Towards Cross-Physical-Domain Threat Inference for Industrial Control System Defense Adaptation

Jainta Paul
u1471999@utah.edu
University of Utah
Salt Lake City, Utah, USA

Lawrence Ponce
u1384059@utah.edu
University of Utah
Salt Lake City, Utah, USA

Mu Zhang
muzhang@cs.utah.edu
University of Utah
Salt Lake City, Utah, USA

Luis Garcia
la.garcia@utah.edu
University of Utah
Salt Lake City, Utah, USA

Abstract

Safety-critical Industrial Control Systems (ICS) are increasingly targeted by Advanced Persistent Threats (APTs), exemplified by attacks like Stuxnet, the Ukraine power grid breaches, and recent U.S. water treatment facility intrusions. These sophisticated attacks often target common sensors and actuator abstractions across different ICS environments. While frameworks like MITRE ATT&CK for ICS categorize attacker Tactics, Techniques, and Procedures (TTPs), they fall short in assessing the physical impact on operational technology (OT). To bridge this gap, we introduce OTTHREAT, a novel ontology that extends the Semantic Sensor Network (SSN) framework by incorporating cyber attack abstractions and the safety properties they target. Our approach enables the mapping of similar physical processes across different ICS domains, facilitating the adaptation of existing mitigations to new threats. We implement and validate a proof-of-concept threat inference framework on three ICS use cases representing different physical domains, including water treatment ICS and oil treatment ICS, that share common sensor and actuator abstractions and demonstrate how both threat assessment and potential mitigations for discovered threats can be adapted across physical domains.

CCS Concepts

• **Computer systems organization → Sensors and actuators**.

Keywords

industrial control systems, knowledge graphs

ACM Reference Format:
Jainta Paul, Lawrence Ponce, Mu Zhang, and Luis Garcia. 2024. Towards Cross-Physical-Domain Threat Inference for Industrial Control System Defense Adaptation. In *Proceedings of the 2024 Workshop on Re-design Industrial Control Systems with Security (RICSS '24), October 14–18, 2024, Salt Lake City, UT, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3689930.3695210

RICSS '24, October 14–18, 2024, Salt Lake City, UT, USA

ACM ISBN 979-8-4007-1226-5/24/10
https://doi.org/10.1145/3689930.3695210

1 Introduction

Advanced Persistent Threats (APTs) are increasingly targeting safety-critical industrial control systems (ICS). High-profile such as Stuxnet [10], Ukraine power grid attacks [18], attacks on water treatment utilities in the US [20, 24], or even attacks on safety-instrumented systems (SIS)–systems designed to prevent industrial incidents [9], highlight a trend of attackers aiming to cause catastrophic damage in society. These attacks target various physical sensors and actuators guiding safety-critical processes, and adversaries often exploit common vulnerabilities across different systems. To proactively mitigate potential threats, it is crucial to infer potential APTs for ICS by mapping and understanding these cross-domain threats.

APTs are often characterized by Tactics, Techniques, and Procedures (TTPs) that attackers use to execute these complex threats. The MITRE ATT&CK framework for Industrial Control Systems (ICS) [4] has become an industry standard for systematically categorizing these TTPs, providing a comprehensive knowledge base that spans the entire lifecycle of an attack. Unlike traditional TTP frameworks that focus solely on IT infrastructure, the MITRE ATT&CK for ICS emphasizes the unique characteristics of industrial environments, incorporating both IT and OT components. However, while the framework conceptually maps TTPs, most assessments focus on the tactics and techniques used to breach systems without fully exploring the disruptions to physical processes and safety mechanisms. Moreover, the frameworks serve as guidelines for cybersecurity assessment and manually configuring the security analyses across the complex connectivity of cyber-physical ICS does not scale and requires cross-domain expertise.

