Data Collection Method for Security Digital Twin on Cyber Physical Systems

Jens Doenhoff$^1$, Yu Tamura$^1$, Daisuke Tsuji$^1$, Tomohiro Shigemoto$^1$

$^1$ Hitachi, Ltd., Research & Development Group, Center for Digital Services
a) jens.doenhoff.oq@hitachi.com

Abstract: Through the progress of Digital Twin technology and subsequent realization of Cyber-Physical Systems (CPS), the threat of cyber attacks to those systems is rising. Due to differing requirements in Operational Technology (OT) systems, existing IT security is insufficient to secure CPS. To overcome these challenges, we are developing a highly modular Security Digital Twin (SDT), targeting a wide variety of OT/IoT use-cases and scenarios. Due to its modular design, a naive implementation of required data collection is highly cost-inefficient; to address this, a data collection abstraction layer for the SDT is proposed, utilizing upper layer data requirements through a unified interface. In a qualitative evaluation, the proposed solution is shown to be cost-effective.

Keywords: Cyber Physical System, Internet of Things, OT security, Digital Twin

Classification: Network system

References

[1] IBM Security, X-Force Threat Intelligence Index 2022, IBM Corporation, Armonk, 2022.
[2] M. Niedermaier, T. Hanka, S. Plaga, A. von Bodisco and D. Merli, “Efficient Passive ICS Device Discovery and Identification by MAC Address Correlation,” Electronic Workshops in Computing, BCS Learning & Development, 2018
[3] Y. Tamura, D. Tsuji, J. Doenhoff, Y. Isobe, T. Shigemoto, N. Kawaguchi and H. Nakakoji, “Security Digital Twin(SDT) for ensuring security and business continuity of Cyber Physical System(CPS),” IPSJ SIG Technical Report, vol. 2022-CSEC-96, no. 30, pp. 1-8, March 2022.
[4] J. Doenhoff, Y. Tamura, D. Tsuji and T. Shigemoto, “Data Collection Method for Security Digital Twin on Cyber Physical Systems,” 2022 IEICE General Conference, B-14-17, March 2022.
[5] M. Dietz, M. Vielberth and G. Fennel, “Integrating Digital Twin Security Simulations in the Security Operations Center,” Proceedings of the 15th International Conference on Availability, Reliability and Security, Association for Computing Machinery, 2020
1 Introduction and Background

The research field regarding virtual representation of physical entities using Digital Twin technology, and subsequent possibility of modeling and analysis, has been gaining momentum due to advances in IoT technology, cloud computing infrastructure and low-latency high-capacity wireless transmission technologies like 5G. This allows the realization of Cyber Physical Systems (CPS) with expected improvements in business efficiency, predictive maintenance and automation, to name a few.

In connecting Operation Technology (OT) systems to cyber space, the threat of cyber attacks against such CPS is increasing, especially due to increasing complexity of those systems; for example manufacturing was the top attacked industry in 2021, showing a sharp increase over recent years [1]. Compared with IT systems, OT systems often have strict operational requirements, with physical machinery being at the risk of damage if those requirements cannot be fulfilled. Therefore, traditional IT-security may fall short due to more strict operational requirements in OT systems [2].

To address these shortcomings, we are aiming to realize the Security Digital Twin (SDT) [3]; functionality includes planning, analysis and automatic application of security countermeasures. It is designed with a modular architecture, allowing support of a variety of IT/OT/IoT use cases and environments.

This paper was previously published in [4] ¹. In this paper, after describing the concept of the Security Digital Twin, we propose a data collection abstraction layer for the SDT, to improve efficiency of data collection and lower development cost. Finally, we show the results of a qualitative evaluation of the SDT with the data collection abstraction layer.

2 Security Digital Twin concept

To address issues specific to security in OT/IoT systems, we are develop-

¹This paper additionally describes the SDT concept, the data collection abstraction layer in more detail, and a qualitative evaluation of the described proposal
ing the Security Digital Twin [3]; its concept is shown in Fig. 1. The SDT collects data and state information from real IT/OT/IoT systems to generate an SDT model in cyberspace. It utilizes existing attack and risk analysis and countermeasure planning tools, modularized in the form of “action modules”, to perform mock attacks and countermeasures in simulations on the SDT model. Business continuity-related parameters are exposed in the SDT model as well, and their results during the simulations are analyzed and evaluated, thus becoming able to consider various attacks’ or countermeasures’ effects on business continuity. The evaluation results can be used to adjust model and simulation settings in a feedback loop. This consideration of effects allows a better evaluation, and possibly automatic selection of countermeasures such that the effects on business-related parameters are minimized, or security and business-related parameters can be compared against each other to support the selection and timing of countermeasures.

The proposed design of the SDT targets a variety of IT/OT/IoT use cases and environments, based on a use case/requirements analysis [3]. Its modular architecture makes the SDT highly extensible, to allow focusing on different use cases in the future, for example connected car.

3 Problem

The modular architecture of the SDT supports an agile development, with an initial focus on availability evaluation, being particularly important for OT/mission critical IoT deployments. Due to the broad scope of use cases the SDT has to cover, the different modules of the SDT have different and changing requirements regarding data sources, formats and timings. Putting the responsibility of source and data selection and collection onto each module – even with middleware support for different collection protocols – requires the implementation of nontrivial logic in each module regarding the specifics of each collection protocol. Especially in large-scale IoT deployments, the number of heterogeneous devices (e.g. from different vendors) can be large, each with specific characteristics to support. Furthermore, the number of devices that need to be represented in the SDT can grow very large, requiring specific logic to optimize concurrent data collection [5].

