capturing the nature of how an attacker learns about a CPS as an attack progresses is often ignored in attacker models where an underlying assumption is made that the attacker is aware of (or has vision of) the full system description. In evaluating attacker behavior Deloglos begins by making the assumption that as the attack or probing progresses, the attacker will begin to learn information about the target system. This behavior is captured in the CPS knowledge module which takes as input the action feedback as R_1 and the attacker initial knowledge from P_2. The rules that the CPS knowledge module implements are:

- When starting an attack, the attacker only has knowledge of the system entry points.
- As the attack progresses the attacker will discover new information about the CPS.
- If a node is compromised, all nodes it is connected to are discovered and added to the CPS knowledge.

Information about the CPS is fed into the attacker's CPS knowledge module as feedback. If a node is compromised it is considered owned by the attacker and capable of performing pivoting attacks. The initial attacker CPS knowledge is simplified for demonstration purposes in the preliminary research, but in a more complex application could include behaviors involving the attacker's discovery of the target system.

5.3.4 Target Node Selection

When the attacker goes to perform an action against a system, the attacker must first select a target node. The target node selection module takes as input the full data set of the CPS knowledge variable and produces a set of valid targets as R_2. The rules that the target node selection module implements are:

- The attacker will only target nodes that exist in the attacker's CPS Knowledge.
- The attacker will not target a node if it is already compromised.
- The attacker will not target a node if all qualified actions have been exhausted.
- If an attacker targets a node, the attacker will not change targets until all actions against it are exhausted.
- If more than one target is valid, the attacker will select a target node at random from amongst the valid nodes.

5.3.5 Action Database

The Action Database property contains the set of all actions available to the attacker. Each action within the database contains several fields of information including the action profile, the action description, the target criteria, and a list of prerequisite actions. The action profile contains a quantitative description of the user and use case of the action, which is used for quantifying a relationship between each action and the attacker profile defined for the attacker model as discussed further in section 5.3.9. The action description is a plain-text description of how the action works in as much detail as is possible. The target criteria defines what system(s) the action is valid against. The prerequisite attacks describe any actions that must be completed before this action may be attempted.

A critical component to the viability of the attacker model is the action database population scheme. CAPEC [39], CWE[40], CVE[41], and CPE [42] are amongst the most popular vulnerability databases and provide different approaches to cataloging attacks, attack descriptions, and attack relationships. Search engines that make use of online attack and vulnerability databases aid in effectively generating an action database for the attacker model. One tool that Deloglos applied to populate the action database when demonstrating his AM was the CYBOK tool [38], which is a literal search engine for CAPEC, CWE, and CVE capable of generating vulnerability data for individual queries or entire systems.

5.3.6 One-Step Look-Ahead Generator

The one-step look-ahead generator applies the attacker's knowledge of the CPS to filter out all attacks that are invalid for the current attack state. Filters are non-probabilistic in nature and may depend on any information regarding the current state of the attack or the description of the node. This module takes as input the full set of the attacker's CPS knowledge from $V1$ as well as the selected target from M_2. This attacker model applies three filters as rules and produces as output R_3 which is the full set of actions that are valid given the target node and the state of the attack.

1. The attacker will only consider actions that meet the target criteria
2. The attacker will not consider an action that has already been performed on the target
3. The attacker will not consider an action if the edge relating the current node to the target node is not a viable propagation path for that action

Deloglos defines $\mathcal{A}$ as the set of all m known actions in the action database and $\Phi \subseteq \mathcal{A}$ as the set of actions known by the attacker. The three filters are defined as $\Phi_{target} \subseteq \Phi$, $\Phi_{ex} \subseteq \Phi$, and $\Phi_{vect} \subseteq \Phi$ for filters 1, 2, and 3 respectively. The set of actions that are valid for the attacker to perform in the given state of the attack (Φ_{valid}) are then defined by Equation 5.1, where the attack space can

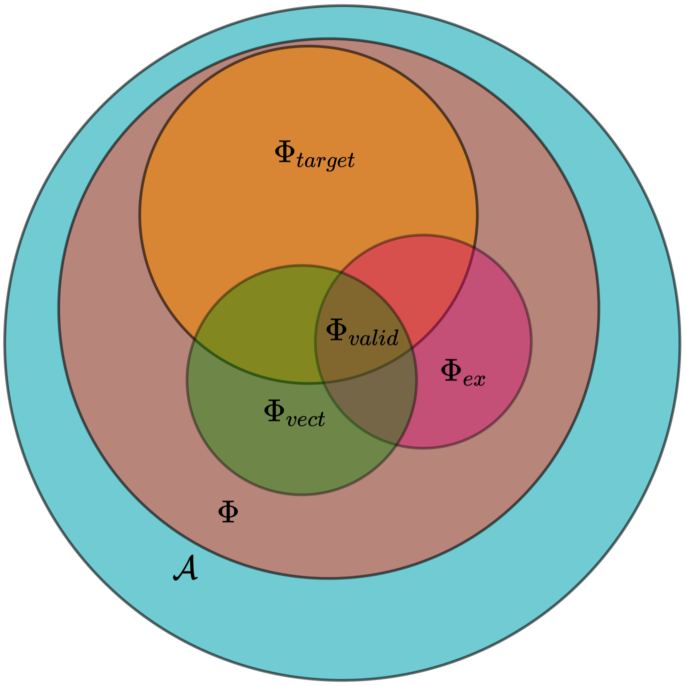


Figure 5.4 The intersection of action selection filters applied to the action database.

be visualized in Figure 5.4.

$$\Phi_{valid} = \Phi_{ex} \cap \Phi_{vect} \cap \Phi_{target} \tag{5.1}$$

5.3.7 Probabilistic Attacker Profile

Attacker profiles are a topic well covered in literature with no recognized standards for what characteristics best model an attacker. The purpose of an attacker profile is to capture characteristics about an attacker that influence the attacker's behavior, thereby describing the expected behavior of the attacker. The characteristics that define the attacker profile are termed attacker properties. Rocchetto et al. [64] performed a literature review on attacker profiles for CPSs in an attempt to find a unifying attacker profiling model to describe various attackers from multiple different research studies. In conclusion, Rocchetto proposed a set of six attacker profiles composed of twenty-nine attacker properties that effectively described the majority of attacker profiles in the referenced literature.

In applying Rocchetto's attacker profiles to an attacker model it is important to note that in a real-world application one cannot assume which attacker will be attacking a system. In order to simulate this non-deterministic behavior, two types of attacker profiles are adopted which are the static attacker profile and the probabilistic attacker profile. A static attacker profile represents one of the six attacker profiles defined by Rocchetto et al. [64]. A probabilistic attacker profile may be represented as a probability mass function (PMF) of the six profiles. The PMF is generated by assigning each of the six attacker profiles ($\Delta_1, \ldots, \Delta_6$) a likelihood of attacking (l_i) such that $0 \leq l_i \leq 1$.The Probability of attack of a specific attacker profile is calculated using:

$$P(\Delta_i) = \frac{l_i}{\sum_{j=1}^{n} l_j} \tag{5.2}$$

where $\sum_{j=1}^{n} P(\Delta_j) = 1$. The PMF in Figure 5.6 is an example probabilistic attacker profile designed to mimic the probability of attackers against a nuclear power plant.

5.3.8 Attacker Profile Selection

The Attacker Profile variable V_2 takes as input the probabilistic attacker profile data from P_4 and uses it to non-deterministically select an attacker profile. This is the mechanism used by Deloglos to simulate the lack of ability to deterministically predict the nature of an attacker. The rules implemented by V_2 are as follows.

- At the beginning of the attack execution, the probabilistic attacker profile is sampled to obtain a discrete attacker profile.
- The discrete attacker profile is only sampled once and is recognized as the attacker for the remainder of the attack process.

5.3.9 Action Assessment

The influence of the attacker characteristics on how the attacker selects an action is a behavior captured in the action assessment module, $M4$. The action assessment model takes as input R_3 from M_3 which contains the full set of actions valid for the attacker to perform given the attack state, as well as the description of the selected target node. The rules applied here are the following.

- Actions may have properties that allow them to be correlated to an attacker.
- An attacker's attack selection decision can be predicted by evaluating the sum of influencing factors between an attacker and an action.

The action assessment module calculates the probability of the attacker performing each of the actions based on probability functions that take as operands the attacker profile, the attack profile, and the current state of the attack.

Action Profiles

An action profile is often represented as a set of properties describing the characteristics of the action [64]. This, however, implies linear proportionality to an attacker profile, which is not universally true. For example, an attacker with a high skill set is not necessarily more likely to perform an

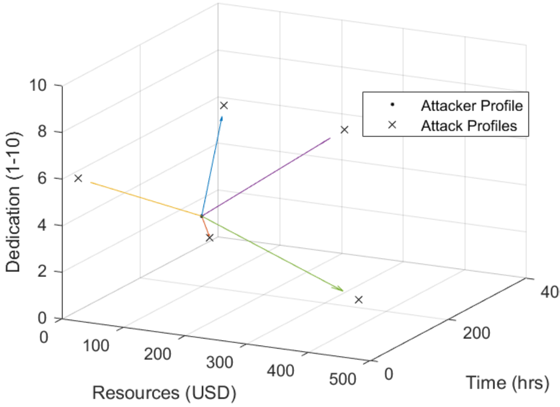


Figure 5.5 An example three-dimensional attack space showing the attacker profile and several action profiles.

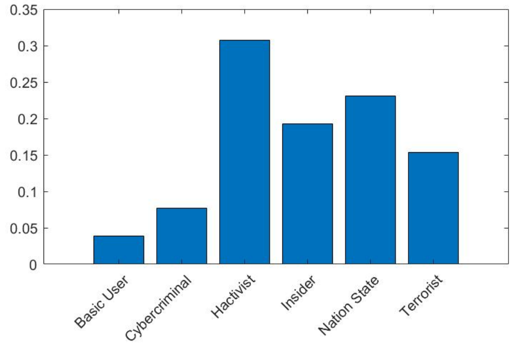


Figure 5.6 An example of a probability mass function for a probabilistic attacker profile against a nuclear power plant.

attack that requires a high skill set when an easier attack may succeed as well. Deloglos captures this behavior by defining an attack profile as the profile of the attacker expected to use that attack. Because attacker behavior is constantly changing as technology evolves, this profiling technique may be reinforced by empirical data from records of attack history. Collaborations such as MITRE's ATT&CK framework [79] may aid in the assessment of current threat actors. This facilitates an attacker model that can better emulate realistic and relevant threats by allowing the user to base the relationship between attackers and their actions off of current attacker data.

As such, Deloglos defines an attacker profile (Δ) as an n-dimensional space of attacker properties (δ_i) such that $\Delta = \{\delta_1, \delta_2, \ldots, \delta_n\}$ for an attacker profile having n properties. An example attack space can be seen in Figure 5.5 where the attacker profile and several action profiles are plotted in the 3-dimensional space. The probability of an attacker performing an action is a function of the distance between the attacker profile and the action profile in n-dimensional space.

Attack Probability Functions

The probability that the attacker will perform an attack at any given time is calculated using the attack probability function. Attacker properties may be one of three types which are sets, bounded ranges, and unbounded ranges. Non-ordered sets are considered to have a scaled property value $\gamma = 1$ if the attacker profile property and the attack profile property match and $\gamma = 0$ otherwise. Ordered set values may be mapped to the scaled property range ($0 \leq \gamma \leq 1$) using fuzzy set theory as demonstrated by Patil et al. in [80].

Bounded ranges are numerical ranges where the value of a property (ε) may only fall between a lower bound (ε_L) and an upper bound (ε_H). Bounded ranges are linearly mapped to the scaled property value ($\gamma : 0 \leq \gamma \leq 1$) using:

$$\gamma = \frac{\varepsilon - \varepsilon_L}{\varepsilon_H - \varepsilon_L} \tag{5.3}$$

Several scaling functions exist for unbounded ranges such as the percent-difference function, the logistic function, and the hyperbolic tangent. The value weighting in these functions, however, is non-linear, which does not properly scale different property values where a score considered median is represented by a numerically large or numerically small value (>100 or <1 respectively). Therefore, Deloglos proposes converting the unbounded property values to a bounded range by first evaluating the maximum (γ_{max}) and minimum (γ_{min}) values for all actions within the database, then using the local maximum and minimum to scale the unbounded range.

Deloglos designates the set of m available actions in the action database as $\mathcal{A} = \{A_1, A_2, \ldots, A_m\}$. Each action A_i ($i = 1, 2, \ldots, m$) has an associated set of scaled property values $\Gamma_i = \{\gamma_i^1, \gamma_i^2, \ldots, \gamma_i^n\}$. For a given attacker profile (Δ) with n scaled property values $\Theta = \{\theta_1, \theta_2, \ldots, \theta_n\}$, the distance ($d_i$) between the attacker and each action is calculated by the distance between the two profiles in n-dimensional space using:

$$d_i = f(\Theta; \Gamma_i) = \sqrt{\sum_{j=1}^{n} \frac{1}{\beta_j^2} \left(\theta_j - \gamma_i^j\right)^2} \tag{5.4}$$

where β_j is a criticality factor such that $\{\beta \in \mathbb{R} | 0 \leq \beta \leq 1\}$ which increases the distance for properties with a $\beta < 1$ criticality. The score of each action (s_i) is inversely proportional to d_i and calculated using the function:

$$s_i = 1 - \frac{d_i}{\sum_{j=1}^{m} d_j} \qquad i = 1, \ldots, m \tag{5.5}$$

This equation is unique in that it calculates the inverse of the distance without applying a nonlinear value-weighting as is observed in the inverse function or exponential functions such as the Softmax function. According to the score for each action, the probability that the attacker will take action A_i is calculated using the function:

$$P[A_i] = \frac{s_i}{\sum_{j=1}^{m} s_j} \tag{5.6}$$

Equation (5.6) has the intuitive interpretation that the higher the score the attacker gets for an action, the higher the probability that this action will be chosen by the attacker.

5.3.10 Action Sampler

The last module in the attacker model is the action sampler module. The action sampler module implements the following rule.

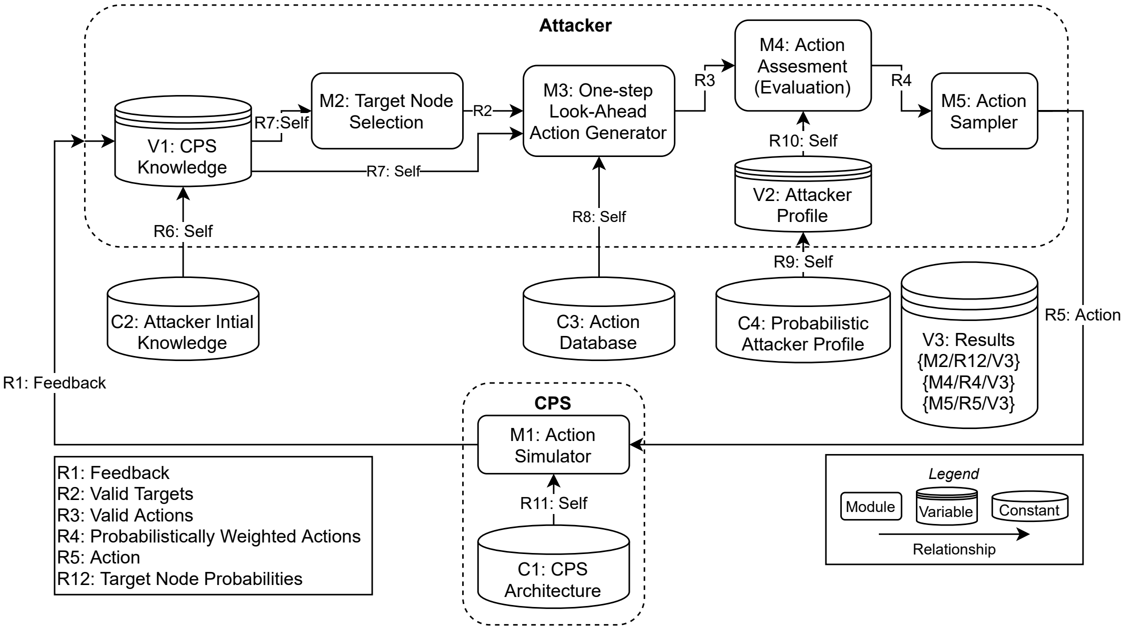


Figure 5.7 Attacker model from Deloglos et al. [5] implemented using the attacker model functional representation.

- After evaluating the actions, the attacker is more likely to choose an action with a high probability than an attack with a low probability.

The action sampler receives as input R_4 which contains all attacks for the target system with their attack probability values and selects one of the actions by sampling a weighted randomizing function ($rand_w()$) mapped to the probabilities of the set of probabilistic actions $\Delta = \{A_i, P[A_i]\}$. This action is then performed by the attacker against the CPS.

5.3.11 Results

The results of the attacker model are captured as a partially observable Markov Model which uses the probabilities of target selection and action selection, along with the decision of the target and action selected to represent the attack path chosen by the attacker. The rule that the Results variable implements is as follows.

- The attack process is modeled as an alternating branching decision tree of the sequence of target selection and action selection.

5.4 Attacker Model Findings

All modules, properties, variables, and relationships necessary to compose the attacker model functional representation were described in the previous section. In order to distinguish the AMFR

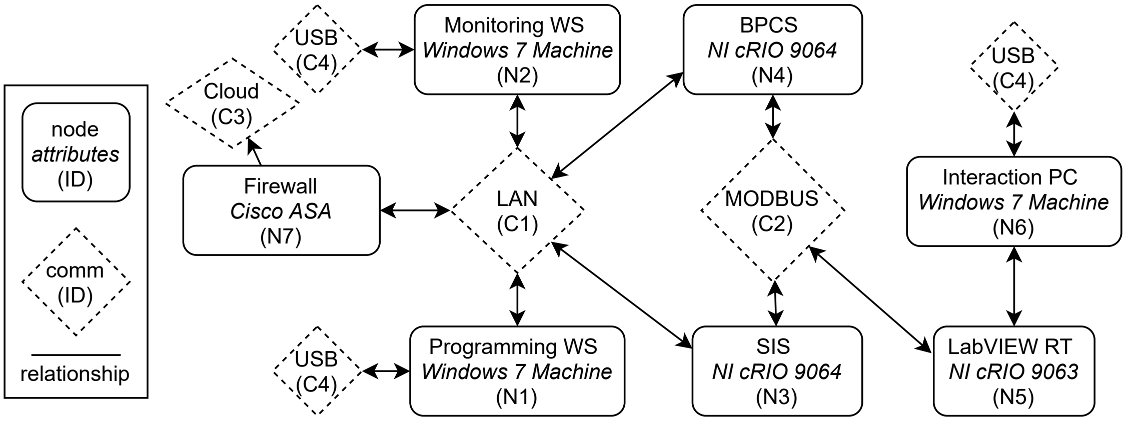


Figure 5.8 Attacker model case study ICS relational diagram.

from the AM in itself, the complete AMFR is captured in Appendix B in Table B.2. The complete diagram for the Deloglos' AMFR can be seen in Figure B.2.

For application of the preliminary Attacker Model, Deloglos demonstrated an attack on the Industrial Control System (ICS) in Figure 5.8 which is composed of nodes and communication channels. This example ICS is used to control a simulated exothermic continuous stirred tank reactor (CSTR) using an NI cRIO controller. The target for the attack is the Basic Process Control System (BPCS, N4). Control or disruption of the BPCS by the attacker indicates a successful attack.

5.4.1 ICS Architecture

The ICS consists of the 7 nodes in Table 5.5, each composed of key attributes included in Figure 5.8. The system is described as having 4 entry points in Table 5.6 which include N1, N2, and N6 via infected USB and N7 via remote access. Six properties are selected as a subset of those described by Rocchetto et al. [64] to describe the attacker and action profiles which include Access, Finances, Knowledge, Manpower, Motivation, and Tools. Access, Motivation, and Tools are defined as set properties with values of {Direct, Wireless, Offsite} for Access and {Low, Medium, High} for Motivation and for Tools. Knowledge is defined as a bounded property with a $0 \leq$ Knowledge ≤ 10 range. Finances and Manpower are defined as unbounded properties. These properties are not intended to be a holistic description of the attacker behavior, but rather to demonstrate the principles and dynamics of the different types of profile properties. The criticality factor is kept at unity (1) for all profile properties. The attacker profile PMF in Figure 5.6 was defined as a set of 6 attacker profiles with property values in Table 5.3. CAPEC, CWE, CVE, and CPE databases were used to search for vulnerability information. The CAPEC and CWE databases were used to identify potential attack patterns and weaknesses respectively, aiding in the discovery of associated

Table 5.3 Attacker profiles and property values

Profile	Access	Finances	Knowledge	Manpower	Motivation	Tools
Basic User	Offsite	100	2	40	Low	Low
Cybercriminal	Offsite	1000	5	160	Medium	High
Hactivist	Wireless	500	6	1500	High	Medium
Insider	Onsite	100	7	10	Medium	Medium
Nation State	Offsite	1000000	9	100000	High	High
Terrorist	Onsite	10000	4	1000	High	Medium

Table 5.4 Case study action profiles

ID	Name	Targets	A	F	K	Mp	Mo	T
V1	Remote-Access Trojan	Windows 7 Machine	Offsite	0	3	20	Low	Low
V2	CVE-2017-2779	Windows 7 Machine	Onsite	10000	9	5000	High	Mid
V3	CVE-2017-2775	Windows 7 Machine	Offsite	6000	10	8000	High	Mid
V4	MODBUS MITM	NI cRIO 9064/9063	Onsite	50000	9	500	High	Mid
V5	MODBUS DOS	NI cRIO 9064/9063	Offsite	40000	6	200	High	Mid
V6	Code Injection	NI cRIO 9064	Onsite	100	4	300	Mid	Low
V7	Watering-Hole	Windows 7 Machine	Offsite	2000	6	300	Mid	Mid
V8	CVE-2014-4115	Windows 7 Machine	Onsite	1200	7	800	High	High
V9	CVE-2010-2568	Windows 7 Machine	Onsite	10000	8	2000	High	High
V10	CVE-2019-1713	Cisco ASA	Offsite	5000	9	100	High	Mid

A=Access, F=Finances, K=Knowledge, Mp=Manpower, Mo=Motivation, T=Tools

CVEs. Table 5.4 contains a sample profile set for the vulnerabilities found for the ICS nodes. In Section 5.4 the variable V represents vulnerability IDs. The variable is reclaimed in later chapters.

5.4.2 Executing the Attacker Model

The following sections detail a single cycle of the attack/feedback process.

Sampling the attacker profile

Sampling the attacker PMF at the beginning of the attack process determines the profile of the attacker performing this attack. This is done by mapping a random function to the probability values of each attacker type. In this instance, the attacker selected is the Nation State.

Table 5.5 Case study nodes

ID	Name	Type
N1	Programming WS	Windows 7 Machine
N2	Monitoring WS	Windows 7 Machine
N3	SIS	NI cRIO 9064
N4	BPCS	NI cRIO 9064
N5	RT Simulation	NI cRIO 9063
N6	Interaction PC	Windows 7 Machine
N7	Firewall	Cisco ASA

Table 5.6 Case study entry points

ID	Node	Mechanism
E1	N1	Infected USB
E2	N2	Infected USB
E3	N6	Infected USB
E4	N7	Remote Access

Table 5.7 Case study step 1 calculated values against node N1

ID	Distance	Score	Probability
V1	2.088	0.6428	0.3214
V7	2.0199	0.6545	0.3272
V8	1.7383	0.7027	0.3513

Target node selection

The attack process in Figure B.2 begins with the evaluation of the Attacker CPS Knowledge and subsequent selection of a node. In the beginning, it is presumed that the attacker has knowledge of components with entry vectors which include N1, N2, N6, and N7. The rule used for node selection in this case study is random. Applying a probabilistic random selection the attacker selects node N2 as the first target.

One-step look-ahead action generation

The one-step look-ahead process evaluates the feasibility of each action based on the set of rules for the attacker model. The rules for this module filter out attacks that do not apply to the node type or have already been performed on the target node. These rules pass V1, V7, and V8 through the one-step look-ahead action generation module.

Action Assessment

The action assessment module applies the attack probability functions to the set of Actions V1, V2, and V8. The probability values for the step, along with intermediate distance and score values, can be observed in Table 5.7.

Action Sampler

The action sampler then selects one of the vulnerabilities at random in accordance with the probability of each action. In this case, the sampler selected V8, which is effectively the attack that the attacker performs.

Feedback

Action V8 has no conditional qualifications for success and therefore results in a successful attack against node N2. Node N2 is now compromised and all communication paths connected to node

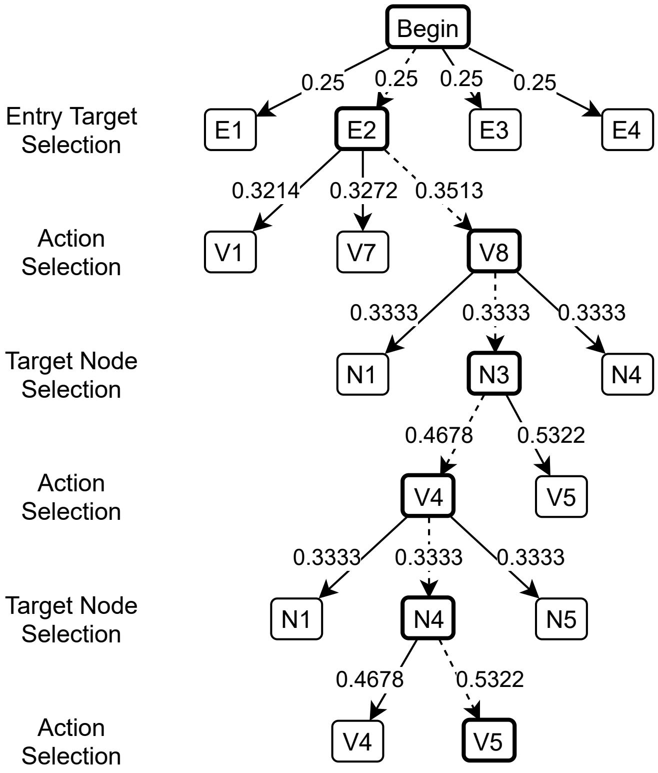


Figure 5.9 Diagram of attacker CPS knowledge upon completion of the attack, including attack progression.

N2 are known. In this attacker model, the nodes connected to N2 are considered known as well. Therefore, information added to the CPS Knowledge model includes the existence of the LAN communication network, the existence of nodes N3 and N4, and LAN communication relationships between N1, N2, N3, N4, and N7.

Attack Progression

The attack cycle is repeated until either the target is reached or there are no actions remaining for the attacker to perform. Figure 5.9 shows the progression of the attack as a POMDP, including each decision the attacker made in the attacker process and the probability of each decision.

5.4.3 Case Study Findings

The steps taken to complete the attack in Figure 5.9 represent one of many possible attacks that may have been performed by the attacker. The attacker was able to compromise the CPS by exploiting three vulnerabilities. Step 1 used an infected USB thumb-drive to gain access to the monitoring workstation. Step 2 used a MODBUS man-in-the-middle attack to take over the SIS cRIO. Step 3 used a MODBUS DOS attack to disrupt the operation of the BPCS.

Chapter Six

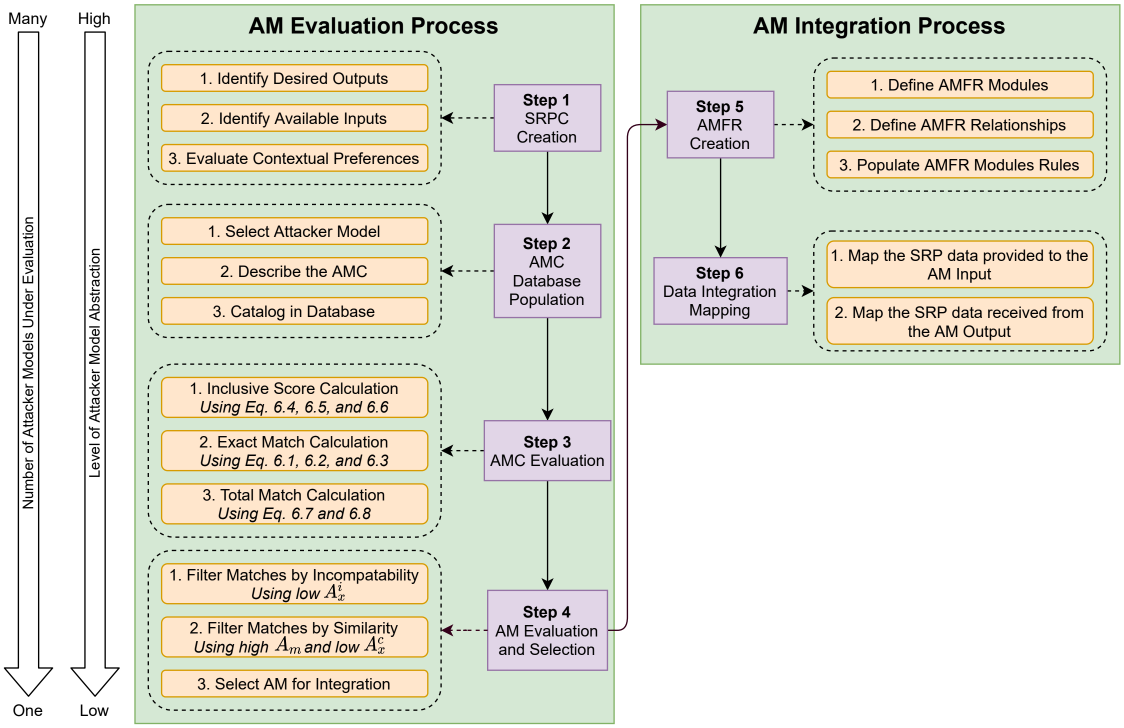
Integrating the Attacker Model Into Security Review Process

One of the primary deliverable of this dissertation is the design of a method for integrating the attacker model perspective into a security review process. In Chapter 3, we identified attacker behavior analysis shortcomings in security review processes and highlighted the potential value of integrating attacker models. In Chapter 4 we presented the AMC, which is a standardized way of describing attacker models, including the input information they require, the output results they provide, their core operating principles, and the contextual assumptions they make. And in Chapter 5 we presented the AMFR, which is a standardized way of representing the functional implementation of attacker models in order to describe at a high-level how an attacker model leverages input information to ultimately produce results. In this chapter, we present the integration workflow, which leverages the AMC and AMFR to integrate attacker models into security review processes.

