Integrating the Technical Level into a Model-based Safety and Security Analysis: Why it is Necessary and How it Can be Done

Sibylle Fröschle
Institute for Secure Cyber-Physical Systems
Hamburg University of Technology
Hamburg, Germany
sibylle.froeschle(at)tuhh.de

***Abstract*—Today's safety-critical systems are both networked to the environment and highly defined by software. Hence, they have become vulnerable to cyber attacks. On the positive side, the great progress in data-centric methods has led to increasingly sophisticated attack detection systems. These typically work and are evaluated at the dynamical system level, decoupled from the technical level. In this paper, we motivate why it is necessary to integrate the technical level into a model-based safety and security analysis at the dynamical system level, and show how this can be done.**

***Index Terms*—Model-based safety; security analysis.**

I. Introduction

Today's safety-critical systems are both networked to the environment and highly defined by software. Hence, they have become vulnerable to cyber attacks. On the positive side, the great progress in data-centric methods allows for increasingly sophisticated attack detection and mitigation measures such as anomaly detection systems based on machine learning or techniques rooted in the area of FDIR (Fault Detection, Isolation, and Reconfiguration). Such data-centric measures are typically modelled and evaluated at the dynamical system level, decoupled from the technical level. However, it is the latter where attacks are realized and shape what an attacker is capable of doing at the dynamical system level.

In this paper, we motivate why it is necessary to integrate the technical level into a safety and security analysis at the dynamical system level, and show how this can be done. In the remainder of the paper, we proceed as follows. In Section II, we explain our setting and provide the motivation. In Section III, we summarize our approach. We conclude this work in Section IV. Throughout, we focus on attacks that act via the computer network. The paper is based on a position paper presented at SafeComp 2025 [1].

II. Setting and Motivation

We consider attacks with respect to a general feedback control system with a detection unit. As illustrated in Fig. 1, such a system consists of the following components. The *plant* is the physical part of the system that is to be controlled. The physical state of the plant can be measured by *sensors* and controlled via *actuators*. Based on the sensor measurements, the *controller* implements a control law and issues control commands to the actuators. The *detection unit* monitors the time series of sensor values and control values. Based on a detection algorithm, it determines when to raise an alarm, and how to handle it. Our setting builds on that of Giraldo et al. [2].

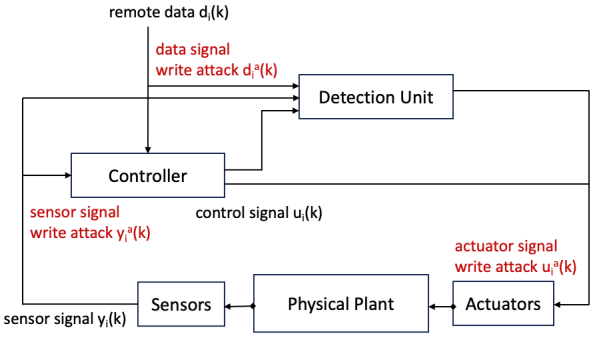
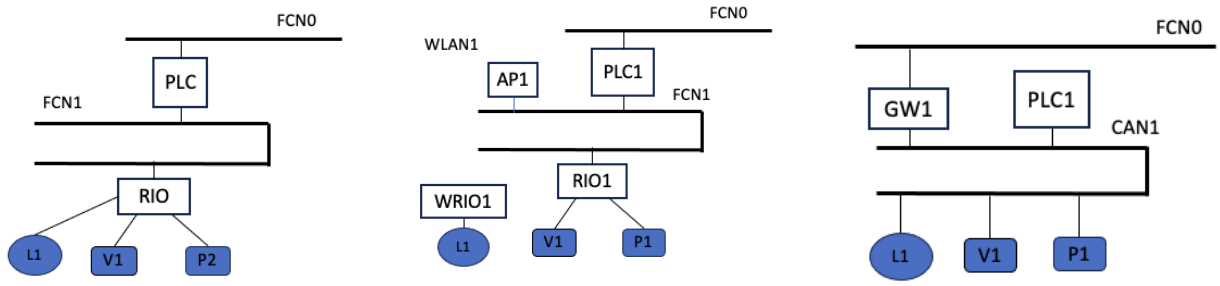


Fig. 1. A general feedback control system.