The naïve collection of all available data for modeling/analysis, with subsequent selection, will increase network congestion and the workload in the modeling/analysis components, often having only limited scalability, in turn causing delayed results and subsequent delay of mitigation measures, compromising timely implementation of security countermeasures and thus increasing risk.

A solution to the mentioned problems has to satisfy the following requirements:

- Ability to collect data using various protocols and data formats.
- Ability to provide collected data to other SDT components/modules.
• Ability to share collected data in various formats, according to the requirements of the respective SDT components/modules.

• Ability to collect data from a large number of devices.

• Ability to hide data collection details (e.g., protocols, data formats) from the data provisioning to other SDT components/modules.

4 Solution proposal

We introduce a data collection abstraction layer (from here “abstraction layer”) between SDT components/modules and the devices to collect data from. The abstraction layer, and its implementation design as an SDT data collection module, is shown in Fig. 2.

The abstraction layer uses device and resource discovery, and subsequent abstracted resource representation to upper layer modules to receive data requirements from those, and distributes relevant data accordingly. This could for example be periodic readings of sensor data to generate an Industrial Control System (ICS) model or adjust its parameters. Access to the abstraction layer is provided through a unified interface, in the form of data collection requirements; this allows dynamic support for different data collection protocols and formats, with the SDT components/modules being agnostic to those protocols and formats. Representable formats, discovered endpoints and data collection requirement requests are stored in databases; collected data or references to it is stored in a database as well. Both module and
database deployment can be distributed, enabling filtering, de-duplication and pre-processing of collected data to optimally satisfy the requesting upper layer modules’ requirements. Access to device data is further abstracted through a protocol-independent interface, enabling data collection protocol implementation changes independent from the other SDT data collection module components. This not only reduces the flow of unnecessary data at an early stage in the SDT, but also allows highly scaled deployments, for scenarios handling large numbers of devices.

5 Evaluation

A qualitative evaluation of the proposed data collection abstraction layer is shown in Table I; the implementation cost of a naïve implementation of the SDT, as described in Section 2, and an implementation utilizing the proposed data collection abstraction layer are compared in terms of the requirements described in Section 3. Evaluation details are described below.

Table I. Qualitative comparison of implementation cost of requirements for the SDT, in a naïve implementation and an implementation with the data collection abstraction layer

| Parameter | Naïve SDT implementation | SDT with data collection abstraction layer |
|-----------|---------------------------|--------------------------------------------|
| Collect data using various protocols and data formats | High | Low |
| Provide collected data to other SDT components/modules | Medium | Low |
| Share collected data in various formats, according to the requirements of the respective SDT components/modules | High | Low |
| Collect data from a large number of devices | Medium | Low |
| Hide data collection details (e.g. protocols, data formats) from the data provisioning to other SDT components/modules | Low | Low |

Collect data using various protocols and data formats
In the naïve implementation, each component must implement existing or new data collection for each protocol/data format separately. Due to the high number of components and supported protocols/data formats, this cost is deemed “high” by the authors. In the proposed solution, each component only implements the abstraction layer usage; changes in protocol/data format require changes in the abstraction layer only.

Provide collected data to other SDT components/modules
In the naïve implementation, each component must implement data collection separately; not every component requires support for every data format, therefore the cost is deemed “medium” by the authors. In the proposed solu-
tion, each component only implements the abstraction layer usage; protocol changes require only changes in the abstraction layer.

**Share collected data in various formats, according to the requirements of the respective SDT components/modules**

In the naïve implementation, each component must implement formats according to its own needs; due to the high number of components and data formats, this cost is deemed “high” by the authors. In the proposed solution, only the abstraction layer interface usage has to be implemented in all components, mostly comprising the exchange of requirements and reception of data in the requested format; format changes require only changes in the abstraction layer.

**Collect data from a large number of devices**

In the naïve implementation, each component collects data directly from each relevant device; the collection is uncoordinated, causing duplicate transfers as well as bottlenecks due to transfer spikes. Coordination requires support in each component, and mitigation would require usage of intermediate caching devices, with the cost deemed “medium” by the authors. In the proposed solution, all data requests are collected through the abstraction layer, allowing deduplication and further optimization of data collection.

**Hide data collection details (e.g. protocols, data formats) from the data provisioning to other SDT components/modules**

In the naïve implementation, data formats have to be adapted to each component’s needs, requiring data transformation implementation at least for a number of components; protocol or data format changes require a similar amount of work. If the number of affected components is manageable, the cost is deemed “low” by the authors. In the proposed solution, data formats provided through the abstraction interface adhere defined rules, which cover all relevant format requirements; therefore components only have to implement the abstraction layer usage; data format changes only require changes in the abstract layer, possibly also requiring configuration changes in affected components, therefore this cost is deemed “low” by the authors.

## 6 Conclusion

With the realization of CPS, the risk of connected OT/IoT systems is rising. To overcome the challenges of traditional IT security being insufficient to secure OT systems, we are developing a highly modular Security Digital Twin, targeting a wide variety of OT/IoT use-cases and scenarios. Due to its modularity and flexibility, a naïve implementation of required data collection is highly cost-inefficient; to address these issues, a data collection abstraction layer for the SDT was proposed, utilizing upper layer data requirements through a unified interface. In a qualitative evaluation, the proposed solution is shown to be more cost-effective when compared to a naïve implementation.