The integration workflow is a structured process which guides the evaluation of one or more attacker models for integration with a security review process, identifying the compatibility and value of each, then guides the integration of a single selected attacker model into the security review process. The workflow is captured in Figure 6.1 and is divided into two primary parts, first the evaluation process, and then the integration process.

6.1 Attribute Selection Motivation

Perhaps the most critically motivating factor in the valuation of an AM is the functionality of the AM and the features it implements. In published literature, attacker models are rarely comprehensive and usually simplify a significant portion of the modeling process to focus on development of a particular feature. While we provide insights on the assumptions and implications of the AM attributes in Chapter 4, we recognize that the value of an attribute is largely dependent on the



Many
High
Number of Attacker Models Under Evaluation
Level of Attacker Model Abstraction
One
Low
AM Evaluation Process
1. Identify Desired Outputs
2. Identify Available Inputs
3. Evaluate Contextual Preferences
Step 1 SRPC Creation
1. Select Attacker Model
2. Describe the AMC
3. Catalog in Database
Step 2 AMC Database Population
1. Inclusive Score Calculation Using Eq. 6.4, 6.5, and 6.6
2. Exact Match Calculation Using Eq. 6.1, 6.2, and 6.3
3. Total Match Calculation Using Eq. 6.7 and 6.8
Step 3 AMC Evaluation
1. Filter Matches by Incompatability Using low A_x^i
2. Filter Matches by Similarity Using high A_m and low A_x^c
3. Select AM for Integration
Step 4 AM Evaluation and Selection
AM Integration Process
Step 5 AMFR Creation
1. Define AMFR Modules
2. Define AMFR Relationships
3. Populate AMFR Modules Rules
Step 6 Data Integration Mapping
1. Map the SRP data provided to the AM Input
2. Map the SRP data received from the AM Output

Figure 6.1 The attacker model integration workflow.

context of application of the SRP and the assumptions made by the SRP. As such, the AMF defines the value of an AM based upon the definition and requirements of the SRP.

We define the motivating criteria for AM selection as the combination of AM compatibility and AM value. AM compatibility is driven by the agreement between information requirements of both the AM and the SRP. This is fully described in Section 6.2.3, but in summary involves the SRP providing the information required by the AM (system architecture descriptions, attack vectors, etc...) and the AM providing the results required by the SRP (attack procedures, risk analyses, etc...). Incompatibility is applied in the AM integration process as a metric for removing AMs from consideration.

The primary factor that influences AM value is the effort required for integration, which represents the amount of manual labor required by the security engineer performing the integration procedure. Attributes that agree between the AM and the SRP are said to have a low labor cost and therefore a high value. In Section 6.2.3 we define a process for remediation of incompatibilities through modification of the AM and/or SRP. Attributes requiring remediation may have a higher labor cost due to the effort involved with modifying the AM and/or the SRP.

6.2 Evaluation Process

The evaluation process accounts for the first four steps of the workflow. At a high level, the evaluation process compares the security review process characterization to the attacker model characterization to evaluate the compatibility between them and the utility that the particular attacker model provides to the SRP. The three interfaces defined by the AMC (the input, output, and contextual interfaces) are at the core of this evaluation. The input interface is used to evaluate if the SRP has available, or can produce, the information that the AM requires to function. The output interface is used to evaluate if the results produced by the AM are valuable to the SRP. Finally, the contextual interface is used to evaluate if the contextual assumptions made by the AM are in agreement with those made by the SRP, as well as to identify neutral assumptions that may not influence the compatibility but should be considered and documented nonetheless.

The four steps of the evaluation workflow in Figure 6.1 can be summarized as follows. In step 1, the SRPC defined in Section 4.6 is created for the security review process. In step 2, the AMC is created for each attacker model under consideration for integration. If more than one attacker model is being evaluated, the AMCs are composed into a database. In step 3, the SRPC is compared to each AMC to produce compatibility scoring metrics as well as scores that inform how valuable the AM is to the SRP. Finally, in step 4 the scoring values are used to evaluate the compatibility of each attacker model and how well each attacker model aligns with the requirements of the SRP. This process includes an evaluation of incompatibility remediation through modification of the SRP or AM. Step 4 concludes with the selection of an attacker model for integration. Each of these steps is described in detail in the following sections.

The evaluation process leverages the information available in the AMC attributes to perform each step of the evaluation at the highest level of abstraction possible. As the evaluation progresses, attacker models that are incompatible or provide a low value to the SRP are filtered out, reducing the number of attacker model integration candidates. As seen in Figure 6.1, as the level of abstraction decreases - and therefore the amount of detailed analysis required for each attacker model increases - the number of attacker models decreases as well. This progression of abstraction drastically reduces the burden of integration by allowing the expert performing the integration to not be required to perform an in-depth dive to the full set of attacker models, but rather only those which are compatible and of considerable value to the SRP.

6.2.1 Creating the Security Review Process Characterization

The first step in the integration workflow is to create a description of the security review process that can be used to compare to the description of the attacker model. The SRPC is described in Section 4.6 and the creation of the SRPC is done by evaluating each attribute in the three AMC interfaces and answering for each the question posited in Section 4.6. A good understanding of the SRP is required to answer the questions for each of the attributes and it may require additional effort to answer questions about attributes that the SRP does not explicitly address.

6.2.2 Creating the Attacker Model Characterization

The second step in the integration workflow is to create an AMC (defined in Chapter 4) for each attacker model in consideration. This results in a pool or database of AMCs that can be evaluated against the SRPC. While the AMF has considerable value in guiding the evaluation of a single attacker model, its true value is realized in its ability to easily scale to evaluations of large numbers of attacker models. The diversity in form and function of different attacker models makes an objective comparison difficult apart from a foundational characterization of AM attributes. The AMC serves is an objective description framework which can be used to evaluate and compare specific attributes of a large number AMs.

This scalability is achieved, in part, by the form of attribute documentation which captures each attribute as a tuple of a short attribute description and either a boolean indicator or a selection indicator, where the boolean indicator is false if the attribute is not realized in the AM or true if it is, the selection indicator is null if the attribute is not realized in the AM or selects from a finite set of options if it is, and the attribute description is a short explanation of how the attribute is realized in the AM. The indicator gives the highest-level perspective, readily allowing a high-level review and comparison of a large number of attributes from multiple attacker models based on attribute inclusion or exclusion. This can be seen Table 4.2 where the table provides the high-level summary all attributes in eleven attacker models for all three AMC interfaces. When considering comparisons of smaller numbers of attributes, the attribute description provides a lower-level perspective, allowing comparison of the methods employed to realize the attribute.

6.2.3 Evaluating the Attacker Model Characterizations

The third step in the integration workflow is to compare the SRPC to the various AMCs to evaluate compatibility and identify which AMC provides the most value to the SRP. We identify three categories of attribute association that are useful in evaluating the compatibility and integration value of the various attacker models. We then derive scoring functions for each of these attribute categories. To derive these equations we define the set of attributes in any given interface of the AMC or SRPC as A, recognizing that the AMC and SRPC will always have identical interface and attribute composition. For any given interface with n attributes, we define A as the set of all attributes such that $A = \{a_1, ..., a_n\}$.

The first category of attributes we identify are matching attributes, which are those attributes which have the same type and description in both the SRPC and the AMC. These attributes indicate similarities between the SRP and the AM and identify functionality that both have in common. These attributes are considered positive for integration as they suggest that the AM and the SRP functionally align and the attribute will not incur a high integration cost. We capture matching attributes as A_m where $A_m \subseteq A$.

We then identify an intermediary value which is the set of interface attributes that do not match as A_x where $A_x \subseteq A$ and $A = A_m \cup A_x$. A mismatched attribute is any attribute that is different between the AMC and the SRPC. The set of mismatched attributes is further partitioned into mismatched compatible attributes (A_x^c) and mismatched incompatible attributes (A_x^i) where $A_m = A_x^i \cup A_x^c$ and therefore $A = A_m \cup A_x^i \cup A_x^c$. For any given attribute, we define compatibility as the fulfillment of the following condition for each respective interface:

- Input Interface: The information specified by the attribute as being required by the AM is produced by the SRP.
- Output Interface: The information specified by the attribute as being required by the SRP is produced from the AM.
- Contextual Interface: The contextual attribute defined by the SRP is realized in the AM.

In effect, attributes that are mismatched may still be compatible if the attribute mismatch results in a surplus of information and functionality rather than a net deficit. An example of this would be an AM that provides additional results that are not required by an SRP. Another example would be an SRP that provides a more detailed system architectural description than is required by the AM. For the contextual interface, this would take the form of the AM having additional contextual attributes to those in the SRP. In these instances the presence of extra information and capability does not hinder the integration, and therefore the compatibility, of the AM and the SRP, but it does identify fundamental differences between the two. These mismatched compatible attributes, in effect, represent the assumptions made by the AM and the SRP that do not align but do not necessarily disqualify the integration.

The final category of attributes, incompatible mismatched attributes, are those attributes which do not match between the AMC and SRPC, and which indicate information required by the AM from the SRP or by the SRP from the AM which is not provided. These attributes indicate capabilities and assumptions of the AM or SRP that depend the form and function of the other and do not align, indicating direct incompatibility and can only be mitigated through modification of the AM or SRP.

Match Score

The scoring function for the matching attributes is captured as the match score. The match score captures the percentage that A_m composes of A. That is, the number of matching attributes relative to the total number of attributes. The purpose of the match score to identify exactly how well the description of the attacker model aligns with the description of the security review process by comparing the indicators of each attribute in the AMC and SRPC, identifying a match if both indicators have the same value else identifying no match. The match score function applies a positive weighting to each attribute that matches between the SRPC and the AMC, and a negative (or zero) weight to each attribute that does not match. The match scoring function is applied to each interface, as well as in summary to all three interfaces. Given the input interface, the contextual interface, and the output interface, each having l, m, and n attributes respectively, the set of attributes in each interface is defined as PI, PC, and PO. The attributes for a given SRPC are defined as PI^{SRP}, PC^{SRP}, and PO^{SRP}, and the attributes for a given AMC are PI^{AM}, PC^{AM}, and PO^{AM}. The score for the match match between a given SRPC and an AMC for the input (SI_{Match}), contextual (SC_{Match}), and output (SO_{Match}) interfaces is a scale of 0-100%.

Attributes are defined as a tuple of a short description and either a boolean indicator or a selection indicator. The boolean indicator has potential values of $TRUE$ and $FALSE$, while the selection indicator can be one of a finite set of options, or $FALSE$. We define three helper functions used to evaluate the attributes, each of which are defined as element-wise functions. The first is the $MATCH(,)$ function which takes two attributes as input and returns a value of $TRUE$ if both input attributes match, and a value of $FALSE$ if both input attributes do not match. The second is the $HAS()$ function which takes one attribute as input and returns a value of $TRUE$ if the attribute does not equal $FALSE$. Finally, we define the $COUNT()$ function as a function that returns the number of attributes in a set. Using these function, we define the match scoring equations for the input, contextual, and output interface attributes as the following.

$$SI_{Match} = \frac{COUNT(MATCH(PI^{AM}, PI^{SRP}))}{l} \tag{6.1}$$

$$SC_{Match} = \frac{COUNT(MATCH(PC^{AM}, PC^{SRP}))}{m} \tag{6.2}$$

$$SO_{Match} = \frac{COUNT(MATCH(PO^{AM}, PO^{SRP}))}{n} \tag{6.3}$$

Inclusive Score

The inclusive scoring function for the input, contextual, and output interfaces are symbolized as $SI_{Inclusive}$, $SC_{Inclusive}$, and $SO_{Inclusive}$ respectively. The inclusive score expands on the matching score by not penalizing attacker models where attribute mismatch is a result of a surplus of information, rather then a deficit. Formulaicaly, this is realized differently for each interface because purpose of the attributes is different between the three interfaces. The inclusive scoring value, in effect, captures the ratio of the number of attributes in the sets A_m and A_x^c relative to the total number of attributes in the set A. The inclusive function for the input interface counts as a match all attributes where the SRPC and AMC attributes are either the same, or where the SRPC has additional input values that the AMC does not require. This function is captured in Equation 6.4,

$$SI_{Inclusive} = \frac{COUNT(HAS(PI^{SRP})\ OR\ MATCH(PI^{SRP}, PI^{AM}))}{l} \tag{6.4}$$

where OR is the respective boolean function that return a value of $TRUE$ or $FALSE$.

The inclusive function for the contextual interface counts as a match all attributes in common between the SRPC and the AMC, as well as all attributes where the AMC has additional contextual characteristics that are not present in the SRPC. The implication behind this inclusive function is that, with regards to contextual attributes, the compatibility of an AM is dependent on the AM agreeing with all the contextual assumptions of an SRP. An AM, however, may bring additional assumptions that do not contradict those of the SRP, but may be either of utility or of no consequence to the SRP. The inclusive matching function is represented as $MC_{Inclusive}$ and can be seen in Equation 6.5.

$$SC_{Inclusive} = \frac{COUNT(HAS(PC^{AM})\ OR\ MATCH(PC^{SRP}, PC^{AM}))}{m} \tag{6.5}$$

Finally, the inclusive function for the output interface counts as a match all attributes the SRPC and the AMC have in common, including attributes that are provided by the AMC but not required by the SRPC. In this function an attacker model is penalized for not providing the results required by the SRP, but is not penalized for providing additional results that the SRP will not utilize. The inclusive matching function for the output interface is captured in Equation 6.6.

$$SO_{Inclusive} = \frac{COUNT(HAS(PO^{AM})\ OR\ MATCH(PO^{SRP}, PO^{AM}))}{n} \tag{6.6}$$

While the score for the set of matching attributes (A_m) is provided directly by the match score, the score for the set of mismatching compatible (A_x^c) and mismatching incompatible (A_x^i) attributes must be derived. For the set A_x^c, the score is calculated as the difference between the inclusive and match scores and applies to all three interfaces as well as to the total scores in the following section. For the set A_x^i, the score is calculated as the percentage of attributes not represented by the inclusive set, which is calculated as the remainder of the inclusive set ($100\% - Score_{Inclusive}$) where the function applies to any of the interface scores as well as the total score in the following section and $Score_{Inclusive}$ is replaced with the respective interface score. These scores are captured formulaically as the following:

$$Score(A_m) = S_{Match} \tag{6.7}$$

$$Score(A_x^c) = S_{Inclusive} - S_{Match} \tag{6.8}$$

$$Score(A_x^i) = 100\% - S_{Inclusive} \tag{6.9}$$

As such, the motivation in applying the attribute scores can be described as follows:

- A_m - A high matching score indicates strong similarity between the form and function of the attacker model and the security review process and is generally considered good.
- A_x^c - A high compatible mismatched score indicates that an attacker model is different in form and function from the security review process (this has no bearing on compatibility).
- A_x^i - A high incompatibility score indicates that the attacker model has significant attribute mismatch that inhibits its compatibility with a security review process and is considered bad.

Total Scoring Functions

The total match score and total inclusive score, S_{Match} and $S_{Inclusive}$ respectively, can be calculated as the average of the matching values for the three interfaces using Equation 6.10 and Equation 6.11 respectively.

$$S_{Match} = \frac{SI_{Match} + SC_{Match} + SO_{Match}}{3} \tag{6.10}$$

$$S_{Inclusive} = \frac{SI_{Inclusive} + SC_{Inclusive} + SO_{Inclusive}}{3} \tag{6.11}$$

Attribute Weighting

In order to preserve the utility and objectivity of the scoring functions, we do not include attribute weightings, but rather evaluate attribute preference later in the workflow. The scoring function are leveraged in the integration workflow as an objective indicator of attribute similarity between attacker models and the SRP, and they are leveraged so as to guide the security expert through the attacker model selection process. During the AM Evaluation and Selection process in Step 4 of the integration workflow, after performing compatibility evaluation, the expert is given the opportunity to apply preference to different attributes to inform the selection of a final attacker model for integration.

6.2.4 Attacker Model Selection Process

The attacker model selection step as seen in Figure 6.1 is a three-step process which involves first filtering out incompatible AMs, secondly filtering out AMs that have low value, and finally selecting an AM for integration from the remaining pool.

Filtering Incompatible AMs

Attribute incompatibility is caused by attributes in the set A_x^i and increases as the number of incompatible attributes increase. The first step for attacker model selection is to use the score for A_x^i to identify and filter our incompatible attacker models. A score of 0% implies that an AM has no incompatibility with the SRP, while an inclusive score of 100% implies that the AM is completely incompatible with the SRP.

A low inclusive score greater than 0% implies only minor incompatibility, which may prompt further investigation on the part of the security engineer performing the integration. In these instances, it may be possible to improve compatibility between the SRP and the AM by considering modification of the AM, the SRP, or both. In these instances, a security engineer evaluates the inclusive matches of the Input, Contextual, and Output interfaces to identify the mismatched attribute(s) and evaluate potential resolution. In the case of the input attribute mismatch, this will require modifying the SRP to make the input information available, or modifying the AM to make the model no longer dependent on that input information. In the case of contextual attribute mismatch, the disagreeing contextual aspects may be removed from the SRP, or the AM may be expanded to account for those contextual aspects. Finally, in the case of output attribute mismatch, the SRP may be modified to not require the mismatched output, or the AM may be evolved to provide the missing output information.

In conclusion, AMs with a 0% score for the incompatible mismatched attribute category (A_x^i) are considered fully compatible and pass the incompatibility filtering step. AMs with a <100% score must be justified by the engineer and pass the filtering step if remediation of the incompatibility can be justified. AMs with a <100% score that cannot be justified are excluded from consideration for the remainder of the evaluation process. We formally capture this in the following equations.

Filtering AMs by Similarity

Once incompatible AMs have been removed, the remaining pool of AMs must be evaluated to identify which is the best match for the security review process. While this process may require more in-depth evaluation of the various attacker models, the scores for the set of mismatched compatible attributes (A_x^c) aid by providing significant context to the integration expert. The score for A_x^c is a measure of the number of attributes that do not cause incompatibility, but are either provided by the SRPC and unused by the AMC or are provided by the AMC and unused by the SRPC. When evaluating AM integration apart from a structured workflow, attributes that enable compatibility or inhibit compatibility such as those in A_m and A_x^i are easier to identify as resolution of those attributes is a functional prerequisite to integration. This scoring method is uniquely valuable as attributes in the set A_x^c are difficulty to identify and often constitute those assumptions made by the AM or the SRP that go unnoticed because they do not inhibit compatibility. This scoring function provides an objective method for identifying those attributes, and therefore the assumptions made by those attributes.

Selecting an AM for integration

Evaluating the merit of an AM for integration with an SRP based on individual attributes represents a significant manual effort. The effect of the previous two filtering steps reduces the burden of effort significantly by eliminating both incompatible and low value AMs from consideration. The remaining AMs, then, undergo a more critical evaluation, where the integration expert evaluates the descriptions of the input, contextual, and output interface attributes, in order to select a single attacker model for integration. If at this stage multiple high-quality attacker models are available that fulfill the requirements of the SRP, then the reviewing expert should perform in-depth evaluation using the attribute descriptions as a reference, and exploring the available literature provided by the authors in order to select a single attacker model for integration.

6.3 Integration Process

Once an attacker model has been selected for integration through the evaluation process, the second part of the workflow focuses on integrating the selected Attacker Model into the SRP. The integration process as seen in Figure 6.1 is a three-step process. In step 5 the AMFR is created for the attacker model. Step 6 characterizes the data provided by and consumed by the SRP. In step 7 the interfaces between the SRPC and the AMC are mapped and the attacker model is integrated into the security review process.

6.3.1 Attacker Model Functional Representation Creation

The AMFR is defined in Chapter 5 and the creation process is demonstrated in the case study which develops the AMFR for Deloglos' AM. This step involves the creation of the full AMFR, including the definition of all modules, variables, constants, relationships, and the rules implemented in the various modules, variables, and relationships. In addition to describing the functional implementation of the AM, the AMFR aids the integration by capturing the functional edges of the AM, identifying where and how information is provided to the AM and where the AM produces results.

6.3.2 Security Review Process Data Mapping

Once the AMFR is created for the AM, it is necessary to identify how data is transferred between the SRP and the AM. The AMFR captures the information required as input to the attacker model from the security review process as input constants. The first part of this step is to identify the source of data from the SRP for each constant of the AMFR. The SRP input data is captured as a tuple of the ID of the AMFR constant the data is provided for, a description of the source of the data in the SRP, and a description of the form of the data being provided.

The AMFR captures the output information provided in conclusion of the attacker model execution to the security review process as a variable. This output data required by the SRP is captured as a tuple of the ID of the AMFR variable the data is provided from, a description of the destination the data is provided to in the SRP, and a description of the form of that the data is being provided in.

6.4 Integration Workflow Conclusion

The conclusion of the integration workflow is a tabulated document identifying the description, form, function, source, and destination of 1) the data that must be provided to the attacker model from the security review process and 2) the data that must be produced from the attacker model to the security review process as the results of the attacker model execution.

The expected next step of applying the now-integrated AM and SRP is to execute the security review process. While the execution steps for every security review process will be unique to that particular review process, practically speaking, this can generally be described as the following steps:

1. Execute the security review process to the point where all data required for the attacker model to execute is available.
2. Extract the information specified in the integration documentation from the security review process and provide it to the documented destinations in the attacker model.
3. Execute the attacker model to completion using the data provided from the security review process.
4. Capture the attacker model results results specified in the integration documentation and provide them to the documented destinations in the security review process.
5. Complete the security review process.

Chapter Seven

Case Study - EPRI's Technical Assesment Methodology

In Chapters 4 and 5 we described the two modules of our attacker modeling framework. In Chapter 6, we described our structured process for integrating attacker models into security review processes using our attacker modeling framework. In order to demonstrate the utility of the attacker modeling framework and the integration workflow, we perform the following integration use case, which explores the integration of an attacker model with the Technical Assessment Methodology (TAM) [1] developed by the Electric Power Research Institute (EPRI).

7.1 TAM Summary

EPRI's Technical Assessment Methodology is a structured security review process used for cyber- and cyber-physical systems. The purpose of the TAM is to facilitate the identification and selection of engineered security control methods and exploit sequence mitigation techniques by characterizing attack surfaces and exploit mechanisms in a CPS. The TAM is part of a larger workflow and integrates into several other EPRI products including EPRI's DRAM (a tool for capturing a comprehensive system description) and EPRI's HAZCAD (a tool for system hazard analysis).

The process of attacker model integration begins by identifying a need for an attacker model, which is often discovered by observing where in the security assessment process the behavior of the attacker is assumed. The workflow of the TAM is partitioned into three primary steps. First, the engineers performing the security review characterize the attack surface and identify exploit sequences. Second, the engineers identify security control methods, which are scored and allocated. Finally, several mitigation and normalization techniques are used to handle residual exploit sequences.

In step one, the characterization of the attack surface is a guided process where engineers follow a guiding workflow to evaluate a system architecture and establish the assessment scope, the asset characteristics, and the attack pathways, which are used to define composite exploit sequences. The attack pathways are identified via a manual effort of observing relationships between components, often using a system diagram, and tracking how different communication protocols relate different physical interfaces. The engineer then applies the attack paths in the context of a finite set of attack scenarios to intuit exploit sequences. The effectiveness of TAM is contingent on the engineer effectively identifying all exploit sequences, as TAM does not claim any assurance for unknown exploit sequences.

A lack of scalability can be a significant limitation in a security review process as the manual effort for completing an SRP can become unrealistic when considered in the scope of a full system architecture. The workflow of the TAM is designed with scalability in mind and aids engineers by decomposing the full assessment process into many smaller steps, each evaluating a narrower scope of the system architecture. While this organizes the assessment process and makes the effort to complete the TAM manageable, the development of exploit sequences still represents a significant manual effort. Attacker models may present a solution to reduce the amount of effort required for AM composition and further improve the scalability of the TAM. It is desirable to offload as much of the exploit sequence composition as possible to the AM.

7.2 Step 1 - SRPC Creation

Step one in the integration process is the creation of the Security Review Process Characterization. This includes evaluation of the various attributes in the three interfaces in accordance with the process in Section 4.6.

7.2.1 Output Interface

We begin by evaluating the output interface to identify what attacker model results may be of use to the TAM. The output interface identifies 5 results, which are Procedures, Security Metrics, Security Properties, Outcome Likelihood, and Risk Assessment. In order to evaluate these, we ask the question, "Do we want the AM to provide these results for the SRP. Considering the integration goal of automating exploit sequence generation, the generation of attack procedures will identify the exploit sequence, and therefore the Procedure result is desirable. We document the Procedure attribute as the tuple {true, Desire the generation of all possible attack exploit sequences}. The remaining 4 results do not contribute to exploit sequence generation and therefore are documented as {false}.

7.2.2 Input Interface

For the input interface, the SRPC identifies all the information available from the SRP that can be provided to the attacker model. This not only requires an understanding of the TAM, but also requires the expected contribution of the AM to be bounded. It was stated at the beginning of the integration that the ideal goal of integrating an attacker model is to automate as much of the exploit sequence generation as possible. The amount of information available in the SRP increases as the exploit sequence generation process progresses. For example, the TAM separates the identification of attack pathways and the composition of exploit sequences into two subsequent steps, both included in the attack sequences generation process. Bounding the role of the attacker model to only the second step means that attack pathways generation will be a preliminary step and attack pathways can be provided to the attacker model as input. Bounding the role of the AM to the whole attacker model generation process, means that the AM will be responsible for both attack pathway generation and exploit sequence generation. Therefore, it is important to bound the expected role and function of the AM before defining the various interfaces.

In this use case, the motivation of automating as much of the attack sequence generation process as possible motivates the AM bounding decision to include attack pathway generation in the expected function of the AM. System component descriptions are captured in the TAM as Cyber Security Data Sheets and component relationships are captured in the form of Relationship Set Data Sheets. System protocols, process properties, and performance metrics are not formal components of the TAM and are not provided to the AM.

In terms of attack vectors, the TAM workflow expects manual identification of vulnerabilities associated with each component. While this process could be delegated to the attacker model, the vast majority of attacker models do not undertake vulnerability identification and discovery, but rather leave that to engineers to provide using more specialized techniques. For the TAM integration we delegate the vulnerability discovery process to the security engineers, and expect as input to the attacker model a vulnerability database composed with CPS component associations. The TAM assessment includes a component-driven vulnerability evaluation. As such, it is expected that vulnerabilities will be associated to specific CPS components and so the assumption of security experts performing the vulnerability analysis is reasonable. The TAM does not explicitly consider different types of attackers, and therefore there are no vulnerability associations to the attacker.

Finally, in regards to attacker definition, the TAM does not consider unique attackers. As such, the entire branch of attacker attributes including *Profile-Defined Attacker*, *Profile Specificity*, and *Model-Defined Attacker* are set to false.

7.2.3 Contextual Interface

The contextual interface is characterized for the SRPC by evaluating each of the attributes and identifying if the SRP recognizes the concept of each attribute as being valid. The TAM begins by first defining the system architecture, then associating vulnerabilities to components in the architecture and using component relationships to associate vulnerabilities together to compose exploit sequences. This corresponds to the system component principle perspective. The objective is described for the TAM in the context of the SRP/AM integration. We capture this as, "Leverages an attacker model to generate exploit sequences".

Time is the first evaluated attribute in the physics attribute category. While the TAM includes the concept of time in later steps of the tool workflow (such as in evaluating exploit sequence criticality), it does not include any standardized time-based characterization of the components, the component relationships, or component processes. Next, the TAM does not have any process that specifically applies unpredictable behaviors to the review process. In order to functionally integrate into a security review process, an attacker model must be executable, else the security experts will need to devise a method to functionally execute the attacker model. For this case study, we assume that the security experts are only interested in attacker models that are executable. With regards to the operational method, the TAM treats exploit sequences as a composition of sequential actions performed by the attacker.

Next, the Attacker/CPS Interaction attribute category is evaluated first considering the Attack Procedure. The TAM does not have a mechanism for evaluating the success or failure of an attack. The workflow is intended to identify the existence of a potential attack path, effectively characterising attack steps as possible or impossible. Therefore, the attack procedure is set to {false,}. The TAM does not consider attack consequences in the exploit sequence generation process and therefore it is set to false as well. The TAM workflow is centered, however, around determining CPS Security Controls. In the TAM workflow, the step following exploit sequence generation includes control method development, which includes consideration of both intrusion detection and countermeasures. Whether or not to include these in the SRPC depends on the bounds set for the role of the AM. Because the scope of the attacker model integration for the TAM was exclusively set to exploit sequence generation, the development of control methods falls outside the scope of the attacker model and therefore the CPS Security Controls attribute and its child attributes are set to false.

The CPS behavior attribute category is the next category evaluated for the contextual interface. The TAM does not explicitly consider CPS Security Protocols. While the TAM does define the communication channels (communication types), it does not actually define the communications that are sent between components. The TAM does not model a functioning system and therefore does not maintain a security state but rather considers a system component insecure if an attack pathway exists for that component. Because the TAM does not model system processes and it does not maintain a CPS Process state, nor does it model cyber-processes or physical processes. The TAM does, however, maintain both a cyber orientation and a physical orientation through documentation of the CPS under review.

Finally, the attacker behavior category is evaluated. The TAM does not specify an attacker but presumes exploit sequences be created for all possible attackers. It also includes an insider as a valid attacker, and for the exploit sequence generation step does not restrict the information available to the insider. Therefore, the exploit sequence generation process presumes the attacker have full awareness of the system architecture, which corresponds with the static attacker knowledge model. The summary of the attribute results can be seen in Table 7.1.

7.3 Step 2 - AMC Database Population

The attacker modeling framework is dependent on the existence of, or ability to create, a database of attacker models, each described with an AMC. For the TAM case study, we use the AMCs for the attacker models evaluated in Chapter 4 which can be seen summarized in Table 4.2 and fully documented in Appendix A. These attacker models were selected because they represent a wide diversity of attacker models in form, function, and assumptions.

7.4 Step 3 - AMC Evaluation

Evaluation of the attacker model database with respect to the SRPC is done via application of the scoring functions defined in Section 6.2.4. The calculated scoring values are presented in Table 7.2. For each scoring category, the average value was calculated and results above the average value were highlighted as green while results below the average value were highlighted as red. The intermediary matching evaluations for each attribute in each interface can be seen in Appendix C as Tables C.1 and C.2.

