Journal of Computing and Information Science in Engineering. Received September 30, 2020;
Accepted manuscript posted March 15, 2021. doi:10.1115/1.4050685
Copyright © 2021 by ASME

A Zero Trust Hybrid Security and Safety Risk Analysis Method

Nikolaos Papakonstantinou
VTT Technical Research Center, Finland
Email: Nikolaos.Papakonstantinou@vtt.fi

Douglas L. Van Bossuyt[*]
Department of Systems Engineering
Naval Postgraduate School
Monterey, CA 93943 USA
Email: douglas.vanbossuyt@nps.edu

Joonas Linnosmaa
VTT Technical Research Centre, Finland
Email: Joonas.Linnosmaa@vtt.fi

Britta Hale
Department of Computer Science
Naval Postgraduate School
Monterey, CA 93943 USA
Email: britta.hale@nps.edu

Bryan O'Halloran
Department of Systems Engineering
Naval Postgraduate School
Monterey, CA 93943 USA
Email: bmohallo@nps.edu

ABSTRACT

Designing complex, socio-technical, cyber-physical systems has become increasingly challenging in recent years. Interdependencies between engineering domains can lead to emergent behavior that is difficult to predict and manage. The recent shift toward model-based design has demonstrated significant advantages for minimizing these challenges [1]. Further, the early identification of safety and security design weaknesses in safety-critical systems leads to reduced redesign costs in later design phases [2, 3]. As a result, this article contributes the the Multidisciplinary Early Design Risk Assessment Framework (MEDRAF) methodology for early combined safety and security assessment based on interdisciplinary dependency models of a system. The focus is on factors con-

[*] Address all correspondence to this author.

Journal of Computing and Information Science in Engineering. Received September 30, 2020;
Accepted manuscript posted March 15, 2021. doi:10.1115/1.4050685

tributing to the estimation of the probabilities of successful attacks on system components. The Zero Trust paradigm is applied in which all humans, hardware, and processes interacting with the system are considered to pose a security risk. A calculation of security-related probability estimates is presented which is dependent on the current global security environment. Subsequently, security and safety probability estimates are combined to present an overall safety-security risk calculation using hybrid safety-security trees. The risk values help designers assess the loss of specific key components and safety functions. The methodology is demonstrated with a case study of a spent fuel pool cooling system in a nuclear reactor. The results of the case study show that the risk of losing one key system component doubles when combining security and safety compared to only assessing safety events. This paper is based on a paper presented at the CIE 2020 conference [4].

1 Introduction

With the emergence of new security risks across the globe, many modern, safety-focused systems have become vulnerable to security threats. Increased system interoperability demands further exacerbate the issue. While vulnerability to security threats has increased, the demand for highly capable systems has also increased. Systems are expected to operate more effectively while they also are being designed on smaller budgets and shorter schedules. This scenario can result in the design of a system that contains safety hazards, security vulnerabilities, or reliability issues.

The field of model-based design has emerged as an approach to deal with the aforementioned scenario. Examples of model-based approaches include model-based systems engineering (MBSE), digital engineering, and digital twin. One benefit of a model-based approach for designing a highly complex system is that it can reduce the time spent performing analyses when compared to a paper-driven approach, offering more engagement between the analyst and the analysis. As a result, the approach in this article leverages a model-based design approach as a foundation for modeling safety and security.

The scientific contribution of this paper is the the Multidisciplinary Early Design Risk Assessment Framework (MEDRAF) methodology which is an extension of the methodology presented in [5] for generating hybrid safety and security trees from interdisciplinary early system dependency models. This paper specifically focuses on performing estimations for the security-related threat probabilities towards a Zero Trust model-driven methodology for hybrid safety and security assessment. Namely, in addition to the core issues of reliability, safety, and security, we also incorporate a Zero Trust approach that assumes the reality of risks posed by individual components, processes, and humans.

A set of security factors is proposed as a basis for the estimation of threat levels extending previous work [6]. As a result of the quantitative approach, data and values for possible hazardous security and safety basic events is needed. Reliability and safety data has been extensively gathered during the past decades; however, security data is less understood or access restricted, amplifying the need to draw knowledge from the system, or in the case of this work, the hybrid model. These security probability calculations, combined with a consequence, enable the estimation of the overall risk of losing a key function or component.

The remainder of this paper is organized the following way. First, related research and the necessary background knowledge is presented in Section 2. In Sections 3 and 4 the methodology is developed and then applied in a case study for

Journal of Computing and Information Science in Engineering. Received September 30, 2020;
Accepted manuscript posted March 15, 2021. doi:10.1115/1.4050685

a spent fuel pool subsystem in a nuclear reactor. Section 5 presents a discussion and Section 6 concludes and offers future direction of the research.

2 Background

A portion of the systems engineering process focuses on defining the system's required functionality early in the lifecycle, proposing and maturing the system, and then verifying and validating (V&V) the final system design against the requirements of the stakeholders. Within the technical systems engineering processes, off-nominal behavior is considered independently through several domain areas including security, safety, and reliability [7]. To successfully minimize unwanted behavior and events from the system, assessments must be applied to the system from the initial design phase.

2.1 Safety Assessment, Risk Analysis

Best practices in systems engineering promote the inclusion of system safety early in a system's design process. Depending on the design phase, the desired results, and the data available, either qualitative or quantitative safety analysis can be carried out to study the risks in the target system. Basic safety and risk analysis methods (e.g. Preliminary Hazard Analysis (PHA), Fault Tree Analysis (FTA), Failure Modes, Effects, and Criticality Analysis (FMECA), Probabilistic Risk Assessment (PRA), Function Failure Identification and Propagation Methodology (FFIP), etc.) have been extensively studied [8, 9, 10, 11, 12]). We use fault trees in this paper; fault trees are "a systematic engineering technique that provides a diagrammatic representation of the relationships between specific events or component failures and an undesirable top event" [13]. Fault trees as a tool for early phase dependence modelling is presented in [14], where fault trees are used for a safety-focused analysis. The concept of fault trees is further expanded with ideas from attack trees and human attacker scenarios into hybrid trees to cover security related topics in [5]. In [15] the hybrid approach is expanded to cover similarities for Defence-in-Depth in safety and security.

2.2 Model-Based Systems Engineering Process

Model-based systems engineering (MBSE) consolidates many models, data sets, and other design process information into one or several inter-operable databases to connect relevant information, analysis, and design efforts together during the system design process [16]. During the construction, commissioning, and operation of systems designed using the MBSE philosophy, it is increasingly common to feed information from these phases of the systems engineering process (which encompasses the system design process) into the same database(s) and re-run analyses conducted during the design phase to V&V the assumptions and requirements of the system [17]. Estefan surveys MBSE methodologies to capture the various uses of MBSE models to achieve the intent of the system engineering design process [18]. Cameron et al. [19] and Weilkiens et al. [20] apply MBSE specifically to system architecture models; however, their work is outside the scope of the interdisciplinary modeling of security, safety, reliability, and similar disciplines. As noted in [21], more research is required to understand how to use MBSE for decision making. Madni and Sievers [22] acknowledge the same shortcoming and further suggest that communication between MBSE models and systems engineering processes is currently lacking. The

Journal of Computing and Information Science in Engineering. Received September 30, 2020;
Accepted manuscript posted March 15, 2021. doi:10.1115/1.4050685

work presented in this manuscript is intended to offer systems engineers appropriate and insightful information during a system's design process with the intent to improve the design process outcome.

2.3 Security Analysis, Zero Trust

Security analyses have long focused on the security of critical components and assumed away all other threats by declaring other aspects of the system out-of-band (OOB). Unfortunately, this leads to a narrow and unrealistic view of system security; as such OOB components are considered perfectly secure in such analyses. To address this issue, security analyses on both the system level, and fine-grained protocol and primitive level have been expanding to encompass an integrated view, with nothing counted as OOB. At the protocol analysis level, integrated models aka. "ceremonies" were introduced to capture multi-disciplinary actions such as human behavior with respect to devices and multiple device components [23,24,25,26,27]. On the cyber systems level, research has expanded to cover a wider variety of attacks and more disruptive attacks within global networks [28].

Parallel to the above analyses is security risk analysis. As in other sub-domains of security, risk analysis has expanded to include traditionally out-of-scope considerations such as insider threats in industrial technological systems [29] and combining security and safety in risk assessment [30]. Under scenarios where security is influenced by all factors in a system – human and machine alike – it is important to consider threat assessment from all directions, namely "the way forward cannot be found solely in mathematics or technology" [31].

Zero Trust (ZT) is an information security framework that states that organizations should not trust any entity inside or outside of their perimeter at any time. Every device, user, app, and network that has access to business data needs to be secured, managed, and monitored – the principle of "never trust, always verify" [32]. Instead of protecting network segments, ZT focuses on protecting resources, i.e. instead of assuming that every user inside a network is trustworthy and cleared for access, no user is trusted, whether inside or outside of the network. This is specifically in response to enterprise trends that include remote users and cloud-based assets that are not located within an enterprise-owned network boundary.

The American Council for Technology-Industry Advisory Council (ACT-IAC) has published a report about new and more effective IT security architectures for governmental agencies [33]. In their work they found that ZT solutions are widely available and currently in use in the private sector, and that there are many companies developing new capabilities and solutions to support ZT. In the U.S., NIST (National Institute of Standards and Technology) is preparing to publish a NIST Special Publication (SP) 800-207 about Zero Trust Architecture (ZTA) [34]. In research and industry ZT has been used, for example, for Big Data security [35], IoT [36], and Cloud computing [37].