Prior works have aimed to structure the unstructured knowledge provided by TTP frameworks and other cyber threat intelligence reports for cyber-security assessment using knowledge graphs (KGs) to identify or infer attack techniques [19, 23, 30], including in the ICS domain [26]. However, these works mainly focus on mapping intrusions on the IT infrastructure rather than the impact on the operational technology (OT). A large body of non-KG-based approaches emerged aiming to understand the IT/OT cross-domain impact of attacks and mitigations [2, 8, 12, 15]. However, these are often tailored analyses for physical domain-specific solutions. More critically, all of the previous frameworks neglect the targeted sensor and actuator semantics of any attack *across* physical domains, e.g., an attack that overflows a water treatment plant by continuously opening a valve will most likely apply to an oil treatment facility controlled by similar industrial pumps and valves. Conversely, several works from the cyber-physical systems design space are aiming to leverage common physical representations as a means to semantically ground ICS components for seamless integration in an increasingly connected world [16, 25], but neglect threat models.

In this paper, we seek to bridge the aforementioned gaps by integrating physical impact assessments into cyber threat ontologies, enhancing our understanding of how APTs can propagate through ICS environments and identifying candidate physical targets using threat intelligence *across* physical domains. As an initial step, we focus on how to represent the physical impact of threats across OT environments. We introduce a formal ontology, OTThreat, that builds upon a standard ontology framework for representing sense-to-actuate relationships in cyber-physical systems, the semantic sensor network (SSN) ontology [22]. OTThreat includes abstractions for attacks, as well as relations mapping the compromised sensors or actuators, as well as abstractions for the safety properties targeted by the attacks for a given process. We provide the preliminary formalization for how an ICS and an associated attack dataset can be mapped to the OTThreat ontology from a variety of unstructured sources, including ICS source operational manuals and attack reports. Given a knowledge base of known ICS attacks and their target ICSs represented as a KG using the OTThreat ontology, we show how similar physical processes can be mapped across physical domains. Moreover, we demonstrate how the KGs can be used to not only infer and adapt known physical threats to a target OT but also to adapt mitigations (e.g., anomaly detectors) for said threats. We demonstrate the efficacy of our approach on three preliminary ICS use cases with known attacks or datasets: the Secure Water Treatment Testbed (SWaT) [21], the MiniSWaT testbed–a scaled-down testbed of SWaT using Raspberry Pi's, and an oil treatment ICS cybersecurity testbed simulator [28].

Contributions. The contributions of this paper are summarized as follows:

- We introduce a cyber-physical threat ontology, OTThreat, to map and infer ICS physical threats across physical domains–combining traditional semantic CPS ontology frameworks with threat abstractions.
- We formalize a framework to map ICS representations to knowledge graphs using the OTThreat ontology, which can then be used to adapt threat intelligence across physical domains. Additionally, we show how the KG representations can be used to adapt existing mitigations across different ICS domains.
- We demonstrate the feasibility of the proposed approach through preliminary demonstrations and experimental validation across 3 ICS physical domains: two different water treatment ICSs and an oil treatment plant.

2 Background and Related Work

2.1 CPS Ontological frameworks

Many prior works have focused on representing Cyber-Physical Systems and their components, primarily for the sake of semantic integration of designs. The Semantic Sensor Network (SSN) [22] ontology is an extension of the Sensor, Observation, Sample, and Actuator (SOSA) [17] ontology, both of which are combined and provide a standard vocabulary for describing sensors, their capabilities, observations, and the context in which measurements are made. These ontologies have become the standard for integrating heterogeneous sensor data from various sources, making combining and analyzing data from sensors and systems seamless and have been adopted in industrial applications for asset tracking, condition monitoring, and process optimization in various domains such as manufacturing, energy, and transportation [29]. The SemCPS [14] framework integrates different perspectives, including mechanical, electrical, and software to represent a system. Other frameworks have been proposed for semantic grounding of interoperability across ICS [16] or providing grounded abstractions for ICS and their digital twins [23]. In all cases, the ontologies typically focus on semantic grounding across ICS designs but typically do not consider CPS threats nor the physical impact an attack may have. Any of these CPS ontological frameworks are amenable to integrating threat intelligence, and in this paper, we choose to extend SSN/SOSA ontologies[1] to represent safety properties, the associated physical threats, and mitigation solutions given that the SSN/SOSA ontology is the most expressive and granular for modeling sense-to-actuate pipelines and is an adopted industry standard.