Example 1. *In a write attack on a sensor signal $y(k)$, the attacker manages to feed a fake sensor signal $y^a(k)$ to the controller. In the worst case, the attacker has full control of the sensor signal, and can deceive the controller about the real state of the plant. Hence, the controller may issue control commands that are inappropriate for the real state, and the attacker may indirectly drive the system into an unsafe state.*

Feedback control systems can be realized by different technical architectures. In Fig. 2, we show three examples. In all of them, the controller and detection unit are both hosted on a Progammable Logic Controller (PLC). The PLC is connected via field network FCN1 (where FCN stands for Field Communications Network) to two actuators, P1 and V1, and one sensor, L1. Remote data may be received via field network FCN0. The first example is close to the first stage of the water treatment system of [2].

Example 2 (Ethernet and Wired PitM)**.** *In TA1 (where TA*



(a) TA 1 (b) TA 2 (c) TA 3

Fig. 2. Three different technical network architectures.

stands for Technical Architecture), the sensor and actuators are linked to the field network via a Remote Input/Output (RIO) module. The field network FCN1 is realized as IEEE 802.3 Ethernet. An attacker with physical access to FCN1 can cut the Ethernet between the RIO and the PLC, and insert their own device. This means they can carry out a Person-in-the-Middle (PitM) attack, and thereby gain complete control over the communication.

Example 3 (WiFi and Wireless PitM)**.** *In TA2, the signal of the sensor (and only of the sensor) is transmitted wirelessly via standard IEEE 802.11 WiFi secured by WiFi Protected Access 2 (WPA2) with Pre-Shared Key (PSK) authentication. To this end, the sensor is connected to a Wireless Remote Input/Output (WRIO) module, which is configured to connect to a WiFi Access Point (AP). Due to a vulnerability on how the keys are derived from the password in WPA2, it is likely that an attacker can brute-force the password by an offline dictionary attack unless it is at least 20 characters long. Then, an attacker without physical access to the system can place themselves as a PitM between the WRIO and the AP and thereby control the sensor signal.*

Example 4 (CAN Bus and CAN Remote Attacks)**.** *TA3 employs Controller Area Network (CAN) as the field network, and all components are directly linked to the CAN. Moreover, the connection to FCN0 is provided via a gateway rather than via the PLC. If an attacker manages to compromise the gateway via FCN0 then, as a direct consequence of the CAN protocol, they can eavesdrop and inject messages. This was made use of in first generation automotive attacks. More detailed investigations have shown that, by abusing CAN error handling and failure confinement, an attacker can go beyond such attacks, and e.g., impersonate nodes without leaving any traces of data frames on the bus [3].*

In Example 2, any detection algorithm can be bypassed since the PitM attacker can send fake signals to the PLC. In Example 3, an attacker who has no physical access to FCN1 can mount sensor attacks but no actuator attacks. Therefore, measures such as physics-based attack detection [2] can prevent that an attacker can manipulate actuation drastically without being discovered. In Example 4, a first generation CAN attacker can use pure injection attacks to perform both sensor and actuator write attacks but with the constraint that the authentic signals cannot be overwritten. Hence, such attacks can easily be detected by anomaly detection algorithms while this is no longer the case for stealthy second generation CAN attacks. The same applies to the Enhanced Remote Attacker Model (ERAM), in which the attacker can also act at the transceiver level [4]. In CAN networks, detection algorithms work best when combined with other security measures. This example also highlights that attacker models and the countermeasures may have to be adapted over time when new attack capabilites are revealed.