To do quantitative/probabilistic analysis for security, value data (or estimates) for security threats related basic events are needed. According to Buldas et al. [38], getting this information is not as easy as largely believed. Buldas et al. state that assigning a reliable attribute value, which is sufficiently precise and refined, to each basic attack step has proven difficult and hard to obtain. Often in industry, companies manage to obtain their own statistical historical data for abstract attacks, but might struggle with more refined statistics. Buldas et al. concluded there is a clear tension between the limited availability of data and the need to get values for risk quantification methods. Such tension was observed when reviewing papers from the domain; research papers introducing novel security analysis methods, for example [39, 40], often assume that the data values are readily available and do not discuss the topic further.

One popular public source for estimating security related data values is the Common Vulnerability Scoring System (CVSS) [41], which is a public initiative designed to address the issue of scoring software vulnerabilities by presenting a framework for assessing and quantifying them. For example, [42] use a combination of Markov chains and CVSS to compute the probability distribution of cloud security threats. The same can be observed in [43], which explores attack graphs using Bayesian Networks and the CVSS index. Another often referenced study by [44] identified the biggest security threats in industry and the probabilities relating to them (see Table 1) by interviewing chief information security officers and from literature review. [45] uses data from the United States Department of Health and Human Services (HHS) for an estimation of cyber-security breach probability using Bayes formula. We take a different approach: while these variants consider probabilities of specific attacks, and are therefore tied to system-specific design (e.g. software specific security attacks or experience of individuals working on specific systems), we tie probabilities to human behavior as well. This necessitates a model design paradigm change that uses the human actor as the cyber attacker, instead of considering attackers as devices.

Number of Attacks per Month	> 100	$51-100$	$10-50$	< 10	None	No Answer
1. Act of Human Error or Failure	5.2%	2.1%	14.6%	41.7%	24.0%	12.5%
2. Compromises to Intellectual Property	1.0%	2.1%	3.1%	25.0%	61.5%	7.3%
3. Deliberate Acts of Espionage or Treason	4.2%	3.1%	3.1%	20.8%	68.8%	
4. Deliberate Acts of Information Extortion			1.0%	8.3%	90.6%	
5. Deliberate Acts of Sabotage or Vandalism	1.0%		3.1%	31.3%	64.6%	
6. Deliberate Acts of Theft			7.3%	38.5%	54.2%	
7. Deliberate Software Attacks	11.5%	9.4%	14.6%	47.9%	16.7%	
8. Forces of Nature	1.0%		2.1%	34.4%	62.5%	
9. Quality of Service Deviations from Service Providers		1.0%	8.3%	43.8%	46.9%	
10. Technical Hardware Failures or Errors		3.1%	11.5%	51.0%	34.4%	
11. Technical Software Failures or Errors		5.2%	18.8%	45.8%	30.2%	
12. Technological Obsolescence		1.0%	15.6%	21.9%	60.4%	1.0%
Average Responses	4.0%	3.4%	8.6%	34.2%	51.2%	6.9%

Table 1. NUMBER OF ATTACKS PER MONTH AS REPORTED BY WHITMAN [44]

2.4 Combined safety and security assessment

While work has been done on the security of industrial technology systems [29] and smart grids [46], these works focused strictly on the security aspects of critical infrastructure. The usual approaches to analysis that these works apply include attack trees [47, 48, 49] and game theoretic approaches [50, 51]. We build on the attack tree line of research but extend the cyber security risk focus to safety.

Journal of Computing and Information Science in Engineering. Received September 30, 2020;
Accepted manuscript posted March 15, 2021. doi:10.1115/1.4050685

One piece of work that similarly takes on the complicated, combined problem of security and safety risk analysis does so by combining bowtie analysis and attack trees for respective safety and security analysis [30]. The two components are separately analyzed, and the likelihood is presented as an output pair (safety, security). The bowtie analysis is based on a fault tree (left part of the "bow") leading to an event (center of the "bow") and then follows an event tree (right part of the "bow"). Traditional PRA, at least in the nuclear domain, follows a different structure where a set of initiating events is considered as the design basis of the plant (i.e.: the system should be prepared for the initiating events). For each of these events there is an event tree that estimates the consequences depending on the activation of safety functions or other mitigation measures. Then fault trees are developed for each of the safety functions for the calculation of the probability of the consequences. In this paper we focus on fault trees for the safety analysis, taking security also into account, which allows for smooth integration of attack trees in the hybrid model. As part of the ZT framework, and building on the larger view of integrated analysis within security, we do not separate out the component pieces but allow for the potential of a security attack to influence the safety of a system and vice versa, yielding a single output. This is consistent with real-world scenarios, where a cyber attack could cause system failure, etc., resulting in a safety risk, while failure of physical system safety could open the door to a cyber attack.

3 Methodology

The key concept presented in this section is the development of a methodology to support combined safety and security assessment. The methodology is applied during the early phase of design and is based on the interdisciplinary model of a complex, socio-technical, and cyber-physical system. Previous work [5] focused on the automatic creation of hybrid (safety/security) fault/attack tree topology from a dependency model. In this previous research no consideration was given to the actual calculation of overall risk; it was out of scope. This prior work is extended in this article with the estimation of security-related probabilities, the quantification of an overall combined safety and security risk, and the formalization of a model-based approach. This extension completes the combined safety and security risk assessment and it is closer to the state of the art safety-focused methods like PRA.

The motivation of the proposed methodology is to provide a practical workflow for early complex system design that is based on 3 key principles:

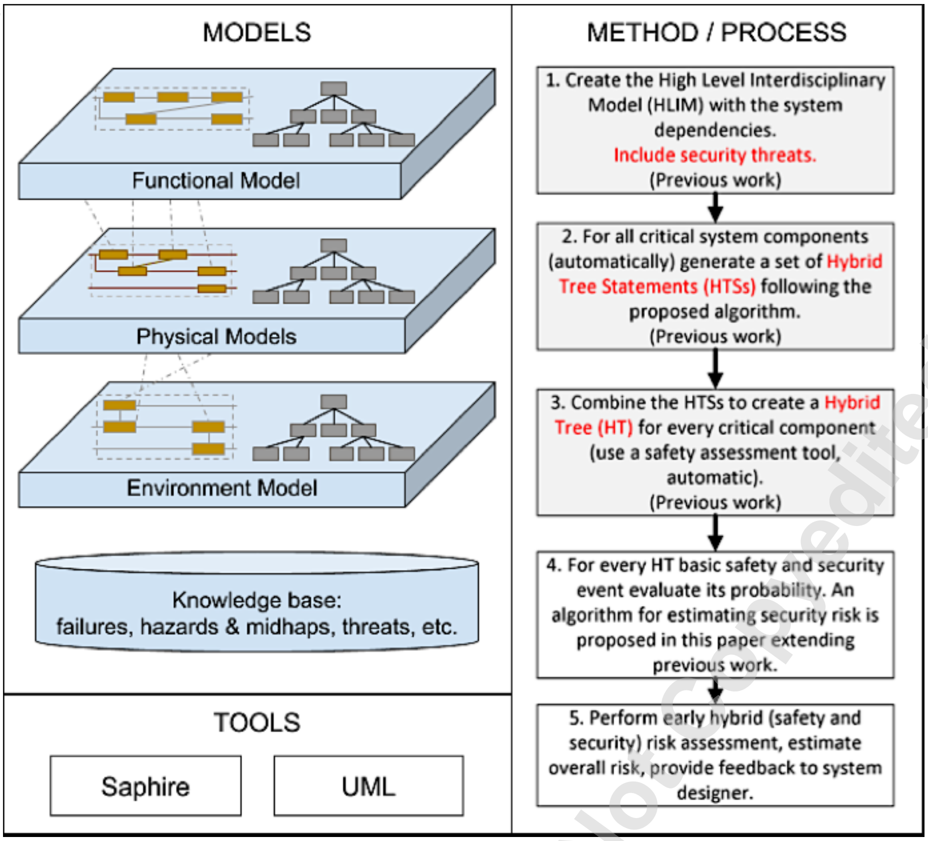
1. **Holistic interdisciplinary modeling** can aid safety and security assessments to identify difficult issues emerging from interdisciplinary relationships between system elements. These dependencies are often discovered late in the design process when system development happens in engineering discipline-based silos.
2. **Combined safety and security assessments**, instead of treating safety and security separately, can highlight safety implications of security incidents as well as promote the concept of overall resilience. There is significant methodology overlap between safety and security engineering during design (e.g. defense in depth, ZT, every component may fail) and during assessment (fault/attack trees, event trees, overall risk calculations). An added benefit is the early identification of trade-offs between safety and security (e.g. access to critical safety equipment).

Journal of Computing and Information Science in Engineering. Received September 30, 2020;
Accepted manuscript posted March 15, 2021. doi:10.1115/1.4050685

3. These early combined assessments need to be **automated** based on past knowledge, prototype software tools for DiD, and PRA modeling (fault trees, event trees). Automation enables the potential to provide near real-time support to designers for safe and secure designs, so that design decisions that increase the overall risk are noticed as soon as possible. This is not easy when assessments are done at major system development milestones down the line; when issues are identified at milestones, the consequence can be a costly redesign – or even worse – an attempt to justify the weakness and ask for an exception.