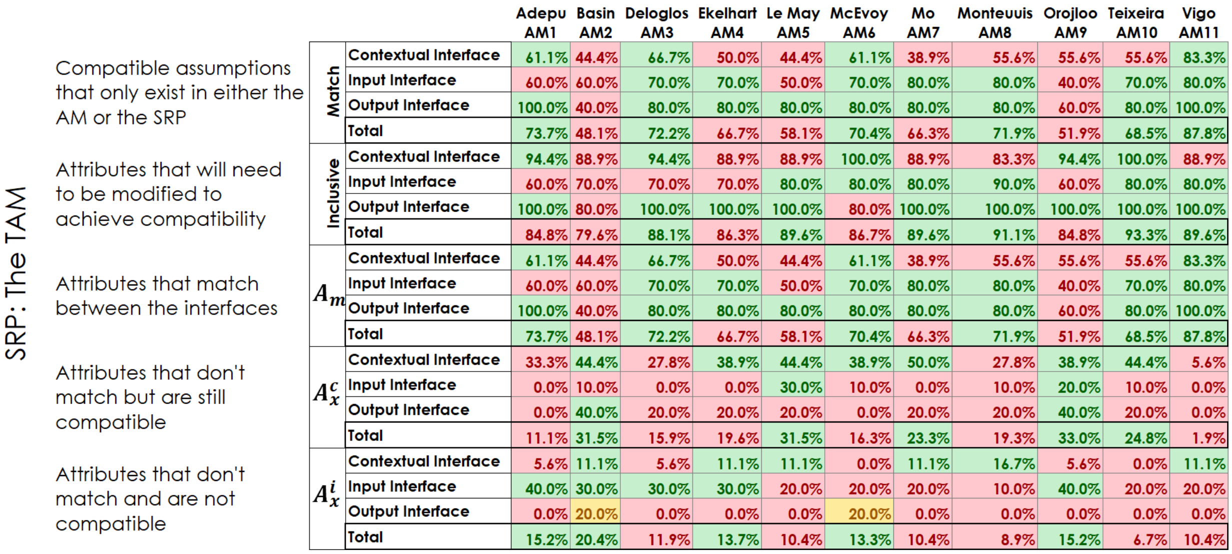
7.5 Step 4 - Attacker Model Selection

The AM selection process follows the process described in Section 6.2.4, beginning with compatibility filtering.

Table 7.1 The security review process characterization for the use-case application of EPRI's Technical Assessment Methodology [1]

	Category	Attribute	Ind	EPRI's TAM Description
Contextual Interface	Objective			Develop exploit sequences from attack pathways and asset characteristics.
	Principle Perspective		C	The assessment process begins with the characterization of the system architecture, then progresses to develop attack pathways from component associations.
	Dimensions	Time	F	
		Unpredictability	F	
		Executable	T	An attacker model must be functionally executable to be considered for integration
		Operational Method	S	Treats exploit sequences as a composition of sequential actions performed by the attacker.
		Attack Procedure	F	
		Attack Consequence	F	
		CPS Security Controls	F	
		Intrusion Detection	F	
		Countermeasures	F	
		CPS Security Protocols	F	
		CPS Communication	F	
		CPS Process State	F	
		CPS Security State	F	
		CPS Process - Cyber	F	
		CPS Process - Physical	F	
		Orientation - Physical	T	Abstractly represented through a Data Flow Diagram.
		Orientation - Cyber	T	Represented through a Data Flow Diagram.
		Attacker Knowledge Model	S	Assumes that the attacker has full knowledge of all available attacks.
Input Interface	CPS	Components	T	Captures system components in the Attack Surface Characterization.
		Component Relationships	T	Captures communication methods and protocols in the Attack Surface Characterization.
		Protocols	F	
		Process Properties	F	
		Performance Metrics	F	
	Attack Vectors	CPS Associations	T	Attack vectors will be defined by security experts with associations to the system components they are applicable to.
		Attacker Associations	F	
	Attacker	Profile-Defined Attacker	F	
		Profile Specificity	F	
		Model-Defined Attacker	F	
Output	Results	Procedure	T	Procedure results would be used to describe exploit sequences.
		Security Metrics	F	
		Security Properties	F	
		Outcome Likelihood	F	
		Risk Assessment	F	

Table 7.2 Scoring values calculated from the comparison of the AMC and SRPC attributes for the TAM use case, highlighting above average values as green, below average values as red, and average values as yellow



SRP: The TAM			Adepu AM1	Basin AM2	Deloglos AM3	Ekelhart AM4	Le May AM5	McEvoy AM6	Mo AM7	Monteuuis AM8	Orojloo AM9	Teixeira AM10	Vigo AM11
Compatible assumptions that only exist in either the AM or the SRP	Match	Contextual Interface	61.1%	44.4%	66.7%	50.0%	44.4%	61.1%	38.9%	55.6%	55.6%	55.6%	83.3%
		Input Interface	60.0%	60.0%	70.0%	70.0%	50.0%	70.0%	80.0%	80.0%	40.0%	70.0%	80.0%
		Output Interface	100.0%	40.0%	80.0%	80.0%	80.0%	80.0%	80.0%	80.0%	60.0%	80.0%	100.0%
		Total	73.7%	48.1%	72.2%	66.7%	58.1%	70.4%	66.3%	71.9%	51.9%	68.5%	87.8%
Attributes that will need to be modified to achieve compatibility	Inclusive	Contextual Interface	94.4%	88.9%	94.4%	88.9%	88.9%	100.0%	88.9%	83.3%	94.4%	100.0%	88.9%
		Input Interface	60.0%	70.0%	70.0%	70.0%	80.0%	80.0%	80.0%	90.0%	60.0%	80.0%	80.0%
		Output Interface	100.0%	80.0%	100.0%	100.0%	100.0%	80.0%	100.0%	100.0%	100.0%	100.0%	100.0%
		Total	84.8%	79.6%	88.1%	86.3%	89.6%	86.7%	89.6%	91.1%	84.8%	93.3%	89.6%
Attributes that match between the interfaces	A_m	Contextual Interface	61.1%	44.4%	66.7%	50.0%	44.4%	61.1%	38.9%	55.6%	55.6%	55.6%	83.3%
		Input Interface	60.0%	60.0%	70.0%	70.0%	50.0%	70.0%	80.0%	80.0%	40.0%	70.0%	80.0%
		Output Interface	100.0%	40.0%	80.0%	80.0%	80.0%	80.0%	80.0%	80.0%	60.0%	80.0%	100.0%
		Total	73.7%	48.1%	72.2%	66.7%	58.1%	70.4%	66.3%	71.9%	51.9%	68.5%	87.8%
Attributes that don't match but are still compatible	A_x^c	Contextual Interface	33.3%	44.4%	27.8%	38.9%	44.4%	38.9%	50.0%	27.8%	38.9%	44.4%	5.6%
		Input Interface	0.0%	10.0%	0.0%	0.0%	30.0%	10.0%	0.0%	10.0%	20.0%	10.0%	0.0%
		Output Interface	0.0%	40.0%	20.0%	20.0%	20.0%	0.0%	20.0%	20.0%	40.0%	20.0%	0.0%
		Total	11.1%	31.5%	15.9%	19.6%	31.5%	16.3%	23.3%	19.3%	33.0%	24.8%	1.9%
Attributes that don't match and are not compatible	A_x^i	Contextual Interface	5.6%	11.1%	5.6%	11.1%	11.1%	0.0%	11.1%	16.7%	5.6%	0.0%	11.1%
		Input Interface	40.0%	30.0%	30.0%	30.0%	20.0%	20.0%	20.0%	10.0%	40.0%	20.0%	20.0%
		Output Interface	0.0%	20.0%	0.0%	0.0%	0.0%	20.0%	0.0%	0.0%	0.0%	0.0%	0.0%
		Total	15.2%	20.4%	11.9%	13.7%	10.4%	13.3%	10.4%	8.9%	15.2%	6.7%	10.4%

7.5.1 Filtering by Incompatibility

First, the score for incompatible attributes (A_x^i) defined in Equation 6.9 is used to filter out incompatible attacker models. In order to be compatible an attacker model must either meet an incompatibility score of 0%, or else the expert performing the review must be able to justify each mismatched attribute of the attacker model. While several of the attacker models achieved very high inclusive matching scores, none of them scored 0% and therefore each requires consideration and justification of the conflicting attributes. We evaluate the attacker models in order of highest-scoring inclusive total match to lowest.

Teixeira's AM in [63] achieved a score of 8.5% with 3 incompatible attributes. The first is the operational method which is sequential for the SRP but discrete-time based for the AM. This is more critical in SRPs where the review process depends on intermediary values in the attacker modeling process. In the TAM case study, the deliverable of generating exploit sequences does not require a particular operating method, so long as the AM can capture those exploit sequences and provide them as output. For the second attribute, the AM requires as input process properties for the system architecture, which are not explicitly defined in the TAM. Inspecting the AM, the process properties expected include control actions, sensor measurements, process and measurement noise, and measurements of the discrepancies between the model and the real process. Remediation of this attribute would require the engineers develop or acquire these process properties for the system under evaluation. While this may be possible for certain systems, it would be a laborintensive process and would conflict with the integration goal of reducing the effort required by the security review experts. Therefore, this attribute cannot be justified and Teixeira's AM removed from consideration.

Monteuuis' AM in [56] achieved a score of 8.9% with 4 incompatible attributes. The first is the principle perspective which is the component perspective for the SRP but is the individual attack perspective for the AM. AMs developed from the individual attack perspective are created around an integral use case and may be valuable to an independent use case if the AM can be modified sufficiently to apply to the new use case. Evaluating Monteuuis' attacker model, the model was designed around the perspective of attackers against connected and automated vehicles. Built on this foundation, the relationship between the attacker and the system is explicitly derived around the architecture of autonomous vehicles and does not translate to other system architectures. Therefore, while Monteuuis' AM may work for TAM if the architecture under review is an autonomous vehicle, it does not work as a generic solution.

Vigo's AM in [73] achieved an incompatibility score of 10.4%, with 5 incompatible attributes. The first two are Uses Profiling and Profile Specificity. These attributes are incompatible because the AM requires as input profiles of expected attackers, whereas the TAM does not explicitly define attacker profiles. Remediation of this is possible if the expert performing the integration adds a step to the TAM where an engineer identifies profiles for various attackers according to the profiling scheme in the AM. While this would require additional work on the part of the security expert, the TAM assumes the export is manually evaluating the range of potential threats in the process of creating exploit sequences, and so it a reasonable assumption. The third attribute is the attacker knowledge model which is static for the SRP but dynamic for Vigo's AM. This can be resolved if the engineer defines the initial knowledge expectation of the attacker, which could be set to a complete initial knowledge for this attribute to have no effect. The final attribute is the executable attribute. While Vigo presents a well-formulated attacker model, it cannot readily be applied to simulate attacker behavior without being executable and is therefore not compatible with the SRP.

Mo's AM in [72] achieved a score of 12.2% with 5 incompatible attributes. The first is the principle perspective which is also the individual attack perspective. Mo's attacker model is designed around smart power grid infrastructure and, similar to Moneuuis', is uniquely designed around the architectural characteristics of power-grids and cannot readily be applied to a generic security review.

Le May's AM in [54] also achieved a score of 12.2% with 5 incompatible attributes. The principle perspective of Le May's attacker model is the vulnerability perspective, in contrast to the system component perspective used in the TAM. Attacker models that use the vulnerability perspective begin with an association of vulnerabilities within the system and then relate those vulnerabilities to components and to the attacker. This vulnerability association can take many forms, but is realized as the exploit sequence in TAM or as an attack execution graph in Le May's model. This is an inverse workflow of that presented by TAM where the system architecture is first defined including relationships between components, then vulnerabilities are associated to components to produce exploit sequences. Therefore, Le May's AM is fundamentally incompatible with the requirements of the TAM use case.

Deloglos' AM in [5] achieved a score of 13.7% with 5 incompatible attributes. The first is the physical orientation attribute which is true in the TAM but not in Deloglos' AM. While the system architectural description provided as input to the TAM captures the physical orientation of the system and exploit sequences take into account how physical orientation influences different attack vectors. Deloglos' AM does not explicitly model physical orientation, but rather requires as input a list of all vulnerabilities of the system, which is expected to take into account vulnerabilities relating to physical orientation. Because the TAM workflow requires discover of all attack paths as a prerequisite step to exploit sequence generation, it is not unreasonable to expect the experts to provide the same list in the form of vulnerabilities to the AM. Therefore this attribute mismatch is acceptable. The TAM defines the Attacker Knowledge Model as static assuming that the attacker may have knowledge of the system architecture and the system details, whereas Deloglos' AM models a dynamic attacker knowledge model. In this instance, either a step can be added to the TAM to define the initial attacker knowledge, or the initial attacker knowledge can be set to knowledge of the complete system. The remaining three mismatched attributes in Deloglos' AM are Uses Profiling, Profile Specificity, and Attacker Associations. Deloglos uses a profiling-based attacker model which requires as input attacker profiles as well as a similar characteristics for each attack action, and then associates the attacker profiles to action characteristics to predict attacks. This could be mediated by adding a step to the TAM where an attacker profile is defined as input for the AM.

McEvoy's AM in [62] achieved a score of 15.2% with 5 incompatible attributes. McEvoy's AM leverages a variant of pi-Calculus to develop a system-theoretic model of a SCADA system. One of the incompatible attributes is the Process Properties attribute. In order to model the SCADA system McEvoy's AM requires descriptions of SCADA supervisor, control, and communication processes. This level of process detail is beyond the requirements of the AM and cannot necessarily be produced or acquired by the experts performing the SRP. Therefore, McEvoy's AM is not compatible.

Adepu's AM in [57] achieved a score of 17.0% with 6 incompatible attributes. Adepu's AM is a generalized attacker and attack model for modeling a diverse set of systems and a diverse variety of attacks and attackers. Adepu's model, however, models the system and the attacker to a significant level of detail, requiring both process properties and system performance metrics as inputs, both of which are not explicitly defined in the TAM, and unrealistic to expect the security review experts to produce without a significant effort. Therefore, Adepu's AM is not a viable AM solution for the TAM.

Orojloo's AM in [55] achieved a score of 17.0% with 6 incompatible attributes. Orojloo's AM, much like Adepu's, requires a process model of the system under review. Because the detail required for composing a process model is not available in the TAM, the AM is not compatible.

Ekelhart's AM in [59] achieved a score of 15.6% with 6 incompatible attributes. Ekelhart's AM use a vulnerability perspective as its principle perspective, taking as input a series of vulnerabilities defined with execution preconditions and post-conditions, as well as an attacker model, and compiles an abstract attack graph. The attack graph is then associated to a system model in order to determine attack procedures. The requirement of vulnerabilities is reasonable given that the TAM requires the population of attack pathways which can be translated to a list of vulnerabilities. The preconditions and post-conditions can be manually created by security experts with minimal additional effort given the simplistic design of Ekelhart's condition scheme. In addition, the creation of the attacker profile is a low-effort requirements which is reasonable for the security experts to produce. The attacker knowledge model is the last conflicting attribute. Ekelhart's AM implements a dynamic attacker knowledge model where attack steps are only visible to an attacker when all preceding attack steps are accomplished. This characteristic of the model is self-contained and does not require additional AM input and therefore is acceptable to the SRP.

Finally, Basin's AM in [74] achieved a score of 22.2% with 7 incompatible attributes. Basin's model uses system-theoretic operational semantics to develop system protocols. Basin's model does not provide attack procedures and is therefore incompatible with the requirements of the SRP.

In conclusion to the filtering step, we identified that two attacker models are compatible with the TAM, which are Deloglos', and Ekelhart's attacker models with incompatibility scores of 13.7% and 15.6% respectively. Resolutions were identified for each incompatibility, requiring modification of the AM or the SRP, or both. The magnitude of effort required for attribute compatibility is the first contributor to the AM valuation. Vigo's AM requires the least effort, requiring the TAM to be extended to define attacker profiles. Deloglos' AM likewise requires the TAM to be extended to define attacker profiles. Deloglos' AM also requires the as input a list of vulnerabilities, which can be provided from the list of attack paths developed in the TAM. Ekelhart's AM requires the definition of an attacker profile by the TAM, as well as a list of vulnerabilities, but requires the engineer performing the TAM to develop preconditions and post-conditions for all vulnerabilities.

7.5.2 Evaluating Assumptions

Once incompatible attacker models have been filtered out, the compatible mismatched attributes (A_x^c) of the remaining attacker models are evaluated in order to identify which attacker model provides the most value to the SRP using Equation 6.8 . The remaining pool of attacker models under consideration includes those created by Deloglos and Ekelhart which have compatible mismatched attribute scores of 15.9% and 17.8% respectively, which indicate what percentage of the attributes that are mismatched do not inhibit compatibility.

Deloglos' AM includes in the set A_x^c the attributes Unpredictability, Attack Procedure, CPS Communication, CPS Process State, CPS Security State, and Outcome Likelihood. The unpredictability in Deloglos' AM is applied by creating a non-deterministic attacker profile, which results in a non-deterministic attack procedure. Deloglos' AM can be leveraged to identify different attack paths by repeatedly executing the AM and collecting the results of different exploit sequences. An attack procedure is not specified in the TAM but rather left to the security expert to intuit. As such, we note that Deloglos' AM uses a qualification-based AM. The information necessary to qualify attack success is provided through the CPS Associations and Attacker Associations attributes which were resolved in the compatibility filtering step by having the SRP expert define profiles for the attacker and each attack path. CPS Communication is realized in Deloglos' AM much like in Vigo's, where communication relationships between components including communication types (MODUBS, Ethernet, etc...) are expected as part of the system architecture. The CPS Process State and CPS Security State are not reasoned about by the TAM and are used as internal mechanisms in the AM to execute the attacker model. The outcome likelihood is an additional value that Deloglos' AM provides, which includes the probability of an exploit sequence being used given the range of attacker profiles.

Ekelhart's AM includes in the set A_x^c the attributes Attack Procedure, Attack Consequence, CPS Security Controls, Intrusion Detection Countermeasures, CPS Process State, CPS Security State, and Outcome Likelihood. Ekelhart's AM uses a qualification-based Attack Procedure similarly to Deloglos' and Vigo's AMs which is internal to the AM. The model also evaluates attack consequences using confidentiality metrics provided to the system. The confidentiality is associated with attack vectors and is expected to be provided as a property for each attack vector that affects confidentiality. Ekelhart's AM models the effect of CPS Security Controls as Intrusion Detection and Countermeasures in the system. The purpose of the TAM is to use the exploit sequences to develop security controls and therefore at the step where the TAM provides information to the AM, security controls do not yet exist. However, Ekelhart's AM is capable of modeling the attack in the absence of security controls which is a viable solution for the TAM integration. The CPS Process State and CPS Security State are similarly attributes that the TAM does not consider. Finally, Ekelhart's AM provides as a result Outcome Likelihood which includes statistics such as if the target was reached, attack duration, total actions, and confidentiality impact.

Both AM's are compatible with the TAM, are of high value to the TAM, and produce the exploit sequences required by the TAM. While Ekelhart's AM has more features, its simulation engine executes the attacker model at a lower interaction level, which requires the definition of detailed properties for the component descriptions, such as installed software, firmware, patches, and other vulnerability-related properties. The manual exploit sequence generation process advised by the TAM is modeled as a higher-level process which does not necessarily consider the more detailed properties of Ekelhart's AM. This is more in-line with the higher-level perspective of Deloglos' AM which uses more generic properties for characterize vulnerability. Therefore, Deloglos' AM is selected for the integration.

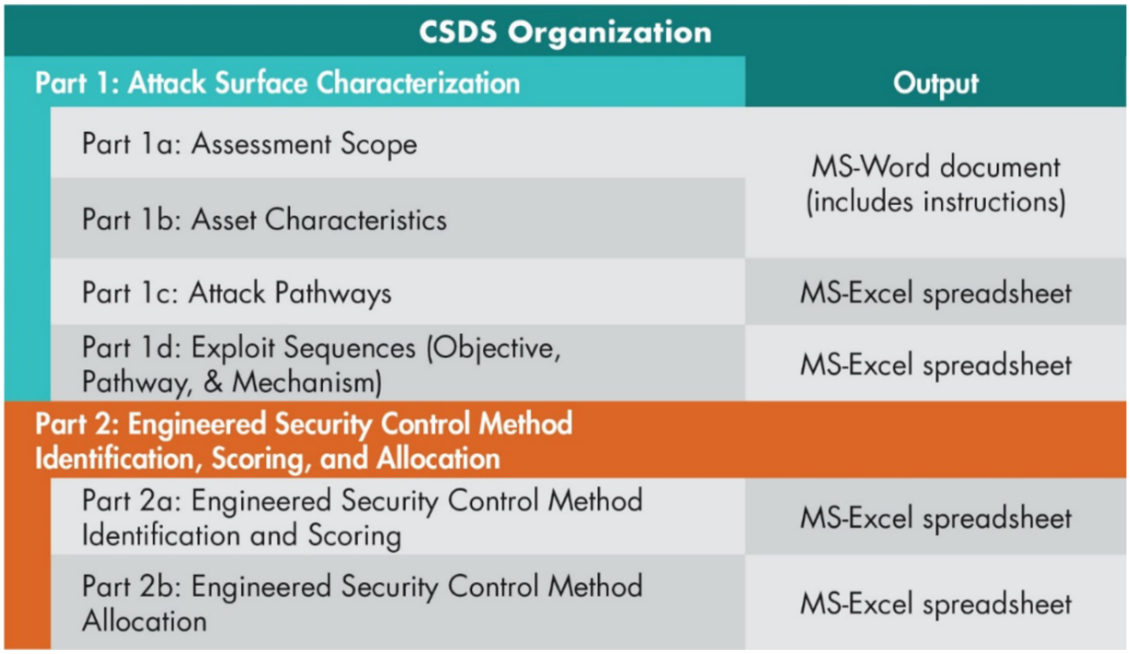
7.6 Step 5 - Attacker Model Functional Representation Creation

The first step in the integration process is the creation of the AMFR for Deloglos' AM. The full AMFR development process for Deloglos' AM can be observed Section 5.3. A complete listing of the AMFR is captured as a table of the modules, variables, constants, and relationships of the AMFR and can be seen in Table B.2 of Appendix B. In addition, the AMFR diagram can be observed in Appendix B in Figure B.2.

7.7 Step 6 - Data Integration Mapping

The AMFR for Deloglos' AM defines four constants of input information, all of which are expected to be provided by the security review process. These constants include the CPS Architecture, the Attacker Initial Knowledge, the Action Database, and the Probabilistic Attacker Profile. For each of these constants we document the mapping of the data provided from the SRP.

The CPS Architecture constant of the AMFR is described as the set of all nodes and edges in the CPS. This is captured captured in the TAM through part 1 of step 1 of the creation of the Cyber Security Data Sheet (CSDS), which can be seen in Figure 7.3. The CSDS requires the creation of the Assessment Scope in Part 1a and the Asset Characteristics in Part 1b. Part of the assessment scope includes the Asset Composition, the Asset Decomposition, and the Installed Configuration and Data Flow. The TAM emphasizes the importance of identifying the bounds of an asset description in attack surface characterization. The asset composition identifies what is included in the asset, including the description of all components and sub-components. Asset decomposition is the process of identifying and documenting the sub-components of the asset. In the TAM, part of the documentation of asset composition and decomposition includes general component descriptions, sub-component descriptions, a list of component and sub-component manuals and documentation, and device model numbers. The installed configuration and data flow in the asset composition captures the relationship between the different components based on the set of a dataflow diagram and



CSDS Organization	
Part 1: Attack Surface Characterization	Output
Part 1a: Assessment Scope	MS-Word document (includes instructions)
Part 1b: Asset Characteristics	
Part 1c: Attack Pathways	MS-Excel spreadsheet
Part 1d: Exploit Sequences (Objective, Pathway, & Mechanism)	MS-Excel spreadsheet
Part 2: Engineered Security Control Method Identification, Scoring, and Allocation	
Part 2a: Engineered Security Control Method Identification and Scoring	MS-Excel spreadsheet
Part 2b: Engineered Security Control Method Allocation	MS-Excel spreadsheet

Table 7.3 Organization and work products from Part 1 of Step 1 of the Cyber Security Data Sheet from EPRI's TAM. Table taken from [1]

.

a data topology and data flow description. Step 1b of the TAM captures the Asset Characteristics, which includes component data such as firmware, operating systems, installed application software, installed configurations, maintenance methods, and site characteristics which include physical and logical component orientation.

The AMFR identifies the required data for the CPS architecture as a set of nodes, each containing an ID, Name, and Type, and a set of edges, each containing an ID, Name, a Source, a Destination, and a Type. For the node, the data maps from the output of the SRP to the input of the AM as follows. The CPS ID and Name are provided by the list of items in the TAM asset decomposition. The node Type is defined by Deloglos' AM as the attribute used to associate attack vectors to nodes. The type can be populated as any attribute that the engineer designing the dataset of attack actions can used to associate attack actions to nodes. This includes physical interfaces and firmware, operating systems, installed application software, installed configurations, maintenance methods, and site characteristics. This mapping can be seen in Table C.3 in Appendix C.

The Asset Characteristics of the TAM also include edge-related information such as physical communication ports and terminals, removable media and portable devices, HMI capabilities, data communication protocols, and services and logical communication ports. This effectively maps to the edge descriptions required by Deloglos' AMFR. The AMFR edge ID, Name, and Type maps from the TAM identifier of the respective edge-related target asset characteristic. This can be seen documented in Table C.3.

In the AMC compatibility filtering part of Step 4, the resolution for the AM requirement of a database of vulnerabilities was to modify the TAM to produce this database. The TAM provides a subset of vulnerabilities by identifying the points on the attack surface where an attack could originate, documenting these as attack vectors. Attack vectors are then used to reason out attack paths. Creation of an action database would require an intermediary step where a vulnerability assessment process is performed on all components of the asset decomposition, as well as on all communications-related asset characteristics. This would provide the action ID, Name, and Target data. The concept of action profile properties used in the AM to associate the attacker to actions does not exist in the TAM. This incompatibility was resolved in Step 4 by modifying the TAM to include a step where the security expert creates for each action in the database a profile in accordance with the profile properties selected in the pre-configuration of the attacker model. This mapping can be seen in Table C.3.

The final constant of the AMFR is the probabilistic attacker profile which is composed of three components: pre-configuration properties for the AM profiling scheme, the set of profile properties for each of the six attacker types, and the likelihood of attack of each of the attacker types. Attacker profiles are not a native concept to the TAM and in the compatibility evaluation in Step 4 this was resolved by modifying the TAM to include a step that creates this data according to the format described in Deloglos' AM. This mapping can be seen in Table C.3.

Through this mapping process, we were able to provide descriptions of the data source for all input data required by the AM from the SRP. The next step is to map how the output results from the AM are returned to the SRP. The results of the AM are documented as the attack path taken by the attacker informed by probabilities for target selection and action selection. The exploit sequence for the TAM is defined as the set of the exploit objective, the attack pathway, and the exploit mechanism. The attack paths from Deloglos' AM are map to the attack paths of the TAM exploit sequence. The vulnerabilities exploited map to the TAM exploit mechanism.

The AMF leverages a high-level description of the attacker model to facilitate an evaluation and integration process without requiring an in-depth dive into the meticulous inner workings of attacker models. A limitation of this workflow is that during integration, different data formats may require additional work in effectively mapping low-level information. This is observed when integrating the third component of the TAM's exploit sequence, which is the exploit objective. In the SRPC, the exploit objective was identified as a result requirement by the TAM. In the AMC, Deloglos' AM was identified as providing attack procedures which included the exploit objective. Deloglos' AM captures the exploit objective as ending target node in the attack procedure. The TAM, however, has a more specific definition for exploit objectives, which is a set of 28 distinct exploit objectives which identify the all possible goals of the attack. Integration of the exploit objective will require a translation of the exploit objective from Deloglos' AM to the proper selection of the exploit objectives defined by the TAM. We capture this in the integration documentation in Table C.4.

7.8 Step 7 - AM Integration

To verify the integration of Deloglos' AM into the TAM, we evaluate the execution of the integrated pair at the boundaries of integration. The component mapping from the TAM to Deloglos' AM can be seen in Table 7.4, and the mapping from Deloglos' AM back to the TAM can be seen in Table 7.5. The first table identifies the data that the TAM is expected to provide to the attacker model. The second identifies the data that the AM is expected to provide back to the TAM. The condition we set for validating the integration is that all fields of data in 7.4 and 7.5 map correctly. The TAM workflow implements a tabulated documentation scheme which can be seen in Tables D.1-D.6. In order to verify the mapping, we expand Table 7.4 to contain a data field for each mapping relationship. We then extract the data from the source listed in the table and document it in the respective location.

The first datum is the mapping of system component node ID and Name information from the Asset Decomposition Description ID in Part 1a, which can be seen in Table D.2. The documentation describes the level of decomposition as being down to the circuit board level, where the individual components can be seen in the data flow diagram in Table D.3. The data flow diagram defines the names of the components but not the IDs. The expert performing the integration may decide what amount of format manipulation is acceptable when transposing the data from the SRP to the AM. In this instance, the lacking of ID values are not a significant issue as the TAM recognizes components by name, and we can manually define IDs in the data transfer process. For the validation datum, we document the RAM on the simplified loop controller as the first component, define the ID as NODE1, and document it as {RAM,NODE1} in Table 7.4.