2.2 Threat Model and Assumptions

System Model. The goal of our framework is to provide security analysts with candidate physical threats of a given ICS. Thus, to build a KG representation of a given ICS, we initially assume that we have access to various structured and semi-structured documentation of the target ICS–including natural language operational manuals, technical documentation, and even source code. Additionally, we assume access to a large knowledge base of known threats and high-profile attacks [5, 11], as well as academic papers describing cyber-physical attacks [13, 27, 31]. The data sources enable the utility of the OTThreat ontology. A key feature of our approach is that our ontology can provide graphical representations with different levels of granularity. This flexibility allows for adaptable threat modeling based on the available information and the specific needs of the analysis. Future work will focus on scenarios with an incomplete view of system architectures, addressing more realistic situations where complete system information may not be available. It is important to note that our model focuses on the operational phase of an engineered system. We are considering the perspective of a security analyst examining potential threats to a system that's already up and running. This means we are dealing with real-world scenarios where the system actively processes data and controls physical processes. **Adversary Capabilities and Goals.** We assume that the goal of an attacker is to maximize physical impact on the ICS while maintaining stealthiness. An attacker has various levels of access to system components, including but not limited to sensors, actuators, PLC, SCADA, and HMI. Thus, an attacker can implement any necessary measures to compromise particular components, but not all. The threat vectors may include sensor spoofing, false data injection into sensors, actuator manipulation, unauthorized commands to actuators, and control logic tampering.

[1] In this paper, we refer to SSN and SOSA ontologies together since they are used in tandem.

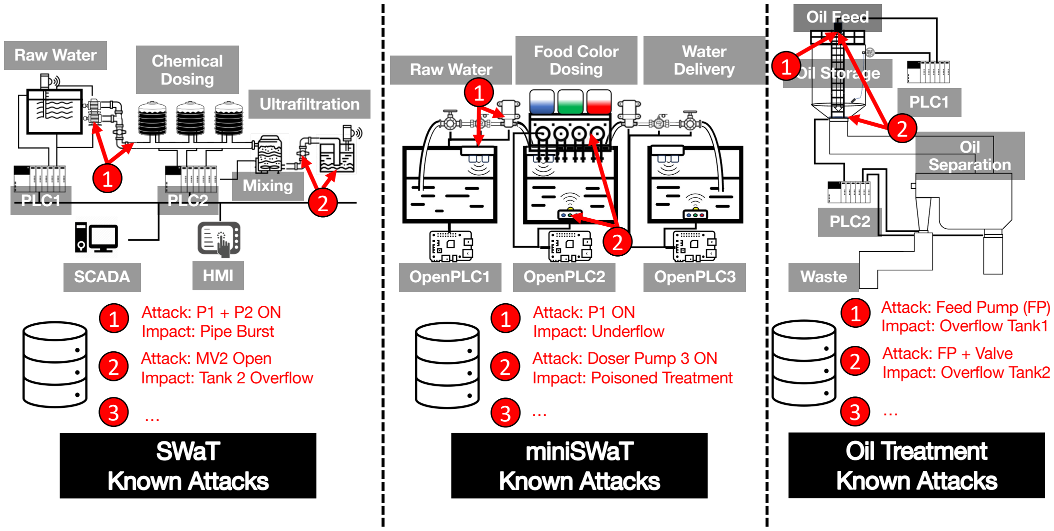

Figure 1: Overview of three use case ICSs, highlighting cross-domain threats and commonalities.

Additionally, we assume that an attacker would aim to repeat behaviors across physical domains based on previously successful incidents.

2.3 Use Case Domains

In our study, we selected three ICS use cases to evaluate our approach comprehensively, depicted in Figure 1. Each use case is associated with a set of known attacks. The first is the Secure Water Treatment (SWaT) testbed [21], a well-established water treatment testbed extensively used in cybersecurity research. SWaT offers a realistic representation of a full-scale water treatment plant, making it an ideal benchmark for our analysis. To complement SWaT, we developed MiniSWaT, a surrogate testbed that mirrors the processes of SWaT but utilizes different hardware components. We incorporated food coloring with RGB sensors in MiniSWaT to simulate the closed-loop chemical dosing process, providing a cost-effective yet functional alternative to actual chemical treatments. Our third use case, an oil treatment system [28], introduces different processes with common components and safety properties similar to water treatment systems.