To clarify the methodology's application, we first introduce and discuss a generalized framework (the Multidisciplinary Early Design Risk Assessment Framework (MEDRAF)) to relate the methodology (system-agnostic) with the dependency model (system-specific). The methodology ("METHOD & PROCESS" and "TOOLS" in Figure 1) developed in this work is inherently system-agnostic. We adhere to the following definitions, which are based on [52]; a methodology itself is a composition of a process, a method, and a tool; a process is a set of tasks designated to accomplish a specific goal; a method is a set of techniques applied within a process to accomplish a specific goal; and a tool is the instrument utilized by the method to accomplish a specific goal [52]. In contrast to the methodology, the dependency model ("MODELS" in Figure 1) is a representation of the system. Within the dependency model is a set of common systems engineering design models. These models generally fall into the following categories: functional models (e.g. functional block diagram and functional hierarchy) to relate levels of the functional hierarchy to one another and functions within a hierarchy level to one another, including functional redundancy; discipline-specific models to capture the specific relationships between physical items and their associated principles (e.g. master interconnect diagram, electrical diagrams, mechanical diagrams, etc.); knowledge bases to use as source data (e.g. list of failures, list of hazards, historical occurrence, etc.); and interoperability models that capture the mapping between the models (indicated with dashed lines in Figure 1). Figure 1 presents an overview of the method and process, the models, and the tools in the context of MEDRAF. The remainder of this section describes the method; the approach used is to present the process by discussing the method's sttiff. The techniques, tools, models, etc. are then discussed within each step.

Journal of Computing and Information Science in Engineering. Received September 30, 2020;
Accepted manuscript posted March 15, 2021. doi:10.1115/1.4050685



1. Create the High Level Interdisciplinary Model (HLIM) with the system dependencies. Include security threats. (Previous work)

2. For all critical system components (automatically) generate a set of Hybrid Tree Statements (HTSs) following the proposed algorithm. (Previous work)

3. Combine the HTSs to create a Hybrid Tree (HT) for every critical component (use a safety assessment tool, automatic). (Previous work)

4. For every HT basic safety and security event evaluate its probability. An algorithm for estimating security risk is proposed in this paper extending previous work.

5. Perform early hybrid (safety and security) risk assessment, estimate overall risk, provide feedback to system designer.

Fig. 1. OVERVIEW OF THE MULTIDISCIPLINARY EARLY DESIGN RISK ASSESSMENT FRAMEWORK (MEDRAF).

Stiff 1-3 are described in detail in [5]. The **first step** calls for the development of an interdisciplinary model of the system that maps dependencies between system components and across disciplines. Within the dependency model are several design models to capture different aspects of the system including functionality, processes (e.g. pipelines, process components), electrical distribution, instrumentation and automation software, environment (e.g. floors, rooms), human factors, etc. A variety of software packages such as Papyrus [53] support constructing these models using the Universal Modeling Language (UML) and similar modeling languages. Security-related elements are also included, such as external attackers, to extend the model and allow it to be used in **Step 2** as a basis for the generation of Hybrid (fault/attack) Tree Statements (HTS) . Each HTS is an individual path between the top event and a cause of the top event. An algorithm outlined in [5] is used to generate the HTS. The set of HTS are merged to generate the complete Hybrid Tree in **Step 3** of the methodology. This process is assisted by software that parses the interdisciplinary model in an open standard modelling language such as UML. Software such as [54] and [55] can be used to model the Hybrid Trees.

Step 4 of the methodology (the first primary contributions of this paper) is the estimation of the threat risk level of security-related basic events. The source of the attack is always considered to be a human, either as part of the system (following the ZT principle) or external to the system. Figure 2 presents a set of key factors involved in the probability of a successful attack.

Journal of Computing and Information Science in Engineering. Received September 30, 2020;
Accepted manuscript posted March 15, 2021. doi:10.1115/1.4050685

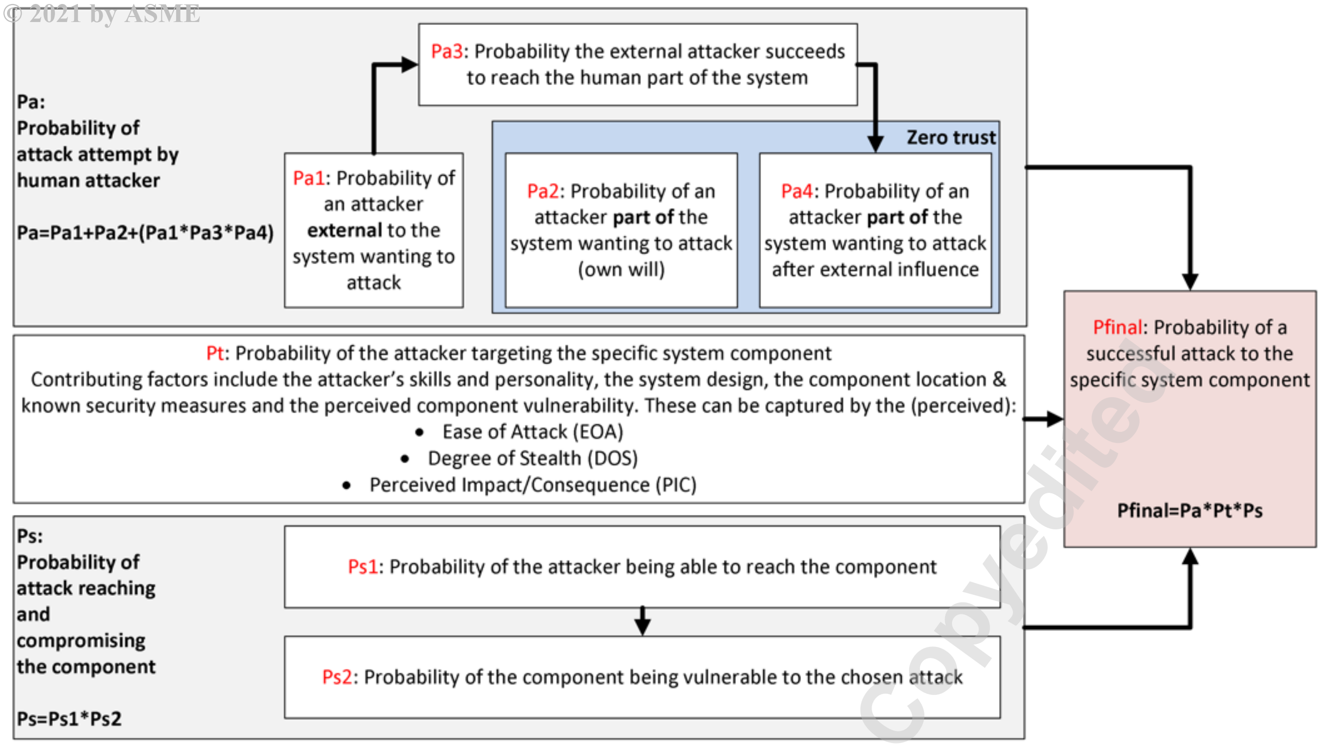

Pa:
Probability of attack attempt by human attacker
Pa=Pa1+Pa2+(Pa1*Pa3*Pa4)
Pa3: Probability the external attacker succeeds to reach the human part of the system
Zero trust
Pa1: Probability of an attacker external to the system wanting to attack
Pa2: Probability of an attacker part of the system wanting to attack (own will)
Pa4: Probability of an attacker part of the system wanting to attack after external influence
Pt: Probability of the attacker targeting the specific system component
Contributing factors include the attacker's skills and personality, the system design, the component location & known security measures and the perceived component vulnerability. These can be captured by the (perceived):
• Ease of Attack (EOA)
• Degree of Stealth (DOS)
• Perceived Impact/Consequence (PIC)
Pfinal: Probability of a successful attack to the specific system component
Pfinal=Pa*Pt*Ps
Ps:
Probability of attack reaching and compromising the component
Ps=Ps1*Ps2
Ps1: Probability of the attacker being able to reach the component
Ps2: Probability of the component being vulnerable to the chosen attack

Fig. 2. KEY FACTORS CONTRIBUTING TO THE ESTIMATION OF THE PROBABILITY OF A SUCCESSFUL ATTACK TO A SYSTEM COMPONENT (Pfinal). PREVIOUS WORK [6] HAS DISCUSSED THE ESTIMATION OF THE PROBABILITY Pt.

Firstly, it is important to estimate the probability of a human wanting to attack the system (**Pa**). This can be broken down to different scenarios:

* A human external to the system wanting to attack directly (Pa1) or by being successful in reaching and influencing a human part of the system to attack using social engineering methods (Pa1*Pa3*Pa4).
* A human part of the system wanting to attack on her own will (Pa2).
* This results to $\boldsymbol{Pa} = Pa1+Pa2+(Pa1*Pa3*Pa4)$.

Then follows an estimation of the attacker choosing to attack a specific system component (Pt). Previous work [6] presented three key factors for the “attractiveness” of attacking a specific component based on the perception of the attacker:

* Ease of Attack (EOA).
* Degree of Stealth (DOS), probability of the attack being undetected.
* Perceived Impact/Consequence (PIC) for how severe the attack would be to the system.

The final factors are the estimation of the attacker being able to reach (using physical or cyber means) the system component under attack (Ps1) (e.g. depending on the relevant location to the attacker) and for the component being vulnerable to that specific attack (Ps2). Note the difference of the attacker's perception on how easy an attack is (EOA parameter, part of Pt probability) and the estimation of how easy the attack is (Ps1). The total probability (**Ps**) to reach and compromise the component is:

$$\mathbf{Ps} = \mathrm{Ps1*Ps2}$$

Journal of Computing and Information Science in Engineering. Received September 30, 2020;
Accepted manuscript posted March 15, 2021. doi:10.1115/1.4050685

The combination of these probabilities gives the overall probability of a successful attack on a system component (**Pfinal**):

$$\mathbf{Pfinal} = \mathrm{Pa{*}Pt{*}Ps}$$

The estimation of specific numerical values for the different probabilities strongly depends on the domain (e.g. defense systems, safety critical infrastructure, etc.) and the current security environment (e.g.: attackers, vulnerabilities). It is important to note that **security assessments are not static** because the causes (attackers) and the possible vulnerabilities change/evolve over time as new types of attacks are developed.

