Technical Report: Towards a Systematic Threat Modeling Approach for Cyber-physical Systems

Goncalo Martins[1], Sajal Bhatia[1], Xenofon Koutsoukos[1], Keith Stouffer[2], CheeYee Tang[2], and Richard Candell[2]

[1]Institute for Software Integrated Systems (ISIS), Department of Electrical Engineering and Computer Science
Vanderbilt University, Nashville, Tennessee, USA
Goncalo.Martins@vanderbilt.edu
Sajal.Bhatia@vanderbilt.edu
Xenofon.Koutsoukos@vanderbilt.edu
[2]National Institute of Standards and Technology (NIST)
Gaithersburg, Maryland, USA
Keith.Stouffer@nist.gov
CheeYee.Tang@nist.gov
Richard.Candell@nist.gov

***Abstract*—Cyber-Physical Systems (CPS) are systems that integrate physical, computational, and networking components. These systems have an impact on the physical components; it is critical to safeguard them against a range of attacks. In this paper, it is argued that an effective approach to achieve this goal is to systematically identify the potential threats at the design phase of building such systems, commonly achieved via threat modeling. In this context, a tool to perform systematic analysis of threat modeling for CPS is proposed. A real-world wireless railway temperature monitoring system is used as a case study to validate the proposed approach. The threats identified in the system are subsequently mitigated using the National Institute of Standards and Technology (NIST) SP 800-82 guidelines.**

***Keywords*-Threat Modeling, Systematic Analysis, Cyber-Physical Systems, Case Study**

I. Introduction

The exponential growth of information and communication technologies over the last decade has given rise to their expansion in real-world applications involving physical processes. This expansion has led to the emergence of closed-loop systems involving strong integration and coordination of physical and cyber (computational and communication) components, often referred to as Cyber-Physical Systems (CPS). These systems are rapidly finding their way into various aspects of the contemporary society such as transportation, healthcare, and critical infrastructure. Increasing dependence on CPS and their potential effects on the physical world, demands them to be inevitably secure, robust, reliable, and trustworthy. Ironically, it also makes such systems very attractive targets for ever increasing, both in number and complexity, cyber attacks.

The complex nature of CPS makes securing such systems go beyond securing each of these components in isolation. A multi-vector attack, exploiting a combined set of vulnerabilities from each of these individual components, can have damaging effects. A prominent recent example of such multi-vector attack was the Stuxnet attack that targeted nuclear centrifuges at the Iranian uranium enrichment plant [1]. In this attack, a worm propagating via Universal Serial Bus (USB) and local network, exploited a zero-day vulnerability of Windows machines and thereby infected the Programmable Logic Controllerss (PLCs). Another example of a multi-vector attack was the Slammer SQL worm which infected a private network at the Davis-Besse nuclear power station and resulted in a substantial time loss of safety monitoring systems [2].

Efforts in securing these CPS have mainly been towards extending the existing approaches to secure their individual components – cyber and physical. This paper, however, argues that it is imperative to simultaneously consider both these components to achieve the desired security of such systems. This goal can be achieved by identifying potential vulnerabilities of such systems, preferably during the design-phase, to minimize the overall costs involved in providing and maintaining their security and reliability. One of the ways in which this identification can be performed is *threat modeling*. In this context, various approaches have been proposed in the literature. Attack tree based approaches [3] are widely used mainly due to their simplistic design. However, static nature and state space explosion considerably restricts their modeling capabilities. Moreover, the reviewed literature also indicates a scarcity of systematic threat modeling approaches and software tools that can be used to perform a comprehensive analysis of a wide range of threats to a variety of CPS. This paper addresses these limitations and it makes the following key contributions.

- Presents a tool to perform systematic threat modeling for CPS using a real-world railway temperature monitoring system as the case study.
- Identifies threats and the corresponding mitigation using the National Institute of Standards and Technology (NIST) guidelines [4].

Another contribution of this work is the adaptation of Microsoft's Security Development Lifecycle (SDL) Threat Modeling Tool [5] for threat identification in the CPS domain. Currently, the SDL tool can be used for analyzing threats in web applications. The paper models software-related threats within the CPS domain in a systematic manner. Modeling of hardware-related threats and combining them with currently identified software-related threats constitutes a part of the future research work in this direction.