2.3.1 SWAT. We used the first process of SWaT. This process takes water from a raw water source and feeds it into a tank. This water will then be pumped into a second tank and treated with chemicals. For this first process, the PLC::PLC1 controls the water inflow by opening or closing the valve, Motorized Valve:MV101, and water outflow by running the Pump::P101. The PLC::PLC1 monitors the water level of the Tank::T101 using a level sensor, Level Indicator::LIT101. The PLC::PLC1 operates another Pump::P102, which works as a backup for the Pump::P101 in case the Pump::P101 does not work. The PLC::PLC1 is responsible for keeping the water level in Tank::T101 within a specific range: it should never hit an "underflow" or an "overflow."

2.3.2 MiniSWaT. The first process of MiniSWaT is very similar to the first process of SWaT. This process takes water from a raw water Tank::T1 and feeds it into a Tank::T2 to be mixed with food coloring. The PLC::PLC1 controls the water outflow by opening or closing the Valve::V1 and running the Pump::P1. The PLC::PLC1 monitors the water level of the Tank::T1 using a Range Sensor::RS1, and the outflow rate using a Flow Sensor::FS1. The PLC::PLC1 ensures that the water level in the tank never hits an "underflow."

2.3.3 Oil Treatment Plant. For our final use case, we used the first process of an oil treatment ICS cybersecurity simulator. The process is controlled by a PLC, which monitors the Oil Storage Unit's oil level with a Tank Level Sensor. The PLC controls a Feed Pump that fills the oil storage. The PLC is additionally tasked with operating an outlet valve to facilitate the release of oil from the storage unit once it has reached its maximum capacity.

3 Semantic Mapping of ICS Threats

In this section, we demonstrate the process of semantically mapping ICS threats to our proposed ontology. This mapping is crucial for translating real-world ICS processes and their associated threats into a structured, machine-readable format. To illustrate this approach, we first provide a detailed example of mapping a single process scan cycle of SWaT, a real-world ICS according to our ontology. We then show, how we can represent safety properties and potential threats to that particular process. Moreover, we introduce an approach that leverages knowledge graph representations of two distinct ICSs to infer potential threats to one system based on known attacks to the other, thereby enabling cross-system threat analysis and prediction. Finally, we describe how such threats can be used to infer potential threat mitigations.

3.1 Use case: Mapping a single process

A PLC monitors and controls a system through a "scan cycle." During each scan cycle, the PLC reads the state of all input devices, such as sensors, and saves them in memory. Next, it executes the programmed logic to control the actuators, analyzing these inputs

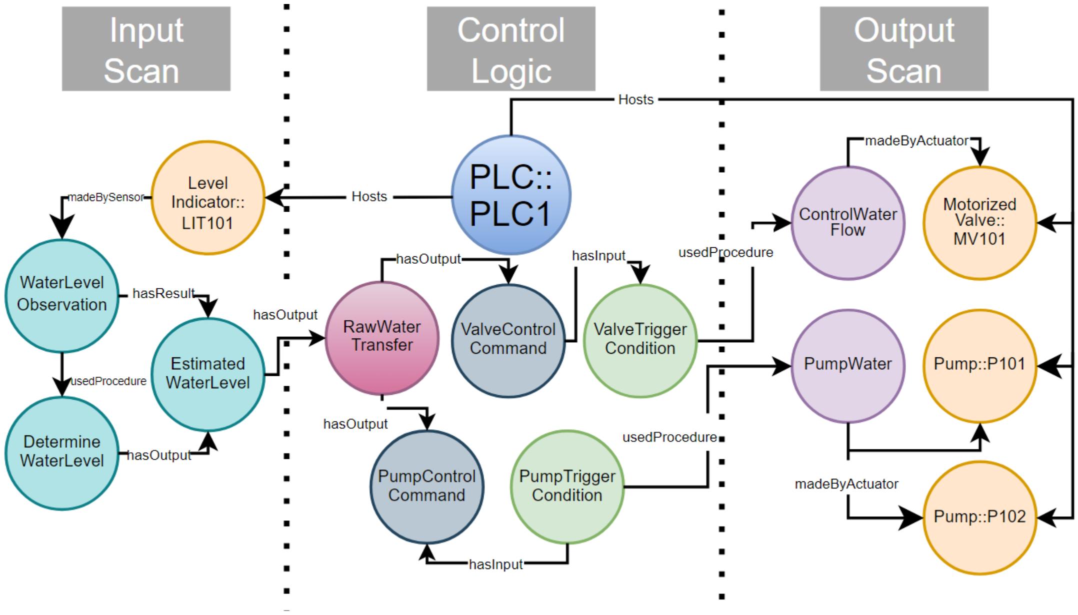

Figure 2: SSN/SOSA representation of a single process of a PLC scan cycle.