The sensitivity of the outcome of Step 4 to changes in the above probabilities is highly system-specific. It is recommended to conduct a sensitivity study of a system to understand which probability variables may require additional uncertainty reduction.

In **Step 5** (the second primary contribution of this paper) the reliability probabilities and the attack probabilities are combined in a common fault/attack tree for the estimation of the overall risk of losing a specific component/function.

The dependencies between the components of a simple process example are presented in Figure 3. The basic system components are named A to E, where components C and D are redundant. Attackers AttA and AttD (see Figure 3) are also included in the model. A combined safety and security fault/attack tree model can be developed in a safety assessment tool like SAPHIRE (see Figure 4) focusing on top event, e.g. the failure of component A. Some basic events of this tree are reliability related and existing knowledge from databases or expert knowledge can be applied to add failure probabilities. The process described above (also see Figure 2) can help to estimate the probability of components A and D being successfully attacked by AttA and AttD respectively. Even using arbitrary estimations for probabilities Pa, Pt, and Ps for attacks to system components, a risk assessment that includes just reliability aspects does give a lower overall risk of a top event compared to a hybrid (safety/security) assessment. In this simple example, using a fixed probability for reliability and security basic events (10^{-3}), the fault tree (only faults) gave an overall risk of $6 \cdot 10^{-3}$ while the model including faults and attacks gives an overall risk of $7 \cdot 10^{-3}$. The main additional contribution was from the basic event of AttA causing harm to Component A with a probability of 10^{-3}).

Journal of Computing and Information Science in Engineering. Received September 30, 2020;
Accepted manuscript posted March 15, 2021. doi:10.1115/1.4050685

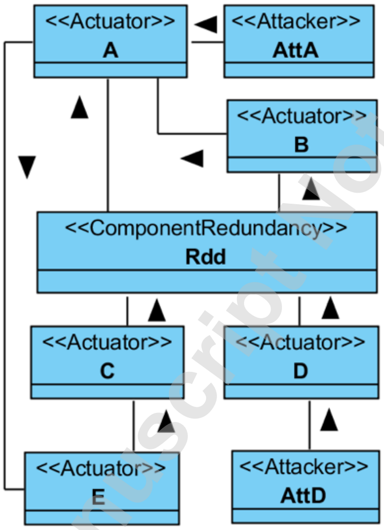


Fig. 3. DEPENDENCY MODEL OF A SIMPLE PROCESS INCLUDING ATTACKERS [5].

Journal of Computing and Information Science in Engineering. Received September 30, 2020;
Accepted manuscript posted March 15, 2021. doi:10.1115/1.4050685

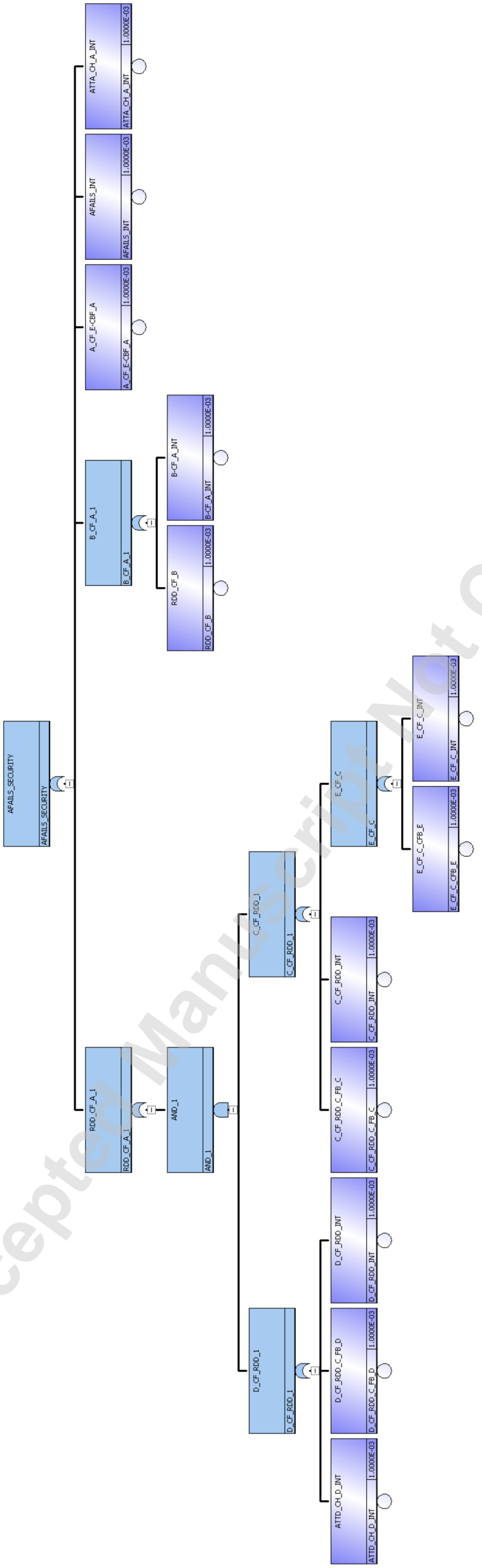


Fig. 4. HYBRID FAULT/ATTACK TREE OF THE SIMPLE PROCESS IN THE SAPHIRE PRA TOOL.

Journal of Computing and Information Science in Engineering. Received September 30, 2020;
Accepted manuscript posted March 15, 2021. doi:10.1115/1.4050685

4 Case Study

The case study demonstrating the proposed MEDRAF methodology for hybrid safety/security assessment is a spent fuel pool cooling system (also used in previous relevant work [5, 15]). The temperature of a water pool containing spent nuclear fuel is regulated using two redundant cooling loops and an additional emergency water supply. An overview of the system is given in Figure 5.

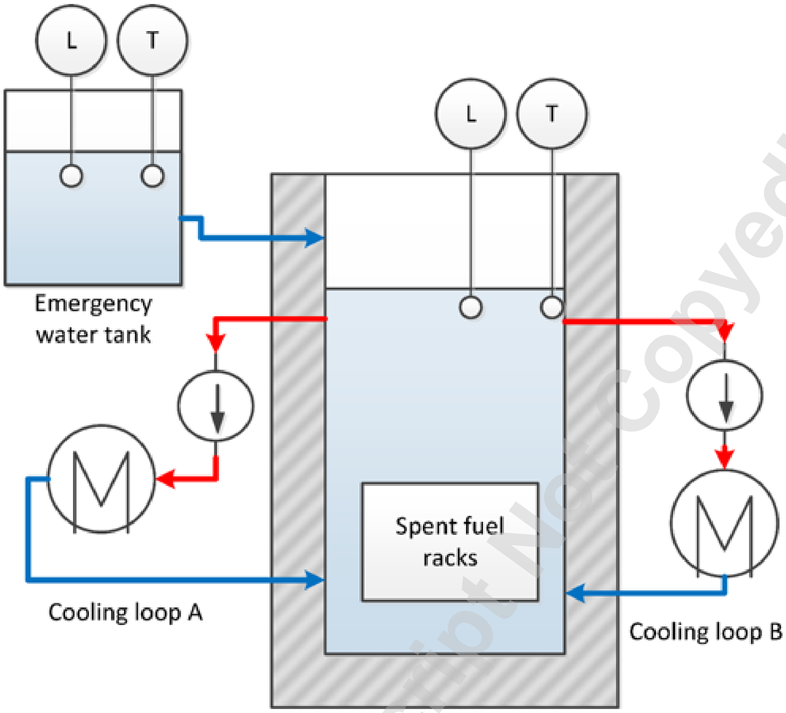


Fig. 5. SYSTEM OVERVIEW OF THE SPENT FUEL POOL COOLING CASE STUDY. THIS SYSTEM INCLUDES TWO REDUNDANT COOLING LOOPS AND AN EMERGENCY WATER SUPPLY.

The interdisciplinary dependency model of the system includes several diagrams as follows. Figure 6 describes the topology of the process where part of the process model is shown with arrows indicating flow through the system. Figure 7 shows the human factors model where part of the model of how humans interact with the system is provided indicating the internal and external threats to the system posed by the humans. Figure 8 shows the allocation of components to rooms in the power plant. Figure 9 is a model of how the automation software interacts with the system. Human attackers external to the system are also added to the model. These diagrams are commonly generated using software packages such as Cameo [56], Papyrus [53], and similar tools as part of a digital engineering and MBSE-driven system design process [19, 17, 57]. Following the Zero Trust paradigm, security attacks can originate from humans outside or within the system (e.g. through their own will or via coercion).

Journal of Computing and Information Science in Engineering. Received September 30, 2020;
Accepted manuscript posted March 15, 2021. doi:10.1115/1.4050685

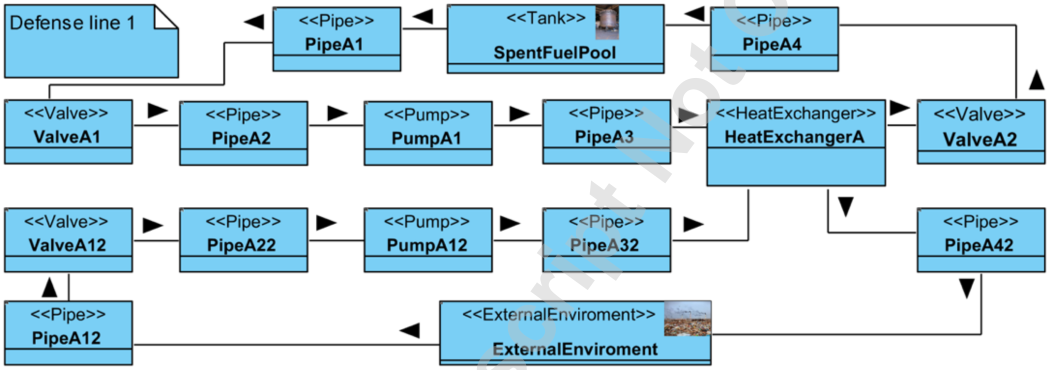


Fig. 6. PART OF THE PROCESS MODEL OF THE SPENT FUEL POOL COOLING SYSTEM CASE STUDY SHOWING ONE OF THE REDUNDANT COOLING LOOPS [5].