The remainder of the paper is organized as follows: Section II gives an overview of the related work in the area of threat modeling. This section also outlines the security guidelines from NIST used to address the threat identified in the case study. Section III discusses the modeling paradigm, including the metamodel and interpreters, developed for this work. Section IV describes the case study used in this paper. This section also presents the resulting modeling environment, threats identified, and addressed using NIST standards. Finally, Section V summarizes the work and gives directions for future work in this area.

II. Background and Related Work

A. Threat Modeling

Threat modeling is an approach for analyzing the security of an application. It is a structured approach that allows a systematic identification and rating of all the security-related threats that are most likely to affect the system under consideration.

Threat modeling is based on a comprehensive understanding of the underlying architecture and implementation details of the system; and provides a way to address these identified threats with appropriate countermeasures. During threat modeling, two types of models are commonly used: a model of what it is being built, and a model of the threats.

For threat models, an approach centered on asset models, attacker models, or software models is used. It is more beneficial to model threats using an individual approach at the time rather than to combine all the models [5].

The attacker-centric approach focuses on identifying the attacker, evaluating their goals, and attempting to predict how these goals might be achieved by the attacker. Software-centric threat modeling, also referred to as system-centric, design-centric, or architecture-centric, begins with the design model of the system under consideration. It focuses on all possible attacks that target each of the model elements. The asset-centric approach focuses on all the individual assets (a system or user level resource associated with certain value) entrusted to the system.

The author in [5] points out the advantages and disadvantages of assets models, attacker models, and software models. However, one of the strong motivations to apply software models for threat modeling relies on software being the foundation of any application, which makes it an ideal place to start the threat-modeling task. Moreover, almost all software development is done with software models that help understand the application. Developers are encouraged to make them good enough to allow effective threat modeling.

B. Threat Modeling Approaches

A majority of existing approaches for threat modeling can be broadly divided into two main groups – attack tree-based approaches and stochastic model-based approaches.

Attack tree-based modeling was presented in [3]. Attack trees formally describe the security of the system under consideration against a variety of attacks. They represent all possible attacks against a system in a tree structure, with the root node representing the overall goal of the attack and leaf nodes representing the different ways of achieving that goal.

Attack trees have been used in a variety of applications. Fung et al. [6] used attack trees to model three fundamental security mechanisms – confidentiality, integrity, and availability of MANET networks. Higuero et al. [7] used attack trees to model digital content security. Bistarelli et al. [8] proposed an extension to attack trees that incorporated defense mechanisms against intrusions on leaf nodes, which they termed as defense trees. This work was further extended by Kordy et al. [9] by formally introducing an attack-defense tree (ADTree). ADTrees not only took into account measures taken by an attacker to compromise a system but also incorporated defense mechanisms employed by a defender to protect the system.

Stochastic model-based threat modeling approaches commonly convert system models to Markov chains and analyze them using state transition matrices. This approach was used by Madan et al. [10] to conduct behavioral analysis of a intrusion tolerant system. Sallhammar et al. [11] presented an integrated security and dependability evaluation approach based on stochastic modeling using game theory to model attackers' behavior. Even though stochastic modeling-based approaches provide stronger and more formal modeling power than attack tree-based approaches, lack of precise representation of an attackers behavior to known distribution functions used in such models limits their usability.

C. Threat Modeling for CPS

This section outlines some of threat modeling techniques that have been applied to the CPS domain. Yampolskiy et al. [12] assessed the applicability of a data flow diagram (DFD) based approach for systematically analyzing cyber-attacks on CPS. In this context, [12] proposed a number of extensions to DFD and evaluated their proposed approach using a quad-rotor unmanned aerial vehicle (UAV) as a case study. The security assessment approach presented by the authors was manual in nature and was strongly dependent on the knowledge-base of the domain expert. Zalewski et al. [13] used Discrete Time Markov Chains (DTMC) to obtain state change (from secure to insecure) probabilities of security violations of a Cooperative Adaptive Cruise Control (CACC) system (considered a CPS system). The authors analyzed and compared two methods (Damage, Reproducibility, Exploitability, Affect Users, and Discoverability (DREAD) model [14] and Common Vulnerability Scoring System (CVSS) base metric [15]) of threat modeling of an inter vehicular communication (IVC) system using Microsoft's SDL threat modeling tool [5]. The authors in [13] acknowledged that both of the methods were developed for security analysis of Internet based applications and may not be directly applicable to CPS domain.

