Impact Assessment of IT Security Breaches in Cyber-Physical Systems

Short paper

András Földvári*, Gergely Biczók†, Imre Kocsis‡, László Gönczy‡, András Pataricza‡
Budapest University of Technonolgy and Economics
*andras.foldvari@edu.bme.hu
†biczok@crysys.hu
‡{kocsis.imre, gonczy.laszlo, andras.pataricza}@vik.bme.hu

***Abstract*—The increased cyber-attack surface in Cyber-Physical Systems, the close coupling to vulnerable physical processes, and the potential for human casualties necessitate a careful extension of traditional safety methodologies, e.g., error propagation analysis (EPA), with cybersecurity capabilities.**

We propose a model-driven information technology-operation technology impact analysis method that supports identifying vulnerabilities, most critical attack strategies, and most dangerous threat actors by analyzing attack scenarios on an abstract functional model of the system. Our solution extends EPA, initially developed for dependability and safety analysis, with cybersecurity aspects to explore the safety impact of a cyberattack on a cyber-physical system.

The paper presents the impact analysis workflow, threat modeling, the pilot analysis tool, and a case study.

***Index Terms*—cyber-physical systems, impact analysis, error propagation analysis**

I. Introduction

Operational Technology (OT) has been a key enabler of industrial societies for centuries. In its modern form, OT manages the operation of physical processes and the machinery used to carry them out; OT systems control anything from nuclear power plants to shoe manufacturing. At the basic level, OT refers to technology that monitors and controls specific devices and processes within industrial workflows. OT is unique in that *Industrial Control Systems (ICS)* perform very specific functions: monitor mechanical performance, close a valve or trigger an emergency shutdown. Historically built on the safety-reliability-productivity paradigm, these systems were closed and purpose-built for the given physical process, therefore hard for malicious actors to interfere with their operation.

On the other hand, *Information Technology (IT)* deals with information: it controls the flow of data, and its usage is pervasive in enterprise environments for efficiency reasons. One of the main strengths of IT (as opposed to OT) is its general applicability to very different processes. As most IT systems are by definition interconnected and open and manage valuable information, system administrators are well-aware of cybersecurity risks. Experts have developed multiple overarching risk management frameworks, best practices, and technical security measures based on the confidentiality-integrity-availability paradigm to deal with the ever-increasing number of security threats.

In the last decade, OT and IT have converged significantly, mainly by the proliferation of Industrial Internet of Things (IIoT) devices, which generate troves of data and enable more efficient control on the factory floor. Essentially, IIoT turns factories into Cyber-Physical Systems (CPS) [1], which are highly interconnected and mutually dependent. The *increased attack surface*, its close coupling to vulnerable physical processes, and the potential for human casualties necessitate that traditional safety methodology, e.g., Error Propagation Analysis (EPA) [2]–[4], is carefully extended with cybersecurity capabilities [5]. There are many publicized incidents demonstrating this necessity, including the 2010 Stuxnet attack on Iran's uranium enrichment centrifuges [6], and the 2014 cyberattack on a German steel mill that resulted in significant damage to blast furnaces [7].

To integrate the EPA and the cybersecurity aspect, we propose a *model-driven impact analysis method* that supports identifying vulnerabilities, most critical attack strategies, and most dangerous threat actors.

The impact analysis uses *qualitative reasoning* [8]–[10] which is well understandable and interpretable by an expert as it corresponds to the typical engineering thinking in which knowing the exact values of variables during analysis is less necessary for understanding the phenomena. The core idea of qualitative reasoning is the aggregation of continuous variable values into ordinal variables that describe *operation domains* in which the data points describe similar behaviors. For instance, a qualitative model describes that the values of the utilization of a particular resource $\{low, medium, high\}$ lead to system states $\{underload, normal, overload\}$.

Our method uses qualitative EPA as the approach for impact analysis. In EPA, the effect of a core incident is propagated through qualitative causal relations between the components.