Journal of Computing and Information Science in Engineering. Received September 30, 2020; Accepted manuscript posted March 15, 2021. doi:10.1115/1.4050685

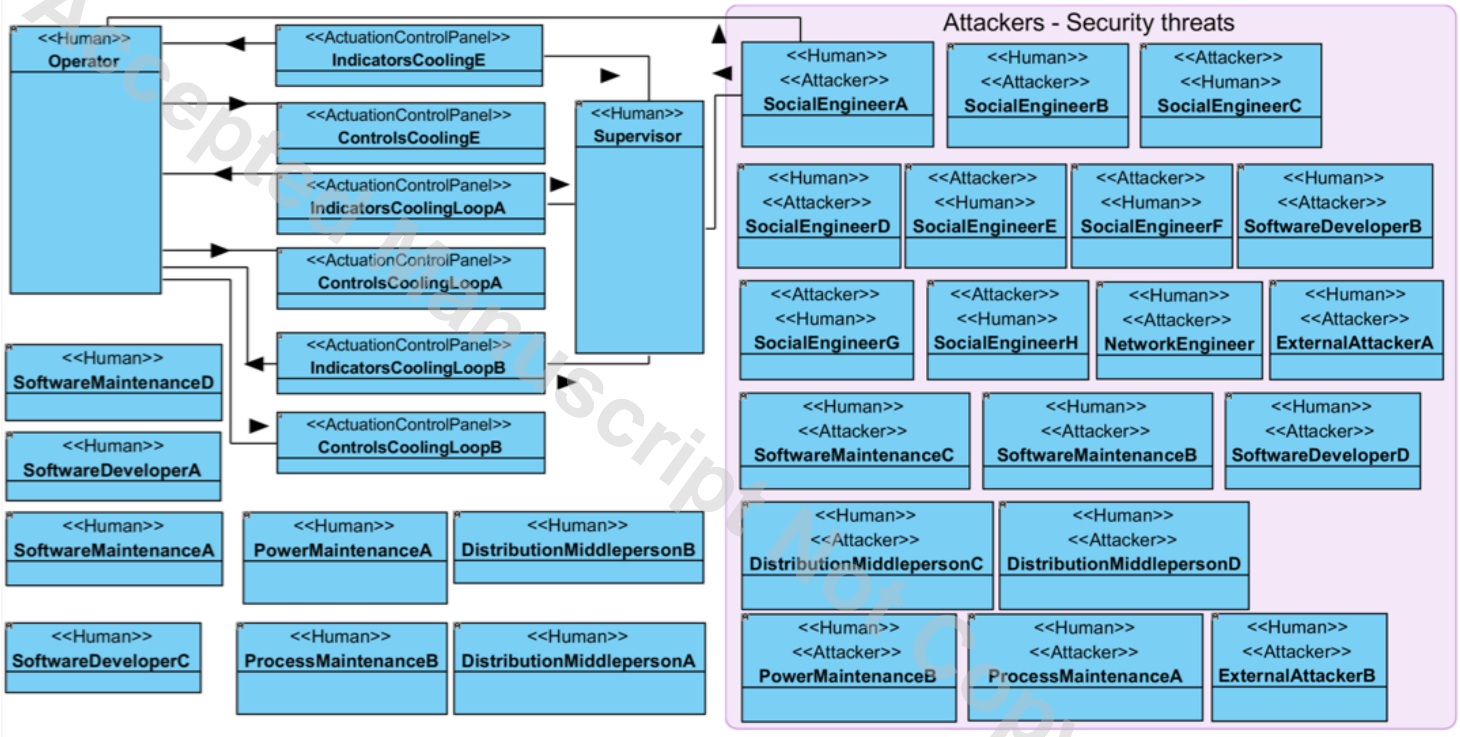


Fig. 7. THE HUMAN FACTORS MODEL OF THE SPENT FUEL POOL COOLING CASE STUDY INCLUDING HUMANS PART OF THE SYSTEMS BUT ALSO HUMANS EXTERNAL TO THE SYSTEM [5].

Journal of Computing and Information Science in Engineering. Received September 30, 2020;
Accepted manuscript posted March 15, 2021. doi:10.1115/1.4050685

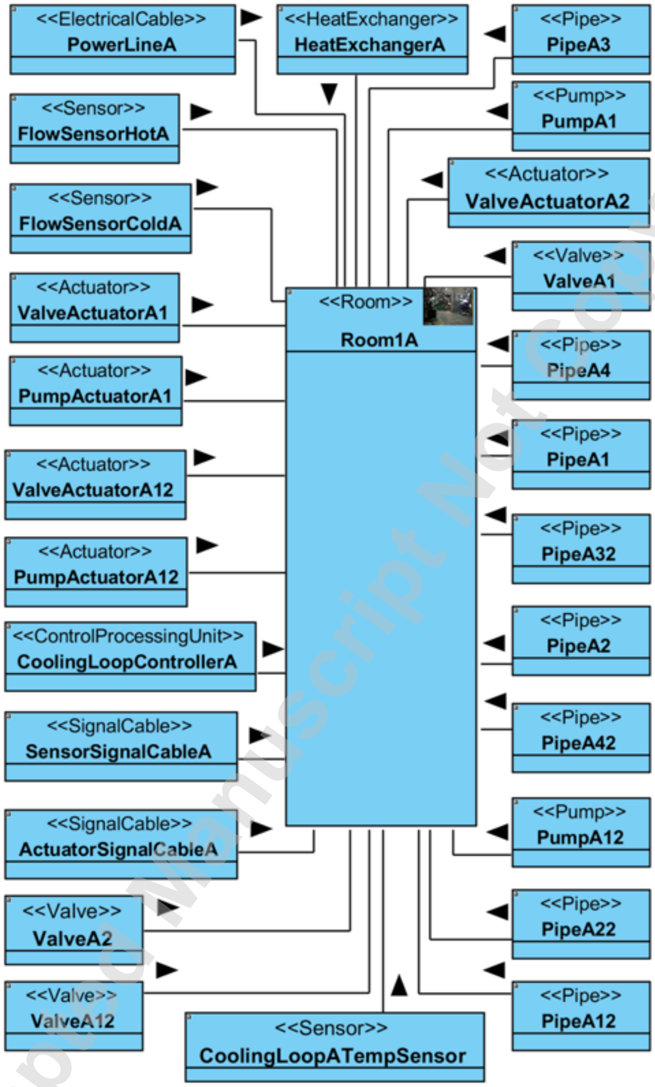


Fig. 8. DEPENDENCIES BETWEEN THE ENVIRONMENT AND SYSTEM COMPONENTS

Journal of Computing and Information Science in Engineering. Received September 30, 2020;
Accepted manuscript posted March 15, 2021. doi:10.1115/1.4050685

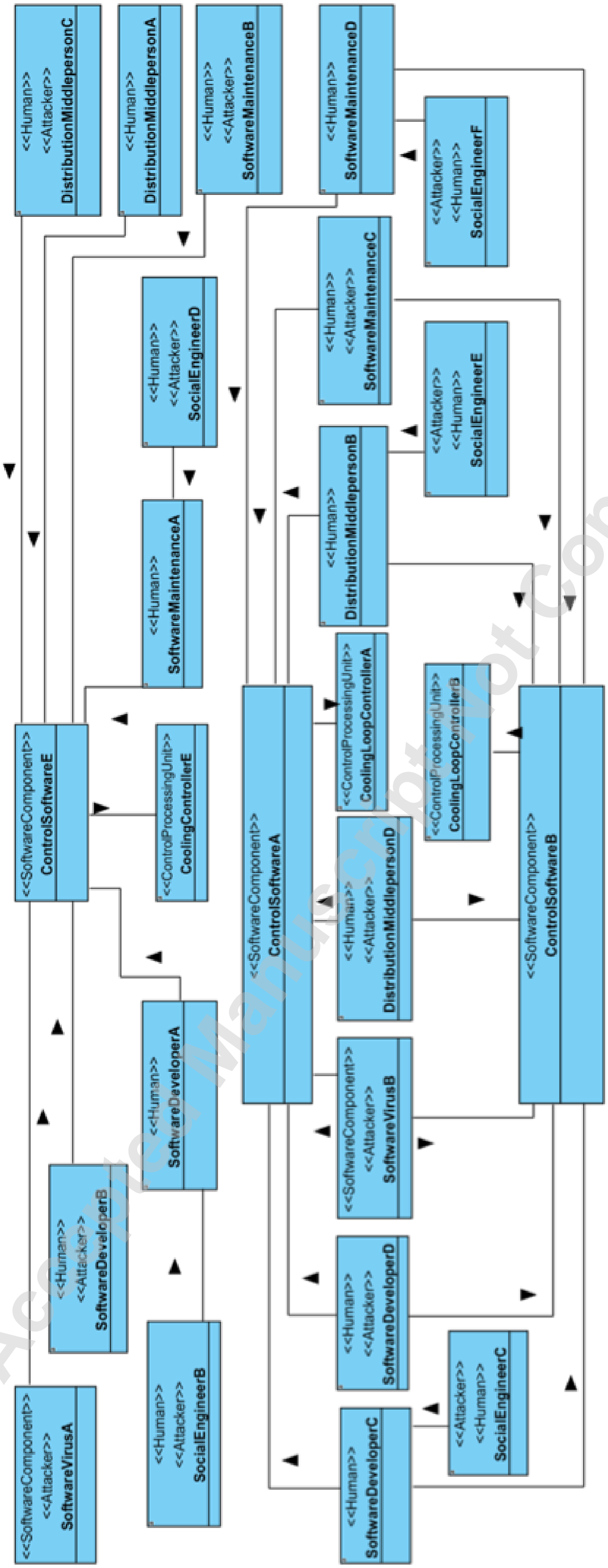


Fig. 9. SIMPLE AUTOMATION MODEL OF THE SPENT FUEL POOL COOLING CASE STUDY INCLUDING RELATIONS TO HUMANS (E.G. SOFTWARE DEVELOPERS AND MIDDLE PERSONS) [5].