There are several asset characteristics to choose from for the node Type. The Type datum is used in the AM to associate vulnerabilities and attack actions to components. For the validation data, we select the Firmware version of the SRL controller, "Firmware Version 2.1". Next is the Edge component with the set of Name, ID, and Type. In the Physical Communication Ports and Terminals section of Table D.6 we see that there are three communication paths to the RAM, which are the Analog Input, Analog Output, and JTAG PINS. For a validation Edge, we select the Analog Inputs. This is defined with several characteristics that may be used for vulnerability association, such as the protocol type and version, which is HART Version 7. We then document this datum as {"Analog Input","EDGE1","HART Version 7"}. We use the Data Topology and Data Flow section from Table D.3 to identify the Edge source and destination. In the diagram, the analog input source

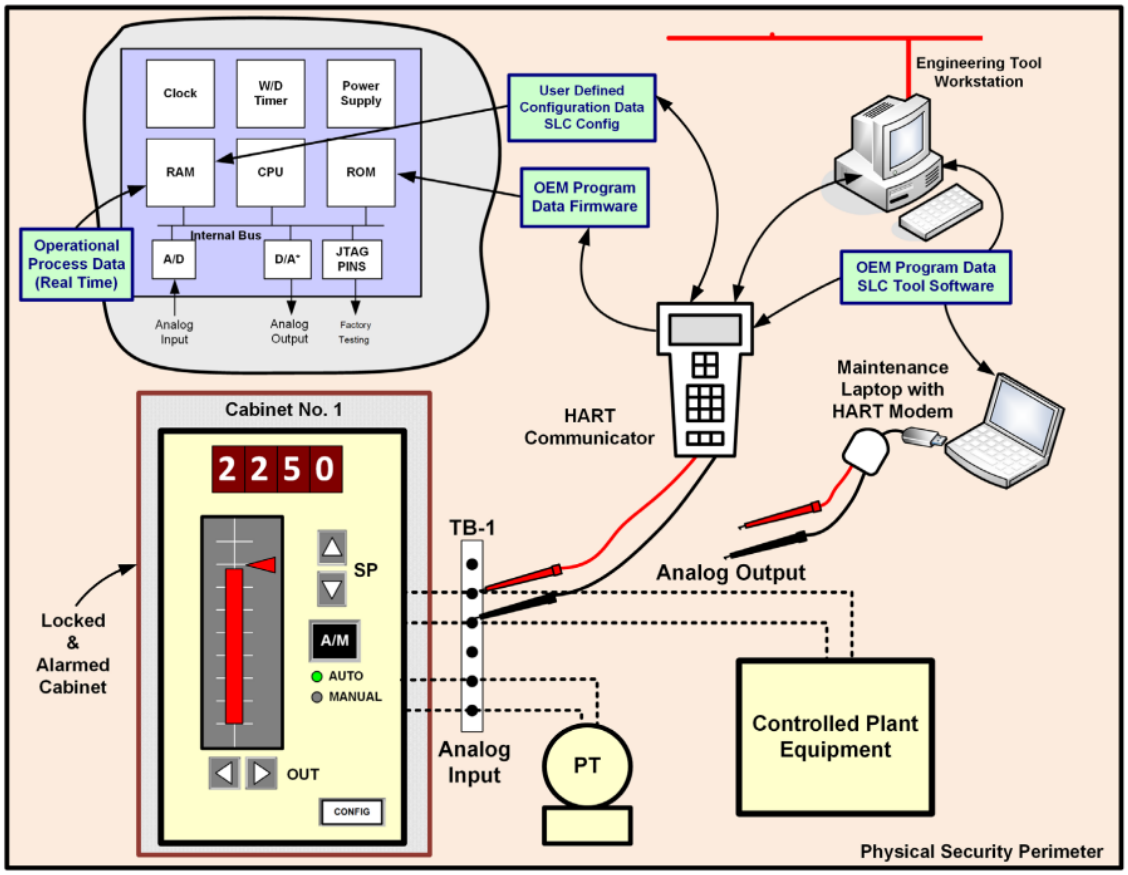
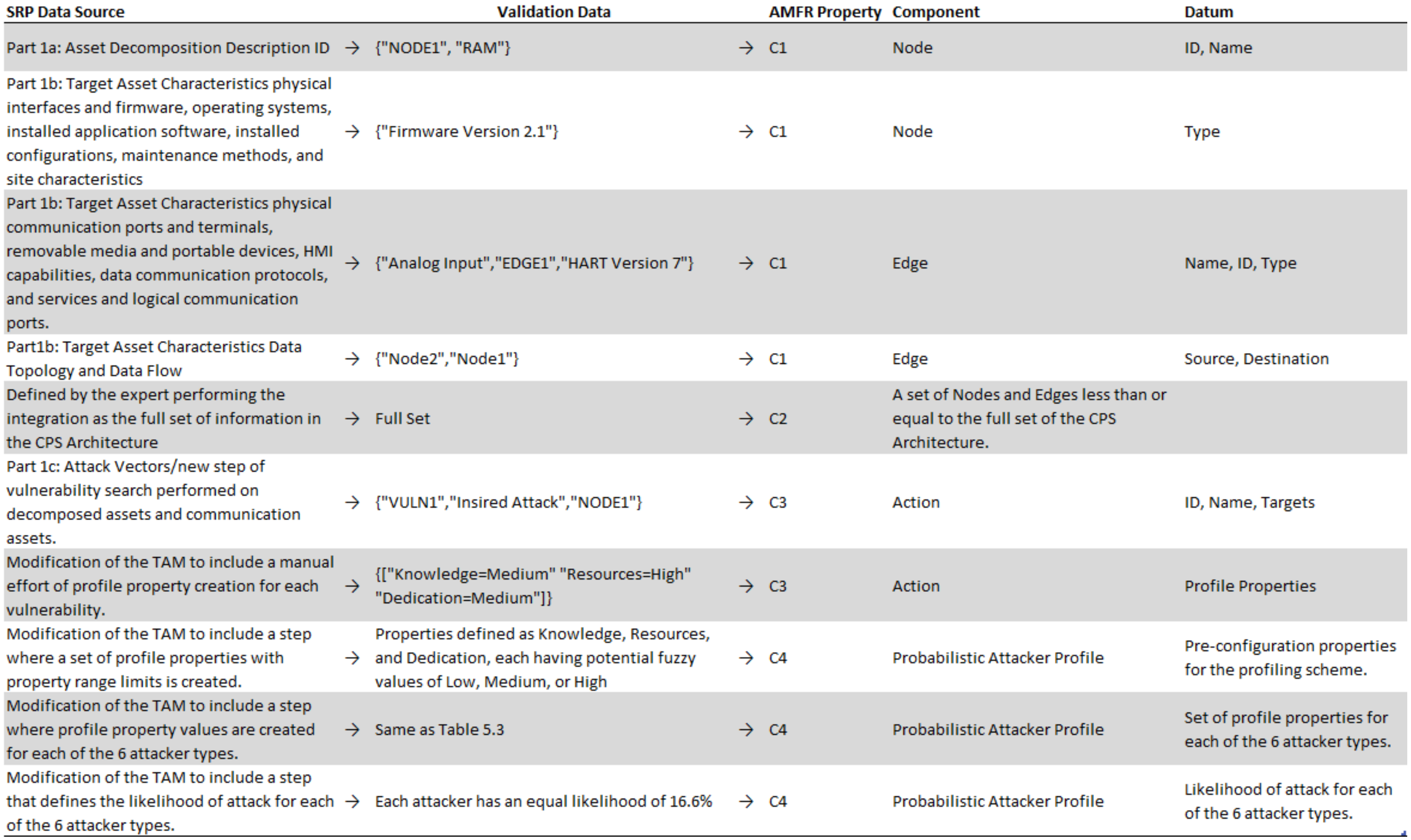

Clock
W/D Timer
Power Supply
RAM
CPU
ROM
Internal Bus
A/D
D/A*
JTAG PINS
Analog Input
Analog Output
Factory Testing
Operational Process Data (Real Time)
User Defined Configuration Data SLC Config
OEM Program Data Firmware
Engineering Tool Workstation
OEM Program Data SLC Tool Software
HART Communicator
Maintenance Laptop with HART Modem
Cabinet No. 1
2250
SP
A/M
AUTO
MANUAL
OUT
CONFIG
Locked & Alarmed Cabinet
TB-1
Analog Output
Analog Input
PT
Controlled Plant Equipment
Physical Security Perimeter

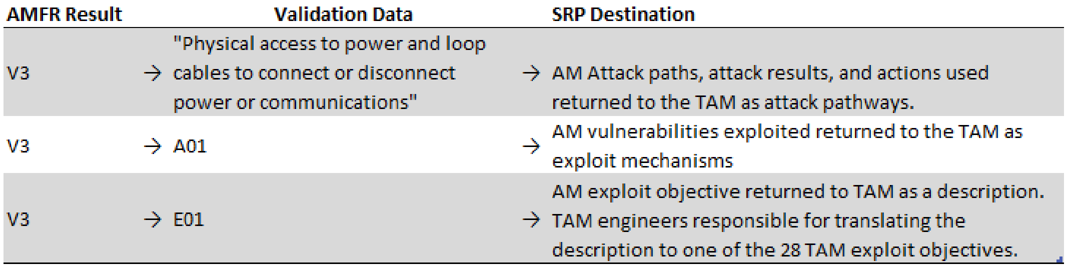
Figure 7.1 An example of a Data Flow Diagram for a simplified single loop controller. Image taken from EPRI's TAM report [1].

Table 7.4 Integration mapping of attacker model input data from the TAM in [1] to Deloglos' attacker model in [5] with validation data



SRP Data Source		Validation Data		AMFR Property	Component	Datum
Part 1a: Asset Decomposition Description ID	→	{"NODE1", "RAM"}	→	C1	Node	ID, Name
Part 1b: Target Asset Characteristics physical interfaces and firmware, operating systems, installed application software, installed configurations, maintenance methods, and site characteristics	→	{"Firmware Version 2.1"}	→	C1	Node	Type
Part 1b: Target Asset Characteristics physical communication ports and terminals, removable media and portable devices, HMI capabilities, data communication protocols, and services and logical communication ports.	→	{"Analog Input","EDGE1","HART Version 7"}	→	C1	Edge	Name, ID, Type
Part1b: Target Asset Characteristics Data Topology and Data Flow	→	{"Node2","Node1"}	→	C1	Edge	Source, Destination
Defined by the expert performing the integration as the full set of information in the CPS Architecture	→	Full Set	→	C2	A set of Nodes and Edges less than or equal to the full set of the CPS Architecture.	
Part 1c: Attack Vectors/new step of vulnerability search performed on decomposed assets and communication assets.	→	{"VULN1","Insired Attack","NODE1"}	→	C3	Action	ID, Name, Targets
Modification of the TAM to include a manual effort of profile property creation for each vulnerability.	→	{["Knowledge=Medium" "Resources=High" "Dedication=Medium"]}	→	C3	Action	Profile Properties
Modification of the TAM to include a step where a set of profile properties with property range limits is created.	→	Properties defined as Knowledge, Resources, and Dedication, each having potential fuzzy values of Low, Medium, or High	→	C4	Probabilistic Attacker Profile	Pre-configuration properties for the profiling scheme.
Modification of the TAM to include a step where profile property values are created for each of the 6 attacker types.	→	Same as Table 5.3	→	C4	Probabilistic Attacker Profile	Set of profile properties for each of the 6 attacker types.
Modification of the TAM to include a step that defines the likelihood of attack for each of the 6 attacker types.	→	Each attacker has an equal likelihood of 16.6%	→	C4	Probabilistic Attacker Profile	Likelihood of attack for each of the 6 attacker types.

Table 7.5 Integration mapping of AM output results from Deloglos' AM in [5] to the TAM in [1] with validation data



AMFR Result	Validation Data	SRP Destination
V3	→ "Physical access to power and loop cables to connect or disconnect power or communications"	→ AM Attack paths, attack results, and actions used returned to the TAM as attack pathways.
V3	→ A01	→ AM vulnerabilities exploited returned to the TAM as exploit mechanisms
V3	→ E01	→ AM exploit objective returned to TAM as a description. TAM engineers responsible for translating the description to one of the 28 TAM exploit objectives.

comes from the component labeled "PT." Assuming the PT to have an ID of "Node2", we define the Edge source and destination datum as {"Node2", "Node1"}. In Table 7.4, C3 represents the initial knowledge known by the attacker. In the compatibility resolution step in Chapter 7 we justified the resolution as providing the full CPS architecture as the attacker's initial knowledge since the TAM does not have the assumption of a growing attacker knowledge.

The next AMFR component is C3, the vulnerability database, which was resolved in the compatibility evaluation to be manually created by the security expert. As an example vulnerability, we select an insider attack where a company employee can access the Analog Input. Next, the AM incompatibility with profile properties was justified as being provided by the security engineer. While we did not yet define the set of profile properties, a good set might be the array {["Knowledge" "Resources" "Dedication"]} using a fuzzy value scheme of Low, Medium, High as valid options for each profile property. An example set of property values for the Physical Sabotage might be {["Knowledge=Medium" "Resources=High" "Dedication=Medium"]}. These values are also used for the pre-configuration properties for the profiling schema. For the set of profile properties for the six attacker types, we use the same property values as in Table 5.3. We set each attacker to be equally likely for the set of attacker likelihoods, resulting in a value of $100\%/6 = 16.6\%$.

The next step in verifying the integration is demonstrating the data mapping from the attacker model back to the security review process. The expected output of the attacker model can be seen in Table 7.5 as the SRP Destinations. The exploit sequences in the TAM are the set of the attack pathways, the exploit mechanism, and the exploit objectives. The first output provided by the attacker model is the attack pathway, which is a description of the attack procedure informed by the attack path through the system, the attack results, and the used attack actions. A reasonably expected result from the attacker model would be a multi-step attack progression where the attacker performs an insider attack to gain physical access to the device, then uses physical access to reconfigure the input wires. This data could be mapped to the TAM Attack Pathway Description as "Physical access to power and loop cables to connect or disconnect power or communications," as seen in Table D.8 as the Attack Pathway Description for A01 and documented in Table 7.5. The second result returned to the TAM is the exploit mechanism, which is the physical access attack pathway of A01 in D.8. The final result from the AM to the TAM is the exploit objective that the exploit mechanism is used for. This step requires the security expert to translate the attack path and results of the attack to one of the 28 TAM exploit objectives. In this instance, the physical attack maps to E01 of the TAM, defined as "Component Enable/Disablement-Immediate."

7.9 Case Study Conclusions

Evaluating the execution of the integrated pair of the TAM and Deloglos' AM identifies that the data from the TAM effectively maps into the inputs of Deloglos' AM, satisfying all input interface requirements. Likewise, the outputs of Deloglos AM provide all the data required by the TAM. The objective of integrating an attack model into the TAM was to generate exploit sequences from attack paths. We observe that the data the attacker model takes as input from the TAM is the attack path data. Also, the results provided to the TAM from Deloglos' AM include the Exploit Objective number and the applicable attack pathways, which, when entered into the TAM, constitute the Exploit Sequence.

In conclusion, by applying our Attacker Modeling framework, we evaluated 11 attacker models and identified two that were compatible with EPRI's TAM. From those, we identified that one required input data at a level of detail not readily available in the TAM, resulting in the selection of Deloglos' AM. We then used the AMFR to define the functional representation of the attacker model, which was used to generate a mapping of the attack path data from the TAM to the inputs of the AM, and then from the AM results to the TAM as exploit sequences. Finally, we integrated the TAM and the AM. For each mapping relationship between them, we demonstrated the validity of the integration by performing a mock execution and demonstrating that the data was provided in a proper format to each of the mapping relationships.

The result of the attacker model evaluation and selection process has the potential to incur scrutiny given that, from a pool of diverse attacker models, the attacker model selected was the one that we designed early-on in the preliminary work of this dissertations. We argue, however, that far from undermining the integrity of the Attacker Modeling Framework Integration Workflow, this asserts its value and demonstrates a high level of fidelity. In order to understand the results of the evaluation process, it is necessary to explicitly describe how the expert performing the integration can influence the results.

The integration workflow begins with the expert who is performing the review defining the problem in the security review process that the expert wants to solve via the integration of an attacker model. This, in effect, is captured when the expert creates the SRPC in the step where the expert defines what information is available in the SRP, the fundamental assumptions of the SRP, and what results are desired from the attacker model. When considering the TAM [1], there are several different places where attacker models may be of use. For example, in the subsequent steps of the TAM, control methods are devised to mitigate attack paths. In the review of attacker models in Table 4.2 we noted that several attacker models produce Security Properties as a Result attribute in the Output Interface. In Basin's attacker model [74], this is specifically done where Basin uses known attacks to create security protocols, which includes the creation of control methods. Moreover, the TAM, which was designed to integrate with other products from EPRI's body of knowledge, was designed to accept metrics for attack criticality from the HAZCAD [35] tool. Potential applications for attacker models can be identified in the components of the HAZCAD focused on evaluating attack criticality levels. However, when creating the SRPC, we decided that integrating the attacker model would be to generate exploit sequences from attack paths. While this was not intentionally selected with our attacker model in mind, we recognize now that the theoretical foundations that motivated this integration goal are the same theoretical foundations that motivated the original design and development of our attacker model, which was created to automate attack path generation from a database of attack vectors. As such, the fact that the attacker model scoring functions, which objectively calculate filtering and similarity scores from attribute indicators, identified our attacker model as a solution to the need we identified for the TAM shows a high degree of effectiveness by having associated the underlying roots of our attacker model as being similar to those of the SRP integration case.

Chapter Eight

Conclusions

Cyber-physical security review processes need structured methods to reason about and model the behavior of attackers. This dissertation identified a need for structured methods to characterize and evaluate attacker models to facilitate integrating attacker modeling techniques into security review processes. In response to this need, we invented, developed, and presented our solution, a generalized attacker modeling framework for characterizing and evaluating attacker models for integration with CPS security review processes. This dissertation bridges the gap between these fields of research. First, it characterizes and describes attacker models from the perspectives of the experts of both fields and can adapt and expand as attacker modeling techniques evolve. Second, it documents attacker models and security review processes in an accessible, scalable, and manageable tabular documentation scheme that facilitates streamlined attacker model evaluation, comparison, and selection. Finally, it provides a structured workflow for functionally integrating attacker models into security review processes.

8.1 Observations

In Chapters 4 and 5 we presented the two composing modules of the attacker modeling framework, the Attacker Model Characterization module and the Attacker Model Functional Representation module. The Attacker Model Characterization module aids in characterizing attacker models through the developed ontology of attributes which informs their value to and compatibility with a security review process. A significant contribution of this work was the structured review of attacker modeling literature required to identify the proper set of attributes. We demonstrated that the AMC can characterize a wide diversity of attacker models - each varying in form, function, and purpose - and can effectively distinguish how the various attributes of each attacker model influence compatibility and utility to different security review processes.

We also demonstrated the utility of our tabular documentation scheme which is used to capture attribute descriptions and leverages different levels of data abstraction to facilitate reviews and comparisons of large numbers of attacker models. This is particularly valuable to security assessment professionals in the industry for whom finding, understanding, and applying attacker models has historically required such a level of effort that it inhibited the application of attacker models in the field.

In Chapter 5 we presented the Attacker Model Functional Representation module, which uses a generalized relationship-based component scheme to capture the functional implementation of a diverse range of attacker models. Using our module-based tabular documentation scheme, we demonstrated that it could represent differing attacker model behaviors, relationships, and execution models. Moreover, we identified how this module's intermediary level of detail lends to bridging the gap between the high-level attacker model evaluation process and the low-level process of functionally integrating an attacker model into a security review process.

We described the application of our Attacker Modeling Framework in our attacker model Integration Workflow in Chapter 6. There, we presented our structured Integration Workflow, a scalable process to objectively evaluate the compatibility and value of a large number of diverse attacker models. The Integration Workflow leverages decreasing data abstraction to reduce the labor required for evaluation and make the integration workflow attractive to a commercial CPS security audience. Finally, we described how we overcame the challenges of objectively evaluating attacker models by using the security review process as the objective standard for attacker model evaluation.

We demonstrated a use case of the Attacker Modeling Framework and its integration workflow using EPRI's Technical Assessment Methodology [1] (TAM), in which we identified an application for attacker modeling in the TAM workflow. In this use case, we demonstrated how the integration workflow used the Attacker Model Characterization to effectively manage and filter down a diverse database of attacker models, each varying in form, function, and purpose. We demonstrated how the scoring methods devised in the Attacker Modeling Framework helped identify a single attacker model which was compatible with the TAM and capable of fulfilling the requirements of the TAM. We then demonstrated how the Attacker Model Functional Representation module captured the functional implementation of the attacker model in enough detail to effectively map the flow of information between the attacker model and the security review process. We then verified the effectiveness of the integration by executing the TAM with the integrated attacker model and verifying the efficacy of the results.

8.2 Limitations

In conclusion to this research, we identified three limitations that should be noted. First, an objective evaluation of attacker models is difficult to develop apart from a context of application. In this dissertation, we use the security review process as the context of application, making it the objective standard by which the attacker model is evaluated. As such, the selection of an attacker model in conclusion to the evaluation part of the integration workflow does not constitute evidence of the efficacy of the attacker model itself, but rather with respect to the security review process under consideration.

Second, while the AMF integration workflow guides the attacker evaluation process, the value of different attributes in different attacker models is ultimately decided by the security expert performing the integration process. During the integration, if more than one attacker model is proven 100% compatible with the security review process it is necessary for the expert performing the integration workflow to evaluate individual attributes of the attacker models to decide which is the better fit for the security review process. While the AMC aids this evaluation significantly by identifying attributes that should be considered, the expert performing the evaluation must be informed to be able to make a final selection. While it would be desirable to produce an objective metric that describes the value of different attacker models, this would require a significantly more detailed characterization of both attacker models and security review processes. While this appeals in theory, the level of effort that would be required for such a detailed evaluation of a large number of attacker models makes it unrealistic for real-world application.

Finally, in our AMF development workflow in Section 3.2 we described how the AMF was developed and how it can be maintained through continued application of the development workflow. This process requires a manual effort to analyze new attacker modeling literature and techniques as well as a well-informed understanding of the fields of attacker modeling and security review processes.

8.3 Future Work

The conclusion of this work lays a foundation for the beginnings of many other works. Looking at the contributions of this dissertation to the research community, we identify several applications for future work.

8.3.1 More Integration Case Studies

The case study in Chapter 7 demonstrated the ability of the AMF to guide a security expert through the evaluation, selection, and integration of an attacker model into a professional security review process, the TAM [1]. Completing additional integration case studies with alternate security review processes may provide insights to refinements of the AMF and will further establish the efficacy and utility of the tool.

8.3.2 Attacker Model Validation

Our attacker modeling framework fundamentally bridges the research fields of security review processes and attacker modeling. In order to objectively establish the efficacy of an attacker model, one must first have a way to validate the results provided by the attacker model. In Section 2.2.4 we discuss the challenges inhibiting attacker model validation and explain how integrating attacker models into security review processes is a promising next step in performing evidence-based validation. Indeed, the formation of this dissertation topic was partly inspired by the lack of research available for introducing attacker modeling into security review processes for attacker model validation. Historically, the consensus in the attacker modeling community has been that the effort required inhibits the integration of attacker models into security review processes for validation purposes. Similarly, the consensus of the CPS security community has been that the effort required inhibits integrating attacker models to achieve greater cyber-physical security threat model assurance. Our attacker modeling framework reduces the integration burden, making attacker model integration a viable research path for attacker model validation research and security review process threat model assurance research.

8.3.3 Attacker Modeling Framework Shared Database

The diversity in form and function of attacker models makes manually evaluating large numbers of attacker models a daunting task. Our attacker modeling framework reduces this burden by guiding security experts through the AMC and AMFR development process and then using the integration workflow to compare large numbers of attacker models. Establishing a database of attacker models, each described according to our attacker modeling framework with an AMC and AMFR, would be valuable to both attacker modeling and security assessment communities. For the attacker modeling community, this could serve as a unifying foundation where there is currently little consensus on best modeling practices and could accelerate the process of attacker model refinement via feedback from integration studies. For the security assessment community - which currently demonstrates a critical need for objective attacker modeling practices - a database of AMFs that represents a significant portion of current attacker modeling methods and practices would not only reduce the burden of attacker model evaluation and integration but would also serve as a common communicating platform between the two fields of research.

8.3.4 Evolution of Attacker Models

In Section 3.2 we describe the workflow used to determine the set of attributes that compose the Attacker Model Characterization interfaces. In this process, we described how new attacker models that we discovered were integrated into an existing AMC through an attribute refinement process.

While the set of attributes we propose in this dissertation holds for the current state of attacker modeling research, it is likely to evolve as new attacker models are developed with new forms, functions, and purposes. As the state-of-the-art attacker modeling literature evolves, our attribute refinement process allows the AMC to evolve in kind.

REFERENCES

[1] "Cyber Security Technical Assessment Methodology: Risk Informed Exploit Sequence Identification and Mitigation, Revision 1", EPRI, Palo Alto, CA, Tech. Rep., 2018. DOI: 3002012752. [Online]. Available: www.epri.com.

[2] G. Bakirtzis, B. T. Carter, C. H. Fleming, and C. R. Elks, "MISSION AWARE: Evidence-Based, Mission-Centric Cybersecurity Analysis", *Accepted to Wiley Systems Engineering Journal*, 2021.

[3] W. Young, "System-Theoretic Process Analysis for Security (STPA-SEC): Cyber Security and STPA", in *2017 Stamp Conference*, 2019. [Online]. Available: http://psas.scripts.mit.edu/home/wp-content/uploads/2017/04/STAMP_2017_STPA_SEC_TUTORIAL_as-presented.pdf.

[4] I. Friedberg, K. McLaughlin, P. Smith, D. Laverty, and S. Sezer, "STPA-SafeSec: Safety and security analysis for cyber-physical systems", *Journal of Information Security and Applications*, vol. 34, pp. 183–196, Jun. 2017, ISSN: 22142126. DOI: 10.1016/j.jisa.2016.05.008.

[5] C. Deloglos, C. Elks, and A. Tantawy, "An Attacker Modeling Framework for the Assessment of Cyber-Physical Systems Security", in *Lecture Notes in Computer Science*, A. Casimiro, F. Ortmeier, F. Bitsch, and P. Ferreira, Eds., vol. 12234 LNCS, Cham: Springer, 2020, pp. 150–163, ISBN: 9783030545482. DOI: 10.1007/978-3-030-54549-9{_}10.

[6] C. Deloglos, A. Tantawy, and C. Elks, "A Framework for Describing Attacker Models", in *12th Nuclear Plant Instrumentation, Control and Human-Machine Interface Technologies*, vol. 12, NPIC&HMIT 2021, 2021, pp. 1349–1360. DOI: 10.13182/T124-34535.

[7] C. R. Elks, A. Tantawy, R. Hite, S. Gautham, A. Jayakumar, and C. Deloglos, "Realizing Verifiable I&C and Embedded Digital Devices for Nuclear Power Design, Verification and Demonstration of the SymPLe Architecture", U.S. Department of Energy, Tech. Rep., 2019. DOI: USDepartmentofEnergyDE-NE0008445.

[8] M. Gibson, C. Elks, A. Tantawy, R. Hite, S. Gautham, A. Jayakumar, and C. Deloglos, "Achieving Verifiable and High Integrity Instrumentation and Control Systems through Complexity Awareness and Constrained Design", Idaho Operations Office, Idaho Falls, ID (United States), Tech. Rep., Jul. 2019. DOI: 10.2172/1547345. [Online]. Available: http://www.osti.gov/servlets/purl/1547345/.

[9] G. Bakirtzis, G. L. Ward, C. J. Deloglos, C. R. Elks, B. M. Horowitz, and C. H. Fleming, "Fundamental Challenges of Cyber-Physical Systems Security Modeling", Apr. 2020. DOI: 10.

1109/DSN-S50200.2020.00021. [Online]. Available: http://arxiv.org/abs/2005.00043%20http://dx.doi.org/10.1109/DSN-S50200.2020.00021.

[10] C. Elks, C. Deloglos, A. Jayakumar, A. Tantawy, R. Hite, and S. Guatham, "Specification of a Bounded Exhaustive Testing Study for a Software-based Embedded Digital Device", Tech. Rep., 2018. [Online]. Available: http://www.inl.gov.

[11] C. R. Elks, A. Tantawy, M. Gibson, R. Hite, S. Gautham, C. Deloglos, A. Jayakumar, S. Khairullah, J. Moore, and A. Nack, "Lessons and Experiences Learned Applying Model Based Engineering to Safety Critical FPGA Designs", 11th International Workshop on the Application of FPGAs in NPPs, Dallas, Tech. Rep., Aug. 2018. [Online]. Available: https://www.researchgate.net/publication/332370279.

[12] D. C. Elks, C. Deloglos, A. Jayakumar, D. A. Tantawy, R. Hite, and S. Gautham, "Realization of a Automated T-Way Combinatorial Testing Approach for a Software Based Embedded Digital Device", Idaho National Laboratory, Idaho Falls, ID (United States), Tech. Rep., Jun. 2019. DOI: 10.2172/1606019.

[13] S. Gautham, A. Varma Jayakumar, R. Hite, C. Deloglos, A. Tantawy, M. Gibson, A. D. Rajagopala, and C. Elks, "MODEL-BASED DESIGN ASSURANCE AND VERIFICATION IN THE CONTEXT OF IEC-61508 SIL-4 STANDARD", in *Safety Critical Software Development, Qualification, and V&V*, 2021. DOI: 10.13182/T124-34548.

[14] P. Beling, B. Horowitz, C. Fleming, S. Adams, G. Bakirtzis, T. Sherburne, C. Elks, A. Collins, C. Deloglos, and B. Simon, "Security Engineering – Decision Support Tool", Systems Engineering Research Center, Tech. Rep., Jun. 2020. DOI: ERC-2020-TR-008.

[15] G. Bakirtzis, B. T. Carter, C. H. Fleming, and C. R. Elks, "MISSION AWARE: Evidence-Based, Mission-Centric Cybersecurity Analysis", *CoRR*, vol. abs/1712.01448, pp. 1–12, 2017. [Online]. Available: http://arxiv.org/abs/1712.01448.

[16] A.-M. Jamil, L. b. Othmane, and A. Valani, "Threat Modeling of Cyber-Physical Systems in Practice", 2021. [Online]. Available: http://arxiv.org/abs/2103.04226.

[17] N. O. Tippenhauer, W. G. Temple, A. H. Vu, B. Chen, D. M. Nicol, Z. Kalbarczyk, and W. H. Sanders, "Automatic Generation of Security Argument Graphs", May 2014. [Online]. Available: http://arxiv.org/abs/1405.7475.

[18] N. Sayfayn and S. Madnick, "Cybersafety Analysis of the Maroochy Shire Sewage Spill Cybersafety Analysis of the Maroochy Shire Sewage Spill", 2017.

[19] V. PREVELAKIS and D. SPINELLIS, *The Athens Affair*, Jun. 2007. [Online]. Available: https://spectrum.ieee.org/the-athens-affair.