This research was funded by the PrOTectME (Protecting Operational Technologies of Medium Enterprises) (EIT Digital 21293) project.

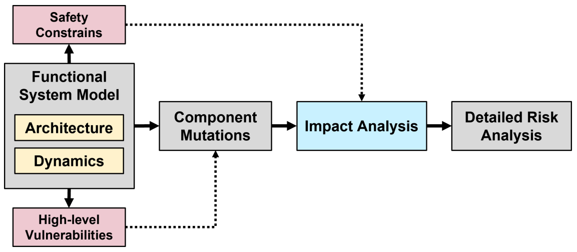

Fig. 1. Impact analysis workflow

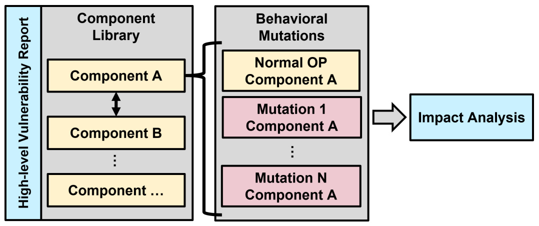

Fig. 2. Component mutations

Section II introduces the impact analysis workflow by highlighting its main steps. Section III describes high-level threat modeling, which is the basis of the component mutations in the impact analysis tool. Section IV presents the impact analysis tool, its features, and the properties of the core engine. Section V shows a case study that covers the main functions of the tool. Section VI concludes the paper and presents the future directions of the research and its applications.

II. Impact Analysis Workflow

The impact analysis tool supports the *detailed risk analysis* by analyzing attack scenarios on an abstract functional model of the system. The basis of the impact analysis is the *high-level functional model* of the system that describes the system functions and the interactions of the components. This model does not cover any implementation-specific details about the system (e.g., the type of the micro-controllers or other software-specific details).

The *dynamic description* of the system describes the internal behavior and the interaction between the components. In the qualitative model, these interactions correspond to causal qualitative relations. Initially, the dynamics of the system describe the case where every component operates as intended. The *attack scenarios* are described by using high-level *cybersecurity frameworks* (e.g., ATT&CK for ICS). The high-level vulnerabilities can be mapped to the components in the functional system model.

Based on the high-level vulnerabilities, the *faulty mutations* of the component behaviors can be modeled and added to the dynamic description of the system. These mutations describe the behavior of the components in case of a successful attack. The impact analysis requires the joint handling of the *functional architecture*, the *dynamic behavior*, and the *faulty mutations* of the components.

The results of the impact analysis were carried out by using error propagation analysis. The outcome of the EPA covers the consequences of the attack scenarios where the *safety constraints* were violated.

A. Behavioral mutations

The different mutations of the components (Fig. 2) are based on the explored attack scenarios. These mutations describe the compromised component behaviors (*fault modes*). The analysis tool selects the different mutated (or original) component *combinations* and runs the analysis on these selected combinations. This way, the analysis tool can select multiple compromised components like in a real attack scenario where the attacker targets many components.

The combinations can be restricted by using *constraints*. For example, if there is no such a combination that Component A with Mutation 1 occurs together with Component B with Mutation 2, the tool can ignore these combinations.

III. Threat modelling

In order to define and inspect attack scenarios (and later, do a proper risk analysis) in any complex IT-enabled system, threat modeling has to be performed. To this end, the cybersecurity community has devised multiple frameworks which, essentially, define dictionaries of *potential attack types* at different levels of detail. One of the best known high-level frameworks is *ATT&CK* [11], maintained by MITRE. ATT&CK is a frequently updated knowledge base of adversary tactics and techniques based on real-world observations. The ATT&CK knowledge base can be used as a foundation for the development of specific threat models and methodologies in different technological sectors. The essence of the framework is a so-called *attack matrix* that lists tactics, corresponding techniques, and sub-techniques; these are extremely useful for the threat modeling of enterprise systems.