Journal of Computing and Information Science in Engineering. Received September 30, 2020;
Accepted manuscript posted March 15, 2021. doi:10.1115/1.4050685

The risk assessment can start by selecting a set of initiating events that are considered as the design basis for the safety of the plant (i.e. the plant's design should be able to handle the initiating events in a way that minimizes undesirable consequences). For every initiating event (e.g. a loss of coolant accident in the spent fuel pool cooling system), an event tree is developed that captures the different consequences depending on the (or lack of) activation of the safety functions of the plant (see Figure 10). A fault tree is compiled, based on the system design, that captures the overall probability of losing a specific safety function. In a traditional safety assessment, this fault tree has reliability-related basic events with the failure probabilities as leaves. The MEDRAF methodology proposed in this article calls for the development of a hybrid (safety and security) tree with the inclusion of safety and security basic events (attacks to specific system components).

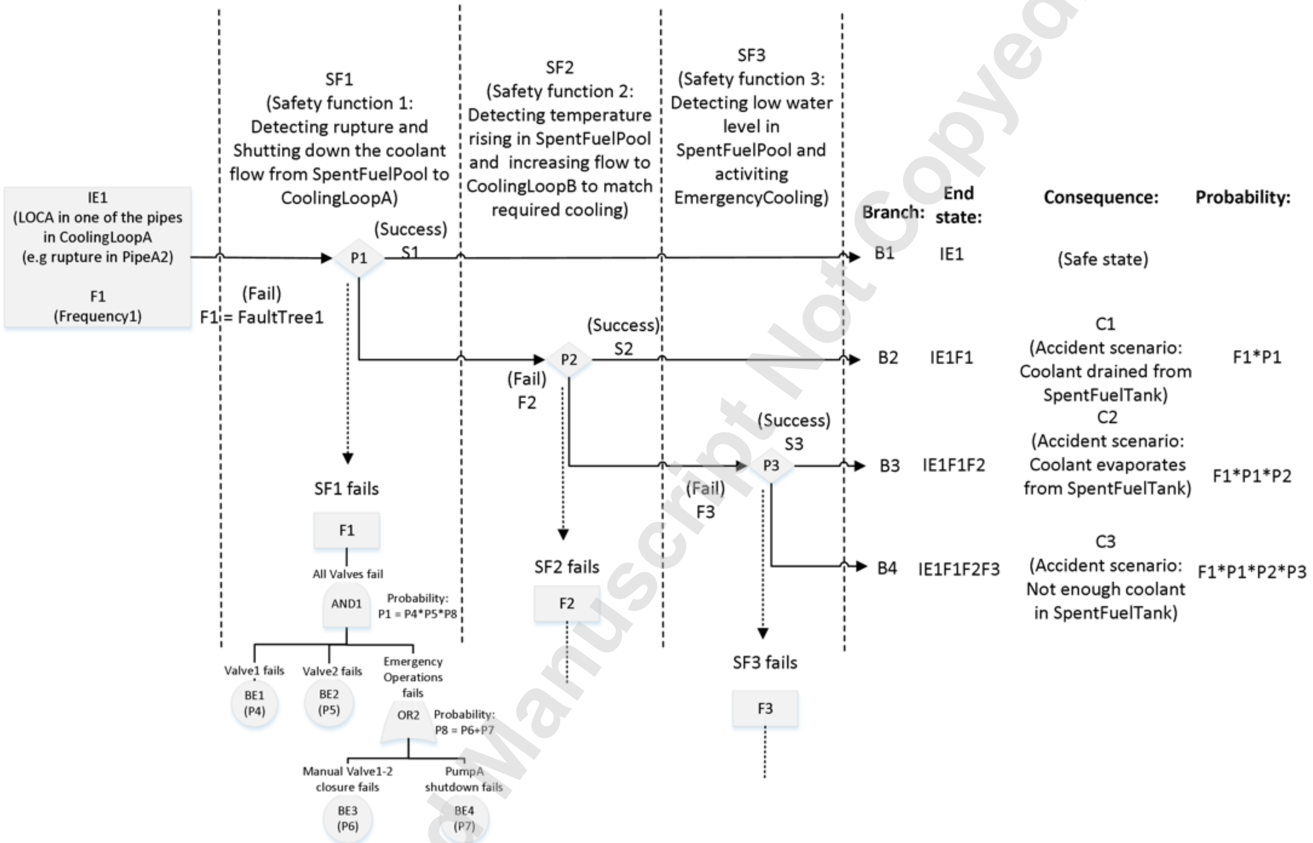

IE1
(LOCA in one of the pipes in CoolingLoopA (e.g rupture in PipeA2)
F1
(Frequency1)
SF1
(Safety function 1: Detecting rupture and Shutting down the coolant flow from SpentFuelPool to CoolingLoopA)
SF2
(Safety function 2: Detecting temperature rising in SpentFuelPool and increasing flow to CoolingLoopB to match required cooling)
SF3
(Safety function 3: Detecting low water level in SpentFuelPool and activiting EmergencyCooling)
Branch:
End state:
Consequence:
Probability:
P1
(Success)
S1
(Fail)
F1 = FaultTree1
B1
IE1
(Safe state)
P2
(Success)
S2
(Fail)
F2
B2
IE1F1
C1
(Accident scenario: Coolant drained from SpentFuelTank)
F1*P1
P3
(Success)
S3
(Fail)
F3
B3
IE1F1F2
C2
(Accident scenario: Coolant evaporates from SpentFuelTank)
F1*P1*P2
B4
IE1F1F2F3
C3
(Accident scenario: Not enough coolant in SpentFuelTank)
F1*P1*P2*P3
SF1 fails
F1
All Valves fail
AND1
Probability:
P1 = P4*P5*P8
Valve1 fails
BE1
(P4)
Valve2 fails
BE2
(P5)
Emergency Operations fails
OR2
Probability:
P8 = P6+P7
Manual Valve1-2 closure fails
BE3
(P6)
PumpA shutdown fails
BE4
(P7)
SF2 fails
F2
SF3 fails
F3

Fig. 10. EVENT TREE FOR A SPECIFIC INITIATING EVENT PART OF THE DESIGN BASIS OF THE PROCESS. IT MODELS THE POTENTIAL CONSEQUENCES GIVEN THE ACTIVATION OF THE AVAILABLE MITIGATION FUNCTIONS. FAULT TREES ARE USED TO ESTIMATE THE PROBABILITY FOR SUCCESSFUL (OR NOT) ACTIVATION OF THE SAFETY FUNCTIONS.

The loss of the control software of the emergency water supply system is selected as an example from the case study. Figure 11 presents a hybrid tree that was developed in the SAPHIRE PRA tool. As in the simple example in the methodology section, overall risk calculations are made for two versions of the tree, one with just reliability basic events (fault tree) and one with reliability and attack basic events (hybrid tree). A fixed probability of 10^{-3} is used for both the safety and security events. In real cases, these probabilities need to vary according to the domain and the specific human attacker under study. The overall risk assessment when only safety is considered gives an overall risk of $5 \cdot 10^{-3}$, with main contributions from

Journal of Computing and Information Science in Engineering. Received September 30, 2020;
Accepted manuscript posted March 15, 2021. doi:10.1115/1.4050685

internal bugs in the software, the software being accidentally damaged by the developer/distributor/maintenance person, or the software being broken by errors in the control automation hardware. The overall calculation when attacks are added to the assessment gives an overall risk of $1.1 \cdot 10^{-2}$ (more than two times higher than the safety-only calculation). The main additional risks come from attacks by the humans that are part of, or external to, the system with either physical or cyber/remote access to the control software.