CPS are a combination of hardware and software modules; security of these systems is still in its infancy. To the best of our knowledge, there aren't any publicly available tools (or techniques) that automatically perform a systematic analysis of security threats in the CPS domain. As previously mentioned, it is believed that an automated hardware and software threat modeling approach, done in the design stage of the system, can help find potential problems that with other approaches would be hard or even impossible to cover.

D. Systems Security Standards

This section describes the standard document, NIST SP 800-82 Revision 2, *Guide to Industrial Control Systems (ICS) Security* [4]. This document provides guidance for establishing system security for industrial control systems (ICS). It provides a notional overview of ICS, reviews typical system topologies and architectures, identifies known threats and vulnerabilities to the ICS systems, and provides recommended security countermeasures to mitigate the associated risks. This document established a framework and process to provide guidance to perform risk assessment, security program development and deployment, and to apply security controls to ICS.

It covers the security controls in the following families: Access Control; Awareness and Training; Audit and Accountability; Security Assessment and Authorization; Configuration Management; Contingency Planning; Identification and Authentication; Incident Response; Maintenance; Media Protection; Physical and Environmental Protection; Planning; Personnel Security; Risk Assessment; System and service Acquisition; System and Communications Protection; and System and Information Integrity.

III. Modeling Paradigm by Domain

As mentioned in Section II, the number of available tools that allow a systematic analysis of threats for CPS is scarce. The scarcity of tools is dependent upon the heterogeneous features of such systems. CPS are composed of hardware and software elements which makes it challenging to model all the security requirements in one tool. To address this challenge, the Generic Modeling Environment (GME) [16] used to support the creation of domain-specific modeling for threat analysis on CPS is proposed.

GME allows the design of metamodels specifying the modeling language of the application domain. The modeling language contains all the syntactic, semantic, and presentation information regarding the domain. Moreover, the modeling language defines the family of models that can be created using the resultant modeling environment.

The proposed modeling paradigm consists of applying and extending the SDL threat modeling tool [5] to model, identify, and mitigate threats in a systematic way for the proposed CPS (Section IV).

A. Metamodel

The first step of developing a metamodel consists of defining a sketch of a metamodel for threat analysis for the proposed CPS. This is achieved by using the MetaGME modeling language, which is installed and registered by default in GME. MetaGME is basically a Unified Modeling Language (UML) Class Diagram extended with some additional concepts, including Object Constraint Language (OCL) constraints and configurable visualization properties.

The CPS components from Section IV are modeled as *first class objects* (FCOs) in GME. The defined FCOs contain both textual Attributes and Constraints. The textual Attributes are related with security aspects from the SDL threat modeling tool (e.g., authentication mechanism attribute). The Constraints are OCL-based expressions to enable verifiability for the models.

Figure 1 presents the metamodel for the proposed CPS model domain. It consist of four FCOs (sensor, repeater, gateway and central station), and two types of connections (WiFi 2.4 GHz and wireless 868 MHz). For this case-study, the components are modeled as processes from the DFD defined in the SDL threat modeling tool. The DFD process attributes were incorporated in the metamodel by implementing them as textual attributes. Figure 2 shows an example of the attributes implemented for data flow connections in GME.