Fortunately, the strong convergence of IT and OT has brought to life a specialized version of the framework referred to as ATT&CK for ICS [12], tailored to the needs of industrial control systems. Its attack matrix has a narrower, *ICS-specific scope*; defined at a conceptual level, this matrix is an excellent starting point to introduce the high-level cybersecurity vulnerabilities into our impact analysis workflow. It covers tactics for gaining access to physical processes (e.g., privilege escalation, lateral movement, etc.) and for tampering with them to inflict loss of different types (e.g., inhibit response function, impair process control, etc.). As the steps for gaining access are similar to general IT systems, in our simple case study, we focus on the tactics (and corresponding techniques) related to controlling the physical processes.

In spite of all its strengths, ATT&CK is not an exhaustive enumeration of attack vectors against specific hardware and software platforms. Such a detailed list of potential, *technology-dependent attack patterns, and weaknesses* is, of course, needed to perform a detailed risk analysis in a real, instantiated operating environment (see the last phase in Fig. 1). Luckily, other MITRE efforts such as *CAPEC* (Common Attack Pattern Enumeration and Classification [13]) and CWE

(Common Weakness Enumeration [14]) are available and suitable to support this endeavor. CAPEC provides a comprehensive dictionary of known patterns of attack employed by adversaries to exploit known weaknesses in cyber-enabled capabilities. Complementarily, CWE lists known software and hardware weakness types.

IV. Impact Analysis Tool

The *impact analysis tool* extracts *error propagation paths* from *initial situations*. The analysis indicates the *relevant components* and *constraint violations* for each error path.

The input model can be extended with metrics like *cost* and *severity*. This way, the analysis can indicate the severity of the attacks, the cost of each scenario, and highlight the *most dangerous hazard* (e.g., cost-severity ratio). *Countermeasures* can be dynamically added to the system to check whether they can mitigate the consequences of an attack.

The *initial situation* can be interpreted as the actual state description of the model. It can cover a specific event or cover multiple threats (e.g., blocking the safety valve controller). *Error propagation paths* cover hazardous situations. Impact analysis focuses on the dysfunctional effects meaning the consequences of the initial situation (e.g., a physical overflow in a system). The *relevance aspect* of the EPA model connects the result of the analysis to relevant components in the original model (e.g., valve controller). The *violation aspect* connects the model to constraints and requirements that the situation or the causal consequences violate (e.g., there should be no overflow in the system).

A. EPA Core Engine

The impact analysis tool uses *Answer Set Programming* (ASP) [15]–[17] as the EPA core engine. In general, ASP is a Prolog-like knowledge representation, knowledge fusion, and logic reasoning framework with optimization mechanisms. ASP can solve *constraint satisfaction problems* and result in one feasible solution corresponding to a particular safety hazard or covering all dangerous situations. The *optimization mechanisms* provide optimized solutions (e.g., the most dangerous hazard).

ASP supports the solution of combinatorial problems with *hard and weak constraints*. Hard constraints correspond to the system's architecture, and they can include other external (environmental) constraints about the system. While hard constraints work as integrity constraints, weak constraints define optimization problems (e.g., risk severity).

The extension of ASP [18] allows the definition of temporal logic programs that handle *linear temporal logic* formulas. With this extension, it is possible to handle the *operational and attack sequences*. The analysis can be extended with external function calls. This way, the extended functionality supports reading the component library and additional properties from external databases and calculating complex metrics.

The EPA engine's *verification and validation* (V&V) power come from the ASP's model checking capabilities. All the risks can be obtained with the EPA engine, and the proof of protection can be validated by checking the solution space against criteria (e.g., safety constraints). Criteria can be formulated by using Linear Temporal Logic (LTL) expressions.