Journal of Computing and Information Science in Engineering. Received September 30, 2020;
Accepted manuscript posted March 15, 2021. doi:10.1115/1.4050685
Copyright © 2021 by ASME

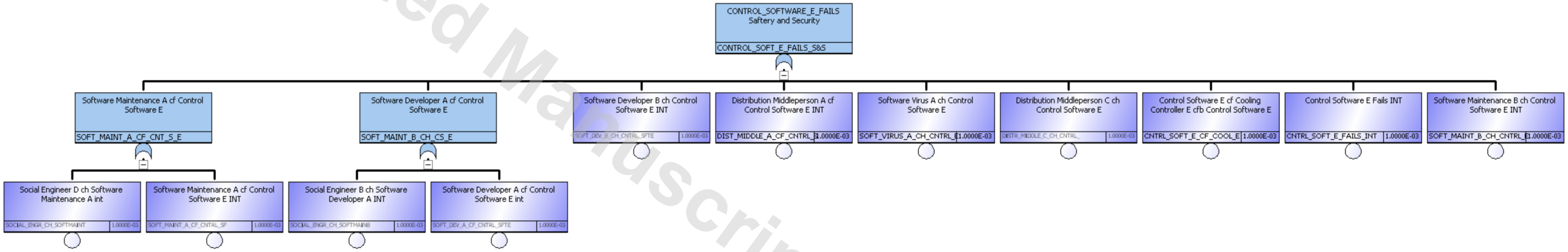

CONTROL_SOFTWARE_E_FAILS
Saftery and Security
CONTROL_SOFT_E_FAILS_S&S
Software Maintenance A cf Control Software E
SOFT_MAINT_A_CF_CNT_S_E
Software Developer A cf Control Software E
SOFT_MAINT_B_CH_CS_E
Software Developer B ch Control Software E INT
SOFT_DEV_B_CH_CNTRL_SFTE
1.0000E-03
Distribution Middleperson A cf Control Software E INT
DIST_MIDDLE_A_CF_CNTRL_E
1.0000E-03
Software Virus A ch Control Software E
SOFT_VIRUS_A_CH_CNTRL_E
1.0000E-03
Distribution Middleperson C ch Control Software E
DISTR_MIDDLE_C_CH_CNTRL_
1.0000E-03
Control Software E cf Cooling Controller E cfb Control Software E
CNTRL_SOFT_E_CF_COOL_E
1.0000E-03
Control Software E Fails INT
CNTRL_SOFT_E_FAILS_INT
1.0000E-03
Software Maintenance B ch Control Software E INT
SOFT_MAINT_B_CH_CNTRL_E
1.0000E-03
Social Engineer D ch Software Maintenance A int
SOCIAL_ENGR_CH_SOFTMAINT
1.0000E-03
Software Maintenance A cf Control Software E INT
SOFT_MAINT_A_CF_CNTRL_SF
1.0000E-03
Social Engineer B ch Software Developer A INT
SOCIAL_ENGR_CH_SOFTMAINB
1.0000E-03
Software Developer A cf Control Software E int
SOFT_DEV_A_CF_CNTRL_SFTE
1.0000E-03

Fig. 11. PART OF THE FAULT/ATTACK TREE OF THE CASE STUDY IN THE SAPHIRE PRA TOOL.

In the case of the control software failure (Figure 9), the dependency model shows that there are many people across the system lifecycle (e.g. design, deployment, operation, maintenance phases) that can have access to the control software. These persons can be divided into those who have direct access due through their role to the system and, for some reason, are motivated to attack the plant (e.g. radicalization, revenge, etc.), and those who are outside the system and try to break in. In cases of collusion between outside attackers and inside attackers, we consider internal attackers to cover both those who are strictly internal and a combination of external and internal. This comes from the fact that a solo internal attacker can also launch attacks from outside, while an external attacker can be assumed to have no internal capabilities. Collusion can be achieved through willful collusion, deception (e.g. fooling the insider into providing information or connecting an unauthorized device to a secured network), benefits (monetary bribes or other incentives), and fear/extortion (e.g. threatening their lives or the lives of their loved ones).

5 Discussion

The MEDRAF methodology presented in this article proposes to integrate safety and security risk analysis from a ZT perspective during the early conceptual design phase of a systems engineering project where a cyber-physical (a system with both hardware and software components) is being developed. Implementing the MEDRAF methodology in a MBSE-driven system design process allows for real-time safety/security risk assessment as design decisions are being made. For instance, a designer choosing to connect both a primary and secondary controller to the same power source could be detected and flagged as a risk that needs to be addressed at the time of the decision versus much later during a design review or even after a system has been fielded. The MEDRAF methodology also provides the possibility of continuously assessing safety/security throughout the system life cycle as new threats emerge and zero-day exploits are uncovered, and as new personnel are brought in to operate and maintain the system among other potential uses.

We assert that a ZT paradigm is appropriate to use in developing modern systems because of the potential for bad actors, state actors, industrial espionage, and other personnel compromises throughout the systems engineering process. Such attacks can come from both internal and external sources, and may occur within the design itself both in hardware and software development or externally via cyber connections. The MEDRAF methodology presented here may help in adopting the ZT paradigm more quickly in risk analysis processes.

While we advocate the MEDRAF ZT hybrid safety/security risk assessment methodology presented above, the methodology is intended to be used as part of a suite of risk assessment analysis methods. For instance, where appropriate, FMECA should still be performed and other efforts such as identifying and mitigating spurious signals into a system should be undertaken [58]. The methodology is meant to augment – not replace – existing analysis methods.

The sensitivity of the results of the MEDRAF methodology to changes in probability of specific attacks is highly dependent on the system design. One potential area of further study is investigating how understanding sensitivity to change in probability could be used to redesign a system to be less impacted by changes in probability. Conducting a sensitivity study may also help to identify which probabilities need further refinement to reduce uncertainty.

The MEDRAF method presented above closes a gap between existing methods to conduct security risk assessments and safety risk assessments, and does so assuming ZT. Existing methods conduct security risk assessments and safety risk assessments separately. In practice, this often means that different people in different organizations within a company conduct the two analyses at different times during a system design process. While some information may cross between the two analyses, we have observed in practice that this is not commonly achieved. Further, identifying risks that cross the security and the safety risk analyses is not common using existing methods. For instance, the fault tree shown in Figure 4 would be developed as two separate fault trees that would not be integrated during separate analyses. Missing potential security and safety risks due to not combining the analyses into the MEDRAF method presented in this paper may open a system to significant system failure events in operation. Identifying such risks much later in the design process after separate analyses are manually compared without the benefit of the proposed method may cause significant cost and schedule overruns as a result of needing either to redesign a portion of the system or develop a remediation subsystem to deal with the potential risk. As discussed in the case study, a simple example of combining the safety and security analyses together in the MEDRAF methodology more than doubled the risk calculation result. Thus, we assert that the proposed method is likely to improve identification of potential security and safety risks, and especially hybrid security and safety risks over existing methods.

6 Conclusions and Future Research

This article presented the MEDRAF methodology for a combined safety and security assessment based on early interdisciplinary models of the system. The foundational argument in favor of a combined safety and security assessment is that security events can have safety consequences and the end goal is a system that is more resilient to both safety and security root causes. The concurrent development and assessment of safety and security can also potentially reveal trade-offs and initiate a discussion among the disciplines in a design process. The MEDRAF methodology is specifically used to for estimating the probability of successful attacks to system components. This is a first step toward improving security and safety assessment; the topic of accurately calculating security risks is still very active and broad. The ZT principle used in the MEDRAF methodology means that no human can be trusted in the same way no component is considered perfectly reliable. The MEDRAF methodology is demonstrated on a spent fuel pool cooling system case study and shows that the overall risk calculation is heavily impacted when security-related basic events are also considered.

Quantifying security risks is complicated and requires current and accurate intelligence (i.e. information) about the current security world and local threat climate, and domain knowledge. The fact that security risks are changing over time needs to be taken into consideration, and a security assessment will need to be updated regularly to include new possibilities for attack.

Acknowledgements

This research is partially supported by the VTT Technical Research Centre and the Naval Postgraduate School. Any opinions or findings of this work are the responsibility of the authors, and do not necessarily reflect the views of the sponsors or collaborators. The case study presented in this publication, while inspired by real systems, is intentionally fictional and idealized in nature.

Journal of Computing and Information Science in Engineering. Received September 30, 2020;
Accepted manuscript posted March 15, 2021. doi:10.1115/1.4050685

References

[1] McDermott, T. A., Hutchison, N., Clifford, m., Van Aken, E., Salado, A., and Henderson, K., 2020. Benchmarking the benefits and current maturity of model-based systems engineering across the enterprise. results of the mbse maturity survey, part 1: Executive summary. Technical Report SERC-2020-SR-001, Systems Engineering Research Center.

[2] Yang, K., and EI-Haik, B. S., 2003. *Design for Six Sigma*. McGraw-Hill, New York City, May 21, 2003.

[3] Clausing, D., and Frey, D. D., 2005. "Improving system reliability by failure-mode avoidance including four concept design strategies". *Systems engineering,* **8**(3), pp. 245–261.

[4] Papakonstantinou, N., Van Bossuyt, D. L., Linnosmaa, J., Hale, B., and O'Halloran, B., 2020. "Towards a zero trust hybrid security and safety risk analysis method". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers.

[5] Papakonstantinou, N., Linnosmaa, J., Alanen, J., Bashir, A. Z., O'Halloran, B., and Van Bossuyt, D. L., 2019. "Early hybrid safety and security risk assessment based on interdisciplinary dependency models". In 2019 Annual Reliability and Maintainability Symposium (RAMS), IEEE, pp. 1–7.

[6] O'Halloran, B. M., Papakonstantinou, N., and Van Bossuyt, D. L., 2018. "Assessing the consequence of cyber and physical malicious attacks in complex, cyber-physical systems during early system design". In 2018 IEEE 16th International Conference on Industrial Informatics (INDIN), IEEE, pp. 733–740.

[7] 2015, I. ., 2015. Systems and software engineering–system life cycle processes. Tech. rep.

[8] Galante, E., Bordalo, D., and Nobrega, M., 2014. "Risk assessment methodology: quantitative hazop". *Journal of Safety Engineering,* **3**(2), pp. 31–36.

[9] Henley, E. J., and Kumamoto, H., 1996. *Probabilistic risk assessment and management for engineers and scientists*, 2nd edition ed. IEEE Press.

[10] Van Bossuyt, D. L., O'Halloran, B. M., and Arlitt, R. M., 2019. "A method of identifying and analyzing irrational system behavior in a system of systems". *Systems Engineering,* **22**(6), pp. 519–537.