[20] *Revenge Hacker: 34 Months, Must Repay Georgia-Pacific $1M*, Feb. 2017. [Online]. Available: https://www.usnews.com/news/louisiana/articles/2017-02-16/revenge-hacker-34-months-must-repay-georgia-pacific-1m.

[21] D. Bilefsky, *Hackers Use New Tactic at Austrian Hotel: Locking the Doors*, Jan. 2017. [Online]. Available: https://www.nytimes.com/2017/01/30/world/europe/hotel-austria-bitcoin-ransom.html.

[22] B. Krebs, *FBI: Smart Meter Hacks Likely to Spread*, Apr. 2012. [Online]. Available: https://krebsonsecurity.com/2012/04/fbi-smart-meter-hacks-likely-to-spread/.

[23] M. Dalli, *Enemalta employees suspended over 1,000 tampered smart meters.* Feb. 2014. [Online]. Available: https://www.maltatoday.com.mt/news/national/35650/enemalta-employees-suspended-over-1-000-tampered-smart-meters-20140211#.YZVoAE7MIuU.

[24] R. Lee, *German Steel Mill Cyber Attack*, Dec. 2014. [Online]. Available: https://assets.contentstack.io/v3/assets/blt36c2e63521272fdc/bltc79a41dbf7d1441e/607f235775873e466bcc539c/ICS-CPPE-case-Study-2-German-Steelworks_Facility.pdf.

[25] D. Kravets, *Feds: Hacker Disabled Offshore Oil Platforms' Leak-Detection System*, Mar. 2009. [Online]. Available: https://www.wired.com/2009/03/feds-hacker-dis/.

[26] J. Leyden, *Polish teen derails tram after hacking train network*, Jan. 2008. [Online]. Available: https://www.theregister.com/2008/01/11/tram_hack/.

[27] S. Kriaa, M. Bouissou, and L. Piètre-Cambacédès, "Modeling the Stuxnet Attack with BDMP: Towards More Formal Risk Assessments", in *7th International Conference on Risks and Security of Internet and Systems, CRiSIS*, 2012, ISBN: 9781467330893. DOI: 10.1109/CRISIS.2012.6378942.

[28] *Gartner Predicts 75% of CEOs Will be Personally Liable for Cyber-Physical Security Incidents by 2024*, 2021. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2020-09-01-gartner-predicts-75--of-ceos-will-be-personally-liabl.

[29] L. Kohnfelder and P. Garg, "The threats to our products", *Microsoft Security Development Blog*, 1999. [Online]. Available: https://www.microsoft.com/security/blog/2009/08/27/the-threats-to-our-products/%0Ahttps://adam.shostack.org/microsoft/The-Threats-To-Our-Products.docx.

[30] M. Hosburgh, "How to Target Critical Infrastructure: The Adversary Return on Investment from an Industrial Control System", SANS Institute, Tech. Rep., 2016.

[31] A. Shostack, *Threat Modeling: Designing for Security*, 9. 2014, vol. 53, pp. 1689–1699, ISBN: 9788578110796. [Online]. Available: https://www.wiley.com/en-us/Threat+Modeling%3A+Designing+for+Security-p-9781118809990.

[32] A. Atzeni, C. Cameroni, S. Faily, J. Lyle, and I. Flechais, "Here's Johnny: A methodology for developing attacker personas", *Proceedings of the 2011 6th International Conference on Availability, Reliability and Security, ARES 2011*, pp. 722–727, 2011. DOI: 10.1109/ARES.2011.115.

[33] A. Steele and X. Jia, "Adversary Centered Design: Threat Modeling Using Anti-Scenarios, Anti-Use Cases and Anti-Personas", in *2008 International Conference on Information and Knowledge Engineering (IKE'08)*, Las Vegas: CSREA Press, Jul. 2008.

[34] C. Moeckel, “Attacker-centric thinking in security”, in *Proceedings of the 15th International Conference on Availability, Reliability and Security*, New York, NY, USA: ACM, Aug. 2020, pp. 1–10, ISBN: 9781450388337. DOI: 10.1145/3407023.3407082.

[35] A. Clark, A. Williams, A. Muna, and M. Gibson, “Hazard and Consequence Analysis for Digital Systems – A New Approach to Risk Analysis in the Digital Era for Nuclear Power Plants”, *Transactions of the American Nuclear Society*, vol. 119, pp. 440–443, 2018.

[36] J. Thomas, “Intro to Systems Theoretic Process Analysis (STPA)”, 2016. [Online]. Available: http://psas.scripts.mit.edu/home/wp-content/uploads/2016/01/Systems-Theoretic-Process-Analysis-STPA-John-Thomas.pdf.

[37] W. Young and R. Porada, “System-Theoretic Process Analysis for Security (STPA-SEC): Cyber Security and STPA”, *2017 Stamp Conference*, 2017. [Online]. Available: http://psas.scripts.mit.edu/home/wp-content/uploads/2017/04/STAMP_2017_STPA_SEC_TUTORIAL_as-presented.pdf.

[38] G. Bakirtzis, B. J. Simon, A. G. Collins, C. H. Fleming, and C. R. Elks, “Data Driven Vulnerability Exploration for Design Phase System Analysis”, *IEEE Systems Journal*, pp. 1–10, Sep. 2019. [Online]. Available: http://arxiv.org/abs/1909.02923.

[39] *CAPEC - Common Attack Pattern Enumeration and Classification*. [Online]. Available: https://capec.mitre.org/.

[40] *CWE - Common Weakness Enumeration*. [Online]. Available: https://cwe.mitre.org/.

[41] *CVE - Common Vulnerabilities and Exposures*. [Online]. Available: https://cve.mitre.org/%20https://www.cve.org/.

[42] *CPE - Common Platform Enumeration*. [Online]. Available: https://nvd.nist.gov/products/cpe.

[43] *Information Technology Definition*. [Online]. Available: https://www.merriam-webster.com/dictionary/information%20technology.

[44] B. Middleton, *A History of Cyber Security Attacks : 1980 to Present*, 1st. Auerbach Publications, Jul. 2017, ISBN: 9781315155852. DOI: 10.1201/9781315155852. [Online]. Available: https://www-taylorfrancis-com.proxy.library.vcu.edu/books/mono/10.1201/9781315155852/history-cyber-security-attacks-bruce-middleton.

[45] *Operational Technology Definition*. [Online]. Available: https://www.gartner.com/en/information-technology/glossary/operational-technology-ot.

[46] M. Iansiti, *The History and Future of Operations*, Jun. 2015. [Online]. Available: https://hbr.org/2015/06/the-history-and-future-of-operations.

[47] *Why Perimeter Security is No Longer Enough*, Jun. 2012. [Online]. Available: https://cyolo.io/blog/why-perimeter-security-is-no-longer-enough/.

[48] C. Salter, O. S. Saydjari, B. Schneier, and J. Wallner, "Toward a secure system engineering methodology", *Proceedings of the 1998 Workshop on New Security Paradigms (NSPW '98)*, vol. Part F129230, pp. 2–10, Jan. 1998. DOI: 10.1145/310889.310900.

[49] B. Kordy, L. Piètre-Cambacédès, and P. Schweitzer, "DAG-based attack and defense modeling: Don't miss the forest for the attack trees", *Computer Science Review*, vol. 13-14, no. C, pp. 1–38, 2014, ISSN: 15740137. DOI: 10.1016/j.cosrev.2014.07.001. [Online]. Available: http://dx.doi.org/10.1016/j.cosrev.2014.07.001.

[50] R. Bell, "Introduction and Revision of IEC 61508", in *Advances in Systems Safety*, C. Dale and A. Tom, Eds., London: Springer London, 2011, pp. 273–291, ISBN: 9780857291332.

[51] "ISO 26262-1:2018 - Road vehicles - Functional safety - Part 1: Vocabulary", International Organization for Standardization, Tech. Rep., 2018. [Online]. Available: https://www.iso.org/standard/68383.html.

[52] B. Brosgol, "Do-178c: the next avionics safety standard", *ACM SIGAda Ada Letters*, vol. 31, no. 3, pp. 5–6, Nov. 2011, ISSN: 1094-3641. DOI: 10.1145/2070336.2070341.

[53] H. Altabbakh, M. A. AlKazimi, S. Murray, and K. Grantham, "STAMP - Holistic system safety approach or just another risk model?", *Journal of Loss Prevention in the Process Industries*, vol. 32, pp. 109–119, Nov. 2014, ISSN: 09504230. DOI: 10.1016/j.jlp.2014.07.010.

[54] E. LeMay and W. H. Sanders, "ADVERSARY-DRIVEN STATE-BASED SYSTEM SECURITY EVALUATION", Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2011.

[55] H. Orojloo and M. Abdollahi Azgomi, "Predicting the behavior of attackers and the consequences of attacks against cyber-physical systems", *Security and Communication Networks*, vol. 9, no. 18, pp. 6111–6136, 2016, ISSN: 19390122. DOI: 10.1002/sec.1761.

[56] J.-P. Monteuuis, J. Petit, J. Zhang, H. Labiod, S. Mafrica, and A. Servel, "Attacker model for Connected and Automated Vehicles", *Acm Cscs*, no. September, 2018. DOI: 10.1145/3273946.3273951. [Online]. Available: https://doi.org/10.1145/3273946.3273951.

[57] S. Adepu and A. Mathur, "Generalized Attacker and Attack Models for Cyber Physical Systems", in *Proceedings - International Computer Software and Applications Conference*, vol. 1, IEEE Computer Society, Aug. 2016, pp. 283–292, ISBN: 9781467388450. DOI: 10.1109/COMPSAC.2016.122.

[58] C. Cheh and W. H. Sanders, "PROTECTING CRITICAL INFRASTRUCTURE SYSTEMS USING CYBER, PHYSICAL, AND SOCIO-TECHNICAL MODELS", Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2019. [Online]. Available: https://www.perform.illinois.edu/Papers/USAN_papers/19CHE02.pdf.

[59] A. Ekelhart, E. Kiesling, B. Grill, C. Strauss, and C. Stummer, "Integrating attacker behavior in IT security analysis: a discrete-event simulation approach", *Information Technology and Management*, vol. 16, no. 3, pp. 221–233, Sep. 2015, ISSN: 15737667. DOI: 10.1007/s10799-015-0232-6.

[60] *Examples - Mobius Wiki*. [Online]. Available: https://www.mobius.illinois.edu/wiki/index.php/Examples.

[61] M. D. Ford, K. Keefe, E. Lemay, W. H. Sanders, and C. Muehrcke, "Implementing the ADVISE security modeling formalism in Möbius", in *Proceedings of the International Conference on Dependable Systems and Networks*, 2013, ISBN: 9781467364713. DOI: 10.1109/DSN.2013.6575362.

[62] T. R. McEvoy and S. D. Wolthusen, "A formal adversary capability model for SCADA environments", in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6712 LNCS, Springer, Berlin, Heidelberg, 2011, pp. 93–103, ISBN: 9783642216930. DOI: 10.1007/978-3-642-21694-7{_}8. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-21694-7_8.

[63] A. Teixeira, D. Pérez, H. Sandberg, and K. H. Johansson, "Attack models and scenarios for networked control systems", *HiCoNS'12 - Proceedings of the 1st ACM International Conference on High Confidence Networked Systems*, pp. 55–64, 2012. DOI: 10.1145/2185505.2185515.

[64] M. Rocchetto and N. O. Tippenhauer, "On attacker models and profiles for cyber-physical systems", in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9879 LNCS, Singapore: Springer Verlag, 2016, pp. 427–449, ISBN: 9783319457406. DOI: 10.1007/978-3-319-45741-3{_}22.

[65] R. Heckman, "Attacker Classification to Aid Targeting Critical Systems for Threat Modelling and Security Review", ROCKYH, Tech. Rep., 2005.

[66] S. M. Dibaji, M. Pirani, D. B. Flamholz, A. M. Annaswamy, K. H. Johansson, and A. Chakrabortty, *A systems and control perspective of CPS security*, 2019. DOI: 10.1016/j.arcontrol.2019.04.011.

[67] C. Cheh, G. A. Weaver, and W. H. Sanders, "Cyber-Physical Topology Language: Definition, Operations, and Application", in *Proceedings - 2015 IEEE 21st Pacific Rim International Symposium on Dependable Computing, PRDC 2015*, 2016, pp. 60–69, ISBN: 9781467393768. DOI: 10.1109/PRDC.2015.20.

[68] J. Y. Choley, F. Mhenni, N. Nguyen, and A. Baklouti, "Topology-based Safety Analysis for Safety Critical CPS", in *Procedia Computer Science*, vol. 95, Elsevier B.V., Jan. 2016, pp. 32–39. DOI: 10.1016/j.procs.2016.09.290.

[69] P. Beling, B. Horowitz, C. Fleming, S. Adams, G. Bakirtzis, T. Sherburne, C. Elks, A. G. Collins, C. Deloglos, and B. J. Simon, "WRT-1013 : Security Engineering - 2019 Technical Report SERC-2020-TR-004", Systems Engineering Research Center (SERC), Tech. Rep., 2020. DOI: SERC-2020-TR-004.

[70] G. Bakirtzis, "Compositional Cyber-Physical Systems Theory", Ph.D. dissertation, Sep. 2021. DOI: 10.18130/xn8v-5d89. [Online]. Available: http://arxiv.org/abs/2109.04858%20http://dx.doi.org/10.18130/xn8v-5d89.

[71] S. Mili, N. Nguyen, and R. Chelouah, "Transformation-Based Approach to Security Verification for Cyber-Physical Systems", *IEEE Systems Journal*, vol. PP, pp. 1–12, 2019, ISSN: 1932-8184. DOI: 10.1109/jsyst.2019.2923818.

[72] Y. Mo, T. H. J. Kim, K. Brancik, D. Dickinson, H. Lee, A. Perrig, and B. Sinopoli, "Cyber-physical security of a smart grid infrastructure", *Proceedings of the IEEE*, vol. 100, no. 1, pp. 195–209, 2012, ISSN: 00189219. DOI: 10.1109/JPROC.2011.2161428.

[73] R. Vigo, "The Cyber-Physical Attacker", in *Lecture Notes in Computer Science, computer safety reliability and security*, vol. 9, Oct. 2012, pp. 347–356, ISBN: 9783540465591. DOI: 10.1007/978-3-642-33675-1. [Online]. Available: http://www.mendeley.com/research/lecture-notes-computer-science-2/.

[74] D. Basin, S. Capkun, P. Schaller, and B. Schmidt, "Formal reasoning about physical properties of security protocols", *ACM Transactions on Information and System Security*, vol. 14, no. 2, pp. 1–28, 2011, ISSN: 10949224. DOI: 10.1145/2019599.2019601.

[75] E. LeMay, M. D. Ford, K. Keefe, W. H. Sanders, and C. Muehrcke, "Model-based security metrics using ADversary VIew Security Evaluation (ADVISE)", *Proceedings of the 2011 8th International Conference on Quantitative Evaluation of Systems, QEST 2011*, no. October, pp. 191–200, 2011. DOI: 10.1109/QEST.2011.34.

[76] G. Bakirtzis, T. Sherburne, S. Adams, B. M. Horowitz, P. A. Beling, and C. H. Fleming, *An Ontological Metamodel for Cyber-Physical System Safety, Security, and Resilience Coengineering*, 2020.

[77] C. Cheh and W. H. Sanders, "The Cyber-Physical Topology Language: Definition and Operations", Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2014.

[78] *SysML Open Source Project*. [Online]. Available: https://sysml.org/.

[79] "ATT&CK for Industrial Control Systems", in *MITRE*, Jan. 2020. DOI: 10.1109/isie.2010.5636886. [Online]. Available: https://collaborate.mitre.org/attackics/index.php/.

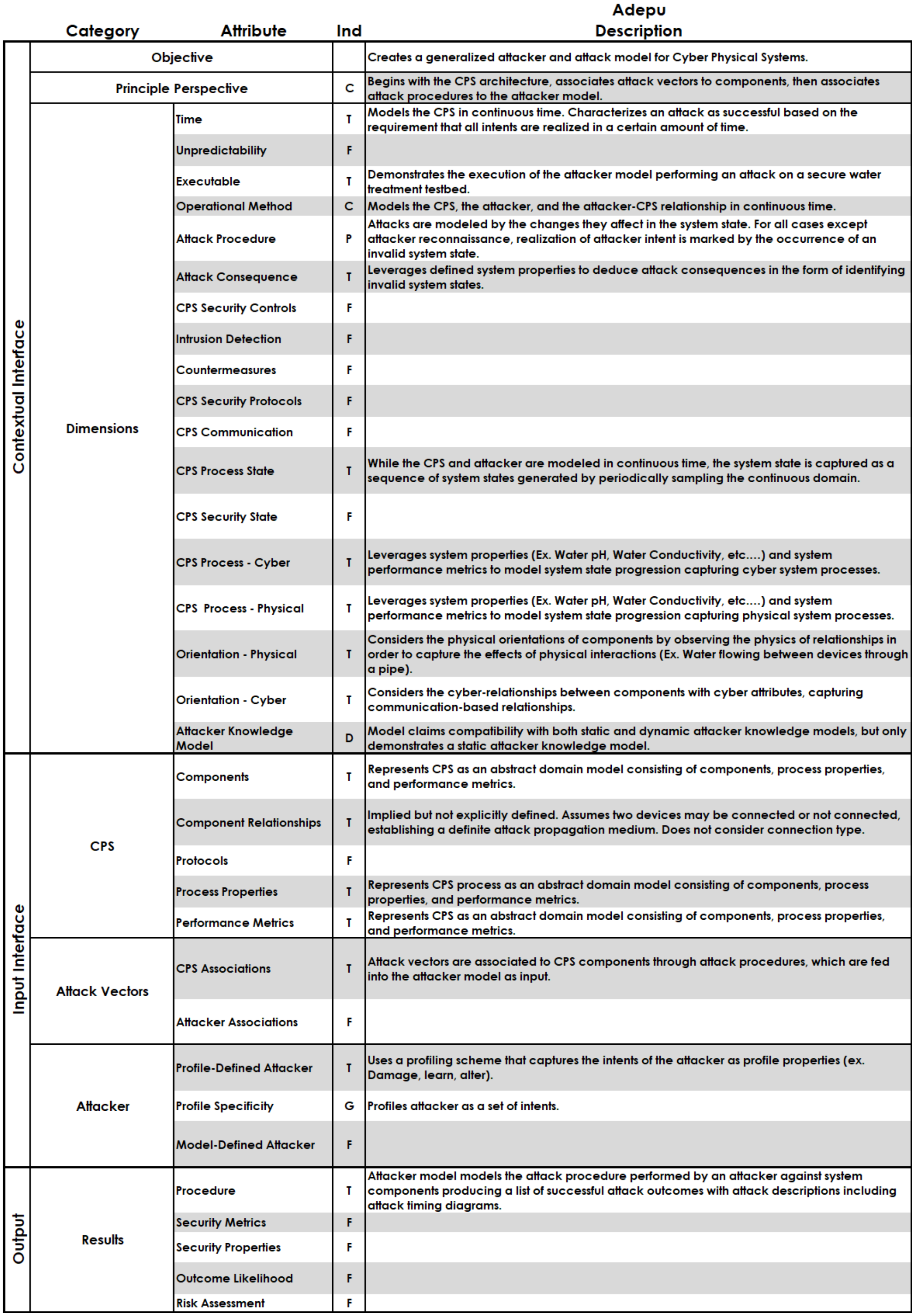
[80] S. K. Patil and R. Kant, "A fuzzy AHP-TOPSIS framework for ranking the solutions of Knowledge Management adoption in Supply Chain to overcome its barriers", *Expert Systems with Applications*, vol. 41, no. 2, pp. 679–693, 2014, ISSN: 09574174. DOI: 10.1016/j.eswa.2013.07.093. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2013.07.093.

APPENDIX

Appendix A

Attacker Model Characterizations

Table A.1 The attacker model characterization for Adepu's attacker model [57]



				Adepu
	Category	Attribute	Ind	Description
Contextual Interface	Objective			Creates a generalized attacker and attack model for Cyber Physical Systems.
	Principle Perspective		C	Begins with the CPS architecture, associates attack vectors to components, then associates attack procedures to the attacker model.
	Dimensions	Time	T	Models the CPS in continuous time. Characterizes an attack as successful based on the requirement that all intents are realized in a certain amount of time.
		Unpredictability	F	
		Executable	T	Demonstrates the execution of the attacker model performing an attack on a secure water treatment testbed.
		Operational Method	C	Models the CPS, the attacker, and the attacker-CPS relationship in continuous time.
		Attack Procedure	P	Attacks are modeled by the changes they affect in the system state. For all cases except attacker reconnaissance, realization of attacker intent is marked by the occurrence of an invalid system state.
		Attack Consequence	T	Leverages defined system properties to deduce attack consequences in the form of identifying invalid system states.
		CPS Security Controls	F	
		Intrusion Detection	F	
		Countermeasures	F	
		CPS Security Protocols	F	
		CPS Communication	F	
		CPS Process State	T	While the CPS and attacker are modeled in continuous time, the system state is captured as a sequence of system states generated by periodically sampling the continuous domain.
		CPS Security State	F	
		CPS Process - Cyber	T	Leverages system properties (Ex. Water pH, Water Conductivity, etc....) and system performance metrics to model system state progression capturing cyber system processes.
		CPS Process - Physical	T	Leverages system properties (Ex. Water pH, Water Conductivity, etc....) and system performance metrics to model system state progression capturing physical system processes.
		Orientation - Physical	T	Considers the physical orientations of components by observing the physics of relationships in order to capture the effects of physical interactions (Ex. Water flowing between devices through a pipe).
		Orientation - Cyber	T	Considers the cyber-relationships between components with cyber attributes, capturing communication-based relationships.
		Attacker Knowledge Model	D	Model claims compatibility with both static and dynamic attacker knowledge models, but only demonstrates a static attacker knowledge model.
Input Interface	CPS	Components	T	Represents CPS as an abstract domain model consisting of components, process properties, and performance metrics.
		Component Relationships	T	Implied but not explicitly defined. Assumes two devices may be connected or not connected, establishing a definite attack propagation medium. Does not consider connection type.
		Protocols	F	
		Process Properties	T	Represents CPS process as an abstract domain model consisting of components, process properties, and performance metrics.
		Performance Metrics	T	Represents CPS as an abstract domain model consisting of components, process properties, and performance metrics.
	Attack Vectors	CPS Associations	T	Attack vectors are associated to CPS components through attack procedures, which are fed into the attacker model as input.
		Attacker Associations	F	
	Attacker	Profile-Defined Attacker	T	Uses a profiling scheme that captures the intents of the attacker as profile properties (ex. Damage, learn, alter).
		Profile Specificity	G	Profiles attacker as a set of intents.
		Model-Defined Attacker	F	
Output	Results	Procedure	T	Attacker model models the attack procedure performed by an attacker against system components producing a list of successful attack outcomes with attack descriptions including attack timing diagrams.
		Security Metrics	F	
		Security Properties	F	
		Outcome Likelihood	F	
		Risk Assessment	F	

Table A.2 The attacker model characterization for Basin's attacker model [74]

				Basin
	Category	Attribute	Ind	Description
Contextual Interface	Objective			Leverages formal methods to reason about physical properties and communications protocols.
	Principle Perspective		V	Attacker model orients around vulnerabilities.
	Dimensions	Time	T	System operational semantics captures essential physical properties of space and time. Models communication time delays based on physical orientation.
		Unpredictability	F	
		Executable	T	Leverages formal methods to execute attacker model.
		Operational Method	C	Models the progression of the attack in continuous time
		Attack Procedure	P	Interaction of honest and dishonest agents in the network is modeled as a continuous-time interaction process where attacker success is marked by system behavior outside of expected behavior specified by security protocols.
		Attack Consequence	F	
		CPS Security Controls	F	
		Intrusion Detection	F	
		Countermeasures	F	
		CPS Security Protocols	T	Communication security protocols are defined for honest agents, whereas dishonest agents can violate the communication protocols.
		CPS Communication	T	Models cyber and physical aspects of component communication, including communication delays based on physical orientation.
		CPS Process State	T	Modeled capturing all system component and attacker functions in real-time as the system state.
		CPS Security State	T	Captures the security state of the system as an evaluation of whether the system is following or violating protocol rules.
		CPS Process - Cyber	T	Models the cyber processes of the system including all agents and communications.
		CPS Process - Physical	T	Models the physical process of communications signals being transmitted between devices.
		Orientation - Physical	T	Models the physical communication aspects based on the physical orientation of various honest and dishonest agents.
		Orientation - Cyber	T	Models the orientation of the agents in the network based on communication capabilities.
		Attacker Knowledge Model	D	Implements a learning attacker knowledge model.
Input Interface	CPS	Components	T	CPS is defined with all components and communications protocols.
		Component Relationships	T	Model is primarily concerned with wireless communications. Expects definitions of all communication paths between devices including expected communications protocols.
		Protocols	T	Formalized describes security protocols implemented at the device level for wireless communications.
		Process Properties	T	Process properties that define communication procedures are expected for each component.
		Performance Metrics	F	
	Attack Vectors	CPS Associations	F	
		Attacker Associations	F	
	Attacker	Profile-Defined Attacker	F	
		Profile Specificity	F	
		Model-Defined Attacker	T	Model defines all communicators as agents, allowing honest and dishonest agents. Behavior of dishonest agents is expected to be defined per-agent.
Output	Results	Procedure	F	
		Security Metrics	T	Models security protocols and detection of security protocol violation.
		Security Properties	T	Produces communications security protocols for several attack scenarios.
		Outcome Likelihood	F	
		Risk Assessment	F	

Table A.3 The attacker model characterization for Deloglos' attacker model [5]

	Category	Attribute	Ind	Deloglos Description
Contextual Interface	Objective			A generalized attacker model for reasoning about security probabilistic attacker models.
	Principle Perspective		C	Begins with a system architecture perspective then associates attack vectors to system components via attacker behavior.
	Dimensions	Time	F	
		Unpredictability	T	Implements a dynamic behavioral attacker model that makes the nature of the attacker unpredictable.
		Executable	T	Is demonstrated as an executable feedback loop.
		Operational Method	S	Executes as a sequential set of operations.
		Attack Procedure	Q	Action selection based on property qualification .
		Attack Consequence	F	
		CPS Security Controls	F	
		Intrusion Detection	F	
		Countermeasures	F	
		CPS Security Protocols	F	
		CPS Communication	T	Models types of communication between components.
		CPS Process State	T	Maintains the state of the CPS process.
		CPS Security State	T	Maintains the security state of the CPS.
		CPS Process - Cyber	F	
		CPS Process - Physical	F	
		Orientation - Physical	F	
		Orientation - Cyber	T	Models the relational orientation of all components in the CPS.
		Attacker Knowledge Model	D	Implements a learning attacker knowledge model.
Input Interface	CPS	Components	T	Expects a system description including all component definitions using a node-edge methodology.
		Component Relationships	T	Implements a node-edge CPS architecture methodology where edges define relationships between components include relevant details such as communication scheme (ethernet, Modbus, analog).
		Protocols	F	
		Process Properties	F	
		Performance Metrics	F	
	Attack Vectors	CPS Associations	T	Associates attack vectors to CPS components through attributization. Each CPS component is assigned attributes (USB, Ethernet, Physical Accessible). Attack vectors are assigned attributes likewise. The attributes are then used to associate components to attack vectors.
		Attacker Associations	T	Applies the same properties to attack vectors that are used to characterize attacker profiles. Vulnerability properties are associated to attacker properties using a probabilistic function to predict vulnerability selection.
	Attacker	Profile-Defined Attacker	T	Uses a property-based profiling scheme that allows users to define properties (cost, knowledge required, etc.…).
		Profile Specificity	G	While the model allows user-define properties, it proposes 6 pre-defined properties that capture the nature of common types of attackers.
		Model-Defined Attacker	F	
Output	Results	Procedure	T	Models attack procedures against the system.
		Security Metrics	F	
		Security Properties	F	
		Outcome Likelihood	T	Calculates the likelihood of attacker decisions and actions using a probabilistic correlation function.
		Risk Assessment	F	

Table A.4 The attacker model characterization for Ekelhart's attacker model [59]

				Ekelhart
	Category	Attribute	Ind	Description
Contextual Interface	Objective			Uses a discrete-event simulation approach to integrate attacker behavior into IT security analysis.
	Principle Perspective		V	Begins with a vulnerability perspective by composing an abstract attack graph.
	Dimensions	Time	F	
		Unpredictability	F	
		Executable	T	Executes the attacker model using the MASON Java discrete-event simulation engine.
		Operational Method	S	Implements attacks using a discrete-event simulation approach.
		Attack Procedure	Q	Assigns preconditions to all possible attack actions. If an action meets all preconditions for a particular target, it is selected.
		Attack Consequence	T	Provides statistics for the number of times attacks succeed and fail, including with respect to countermeasures.
		CPS Security Controls	T	Models control actions, the conditions for activating control actions, and the effects of activating control actions.
		Intrusion Detection	T	Includes intrusion detection as a control method, modeling the effects of intrusion detection on the attack outcome.
		Countermeasures	T	Allows definition of control attributes and simulation of attacks that result in control actions.
		CPS Security Protocols	F	
		CPS Communication	F	
		CPS Process State	T	Models the process state of the CPS, including changes to the system during the attack process based on how the attack effects various components
		CPS Security State	T	Models the security state of the CPS, including how security actions by the intrusion detection system alter the security state of various components
		CPS Process - Cyber	F	
		CPS Process - Physical	F	
		Orientation - Physical	F	
		Orientation - Cyber	T	Models the relational orientation of all components including changes to orientation based on attack progress and security countermeasure application.
		Attacker Knowledge Model	D	Allows attacker to be modeled with varying levels of knowledge and obtain more knowledge as the attack progresses.
Input Interface	CPS	Components	T	Utilizes a system model composed of assets which include system components.
		Component Relationships	T	Defined types of relationships between assets such as "connected" between devices or "installed" between devices and software applications.
		Protocols	F	
		Process Properties	F	
		Performance Metrics	F	
	Attack Vectors	CPS Associations	T	Defined preconditions for attacker actions and associates actions to CPS components with descriptions that satisfy the preconditions.
		Attacker Associations	T	Attacker has to be attributized with motivations, goals, and interaction capabilities.
	Attacker	Profile-Defined Attacker	T	Uses a property-based profiling scheme that uses attributes such as monetary and time budget, motivation, and risk preferences.
		Profile Specificity	S	Uses a property-based profiling scheme that uses attributes such as monetary and time budget, motivation, and risk preferences.
		Model-Defined Attacker	F	
Output	Results	Procedure	T	Integrates a knowledge base, an attack graph generation component, and a discrete-event engine with attacker behavior to simulate attack procedure.
		Security Metrics	F	
		Security Properties	F	
		Outcome Likelihood	T	Implements a discrete-event engine to likely attack outcomes.
		Risk Assessment	F	