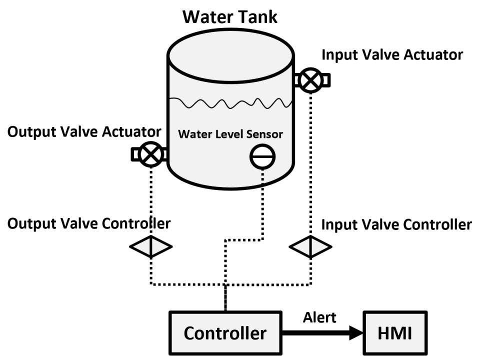

Fig. 3. Water tank architecture

The EPA core engine provides a *compositional modeling* functionality that allows hierarchical modeling and analysis. The tool supports the *hierarchical analysis* of the system by using a *hierarchical component library*. This way, a complex task can be broken down, and the analyst can focus on specific parts of the system.

V. Case study

The case study is presented on a *water tank system* (Fig. 3), elaborated jointly with the experts from ResilTech. The system consists of a main water tank component with input and output valve actuators and their respective controllers. The water tank includes a *water level sensor* that measures the water level in the tank. The water tank *controller* sends control messages to the *valve controllers* based on the measure of the water level sensor. The system also contains a *Human-Machine Interface* (HMI) where the operator can see the status of the system and can react to dangerous events.

The first step of the analysis requires the model of the *intended behavior* of the system. This step includes that if an overflow is detected, the output valve opens, the input valve closes, and the operator receives an alert through the HMI.

The system's *qualitative model* includes the components' operational domains (e.g., for the water level sensor: *empty, normal, full, overloaded*) and the qualitative causal relations between the components. For example, the value correspondence between the water level sensor and the HMI says that if the value of the sensor is overloaded, then the HMI should be in the *alert* qualitative state.

The *safety constraints* for the system are 1) the water tank should not be overflow; 2) alert should be sent to the operator in case of water tank overflow. In the case of a cyberattack, the attacker can create multiple threats in the system. In this example, we consider that the following attacks are carried out simultaneously:

1) Block Command Message (BCM) sent to Output Valve
2) Block Command Message (BCM) sent to Input Valve
3) Spoof Reporting Message (SRM) sent to HMI

TABLE I
IMPACT ANALYSIS RESULTS

Attacks			**Constraints**		**Cost**
BCM Input	**BCM Output**	**SRM HMI**	**Water Overflow**	**Alert to HMI**	
*			-	-	15
	*		Violated	-	10
		*	-	-	15
*	*		Violated	-	20
	*	*	Violated	Violated	25
*		*	-	-	25
*	*	*	Violated	Violated	35

The *attack types* use the terminology from ATT&CK for ICS framework. The dynamic model was extended with the Block Command Message and the Spoof Reporting Message mutation of the corresponding components. The initial dynamic behaviors and the mutations of the components have the same interfaces, but the mutations can describe different qualitative causal relations as initially. The expected result of the simultaneous attack is the following: Output valve remains closed; input valve remains opened; water level value is falsified to the HMI.

The impact analysis engine checks the *violations of the safety constraint* for all the attack scenario combinations. Table I shows the result of the analysis. The first three columns of the table show the selected *combination of the attack scenarios*. The Constrains columns show the *violated safety constraints*. The combination of the three attack scenarios violates both safety constraints. The last columns shows the proportionate cost of the different attack scenarios. However, the *most efficient attack* is when the attacker only blocks the command to the output valve and spoofs the alert message.

The result of the analysis is easily extensible with *countermeasures*. The analysis can check for safety violations whether the countermeasures are active or not. For example, if a safety valve can be activated automatically in case of an overflow, only the ”no alert to operator” constraint will be violated.

VI. CONCLUSIONS

Our solution extends error propagation analysis, initially developed for *dependability and safety analysis*, with *cybersecurity aspects*, to explore the safety impact of a cyberattack on a cyber-physical system. The impact analysis tool highlights critical control paths and reasons for the consequences of security incidents. The *qualitative aspect* of the method helps get a high-level overview of the system's critical operational modes and create and validate countermeasures against safety constraint violations.