III. Approach

Let S be the overall system, and S_C the System under Consideration (SUC). We assume that a dynamical model of S_C is available such as a simulation model (e.g., a Simulink model) or a formal model (e.g., a hybrid automaton model). An attack mode for S_C is given by a specification of which of its signals are under a read and/or write attack, possibly with constraints on how the signals can be manipulated by a write attack.

Central to our approach is that we can model the attack modes into S_C by a generic transformation. The transformation will give rise to the *SUC under attack*, denoted by S_C^A. We have defined the transformation for hybrid automata but this can be done analogously for simulation models. The transformation composes the SUC with an attacker's component, and modifies the SUC itself by some tweaks that ensure that the signals that the attacker can actively interfer with are appropriately fed into the SUC. Moreover, signals that can be read by the attacker can serve the attacker to refine their signal output, e.g., to remain undetected.

Given a potential technical architecture TA_S for the overall system S, we can carry out the safety and security activities for S_C in a systematic and integrated fashion as follows:

(1) Identify all the computer networks and technical attacks relevant for the SUC S_C, and derive the corresponding attack modes. (2) For each identified technical attack A, evaluate and rate the feasibility of A. Explore whether the feasibility can be mitigated by security controls. This can be done by using any

suitable method, e.g., the domain-specific technique of [5] or a style of attack tree analysis [6]. (3) For each identified attack mode for S_C, evaluate the corresponding SUC under attack S_C^A, and rate the safety impact. Explore whether and how the safety impact can be mitigated by attack detection systems and/or other measures. Thereby, elicit new failure and/or fail-safe modes specific to attacks or new causes to existing failure and/or fail-safe modes. (4) Assess the overall risk based on steps 2 and 3. Iterate these steps until risk is mitigated to an acceptable level.

IV. Conclusion and Future Work

We have put forward a new approach that bridges the gap between the dynamical system level and the technical level where attacks actually take place. While we have focused on the analysis of a SUC here, our approach is also geared towards bringing to light information such as new failure and/or fail-safe modes and dependencies between signals (in the sense that they are affected by the same network attacks). Such information is needed as input for the analysis of the overall system, e.g., in terms of a Failure Mode and Effects Analysis (FMEA) or a combined attack and fault tree analysis. In future work, we will extend the approach beyond network attacks: to encompass also attacks via computing platforms and the environment. Moreover, it remains to conduct a larger case study, and explore how the approach scales for real-life systems. For the latter, we intend to develop principles of compositionality.

References

[1] S. Fröschle, "Integrating the technical level into a model-based safety and security analysis: why it's necessary and how it can be done," in *SAFECOMP 2025 Position Paper*, Stockhlom, Sweden, Sep. 2025. [Online]. Available: https://laas.hal.science/hal-05242073

[2] J. Giraldo, D. Urbina, A. Cardenas, J. Valente, M. Faisal, J. Ruths, N. O. Tippenhauer, H. Sandberg, and R. Candell, "A survey of physics-based attack detection in cyber-physical systems," *ACM Comput. Surv.*, vol. 51, no. 4, jul 2018.

[3] S. Fröschle and A. Stühring, "Analyzing the capabilities of the CAN attacker," in *ESORICS 2017*. Springer International, 2017, pp. 464–482.

[4] Z. Tang, K. Serag, S. Zonouz, Z. B. Celik, D. Xu, and R. Beyah, "ERACAN: Defending against an emerging CAN threat model," in *ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 1894–1908. [Online]. Available: https://doi.org/10.1145/3658644.3690267

[5] C. Schmittner, B. Schrammel, and S. König, "Asset driven ISO/SAE 21434 compliant automotive cybersecurity analysis with ThreatGet," in *Systems, Software and Services Process Improvement*. Springer, 2021, pp. 548–563.

[6] S. M. Nicoletti, M. Peppelman, C. Kolb, and M. Stoelinga, "Model-based joint analysis of safety and security: survey and identification of gaps," *Computer Science Review*, vol. 50, p. 100597, 2023.