B. Interpreters

One of the motivations for modeling threats in GME is the desire to describe a system in a structured way and to use the description as a form of identifying threats in a systematic way. Moreover, we also want to analyze the model automatically. Typically, the model analyzes range from simple to sophisticated:

- running queries, generating lists, and writing reports based on the contents of the model;
- generating program code or system configuration;

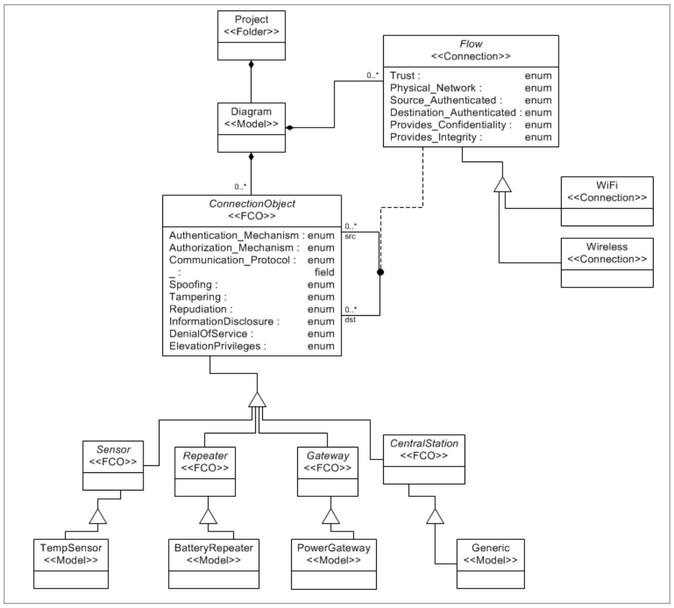

Figure 1. GME metamodel - case study

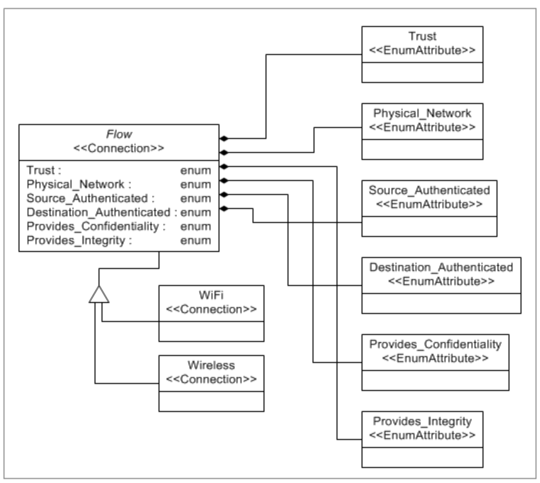

Figure 2. GME metamodel - Data Flow Connection Attributes

- using the models as a data exchange format to integrate tools that are incompatible with each other.

To perform the model analysis, programmatic access to the GME model information is required. To meet this requirement, we are using a technique provided by GME called interpreters. Interpreters are not standalone programs; they are components (usually dynamic link libraries (DLLs)) that are loaded and executed by GME upon a user's request. In this case, we developed the interpreter code responsible for navigating through the model, analyzing the results and extracting the security vulnerabilities present in all data flow connections. The security vulnerabilities are the same as the ones identified in the SDL threat modeling tool, with the exception that in this case the vulnerabilities are adapted and related to the CPS case-study system.

IV. Case Study: eRTM

The CPS case study consists of a wireless sensor network for monitoring of rail temperature[1] (eRTM system). A possible eRTM system architecture is presented in Figure 3 and consists of two main sections: a CPS system section and an Internet Protocol (IP) network section.

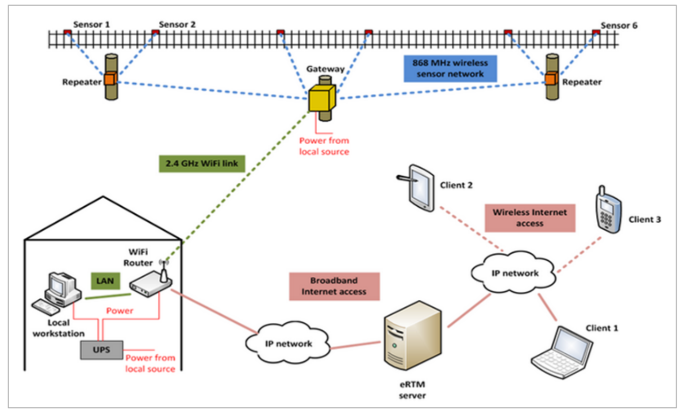

Figure 3. eRTM Generic System Architecture[1]