One of the key advantages of the underlying algorithmic is its potential to incorporate optimization aspects. A straightforward extension of the models can answer questions like the most severe attack in terms of hazards or the efficient protections for consolidating the system.

Technically, we plan to extend the interface of the impact analysis tool to model components and their interactions easily, and we would also like to create a proper visualization for qualitative EPA. Furthermore, we are integrating our solution with the ResilBlockly tool [1], [19], [20] which provides a compositional framework for risk analysis.

ACKNOWLEDGMENT

We wish to acknowledge the help and fruitful discussion with our project partners: ResilTech, Engineering, CEFRIEL.

We greatly appreciate the expert advice of Prof. Levente Buttyán, providing essential help in covering the security aspects. We would also like to show appreciation to our colleague Benedek Szántó who helped in the pilot implementation of the EPA tool.

REFERENCES

[1] A. Bondavalli, S. Bouchenak, and H. Kopetz, *Cyber-Physical Systems of Systems: Foundations–A Conceptual Model and Some Derivations: the AMADEOS Legacy*. Springer, 2016, vol. 10099.
[2] A. Pataricza, "Systematic generation of dependability cases from functional models," in *Formal Methods for Automation and Safety in Railway and Automotive Systems. Proc. Symposium FORMS/FORMAT, October*, 2007, pp. 9–10.
[3] ——, "Towards open modular critical systems," in *FORMS/FORMAT 2010*. Springer, 2011, pp. 41–42.
[4] I. Kocsis, "Qualitative models in resilience assurance." PhD Thesis BME, 2019.
[5] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security—a survey," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1802–1831, 2017.
[6] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security & Privacy*, vol. 9, no. 3, pp. 49–51, 2011.
[7] R. M. Lee, M. J. Assante, and T. Conway, "German steel mill cyber attack," *Industrial Control Systems*, vol. 30, p. 62, 2014.
[8] K. D. Forbus, "Qualitative process theory," *Artificial intelligence*, vol. 24, no. 1-3, pp. 85–168, 1984.
[9] ——, "Qualitative modeling," *Foundations of Artificial Intelligence*, vol. 3, pp. 361–393, 2008.
[10] ——, "Qualitative modeling," *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 2, no. 4, pp. 374–391, 2011.
[11] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, "MITRE ATT&CK: Design and philosophy," *Technical report*, 2018.
[12] O. Alexander, M. Belisle, and J. Steele, "MITRE ATT&CK® for industrial control systems: Design and philosophy," *The MITRE Corporation: Bedford, MA, USA*, 2020.
[13] "CAPEC: Common Attack Pattern Enumeration and Classification," https://capec.mitre.org.
[14] R. A. Martin, "Common weakness enumeration," *Mitre Corporation*, 2007.
[15] V. Lifschitz, *Answer Set Programming*. Cham: Springer, 2019.
[16] G. Brewka, T. Eiter, and M. Truszczyński, "Answer set programming at a glance," *Comm. of the ACM*, vol. 54, no. 12, pp. 92–103, 2011.
[17] F. Calimeri et al., "Asp-core-2: Input language format," *ASP Standardization Working Group*, 2012.
[18] P. Cabalar, R. Kaminski, P. Morkisch, and T. Schaub, "telingo = ASP + time," in *International Conference on Logic Programming and Nonmonotonic Reasoning*. Springer, 2019, pp. 256–269.
[19] M. Mori, A. Ceccarelli, P. Lollini, B. Frömel, F. Brancati, and A. Bondavalli, "Systems-of-systems modeling using a comprehensive viewpoint-based sysml profile," *Journal of Software: Evolution and Process*, vol. 30, no. 3, p. e1878, 2018.
[20] "ResilBlockly Tool," https://resilblockly.resiltech.com:8502/.