Table A.5 The attacker model characterization for LeMay's attacker model [54]

				Le May
	Category	Attribute	Ind	Description
Contextual Interface	Objective			Implements a generalized adversary-driven state-based system security evaluation.
	Principle Perspective		V	Custom attack execution graph begins with a notion of action relationships, with the CPS orientation assumed by action relationship.
	Dimensions	Time	T	Models the amount of time it takes the attacker to attempt an attack step.
		Unpredictability	F	
		Executable	T	Is demonstrated as a full executable program in an external tool.
		Operational Method	S	Sequential Operation
		Attack Procedure	P	Uses probabilistic functions to calculate the probability of attack actions begin selected.
		Attack Consequence	T	Evaluates the effects of action on different system components.
		CPS Security Controls	T	Includes the effects of countermeasures as a variable in the function defining the outcome probability for an attack.
		Intrusion Detection	T	Allows for each attack a probability that the attack will be detected when a certain outcome occurs from the attack.
		Countermeasures	T	Models the deployment of countermeasures via the inclusion of a likelihood to be detected probability weighting.
		CPS Security Protocols	F	
		CPS Communication	F	
		CPS Process State	T	Implements formal method for capturing the system state and the state of components at all points in time.
		CPS Security State	T	Tracks the attack process using event metrics which capture the system security state
		CPS Process - Cyber	F	
		CPS Process - Physical	F	
		Orientation - Physical	F	
		Orientation - Cyber	T	System is defined using a custom attack execution graph (AEG). Cyber orientation for certain components can be inferred from the AEG, but a details CPS architecture is not required once the AEG has been created.
		Attacker Knowledge Model	D	Implements a learning attacker knowledge model.
Input Interface	CPS	Components	F	Custom attack execution graph assumes vulnerability associations are composed from preliminary step of architecture evaluation.
		Component Relationships	F	Custom attack execution graph assumes vulnerability associations are composed from preliminary step of architecture evaluation.
		Protocols	F	
		Process Properties	F	
		Performance Metrics	F	
	Attack Vectors	CPS Associations	F	Custom attack execution graph assumes vulnerability associations are composed from preliminary step of architecture evaluation.
		Attacker Associations	F	Custom attack execution graph assumes vulnerability associations are composed from preliminary step of architecture evaluation.
	Attacker	Profile-Defined Attacker	T	Defines the attacker behavior (titled adversary profile) as a property-based profile which includes as properties the attack skill level, the attack goal, the payoff, and weightings and functional relationships for cost, payoff, and detection probability
		Profile Specificity	S	A specific definition for the attacker profile is provided including all attributes included.
		Model-Defined Attacker	F	
Output	Results	Procedure	T	Models the progression of the attack.
		Security Metrics	T	Reasons about how various attack actions lead to violations of security metrics.
		Security Properties	F	
		Outcome Likelihood	F	
		Risk Assessment	F	

Table A.6 The attacker model characterization for McEvoy's attacker model [62]

				McEvoy
	Category	Attribute	Ind	Description
Contextual Interface	Objective			Implements a formal adversary model using π-calculus to model attacker behavior on a SCADA system.
	Principle Perspective		C	Begins with a component orientation around a SCADA system design.
	Dimensions	Time	F	
		Unpredictability	F	
		Executable	T	Uses π-calculus to implement an executable model.
		Operational Method	S	Executes as a sequence of state.
		Attack Procedure	P	Attacks and attack results are modeled as real-time processes.
		Attack Consequence	F	
		CPS Security Controls	T	Models intrusion detection.
		Intrusion Detection	T	Leverages the AM process model to detect undesired behavior, detecting different types of attacker intrusion.
		Countermeasures	F	
		CPS Security Protocols	F	
		CPS Communication	T	Specific to SCADA systems, captures the communication processes using π-Calculus.
		CPS Process State	T	Captures the behavior of system components and communications between components as part of a π-Calculus modeling process.
		CPS Security State	T	Captures the behavior of system components and communications between components and leverages them to deduce CPS behavior out of acceptable behavioral bounds.
		CPS Process - Cyber	T	Models ongoing SCADA processes and the effects of an attacker on the process.
		CPS Process - Physical	F	
		Orientation - Physical	F	
		Orientation - Cyber	T	Models the cyber orientation of components using system processes and control loops.
		Attacker Knowledge Model	S	Assumes malicious agent begins attack with all capabilities and knowledge of the system.
Input Interface	CPS	Components	T	Characterizes the system explicitly as a SCADA system, composed of processes, sensors, and actuators.
		Component Relationships	T	Defines the System as a SCADA system that formally defines communications between devices as a part of the SCADA system.
		Protocols	F	
		Process Properties	T	Captures expected SCADA operations including supervisor processes, and control loops.
		Performance Metrics	F	
	Attack Vectors	CPS Associations	F	
		Attacker Associations	F	
	Attacker	Profile-Defined Attacker	F	
		Profile Specificity	F	
		Model-Defined Attacker	T	Defines a group of attackers, each being defined as an agent-based adversary capabilities model.
Output	Results	Procedure	F	Models various attack strategies using π-Calculus.
		Security Metrics	F	
		Security Properties	F	
		Outcome Likelihood	F	
		Risk Assessment	F	

Table A.7 The attacker model characterization for Mo's attacker model [72]

	Category	Attribute	Ind	Mo Description
Contextual Interface	Objective			Reasons about the cyber-physical security of a smart grid infrastructure.
	Principle Perspective		I	CPS Architecture and model relationships are explicitly defined using a system-theoretic approach.
	Dimensions	Time	T	Models the system as a function of continuous time. Uses time to describe the attack process, capturing how long it takes the attacker to perform certain actions.
		Unpredictability	F	
		Executable	T	A system theoretic approach is used to simulate attacker behavior.
		Operational Method	C	Attacker/system interactions modeled as a continuous-time process
		Attack Procedure	P	Attacks are modeled using system-theoretic continuous approaches.
		Attack Consequence	F	
		CPS Security Controls	T	Uses system-theoretic models and countermeasures to detect certain cyber attacks.
		Intrusion Detection	T	Detects intrusion based on detection of violations to security requirements using a system-theoretic model.
		Countermeasures	T	Models the effects of countermeasures.
		CPS Security Protocols	F	
		CPS Communication	T	Models communications between components using a system-theoretic model.
		CPS Process State	T	System-theoretic model allows the deduction of the state of all components in the CPS at all times.
		CPS Security State	T	The model implements a system-theoretic technique to detect violations to the security state of the system.
		CPS Process - Cyber	T	AM Models the cyber processes of the system and attacker interactions.
		CPS Process - Physical	F	
		Orientation - Physical	F	
		Orientation - Cyber	T	Models the cyber orientation of components and the attacker.
		Attacker Knowledge Model	S	Attacker behavior is defined as part of the AM and in all examples is defined as static assuming all necessary knowledge to perform the modelled attack.
Input Interface	CPS	Components	T	AM is geared toward smart power grids and defines a static architecture of four categories of components (Generation, Transmission, Distribution, and Consumption).
		Component Relationships	T	Defines a static set of communications between categories of components and the control center which is the set of wide-area networks (WAN), neighbor-area network (NAN), and home-area network (HAN).
		Protocols	F	
		Process Properties	F	
		Performance Metrics	F	
	Attack Vectors	CPS Associations	T	Defines a static set of attack vectors which are explicitly defined in a specific context of system components.
		Attacker Associations	T	Defines a static set of attack vectors which are explicitly defined in a specific context of attackers.
	Attacker	Profile-Defined Attacker	F	
		Profile Specificity	F	
		Model-Defined Attacker	T	The behavior of attackers is reasoned about by the author for the specific attacks on a specific system architecture.
Output	Results	Procedure	T	Models attack procedures using discrete-time calculus for smart-grid infrastructure
		Security Metrics	F	
		Security Properties	T	The attacker model formally derives countermeasures for various attack scenarios as a modeling result.
		Outcome Likelihood	F	
		Risk Assessment	F	

Table A.8 The attacker model characterization for Monteuuis' attacker model [56]

				Monteuuis
	Category	Attribute	Ind	Description
Contextual Interface	Objective			Reasons about attacker behavior in connected and automated vehicles.
	Principle Perspective		I	Reasoning about attackers is developed in the context of automated vehicles.
	Dimensions	Time	F	
		Unpredictability	F	
		Executable	F	
		Operational Method	F	
		Attack Procedure	F	
		Attack Consequence	F	
		CPS Security Controls	T	The AM reasons about countermeasures for the defined attack scenarios.
		Intrusion Detection	F	
		Countermeasures	T	Reasons on the effects of countermeasures on achieving security goals.
		CPS Security Protocols	F	
		CPS Communication	T	The AM reasons about the effect of component communication on various attacks.
		CPS Process State	T	Reasons about the state of system processes during attacks.
		CPS Security State	T	Reasons about the state of system security during the attacks.
		CPS Process - Cyber	F	
		CPS Process - Physical	F	
		Orientation - Physical	F	
		Orientation - Cyber	T	Reasons about cyber relationships between CPS components.
		Attacker Knowledge Model	S	Reasoning about attackers follows a static attacker knowledge model.
Input Interface	CPS	Components	T	The AM is geared toward smart cars and implements a static smart-car architecture, defining all system components
		Component Relationships	T	The AM is geared toward smart cars and implements a static smart-car architecture, defining all communication paths between system components
		Protocols	F	
		Process Properties	F	
		Performance Metrics	F	
	Attack Vectors	CPS Associations	F	
		Attacker Associations	F	
	Attacker	Profile-Defined Attacker	F	
		Profile Specificity	F	
		Model-Defined Attacker	T	Attacker Model is explicitly defined by the author. The other identifies several different attacks and how different attackers perform different attacks on the static system architecture.
Output	Results	Procedure	T	Reasons about attack procedures in automated vehicles.
		Security Metrics	T	Reasons about the effect of attacks on various models of security goals.
		Security Properties	F	
		Outcome Likelihood	F	
		Risk Assessment	F	

Table A.9 The attacker model characterization for Orojloo' attacker model [55]

	Category	Attribute	Ind	Orojloo Description
Contextual Interface	Objective			Applies fuzzy theory to predict the behavior of attackers and the consequences of attacks against cyber-physical systems.
	Principle Perspective		C	Begins with a component perspective of the CPS architecture.
	Dimensions	Time	T	AM models the system using discrete-time functions. Attack model implemented using a discrete-time Markov chain.
		Unpredictability	F	
		Executable	T	Demonstrates the execution of the attacker model as a result of fuzzy operations.
		Operational Method	D	Fuzzy theory modeled in discrete time.
		Attack Procedure	P	Models the attack procedure as a real-time process, evaluating the attack as a real-time interaction with the system.
		Attack Consequence	T	Proposes a method for calculating the risk of individual attacks.
		CPS Security Controls	F	
		Intrusion Detection	F	
		Countermeasures	F	
		CPS Security Protocols	F	
		CPS Communication	F	
		CPS Process State	T	Captures the state of the system using a discrete-time modelling technique.
		CPS Security State	T	Leverages the system model to evaluate the security state of the system as time progresses.
		CPS Process - Cyber	T	Models the cyber processes of the system.
		CPS Process - Physical	T	Models the physical processes of the system.
		Orientation - Physical	F	Unsure? May include: Considers the physical orientation of components and the effects of physical orientation on physical and cyber processes.
		Orientation - Cyber	T	Considers the cyber-relationships between components.
		Attacker Knowledge Model	S	Implements a static attacker knowledge model.
Input Interface	CPS	Components	T	AM begins with an attack tree which is a composite of attack steps, which may or may not be associated to components. The AM provides guidelines for composing the attack tree from a system architecture description .
		Component Relationships	F	
		Protocols	F	
		Process Properties	T	Demonstrates methods for describing system process properties using fuzzy logic.
		Performance Metrics	F	
	Attack Vectors	CPS Associations	F	
		Attacker Associations	T	Both attackers and attack steps are assigned 4-property profile values consisting of knowledge, access, user interaction, and skill, which of formulaically associated.
	Attacker	Profile-Defined Attacker	T	Property-based profiling schema consisting of knowledge, access, user interaction, and skill
		Profile Specificity	S	Property-based profiling schema consisting of knowledge, access, user interaction, and skill
		Model-Defined Attacker	F	
Output	Results	Procedure	T	AM models the attack procedure, specifically identifying how varying attacker behavior influences the actions the attacker takes using formal methods.
		Security Metrics	F	
		Security Properties	F	
		Outcome Likelihood	T	At every potential attacker decision, provides decision likelihoods for attacker behavior.
		Risk Assessment	T	Derives a model to calculate the risk of various attacks.

Table A.10 The attacker model characterization for Teixeira' attacker model [63]

				Teixeira
	Category	Attribute	Ind	Description
Contextual Interface	Objective			Leverages discrete-time state space equations to model the relationship between the attacker and the system for analyzing networked control systems.
	Principle Perspective		C	Begins from the architectural perspective of the system.
	Dimensions	Time	T	Models the physical plant in a discrete-time state-space form.
		Unpredictability	F	
		Executable	T	Executes the system as a discrete-time state-space model.
		Operational Method	D	Models the system and attacker in a discrete-time state-space form.
		Attack Procedure	P	Attack is modeled using discrete-time functions.
		Attack Consequence	T	Models the effects on physical operations caused by attacks.
		CPS Security Controls	F	
		Intrusion Detection	F	
		Countermeasures	F	
		CPS Security Protocols	F	
		CPS Communication	T	Models communications between components using a discrete-time state-space form.
		CPS Process State	T	Maintains a state-space representation of the discrete-time system.
		CPS Security State	T	Leverages the state-space system model to determine the security state of the system.
		CPS Process - Cyber	T	Models the Cyber-processes of the system using the discrete-time state-space system model.
		CPS Process - Physical	T	Models the physical processes of the system using the discrete-time state-space system model.
		Orientation - Physical	T	Captures the physical orientation of some components as a part of the proposed methodology for modeling physical attacks.
		Orientation - Cyber	T	Captures the cyber orientation of system components.
		Attacker Knowledge Model	S	Attacker uses a static knowledge model of consisting of the plant, control algorithms, and the anomaly detector.
Input Interface	CPS	Components	T	Defines a plant as a discrete-time state form which captures controllers, sensors, and actuators as control data, sensor data, and actuator data at a given point in discrete time.
		Component Relationships	T	Models communication in a plant as the discrete-time flow of data between system components.
		Protocols	F	
		Process Properties	T	Defines normal plant behavior as properties represented in discrete-time functions.
		Performance Metrics	F	
	Attack Vectors	CPS Associations	F	Assumes all attack vectors are defined with associations to which components they are valid
		Attacker Associations	F	
	Attacker	Profile-Defined Attacker	F	
		Profile Specificity	F	
		Model-Defined Attacker	T	Attacks are individually defined as functional operations using discrete-time logic.
Output	Results	Procedure	T	Models attack procedure for SCADA systems.
		Security Metrics	F	
		Security Properties	T	Develops security properties for deviations from acceptable system operation boundaries.
		Outcome Likelihood	F	
		Risk Assessment	F	

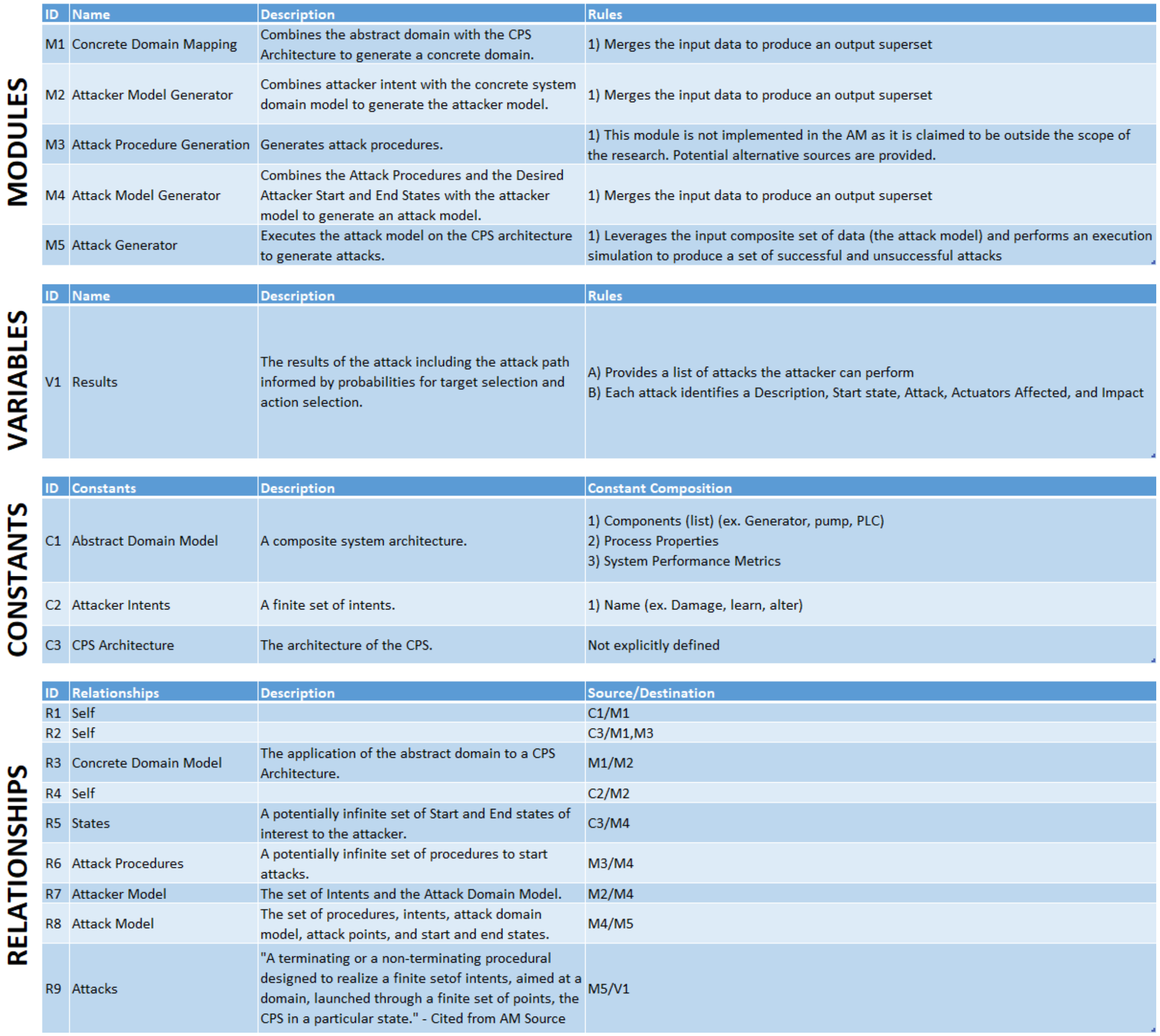
Table A.11 The attacker model characterization for Vigo' attacker model [73]

	Category	Attribute	Ind	Vigo Description
Contextual Interface	Objective			Uses a protocol perspective to reason about the behavior of a cyber-physical attacker.
	Principle Perspective		C	The approach begins with an analysis of a CPS architecture.
	Dimensions	Time	F	
		Unpredictability	F	
		Executable	F	
		Operational Method	F	
		Attack Procedure	F	
		Attack Consequence	F	
		CPS Security Controls	F	
		Intrusion Detection	F	
		Countermeasures	F	
		CPS Security Protocols	F	
		CPS Communication	T	Reference model defines the cyber topology of the system.
		CPS Process State	F	
		CPS Security State	F	
		CPS Process - Cyber	F	
		CPS Process - Physical	F	
		Orientation - Physical	T	Reasons about the physical orientation of system components.
		Orientation - Cyber	T	Reference model defines the cyber topology of the system.
		Attacker Knowledge Model	D	AM assumes a static knowledge model for the attacker model.
Input Interface	CPS	Components	T	Formally defines the CPS as a pair of components of the system and the topology of the system.
		Component Relationships	T	Each component is defined with a list of interfaces (communication methods) toward the environment and toward other nodes.
		Protocols	F	
		Process Properties	F	
		Performance Metrics	F	
	Attack Vectors	CPS Associations	T	Each attack is associated to components of the system using the basic actions: remove, read/write, reveal, reprogram, starve.
		Attacker Associations	F	
	Attacker	Profile-Defined Attacker	T	Attacker profile defined as a generic set of properties
		Profile Specificity	G	Profile properties defined as generic
		Model-Defined Attacker	F	
Output	Results	Procedure	T	AM models the attack procedure performed by an adversary against system components.
		Security Metrics	F	
		Security Properties	F	
		Outcome Likelihood	F	
		Risk Assessment	F	

Appendix B

Attacker Model Functional Representations

Table B.1 Attacker model functional representation data for Adepu's attacker model in [57]



MODULES

ID	Name	Description	Rules
M1	Concrete Domain Mapping	Combines the abstract domain with the CPS Architecture to generate a concrete domain.	1) Merges the input data to produce an output superset
M2	Attacker Model Generator	Combines attacker intent with the concrete system domain model to generate the attacker model.	1) Merges the input data to produce an output superset
M3	Attack Procedure Generation	Generates attack procedures.	1) This module is not implemented in the AM as it is claimed to be outside the scope of the research. Potential alternative sources are provided.
M4	Attack Model Generator	Combines the Attack Procedures and the Desired Attacker Start and End States with the attacker model to generate an attack model.	1) Merges the input data to produce an output superset
M5	Attack Generator	Executes the attack model on the CPS architecture to generate attacks.	1) Leverages the input composite set of data (the attack model) and performs an execution simulation to produce a set of successful and unsuccessful attacks

VARIABLES

ID	Name	Description	Rules
V1	Results	The results of the attack including the attack path informed by probabilities for target selection and action selection.	A) Provides a list of attacks the attacker can perform B) Each attack identifies a Description, Start state, Attack, Actuators Affected, and Impact

CONSTANTS

ID	Constants	Description	Constant Composition
C1	Abstract Domain Model	A composite system architecture.	1) Components (list) (ex. Generator, pump, PLC) 2) Process Properties 3) System Performance Metrics
C2	Attacker Intents	A finite set of intents.	1) Name (ex. Damage, learn, alter)
C3	CPS Architecture	The architecture of the CPS.	Not explicitly defined

RELATIONSHIPS

ID	Relationships	Description	Source/Destination
R1	Self		C1/M1
R2	Self		C3/M1,M3
R3	Concrete Domain Model	The application of the abstract domain to a CPS Architecture.	M1/M2
R4	Self		C2/M2
R5	States	A potentially infinite set of Start and End states of interest to the attacker.	C3/M4
R6	Attack Procedures	A potentially infinite set of procedures to start attacks.	M3/M4
R7	Attacker Model	The set of Intents and the Attack Domain Model.	M2/M4
R8	Attack Model	The set of procedures, intents, attack domain model, attack points, and start and end states.	M4/M5
R9	Attacks	"A terminating or a non-terminating procedural designed to realize a finite setof intents, aimed at a domain, launched through a finite set of points, the CPS in a particular state." - Cited from AM Source	M5/V1

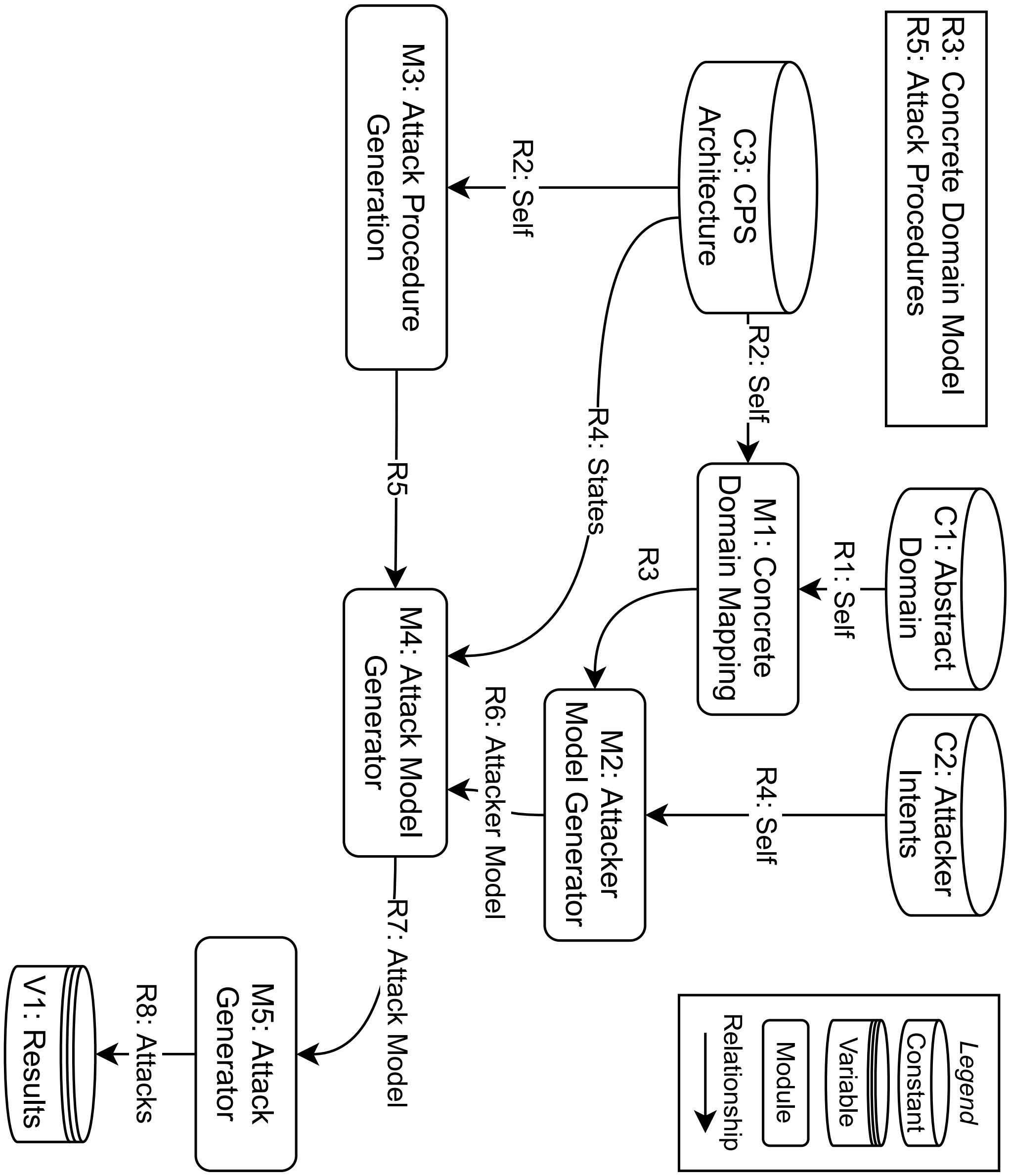
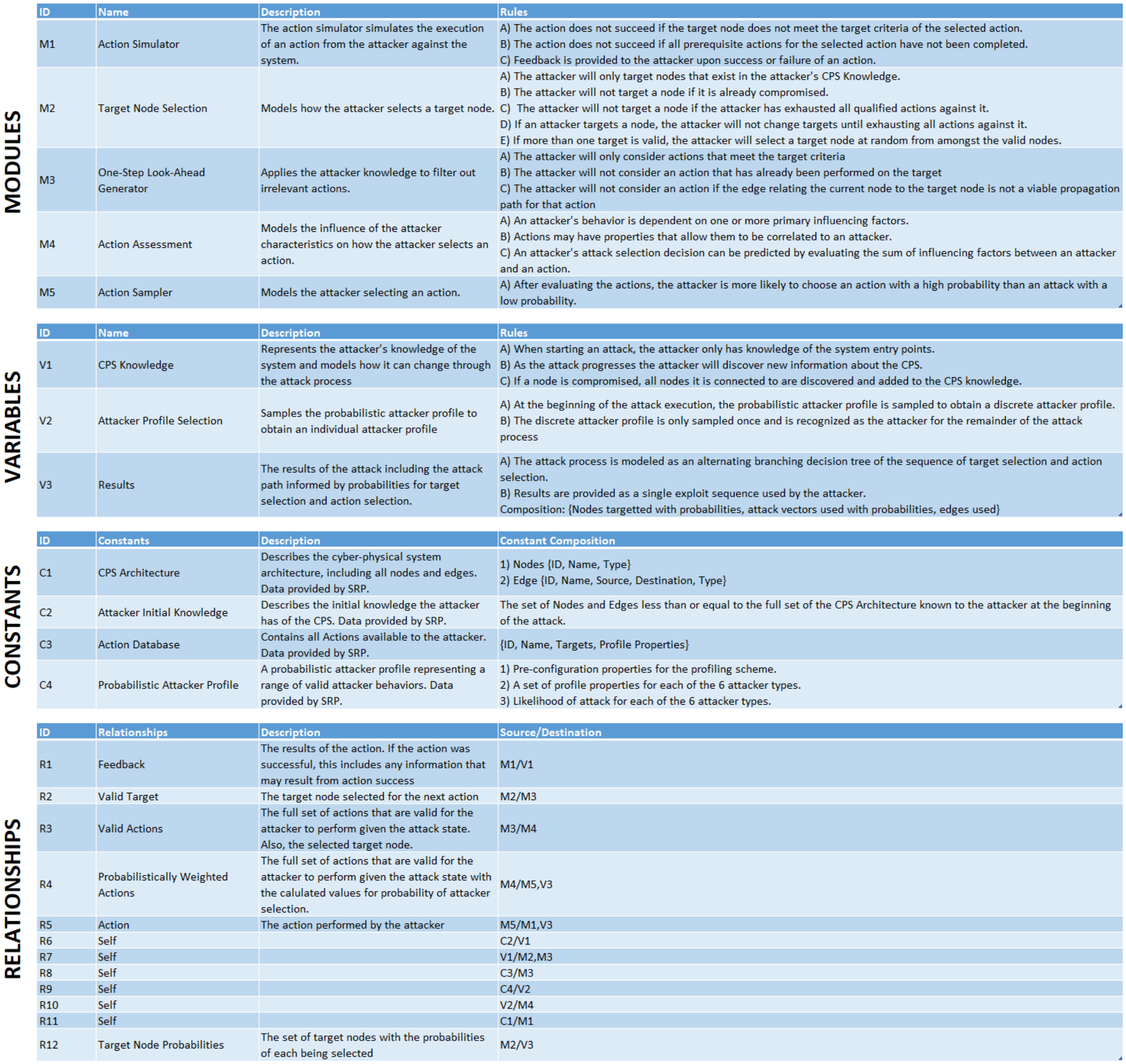


Figure B.1 The attacker model functional representation diagram for Adepu's attacker model in [57].