The CPS system section is comprised of battery-powered temperature-measuring modules (sensors) connected via 868 MHz radio channels. These sensors gather temperature-related information and communicate it to the gateway units via the repeaters. These gateway units subsequently transmit all the collected information to a central station through a 2.4 GHz WiFi link. The processed monitoring information is then communicated to clients via the conventional IP network and is made accessible through browsers and smartphone applications. Based on the temperature limit settings, alarm messages are sent to specified clients. This work covers only the CPS system section of Figure 3.

A. Resultant Modeling Environment

1) System Model: The CPS system components from Figure 3 are modeled in GME. The resultant modeling environment is presented in Figure 4. Table I summarizes the selected model properties for the components present in the modeling environment. The model properties used for each component are the following: code type, which can be native or managed; running code as, either administrator or local user; and accepts inputs, from nothing or any remote user or entity.

2) Finding Threats: There are nine data flow connections and after running the interpreter described in Section III-B, the modeling environment identified ten threats for each data flow (a total of 90 threats).

[1] http://www.evopro.hu/eng/page/ertm

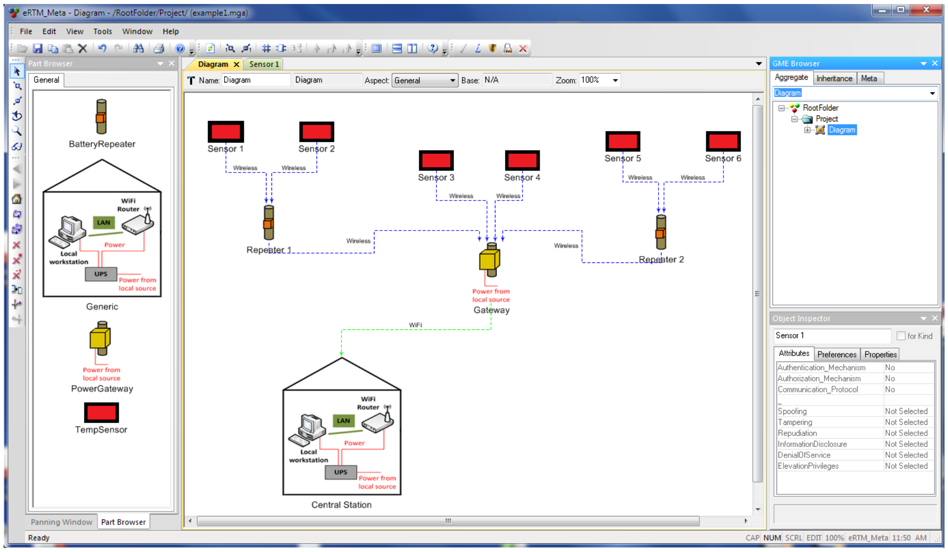

Figure 4. GME eRTM Model

Table I
MODEL PROPERTIES

Component	# Components	Model Properties		
Sensor	6	Code Type: Managed	Running As: Administrator	Accepts Input From: Nothing
Gateway	1	Code Type: Managed	Running As: Administrator	Accepts Input From: Any Remote User or Entity
Repeater	2	Code Type: Managed	Running As: Administrator	Accepts Input From: Any Remote User or Entity
Central Station	1	Code Type: Managed	Running As: Administrator	Accepts Input From: Any Remote User or Entity
WiFi	1	Physical Network: 2.4 GHz	Trust: No	
Wireless	8	Physical Network: 868 MHz	Trust: No	

As an example, two Spoofing threats, one Tampering threat, one Repudiation threat, one Information Disclosure threat, two Denial Of Service threats, and three Elevation Of Privileges threats were identified between a Sensor and a Repeater. Table II summarizes these ten threats.

3) Addressing Threats: Based on the security categorization process, the security control baseline for the eRTM case study was categorized as a moderate impact system, as the impact on confidentiality is low, and the impact on integrity and availability are both moderate [4].