[11] Sierla, S., O'Halloran, B. M., Karhela, T., Papakonstantinou, N., and Tumer, I. Y., 2013. "Common cause failure analysis of cyber–physical systems situated in constructed environments". *Research in Engineering Design,* **24**(4), pp. 375–394.

[12] Nikula, H., Sierla, S., O'Halloran, B., and Karhela, T., 2015. "Capturing deviations from design intent in building simulation models for risk assessment". *Journal of Computing and Information Science in Engineering,* **15**(4), p. 041011.

[13] Wang, J., and Ruxton, T., 1997. "A review of safety analysis methods applied to the design process". *Journal of Engeering Design,* **8**(2), pp. 131–152.

[14] Papakonstantinou, N., Linnosmaa, J., Alanen, J., and O'Halloran, B., 2018. "Automatic fault tree generation from multidisciplinary dependency models for early failure propagation assessment". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 51739, American Society of Mechanical Engineers, p. V01BT02A037.

[15] Papakonstantinou, N., Linnosmaa, J., Bashir, A. Z., Malm, T., and Van Bossuyt, D. L., 2020. "Early combined safety-

security defense in depth assessment of complex systems". In 2020 Annual Reliability and Maintainability Symposium (RAMS), IEEE, pp. 1–7.

[16] Ramos, A. L., Ferreira, J. V., and Barceló, J., 2011. "Model-based systems engineering: An emerging approach for modern systems". *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* **42**(1), pp. 101–111.

[17] Bickford, J., Van Bossuyt, D. L., Beery, P., and Pollman, A., 2020. "Operationalizing digital twins through model-based systems engineering methods". *Systems Engineering*. In Press.

[18] Estefan, J. A., et al., 2007. "Survey of model-based systems engineering (mbse) methodologies". *Incose MBSE Focus Group,* **25**(8), pp. 1–12.

[19] Cameron, B., Crawley, E., and Selva, D., 2016. *Systems Architecture. Strategy and product development for complex systems*. Pearson Education.

[20] Weilkiens, T., Lamm, J. G., Roth, S., and Walker, M., 2015. *Model-based system architecture*. John Wiley & Sons.

[21] Russell, M., 2012. "Using mbse to enhance system design decision making". *Procedia Computer Science,* **8**, pp. 188–193.

[22] Madni, A. M., and Sievers, M., 2018. "Model-based systems engineering: Motivation, current status, and research opportunities". *Systems Engineering,* **21**(3), pp. 172–190.

[23] Ellison, C. M., 2007. "Ceremony design and analysis.". *IACR Cryptol. ePrint Arch.,* **2007**, p. 399.

[24] Bella, G., and Coles-Kemp, L., 2012. "Layered analysis of security ceremonies". In IFIP International Information Security Conference, Springer, pp. 273–286.

[25] Carlos, M. C., Martina, J. E., Price, G., and Custódio, R. F., 2013. "An updated threat model for security ceremonies". In Proceedings of the 28th annual ACM symposium on applied computing, pp. 1836–1843.

[26] Radke, K., Boyd, C., Nieto, J. G., and Brereton, M., 2011. "Ceremony analysis: Strengths and weaknesses". In IFIP International Information Security Conference, Springer, pp. 104–115.

[27] Dowling, B., and Hale, B., 2020. "There can be no compromise: The necessity of ratcheted authentication in secure messaging". *IACR Cryptol. ePrint Arch.,* **2020**, p. 541.

[28] Hooper, E., 2009. "Intelligent strategies for secure complex systems integration and design, effective risk management and privacy". In 2009 3rd Annual IEEE Systems Conference, IEEE, pp. 257–261.

[29] Paté-Cornell, M.-E., Kuypers, M., Smith, M., and Keller, P., 2018. "Cyber risk management for critical infrastructure: a risk analysis model and three case studies". *Risk Analysis,* **38**(2), pp. 226–241.

[30] Abdo, H., Kaouk, M., Flaus, J.-M., and Masse, F., 2018. "A safety/security risk analysis approach of industrial control systems: A cyber bowtie–combining new version of attack tree with bowtie analysis". *Computers & Security,* **72**, pp. 175–195.

[31] Shostack, A., and Stewart, A., 2008. *The new school of information security*. Pearson Education.

[32] Goldstein, P., 2019. "Do zero-trust security frameworks provide top network security?". *FedTech Magazine*, Aug.

[33] Team, I. Z. T. P., 2019. Zero trust cybersecurity, current trends. Tech. rep., American Council for Technology, April.

Journal of Computing and Information Science in Engineering. Received September 30, 2020;
Accepted manuscript posted March 15, 2021. doi:10.1115/1.4050685

[34] Rose, S., Borchert, O., Mitchell, S., and Connelly, S., 2020. Zero trust architecture. NIST Special Publication SP-800, National Institute of Standards.

[35] Tao, Y., Lei, Z., and Ruxiang, P., 2018. "Fine-grained big data security method based on zero trust model". In 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS), IEEE, pp. 1040–1045.

[36] Samaniego, M., and Deters, R., 2018. "Zero-trust hierarchical management in iot". In 2018 IEEE International Congress on Internet of Things (ICIOT), IEEE, pp. 88–95.

[37] Scott, B., 2018. "How a zero trust approach can help to secure your aws environment". *Network Security,* **2018**(3), pp. 5–8.

[38] Buldas, A., Gadyatskaya, O., Lenin, A., Mauw, S., and Trujillo-Rasua, R., 2020. "Attribute evaluation on attack trees with incomplete information". *Computers & Security,* **88**, p. 101630.

[39] Kordy, B., Pouly, M., and Schweitzer, P., 2014. "A probabilistic framework for security scenarios with dependent actions". In International Conference on Integrated Formal Methods, Springer, pp. 256–271.

[40] Schultz, E. E., 2002. "A framework for understanding and predicting insider attacks". *Computers & Security,* **21**(6), pp. 526–531.

[41] Mell, P., Scarfone, K., and Romanosky, S., 2006. "Common vulnerability scoring system". *IEEE Security & Privacy,* **4**(6), pp. 85–89.

[42] Le, N. T., and Hoang, D. B., 2018. "Security threat probability computation using markov chain and common vulnerability scoring system". In 2018 28th International Telecommunication Networks and Applications Conference (ITNAC), IEEE, pp. 1–6.

[43] Gao, N., He, Y., and Ling, B., 2018. "Exploring attack graphs for security risk assessment: a probabilistic approach". *Wuhan University Journal of Natural Sciences,* **23**(2), pp. 171–177.

[44] Whitman, M. E., 2003. "Enemy at the gate: threats to information security". *Communications of the ACM,* **46**(8), pp. 91–95.

[45] Anthony, M., Ishmael, M., Santa, E., Shemyakin, A., Stanull, G., and Vandeweghe, N., 2016. "Estimating probability of a cybersecurity breach". *Risk Management*, 12.

[46] Smith, M. D., and Paté-Cornell, M. E., 2018. "Cyber risk analysis for a smart grid: how smart is smart enough? a multiarmed bandit approach to cyber security investment". *IEEE Transactions on Engineering Management,* **65**(3), pp. 434–447.

[47] Sommestad, T., Ekstedt, M., and Nordstrom, L., 2009. "Modeling security of power communication systems using defense graphs and influence diagrams". *IEEE Transactions on Power Delivery,* **24**(4), pp. 1801–1808.

[48] Liu, N., Zhang, J., Zhang, H., and Liu, W., 2010. "Security assessment for communication networks of power control systems using attack graph and mcdm". *IEEE Transactions on Power Delivery,* **25**(3), pp. 1492–1500.

[49] Hahn, A., and Govindarasu, M., 2011. "Cyber attack exposure evaluation framework for the smart grid". *IEEE Transactions on Smart Grid,* **2**(4), pp. 835–843.

[50] Rao, N. S., Poole, S. W., Ma, C. Y., He, F., Zhuang, J., and Yau, D. K., 2016. "Defense of cyber infrastructures against

Journal of Computing and Information Science in Engineering. Received September 30, 2020;
Accepted manuscript posted March 15, 2021. doi:10.1115/1.4050685

cyber-physical attacks using game-theoretic models". *Risk Analysis,* **36**(4), pp. 694–710.

[51] Rao, N. S., Ma, C. Y., Shah, U., Zhuang, J., He, F., and Yau, D. K., 2015. "On resilience of cyber-physical infrastructures using discrete product-form games". In 2015 18th International Conference on Information Fusion (Fusion), IEEE, pp. 1451–1458.

[52] Alai, S. P., 2019. "Evaluating arcadia/capella vs. oosem/sysml for system architecture development". PhD thesis, Purdue University Graduate School.

[53] , 2021. Eclipse papyrus modeling environment. `https://www.eclipse.org/papyrus/`. Accessed: 2021-02-04.

[54] VTT Technical Research Centre of Finland Ltd. , 2021. Finpsa tool for probabilistic risk assessment. `https://simulationstore.com/finpsa`. Accessed: 2021-02-06.

[55] Smith, C., and Wood, S., 2011. Systems analysis programs for hands-on integrated reliability evaluations (saphire) version 8: User's guide. Tech. Rep. NUREG/CR-7039, Department of Energy.

[56] , 2021. Cameo systems modeler. `https://www.nomagic.com/products/cameo-systems-modeler`. Accessed: 2021-02-04.

[57] Maier, M. W., 2009. *The art of systems architecting*. CRC press.

[58] Van Bossuyt, D. L., and Arlitt, R. M., 2020. "A Functional Failure Analysis Method of Identifying and Mitigating Spurious System Emissions From a System of Interest in a System of Systems". *Journal of Computing and Information Science in Engineering,* **20**(5), 05. 054501.