Table B.2 Attacker model functional representation data for Degloglos' attacker model in [5]



MODULES

ID	Name	Description	Rules
M1	Action Simulator	The action simulator simulates the execution of an action from the attacker against the system.	A) The action does not succeed if the target node does not meet the target criteria of the selected action. B) The action does not succeed if all prerequisite actions for the selected action have not been completed. C) Feedback is provided to the attacker upon success or failure of an action.
M2	Target Node Selection	Models how the attacker selects a target node.	A) The attacker will only target nodes that exist in the attacker's CPS Knowledge. B) The attacker will not target a node if it is already compromised. C) The attacker will not target a node if the attacker has exhausted all qualified actions against it. D) If an attacker targets a node, the attacker will not change targets until exhausting all actions against it. E) If more than one target is valid, the attacker will select a target node at random from amongst the valid nodes.
M3	One-Step Look-Ahead Generator	Applies the attacker knowledge to filter out irrelevant actions.	A) The attacker will only consider actions that meet the target criteria B) The attacker will not consider an action that has already been performed on the target C) The attacker will not consider an action if the edge relating the current node to the target node is not a viable propagation path for that action
M4	Action Assessment	Models the influence of the attacker characteristics on how the attacker selects an action.	A) An attacker's behavior is dependent on one or more primary influencing factors. B) Actions may have properties that allow them to be correlated to an attacker. C) An attacker's attack selection decision can be predicted by evaluating the sum of influencing factors between an attacker and an action.
M5	Action Sampler	Models the attacker selecting an action.	A) After evaluating the actions, the attacker is more likely to choose an action with a high probability than an attack with a low probability.

VARIABLES

ID	Name	Description	Rules
V1	CPS Knowledge	Represents the attacker's knowledge of the system and models how it can change through the attack process	A) When starting an attack, the attacker only has knowledge of the system entry points. B) As the attack progresses the attacker will discover new information about the CPS. C) If a node is compromised, all nodes it is connected to are discovered and added to the CPS knowledge.
V2	Attacker Profile Selection	Samples the probabilistic attacker profile to obtain an individual attacker profile	A) At the beginning of the attack execution, the probabilistic attacker profile is sampled to obtain a discrete attacker profile. B) The discrete attacker profile is only sampled once and is recognized as the attacker for the remainder of the attack process
V3	Results	The results of the attack including the attack path informed by probabilities for target selection and action selection.	A) The attack process is modeled as an alternating branching decision tree of the sequence of target selection and action selection. B) Results are provided as a single exploit sequence used by the attacker. Composition: {Nodes targetted with probabilities, attack vectors used with probabilities, edges used}

CONSTANTS

ID	Constants	Description	Constant Composition
C1	CPS Architecture	Describes the cyber-physical system architecture, including all nodes and edges. Data provided by SRP.	1) Nodes {ID, Name, Type} 2) Edge {ID, Name, Source, Destination, Type}
C2	Attacker Initial Knowledge	Describes the initial knowledge the attacker has of the CPS. Data provided by SRP.	The set of Nodes and Edges less than or equal to the full set of the CPS Architecture known to the attacker at the beginning of the attack.
C3	Action Database	Contains all Actions available to the attacker. Data provided by SRP.	{ID, Name, Targets, Profile Properties}
C4	Probabilistic Attacker Profile	A probabilistic attacker profile representing a range of valid attacker behaviors. Data provided by SRP.	1) Pre-configuration properties for the profiling scheme. 2) A set of profile properties for each of the 6 attacker types. 3) Likelihood of attack for each of the 6 attacker types.

RELATIONSHIPS

ID	Relationships	Description	Source/Destination
R1	Feedback	The results of the action. If the action was successful, this includes any information that may result from action success	M1/V1
R2	Valid Target	The target node selected for the next action	M2/M3
R3	Valid Actions	The full set of actions that are valid for the attacker to perform given the attack state. Also, the selected target node.	M3/M4
R4	Probabilistically Weighted Actions	The full set of actions that are valid for the attacker to perform given the attack state with the calulated values for probability of attacker selection.	M4/M5,V3
R5	Action	The action performed by the attacker	M5/M1,V3
R6	Self		C2/V1
R7	Self		V1/M2,M3
R8	Self		C3/M3
R9	Self		C4/V2
R10	Self		V2/M4
R11	Self		C1/M1
R12	Target Node Probabilities	The set of target nodes with the probabilities of each being selected	M2/V3

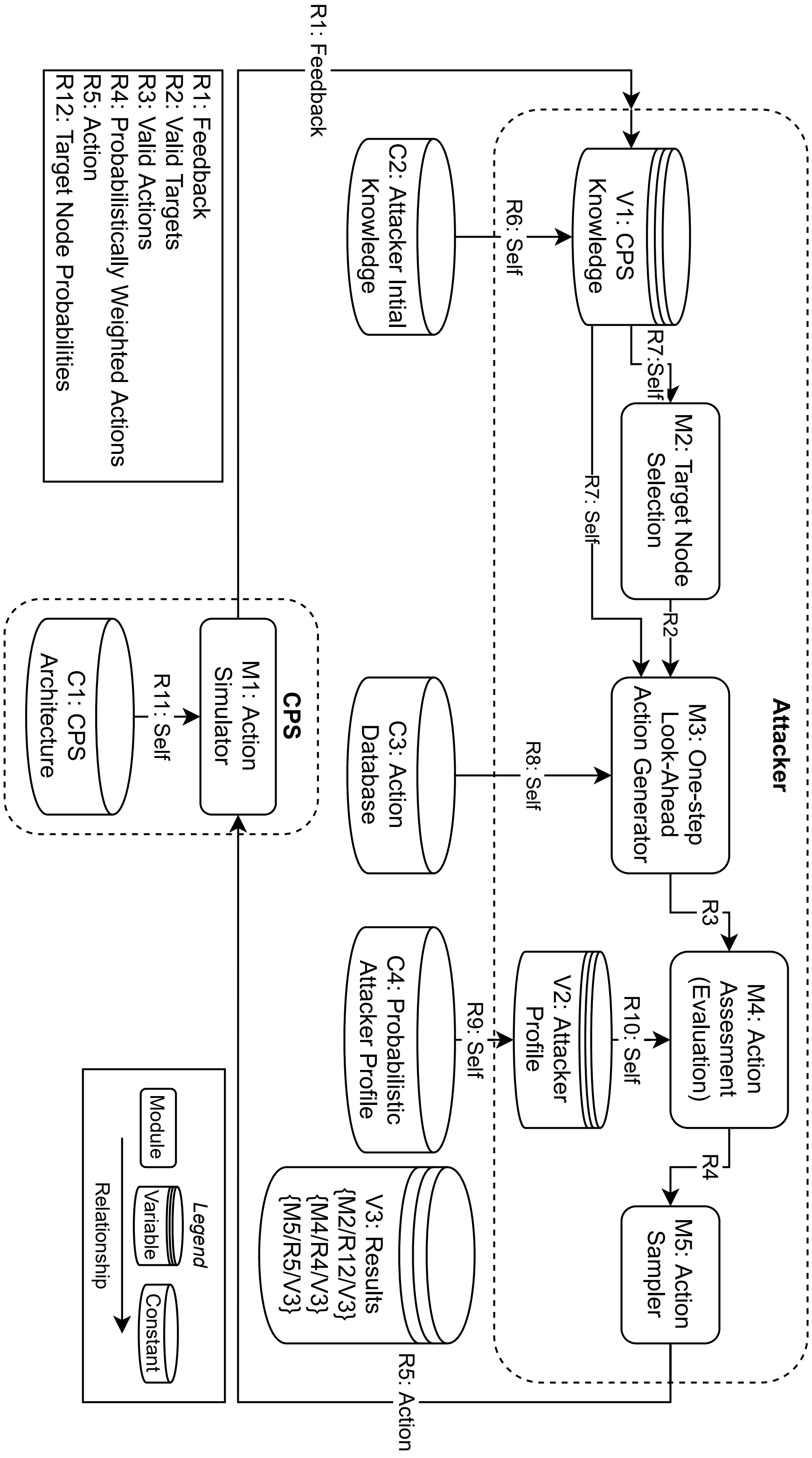


Figure B.2 The attacker model functional representation diagram for Deloglos' attacker model in [5].

Appendix C

Use-Case Intermediate Results

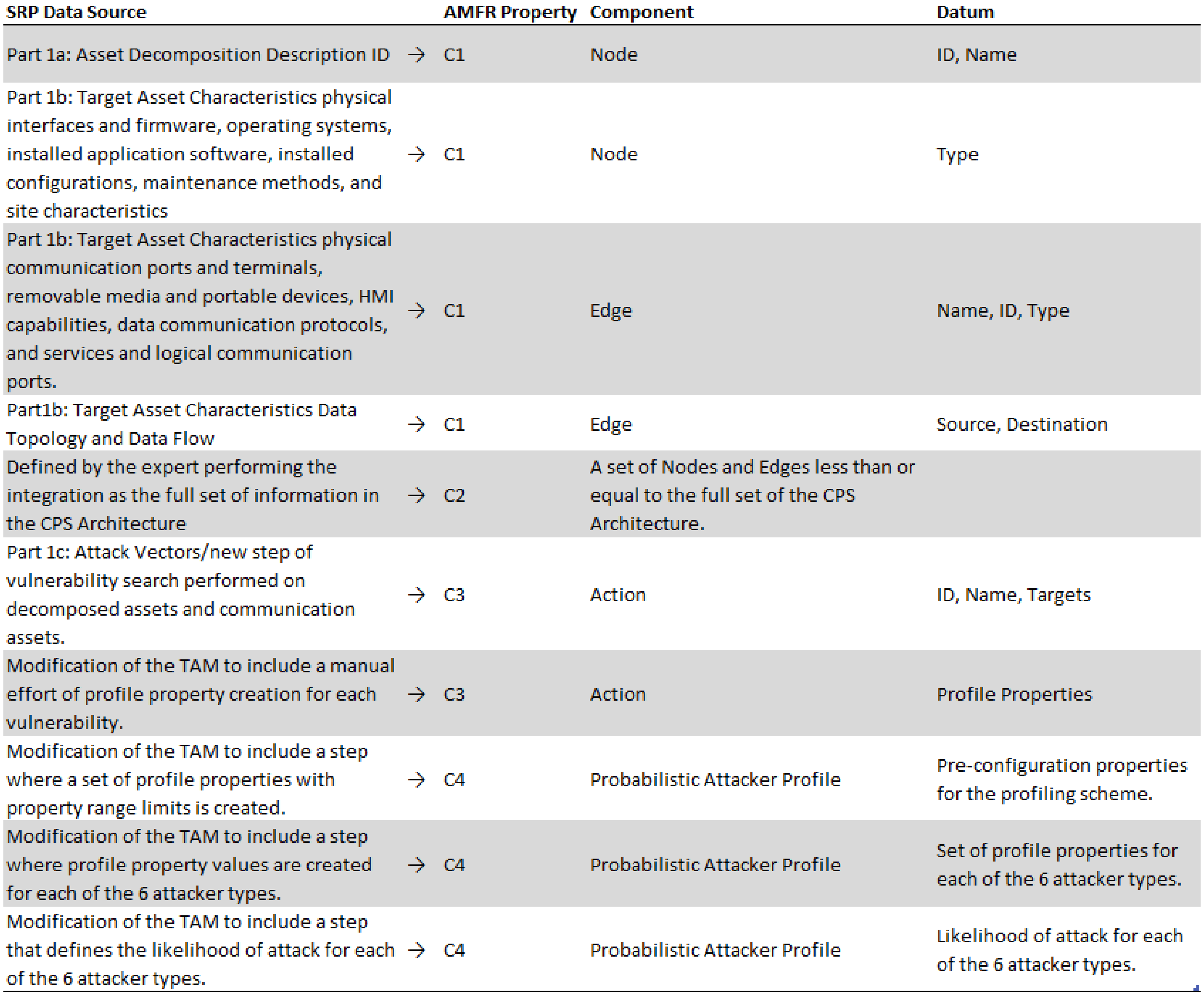
Table C.1 Intermediary results for the exact scoring functions for the TAM use case

			Adepu	Basin	Deloglos	Ekelhart	Le May	McEvoy	Mo	Monteuuis	Orojloo	Teixeira	Vigo
Category		**Attribute**	**AM1**	**AM2**	**AM3**	**AM4**	**AM5**	**AM6**	**AM7**	**AM8**	**AM9**	**AM10**	**AM11**
Contextual Interface		**Principle Perspective**	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE
	Dimensions	**Time**	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE
		Unpredictability	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
		Executable	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE
		Operational Method	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
		Attack Procedure	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE
		Attack Consequence	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE
		CPS Security Controls	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE
		Intrusion Detection	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE
		Countermeasures	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE
		CPS Security Protocols	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
		CPS Communication	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE
		CPS Process State	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
		CPS Security State	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
		CPS Process - Cyber	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE
		CPS Process - Physical	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE
		Orientation - Physical	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE
		Orientation - Cyber	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
		Attacker Knowledge Model	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE
Input Interface	**CPS**	**Components**	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
		Component Relationships	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE
		Protocols	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
		Process Properties	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE
		Performance Metrics	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
	Attack Vectors	**CPS Associations**	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE
		Attacker Associations	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE
	Attacker	**Profile-Defined Attacker**	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE
		Profile Specificity	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE
		Model-Defined Attacker	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE
Output	**Results**	**Procedure**	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
		Security Metrics	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE
		Security Properties	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE
		Outcome Likelihood	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE
		Risk Assessment	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE

Table C.2 Intermediary results for the inclusive scoring functions for the TAM use case

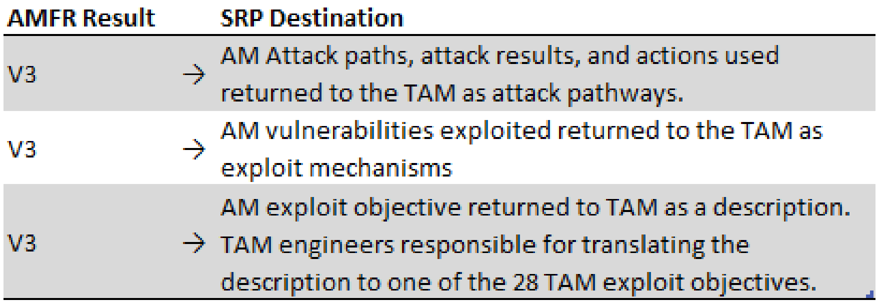
Category		Attribute	Adepu AM1	Basin AM2	Deloglos AM3	Ekelhart AM4	Le May AM5	McEvoy AM6	Mo AM7	Monteuuis AM8	Orojloo AM9	Teixeira AM10	Vigo AM11
Contextual Interface		Principle Perspective	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE
	Dimensions	Time	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
		Unpredictability	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
		Executable	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE
		Operational Method	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
		Attack Procedure	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
		Attack Consequence	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
		CPS Security Controls	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
		Intrusion Detection	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
		Countermeasures	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
		CPS Security Protocols	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
		CPS Communication	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
		CPS Process State	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
		CPS Security State	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
		CPS Process - Cyber	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
		CPS Process - Physical	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
		Orientation - Physical	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE
		Orientation - Cyber	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
		Attacker Knowledge Model	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE
Input Interface	CPS	Components	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
		Component Relationships	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
		Protocols	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
		Process Properties	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE
		Performance Metrics	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
	Attack Vectors	CPS Associations	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
		Attacker Associations	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE
	Attacker	Profile-Defined Attacker	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE
		Profile Specificity	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE
		Model-Defined Attacker	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE
Output	Results	Procedure	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
		Security Metrics	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
		Security Properties	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
		Outcome Likelihood	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
		Risk Assessment	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE

Table C.3 Integration mapping of attacker model input data from the TAM in [1] to Deloglos' attacker model in [5] for the use case in Chapter 7



SRP Data Source		AMFR Property	Component	Datum
Part 1a: Asset Decomposition Description ID	→	C1	Node	ID, Name
Part 1b: Target Asset Characteristics physical interfaces and firmware, operating systems, installed application software, installed configurations, maintenance methods, and site characteristics	→	C1	Node	Type
Part 1b: Target Asset Characteristics physical communication ports and terminals, removable media and portable devices, HMI capabilities, data communication protocols, and services and logical communication ports.	→	C1	Edge	Name, ID, Type
Part1b: Target Asset Characteristics Data Topology and Data Flow	→	C1	Edge	Source, Destination
Defined by the expert performing the integration as the full set of information in the CPS Architecture	→	C2	A set of Nodes and Edges less than or equal to the full set of the CPS Architecture.	
Part 1c: Attack Vectors/new step of vulnerability search performed on decomposed assets and communication assets.	→	C3	Action	ID, Name, Targets
Modification of the TAM to include a manual effort of profile property creation for each vulnerability.	→	C3	Action	Profile Properties
Modification of the TAM to include a step where a set of profile properties with property range limits is created.	→	C4	Probabilistic Attacker Profile	Pre-configuration properties for the profiling scheme.
Modification of the TAM to include a step where profile property values are created for each of the 6 attacker types.	→	C4	Probabilistic Attacker Profile	Set of profile properties for each of the 6 attacker types.
Modification of the TAM to include a step that defines the likelihood of attack for each of the 6 attacker types.	→	C4	Probabilistic Attacker Profile	Likelihood of attack for each of the 6 attacker types.

Table C.4 Integration mapping of AM output results from Deloglos' AM in [5] to the TAM in [1]

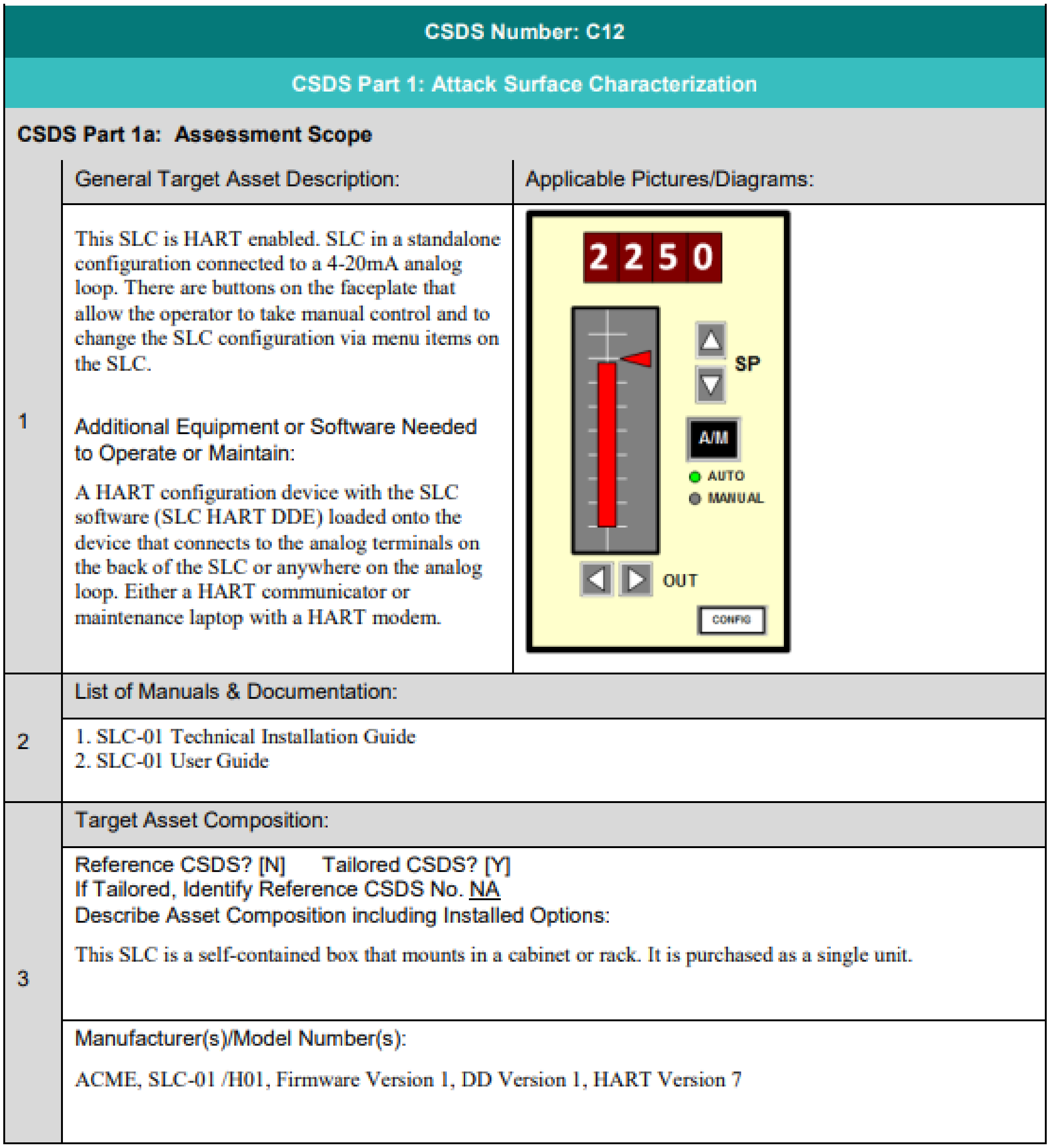


AMFR Result		SRP Destination
V3	→	AM Attack paths, attack results, and actions used returned to the TAM as attack pathways.
V3	→	AM vulnerabilities exploited returned to the TAM as exploit mechanisms
V3	→	AM exploit objective returned to TAM as a description. TAM engineers responsible for translating the description to one of the 28 TAM exploit objectives.

Appendix D

Execution Example Cyber Security Data Sheet

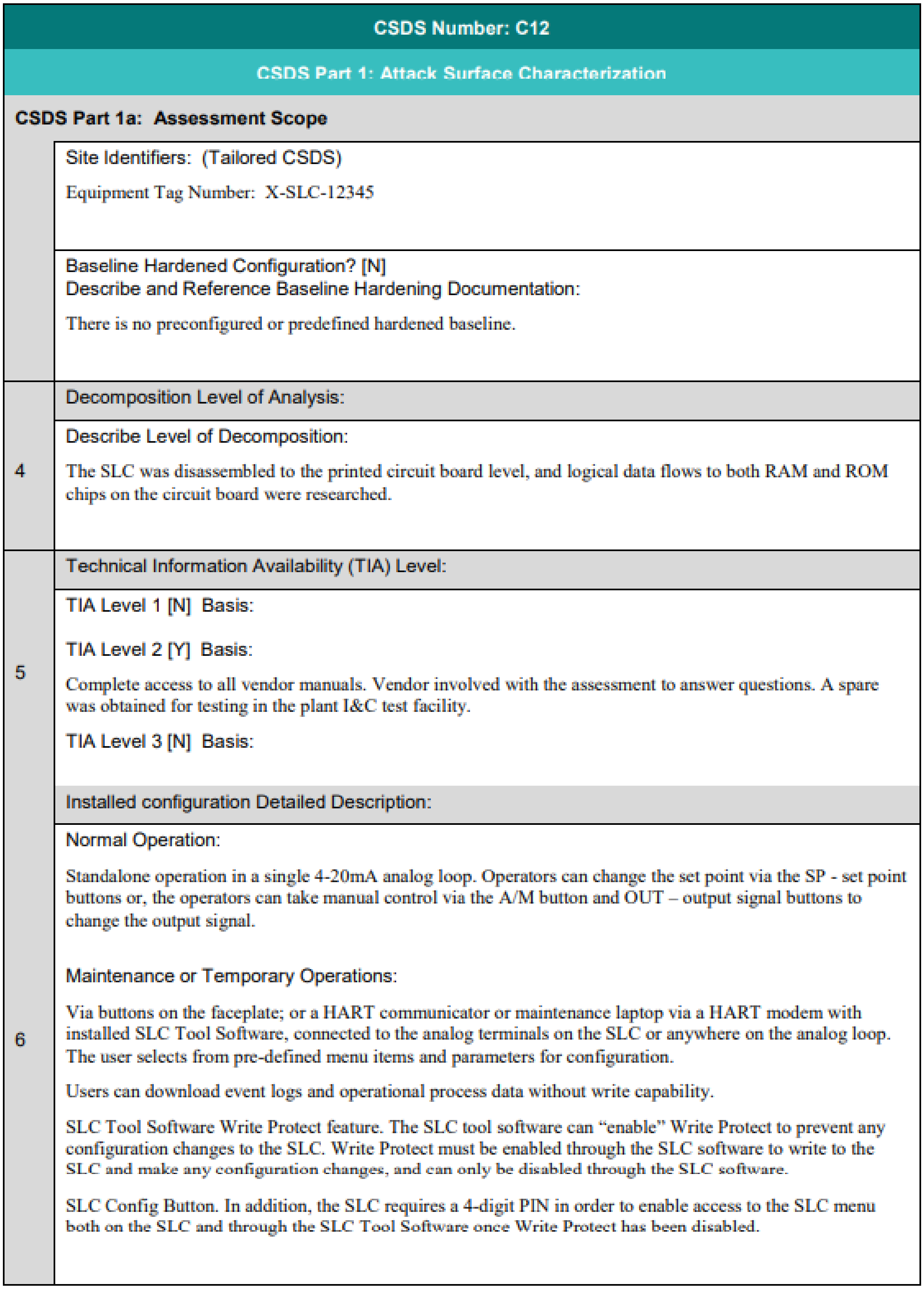
Table D.1 Page 1 of the cyber security data sheet for the integration use-case execution in Section 7.8



2250
SP
A/M
AUTO
MANUAL
OUT
CONFIG

CSDS Number: C12		
CSDS Part 1: Attack Surface Characterization		
CSDS Part 1a: Assessment Scope		
1	General Target Asset Description:	Applicable Pictures/Diagrams:
	This SLC is HART enabled. SLC in a standalone configuration connected to a 4-20mA analog loop. There are buttons on the faceplate that allow the operator to take manual control and to change the SLC configuration via menu items on the SLC. Additional Equipment or Software Needed to Operate or Maintain: A HART configuration device with the SLC software (SLC HART DDE) loaded onto the device that connects to the analog terminals on the back of the SLC or anywhere on the analog loop. Either a HART communicator or maintenance laptop with a HART modem.	
2	List of Manuals & Documentation:	
	1. SLC-01 Technical Installation Guide 2. SLC-01 User Guide	
3	Target Asset Composition:	
	Reference CSDS? [N] Tailored CSDS? [Y] If Tailored, Identify Reference CSDS No. NA Describe Asset Composition including Installed Options: This SLC is a self-contained box that mounts in a cabinet or rack. It is purchased as a single unit.	
	Manufacturer(s)/Model Number(s): ACME, SLC-01 /H01, Firmware Version 1, DD Version 1, HART Version 7	

Table D.2 Page 2 of the cyber security data sheet for the integration use-case execution in Section 7.8



CSDS Number: C12	
CSDS Part 1: Attack Surface Characterization	
CSDS Part 1a: Assessment Scope	
	Site Identifiers: (Tailored CSDS) Equipment Tag Number: X-SLC-12345
	Baseline Hardened Configuration? [N] Describe and Reference Baseline Hardening Documentation: There is no preconfigured or predefined hardened baseline.
4	Decomposition Level of Analysis:
	Describe Level of Decomposition: The SLC was disassembled to the printed circuit board level, and logical data flows to both RAM and ROM chips on the circuit board were researched.
5	Technical Information Availability (TIA) Level:
	TIA Level 1 [N] Basis: TIA Level 2 [Y] Basis: Complete access to all vendor manuals. Vendor involved with the assessment to answer questions. A spare was obtained for testing in the plant I&C test facility. TIA Level 3 [N] Basis:
6	Installed configuration Detailed Description:
	Normal Operation: Standalone operation in a single 4-20mA analog loop. Operators can change the set point via the SP - set point buttons or, the operators can take manual control via the A/M button and OUT – output signal buttons to change the output signal. Maintenance or Temporary Operations: Via buttons on the faceplate; or a HART communicator or maintenance laptop via a HART modem with installed SLC Tool Software, connected to the analog terminals on the SLC or anywhere on the analog loop. The user selects from pre-defined menu items and parameters for configuration. Users can download event logs and operational process data without write capability. SLC Tool Software Write Protect feature. The SLC tool software can “enable” Write Protect to prevent any configuration changes to the SLC. Write Protect must be enabled through the SLC software to write to the SLC and make any configuration changes, and can only be disabled through the SLC software. SLC Config Button. In addition, the SLC requires a 4-digit PIN in order to enable access to the SLC menu both on the SLC and through the SLC Tool Software once Write Protect has been disabled.