According to SP 800-82 Appendix G, ICS Overlay, all security controls for the moderate baseline should be implemented. However, for the illustration purpose of this case study, certain controls are selected to directly address the threat identified by the threat modeling tool, as summarized in Table II.

V. CONCLUSION AND FUTURE WORK

The complex nature of CPSs makes securing such systems a challenge. Efforts in securing CPSs have mainly been focused towards extending the existing approaches to secure their individual components - cyber and physical. However, it is important to identify the potential vulnerabilities during the design-phase in a systematic way to minimize the overall costs involved in providing and maintaining their security and reliability. This paper addresses these challenges by proposing a tool that allows, during a CPS design phase, a systematic analysis of threat modeling for a CPS using a real-world railway temperature monitoring system as the case study. After identifying the possible threats in the modeled CPS system, the proposed approach also addresses them using the NIST security guidelines.

There are two main directions as future work: first, CPS systems are a combination of software and hardware components. To date, the proposed tool only addresses software threats. The combination and/or correlation of software and hardware threats needs to be investigated. The authors will explore the feasibility of including hardware threats in the existing modeling environment. Second, there is more than one way to do threat modeling, and the best way to threat modeling is the way that allows one to find more threats against a system. The authors will investigate ways to merge different threat modeling techniques (e.g., attack tree-based approaches) with the proposed one to enable the expansion of threat identification and system vulnerabilities.

VI. ACKNOWLEDGMENTS

This work was supported in part by NIST (70NANB13H169), the National Science Foundation under Award CNS-1238959, and by the Air Force Research Laboratory under Award FA8750-14-2-0180.

VII. DISCLAIMER

Certain commercial equipment, instruments, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not

Table II
IDENTIFIED THREATS AND RESPECTIVE MITIGATION BETWEEN A SENSOR AND A REPEATER

Threat	Description	Mitigation Control Number	Mitigation Control Name
1. Elevation Using Impersonation	Repeater may be able to impersonate the context of Sensor in order to gain additional privilege.	IA-3 (1) (4)	Device Identification and Authentication
2. Spoofing the Sensor	Sensor may be spoofed by an attacker and this may lead to unauthorized access to Repeater.	SC-7 (3) (4) (5) (7) (18) SC-8 (1)	Boundary Protection Transmission Confidentiality and Integrity
3. Spoofing the Repeater	Repeater may be spoofed by an attacker and this may lead to information disclosure by Sensor.	SC-7 (3) (4) (5) (7) (18) SC-8 (1)	Boundary Protection Transmission Confidentiality and Integrity
4. Potential Lack of Input Validation for Repeater	Data flowing across 868 MHz Wireless may be tampered with by an attacker.	SI-10	Information Input Validation
5. Potential Data Repudiation by Repeater	Repeater claims that it did not receive data from a source outside the trust boundary.	AU-8 (1) AU-9 (4)	Time Stamps Protection of Audit Information
6. Data Flow Sniffing	Data flowing across 868 MHz Wireless may be sniffed by an attacker.	SC-7 (3) (4) (5) (7) (18) SC-8 (1)	Boundary Protection Transmission Confidentiality and Integrity
7. Potential Process Crash or Stop for Repeater	Repeater crashes, halts, stops or runs slowly; in all cases violating an availability metric.	SC-5 SC-6	Denial of Service Protection Resource Availability
8. Data Flow 868 MHz Wireless Is Potentially Interrupted	An external agent interrupts data flowing across a trust boundary in either direction.	SC-5 SC-7 (3) (4) (5) (7) (18) SC-8 (1)	Denial of Service Protection Boundary Protection Transmission Confidentiality and Integrity
9. Repeater May be Subject to Elevation of Privilege Using Remote Code Execution	Sensor may be able to remotely execute code for Repeater.	IA-3 (1) (4) SC-7 (3) (4) (5) (7) (18) SC-8 (1) SI-7 PE-4	Device Identification and Authentication Boundary Protection Transmission Confidentiality and Integrity Software, Firmware, and Information Integrity Access Control for Transmission Medium
10. Elevation by Changing the Execution Flow in Repeater	An attacker may pass data into Repeater in order to change the flow of program execution within Repeater to the attacker's choosing.	IA-3 (1) (4) SC-7 (3) (4) (5) (7) (18) SC-8 (1) SI-7 PE-4	Device Identification and Authentication Boundary Protection Transmission Confidentiality and Integrity Software, Firmware, and Information Integrity Access Control for Transmission Medium

intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

REFERENCES

[1] N. Falliere, L. O. Murchu, and E. Chien, "W32. stuxnet dossier," *White paper, Symantec Corp., Security Response*, 2011.

[2] E. Levy, "Crossover: online pests plaguing the off line world," *Security & Privacy, IEEE*, vol. 1, no. 6, pp. 71–73, 2003.

[3] B. Schneier, "Attack trees: modeling security threats," 1999.

[4] K. Stouffer, S. Lightman, V. Pillitteri, M. Abrams, and A. Hahn, "NIST Special Publication 800-82 Revision 2, Guide to Industrial Control Systems (ICS) Security," tech. rep., National Institute of Standards and Technology, 2014.

[5] A. Shostack, *Threat Modeling, Designing for Security*. Wiley, 2014.

[6] C. Fung, A. Chen, X. Wang, J. Lee, R. Tarquini, M. Anderson, and R. Linger, "Survivability analysis of distributed systems using attack tree methodology," in *Military Communications Conference, 2005. MILCOM 2005. IEEE*, pp. 583–589, IEEE, 2005.

[7] M. Higuero, J. Unzilla, E. Jacob, P. Saiz, M. Aguado, and D. Luengo, "Application of'attack trees' in security analysis of digital contents e-commerce protocols with copyright protection," in *Security Technology, 2005. CCST'05. 39th Annual 2005 International Carnahan Conference on*, pp. 57–60, IEEE, 2005.

[8] S. Bistarelli, F. Fioravanti, and P. Peretti, "Defense trees for economic evaluation of security investments," in *Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on*, pp. 8–pp, IEEE, 2006.

[9] B. Kordy, S. Mauw, S. Radomirović, and P. Schweitzer, "Foundations of attack–defense trees," in *Formal Aspects of Security and Trust*, pp. 80–95, Springer, 2011.

[10] B. B. Madan, K. Gogeva-Popstojanova, K. Vaidyanathan, and K. S. Trivedi, "Modeling and quantification of security attributes of software systems," in *Dependable Systems and Networks, 2002. DSN 2002. Proceedings. International Conference on*, pp. 505–514, IEEE, 2002.

[11] K. Sallhammar, B. E. Helvik, and S. J. Knapskog, "Towards a stochastic model for integrated security and dependability evaluation," in *Availability, Reliability*

and Security, 2006. ARES 2006. The First International Conference on, pp. 8–pp, IEEE, 2006.

[12] M. Yampolskiy, P. Horvath, X. D. Koutsoukos, Y. Xue, and J. Sztipanovits, "Systematic analysis of cyber-attacks on cps-evaluating applicability of dfd-based approach," in *Resilient Control Systems (ISRCS), 2012 5th International Symposium on*, pp. 55–62, IEEE, 2012.

[13] J. Zalewski, S. Drager, W. McKeever, and A. J. Kornecki, "Threat modeling for security assessment in cyberphysical systems," *CSIIRW '13 Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop, Article No. 10*, 2013.

[14] J. A. Ingalsbe, L. Kunimatsu, T. Baeten, and N. R. Mead, "Threat modeling: diving into the deep end," *Software, IEEE*, vol. 25, no. 1, pp. 28–34, 2008.

[15] P. Mell, K. Scarfone, and S. Romanosky, "A complete guide to the common vulnerability scoring system version 2.0," in *Published by FIRST-Forum of Incident Response and Security Teams*, pp. 1–23, 2007.

[16] A. Ledeczi, M. Maroti, A. Bakay, G. Karsai, J. Garrett, C. Thomason, G. Nordstrom, J. Sprinkle, and P. Volgyesi, "The generic modeling environment," *IEEE International Symposium on Intelligence Signal Processing (WISP), May*, 2001.