Table D.3 Page 3 of the cyber security data sheet for the integration use-case execution in Section 7.8

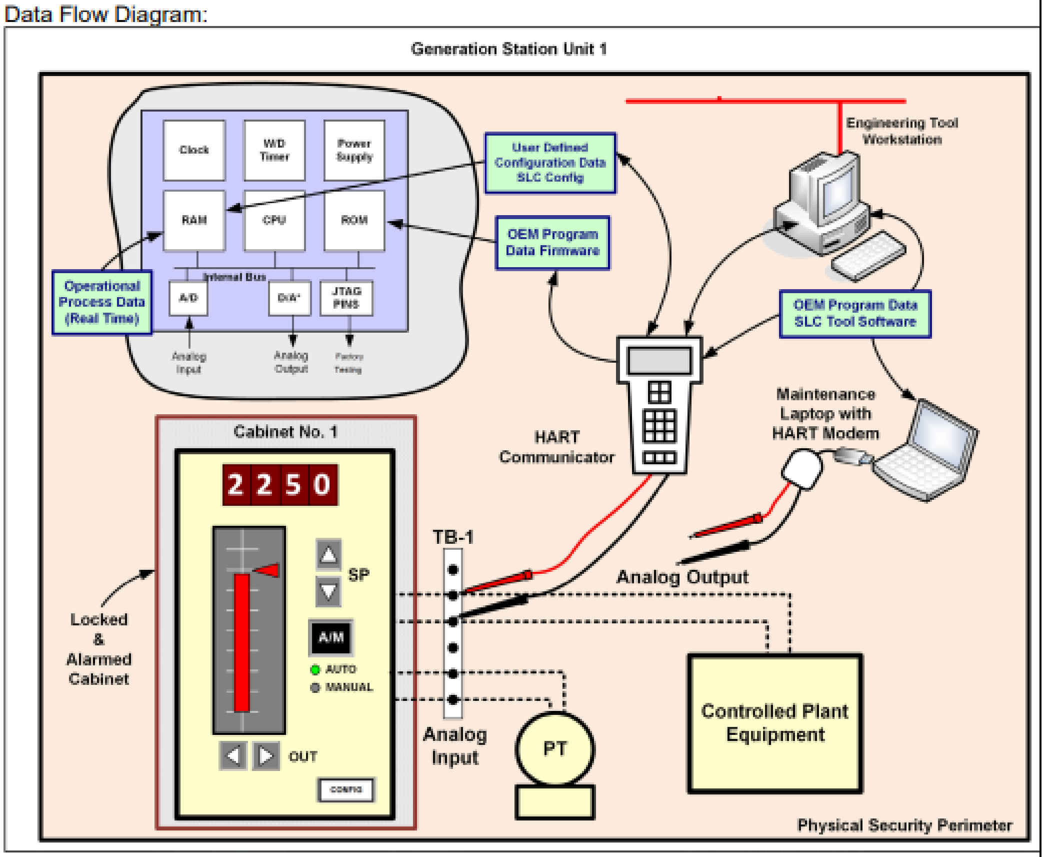
CSDS Number: C12

CSDS Part 1: Attack Surface Characterization

CSDS Part 1a: Assessment Scope

7 Data Topology and Data Flow: Data Flow Diagram Attached? [N]

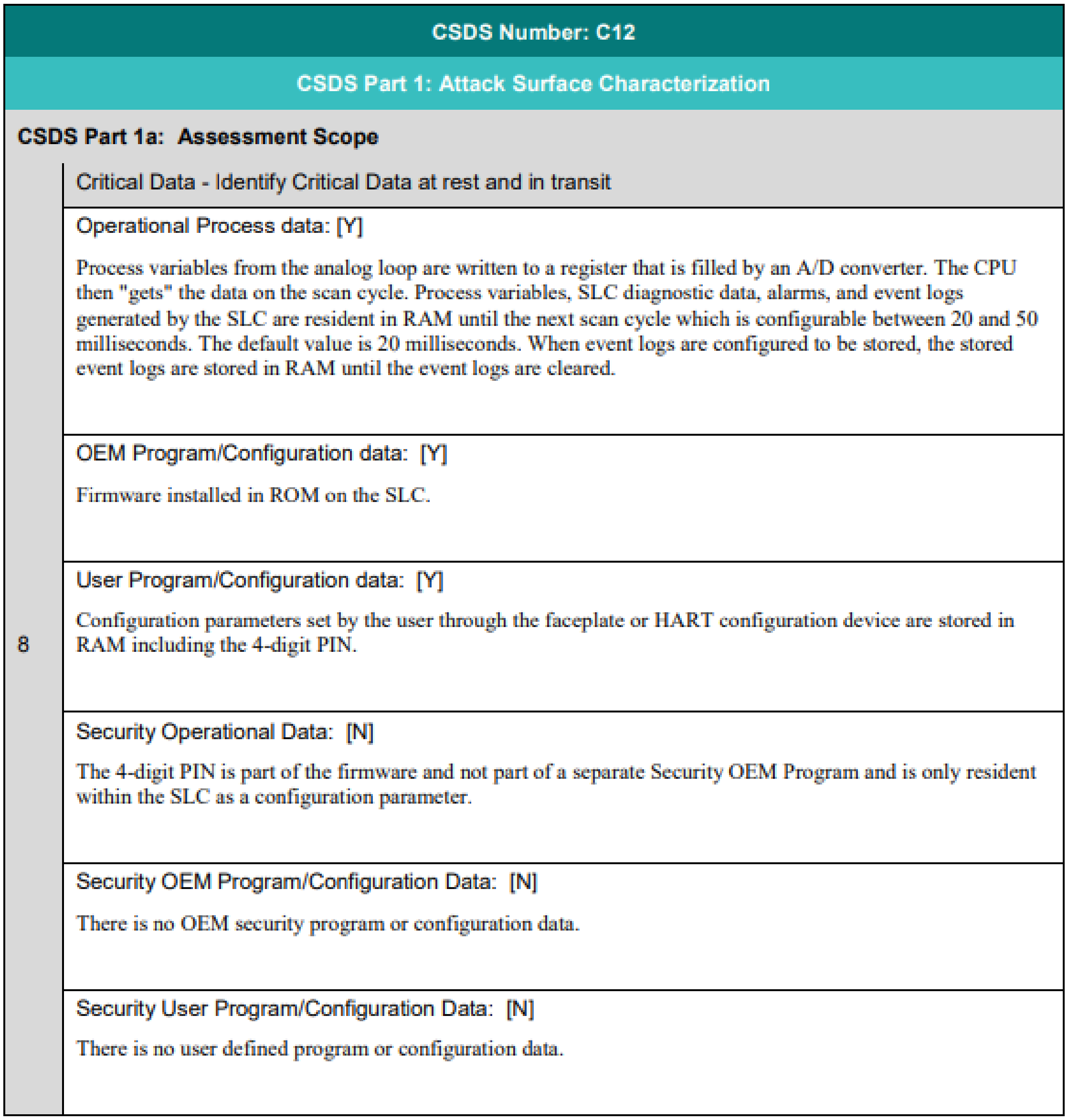
Data Flow Diagram:


Generation Station Unit 1
Clock
WD Timer
Power Supply
RAM
CPU
ROM
Internal Bus
A/D
D/A*
JTAG PINS
Analog Input
Analog Output
Factory Testing
Operational Process Data (Real Time)
User Defined Configuration Data SLC Config
OEM Program Data Firmware
Engineering Tool Workstation
OEM Program Data SLC Tool Software
Maintenance Laptop with HART Modem
HART Communicator
Cabinet No. 1
2 2 5 0
SP
A/M
AUTO
MANUAL
OUT
CONFIG
Locked & Alarmed Cabinet
TB-1
Analog Output
Analog Input
PT
Controlled Plant Equipment
Physical Security Perimeter

Data Topology and Data Flow Description:

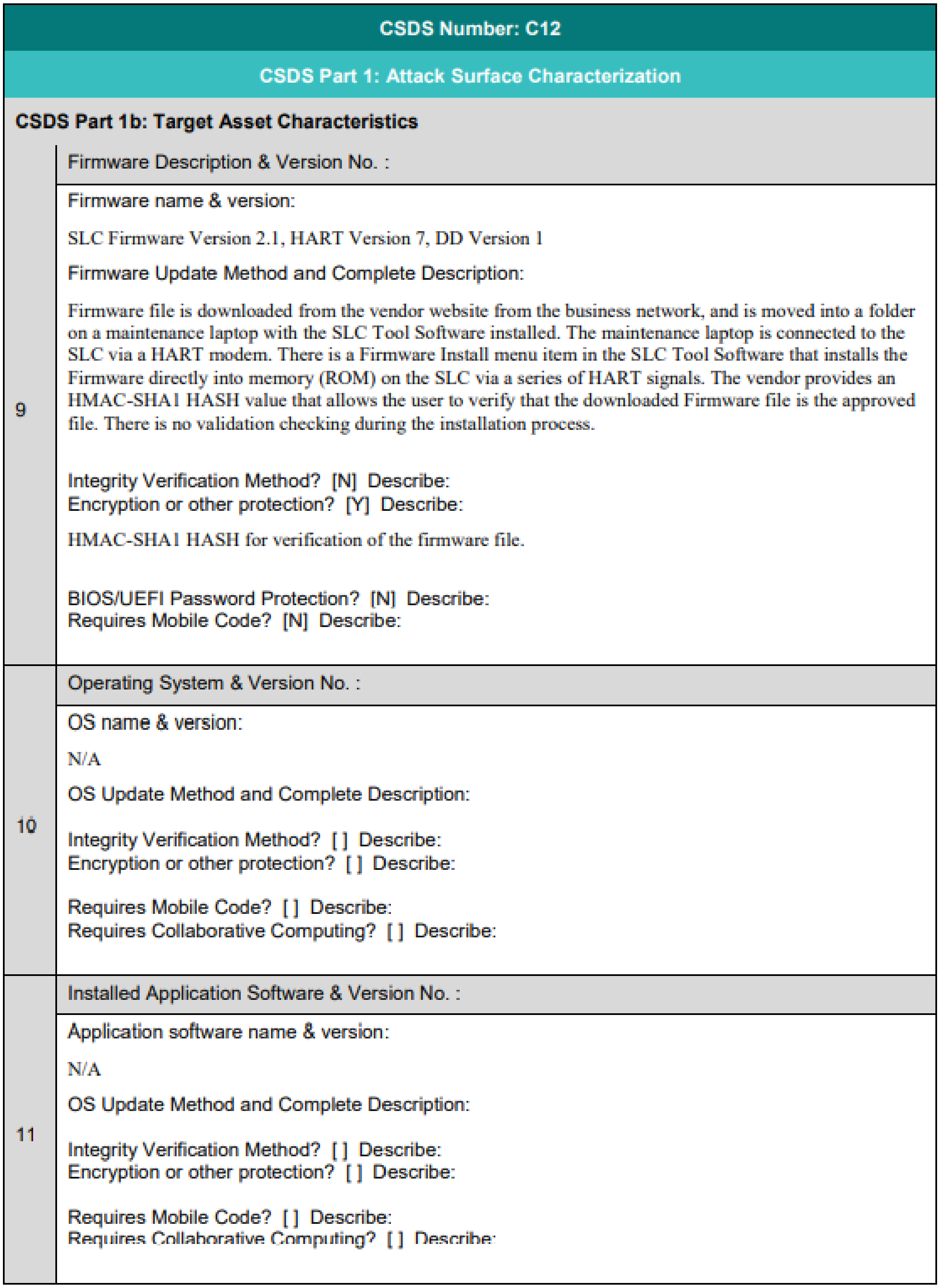
See Line 6

Table D.4 Page 4 of the cyber security data sheet for the integration use-case execution in Section 7.8



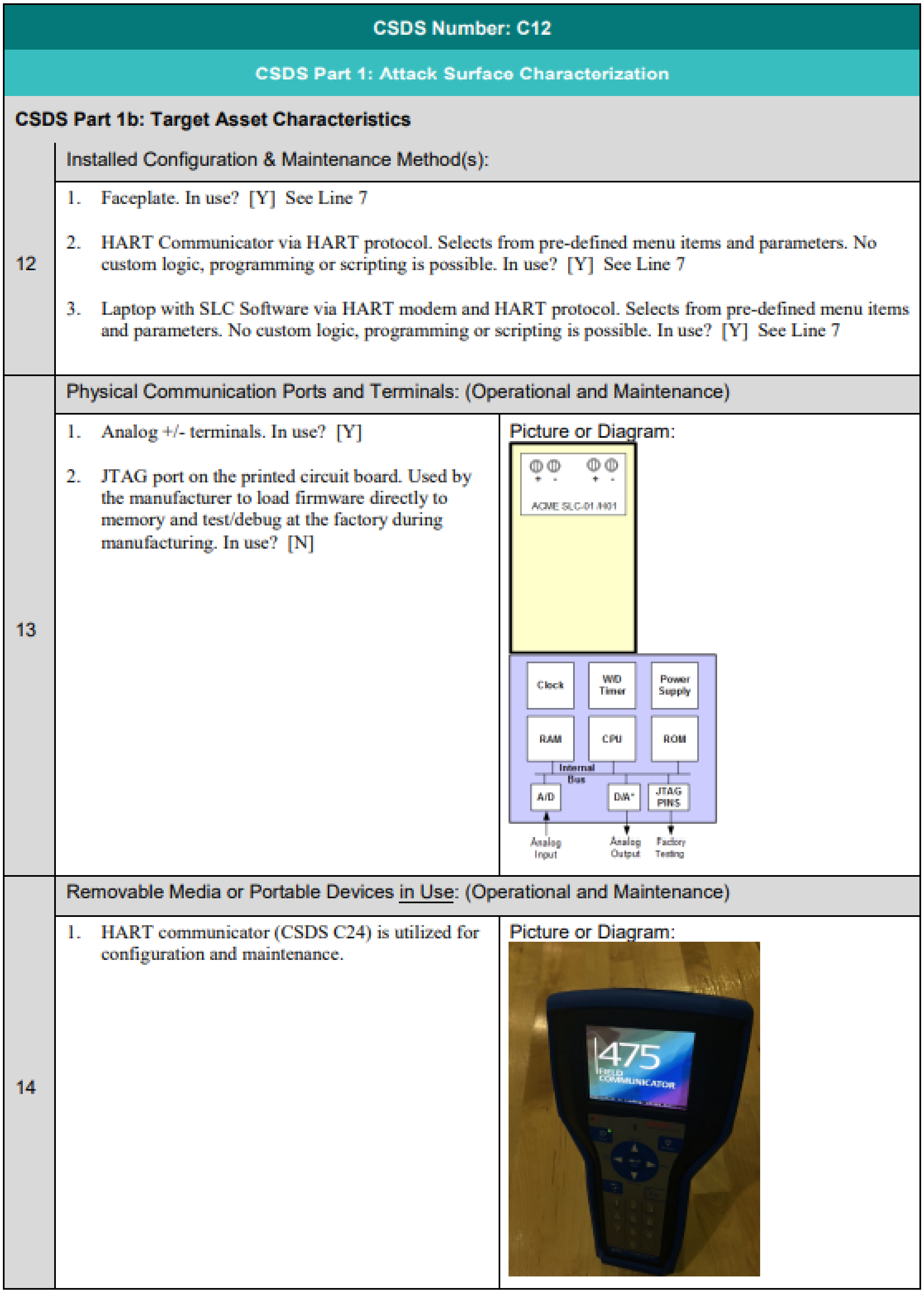
CSDS Number: C12	
CSDS Part 1: Attack Surface Characterization	
CSDS Part 1a: Assessment Scope	
8	Critical Data - Identify Critical Data at rest and in transit
	Operational Process data: [Y] Process variables from the analog loop are written to a register that is filled by an A/D converter. The CPU then "gets" the data on the scan cycle. Process variables, SLC diagnostic data, alarms, and event logs generated by the SLC are resident in RAM until the next scan cycle which is configurable between 20 and 50 milliseconds. The default value is 20 milliseconds. When event logs are configured to be stored, the stored event logs are stored in RAM until the event logs are cleared.
	OEM Program/Configuration data: [Y] Firmware installed in ROM on the SLC.
	User Program/Configuration data: [Y] Configuration parameters set by the user through the faceplate or HART configuration device are stored in RAM including the 4-digit PIN.
	Security Operational Data: [N] The 4-digit PIN is part of the firmware and not part of a separate Security OEM Program and is only resident within the SLC as a configuration parameter.
	Security OEM Program/Configuration Data: [N] There is no OEM security program or configuration data.
	Security User Program/Configuration Data: [N] There is no user defined program or configuration data.

Table D.5 Page 5 of the cyber security data sheet for the integration use-case execution in Section 7.8



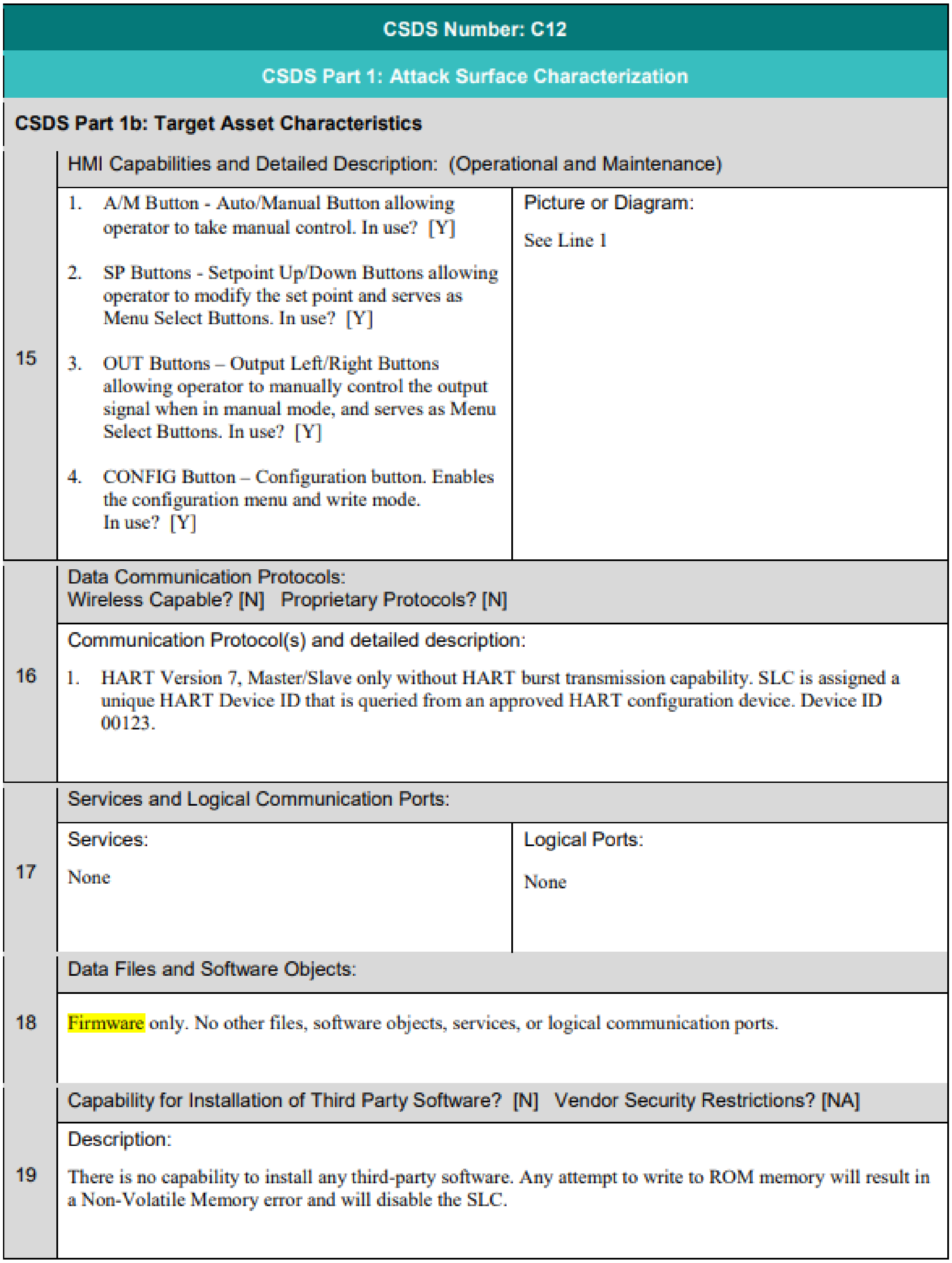
CSDS Number: C12	
CSDS Part 1: Attack Surface Characterization	
CSDS Part 1b: Target Asset Characteristics	
9	Firmware Description & Version No. :
	Firmware name & version: SLC Firmware Version 2.1, HART Version 7, DD Version 1 Firmware Update Method and Complete Description: Firmware file is downloaded from the vendor website from the business network, and is moved into a folder on a maintenance laptop with the SLC Tool Software installed. The maintenance laptop is connected to the SLC via a HART modem. There is a Firmware Install menu item in the SLC Tool Software that installs the Firmware directly into memory (ROM) on the SLC via a series of HART signals. The vendor provides an HMAC-SHA1 HASH value that allows the user to verify that the downloaded Firmware file is the approved file. There is no validation checking during the installation process. Integrity Verification Method? [N] Describe: Encryption or other protection? [Y] Describe: HMAC-SHA1 HASH for verification of the firmware file. BIOS/UEFI Password Protection? [N] Describe: Requires Mobile Code? [N] Describe:
10	Operating System & Version No. :
	OS name & version: N/A OS Update Method and Complete Description: Integrity Verification Method? [ ] Describe: Encryption or other protection? [ ] Describe: Requires Mobile Code? [ ] Describe: Requires Collaborative Computing? [ ] Describe:
11	Installed Application Software & Version No. :
	Application software name & version: N/A OS Update Method and Complete Description: Integrity Verification Method? [ ] Describe: Encryption or other protection? [ ] Describe: Requires Mobile Code? [ ] Describe: Requires Collaborative Computing? [ ] Describe:

Table D.6 Page 6 of the cyber security data sheet for the integration use-case execution in Section 7.8



CSDS Number: C12		
CSDS Part 1: Attack Surface Characterization		
CSDS Part 1b: Target Asset Characteristics		
12	Installed Configuration & Maintenance Method(s):	
	1. Faceplate. In use? [Y] See Line 7 2. HART Communicator via HART protocol. Selects from pre-defined menu items and parameters. No custom logic, programming or scripting is possible. In use? [Y] See Line 7 3. Laptop with SLC Software via HART modem and HART protocol. Selects from pre-defined menu items and parameters. No custom logic, programming or scripting is possible. In use? [Y] See Line 7	
13	Physical Communication Ports and Terminals: (Operational and Maintenance)	
	1. Analog +/- terminals. In use? [Y] 2. JTAG port on the printed circuit board. Used by the manufacturer to load firmware directly to memory and test/debug at the factory during manufacturing. In use? [N]	Picture or Diagram:
14	Removable Media or Portable Devices in Use: (Operational and Maintenance)	
	1. HART communicator (CSDS C24) is utilized for configuration and maintenance.	Picture or Diagram:

Table D.7 Page 7 of the cyber security data sheet for the integration use-case execution in Section 7.8



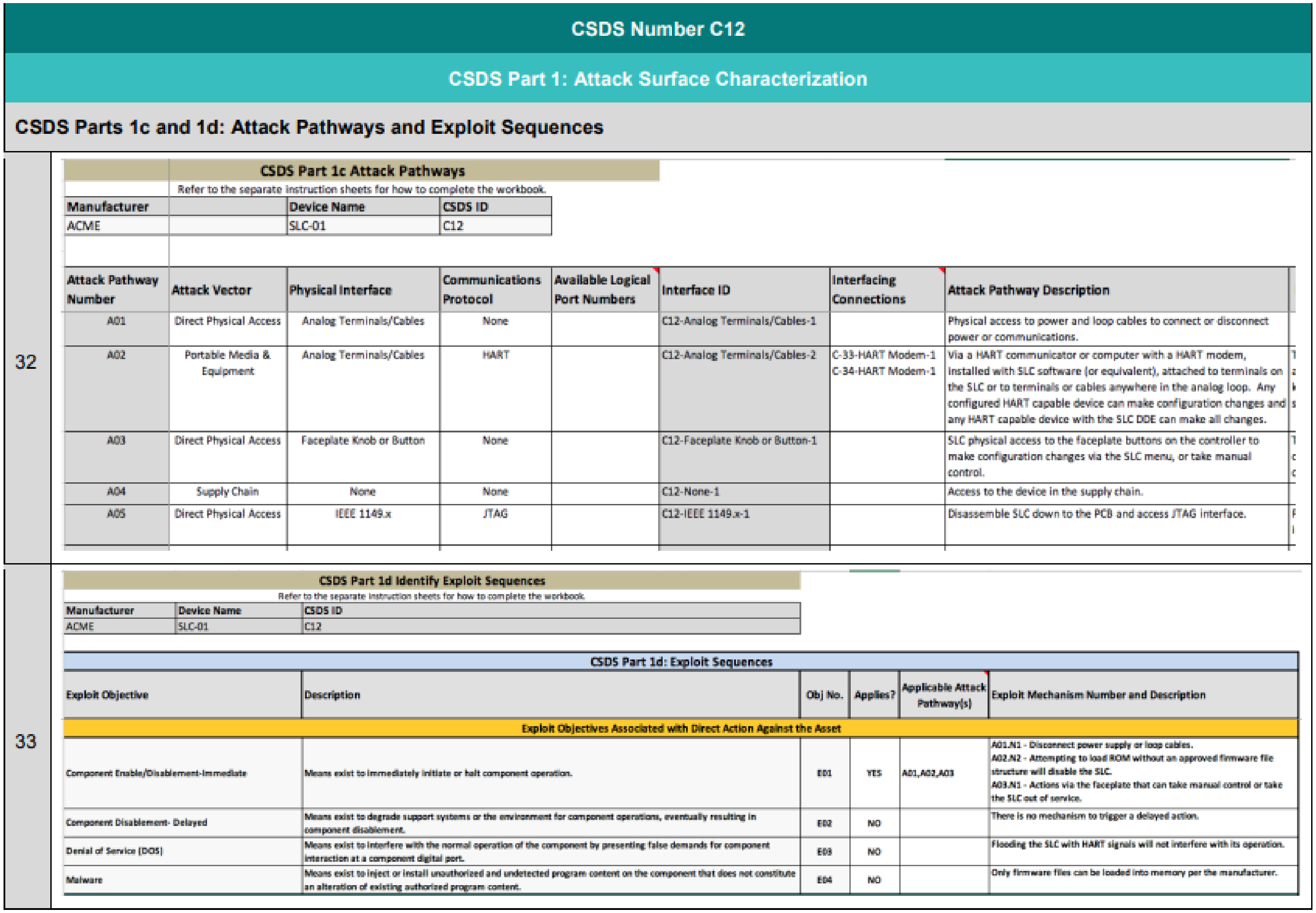
CSDS Number: C12

CSDS Part 1: Attack Surface Characterization

CSDS Part 1b: Target Asset Characteristics

15	HMI Capabilities and Detailed Description: (Operational and Maintenance)	
	1. A/M Button - Auto/Manual Button allowing operator to take manual control. In use? [Y] 2. SP Buttons - Setpoint Up/Down Buttons allowing operator to modify the set point and serves as Menu Select Buttons. In use? [Y] 3. OUT Buttons – Output Left/Right Buttons allowing operator to manually control the output signal when in manual mode, and serves as Menu Select Buttons. In use? [Y] 4. CONFIG Button – Configuration button. Enables the configuration menu and write mode. In use? [Y]	Picture or Diagram: See Line 1
16	Data Communication Protocols: Wireless Capable? [N] Proprietary Protocols? [N]	
	Communication Protocol(s) and detailed description: 1. HART Version 7, Master/Slave only without HART burst transmission capability. SLC is assigned a unique HART Device ID that is queried from an approved HART configuration device. Device ID 00123.	
17	Services and Logical Communication Ports:	
	Services: None	Logical Ports: None
18	Data Files and Software Objects:	
	Firmware only. No other files, software objects, services, or logical communication ports.	
19	Capability for Installation of Third Party Software? [N] Vendor Security Restrictions? [NA]	
	Description: There is no capability to install any third-party software. Any attempt to write to ROM memory will result in a Non-Volatile Memory error and will disable the SLC.	

Table D.8 Page 12 of the cyber security data sheet for the integration use-case execution in Section 7.8



CSDS Number C12

CSDS Part 1: Attack Surface Characterization

CSDS Parts 1c and 1d: Attack Pathways and Exploit Sequences

32

CSDS Part 1c Attack Pathways

Refer to the separate instruction sheets for how to complete the workbook.

Manufacturer		Device Name	CSDS ID
ACME		SLC-01	C12

Attack Pathway Number	Attack Vector	Physical Interface	Communications Protocol	Available Logical Port Numbers	Interface ID	Interfacing Connections	Attack Pathway Description
A01	Direct Physical Access	Analog Terminals/Cables	None		C12-Analog Terminals/Cables-1		Physical access to power and loop cables to connect or disconnect power or communications.
A02	Portable Media & Equipment	Analog Terminals/Cables	HART		C12-Analog Terminals/Cables-2	C-33-HART Modem-1 C-34-HART Modem-1	Via a HART communicator or computer with a HART modem, installed with SLC software (or equivalent), attached to terminals on the SLC or to terminals or cables anywhere in the analog loop. Any configured HART capable device can make configuration changes and any HART capable device with the SLC DDE can make all changes.
A03	Direct Physical Access	Faceplate Knob or Button	None		C12-Faceplate Knob or Button-1		SLC physical access to the faceplate buttons on the controller to make configuration changes via the SLC menu, or take manual control.
A04	Supply Chain	None	None		C12-None-1		Access to the device in the supply chain.
A05	Direct Physical Access	IEEE 1149.x	JTAG		C12-IEEE 1149.x-1		Disassemble SLC down to the PCB and access JTAG interface.

33

CSDS Part 1d Identify Exploit Sequences

Refer to the separate instruction sheets for how to complete the workbook.

Manufacturer	Device Name	CSDS ID
ACME	SLC-01	C12

CSDS Part 1d: Exploit Sequences

Exploit Objective	Description	Obj No.	Applies?	Applicable Attack Pathway(s)	Exploit Mechanism Number and Description
Exploit Objectives Associated with Direct Action Against the Asset					
Component Enable/Disablement-Immediate	Means exist to immediately initiate or halt component operation.	E01	YES	A01,A02,A03	A01.N1 - Disconnect power supply or loop cables. A02.N2 - Attempting to load ROM without an approved firmware file structure will disable the SLC. A03.N1 - Actions via the faceplate that can take manual control or take the SLC out of service.
Component Disablement- Delayed	Means exist to degrade support systems or the environment for component operations, eventually resulting in component disablement.	E02	NO		There is no mechanism to trigger a delayed action.
Denial of Service (DOS)	Means exist to interfere with the normal operation of the component by presenting false demands for component interaction at a component digital port.	E03	NO		Flooding the SLC with HART signals will not interfere with its operation.
Malware	Means exist to inject or install unauthorized and undetected program content on the component that does not constitute an alteration of existing authorized program content.	E04	NO		Only firmware files can be loaded into memory per the manufacturer.