and making decisions depending on the instructions. In Figure 2, we show our envisioned framework that takes the description of a process as an input and outputs the representation of the scan cycle using SSN and SOSA ontologies. As depicted in the figure, the PLC works as a Platform that "hosts" Sensors and Actuators and as a System that "implements" a Procedure. In this case, the PLC::PLC1 "hosts" one Sensor, Level Indicator::LIT101, and three Actuators: Pump::P101, Pump::P102, and Motorized Valve::MV101. The PLC::PLC1 "implements" the Procedure RawWaterTransfer. The input of this Procedure comes from an Observation called WaterLevelObservation, which is "madeBySensor" Level Indicator::LIT101. The Procedure outputs control logic for the Actuators.

3.2 Ontological Development of Cyber-Physical Threat Ontology

As depicted in Figure 2, SSN and SOSA ontologies are not sufficient to represent the safety properties, physical threats, and defense mechanisms associated with a process. Figure 3 illustrates how threats interact with the SWaT process in our extended ontology.
Class :: SafetyProperty. In the context of our ontology, a safety property represents any type of safety requirement typically used as a premise for safety verification and validation, anomaly detection, or other safety-critical operations. These properties are usually functions of observable properties. These safety properties often become targets of attacks. In Figure 3, we can see that the Tank::T101 has a SafetyProperty called WaterLevelThreshold, which indicates the water level in Tank::T101 must always be within a certain range[2].

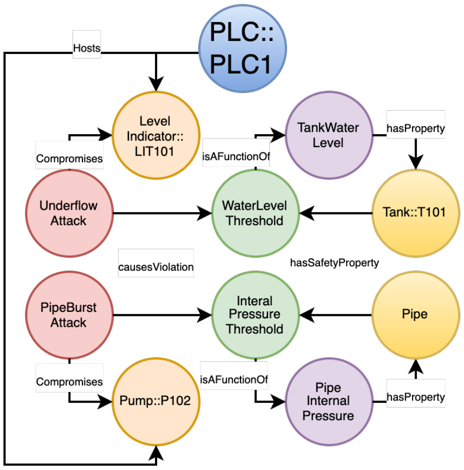

Figure 3: OTThreat KG representation of attacks compromising specific sensors or actuators and causing violations for certain infrastructural safety requirements.

Class :: Attack. An Attack "compromises" a *FeatureOfInterest* and "causesViolation" of a *SafetyProperty*. An ICS attack can compromise any component that is controlled by a computer program. As shown in Figure 3, the UnderflowAttack compromises the Level Indicator::LIT101 and causes a violation of WaterLevelThreshold. It's important to note that our approach is not tied to any specific safety property language, allowing for flexibility in representing various types of safety requirements.

[2] Safety properties in our ontology can represent a wide range of requirements, from simple threshold values to complex relationships between multiple system variables.

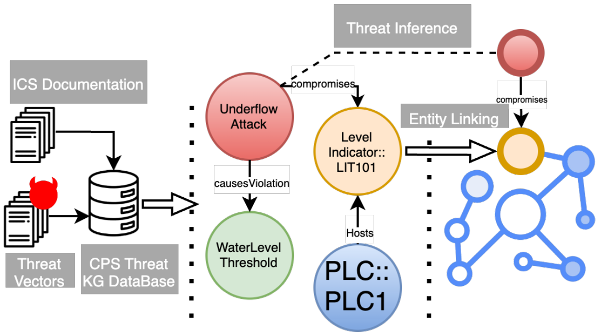

Figure 4: Envisioned KG threat inference pipeline enabled by OTThreat ontology.

Class :: Defense. A Defense "monitors" an *ObservableProperty* and "detects" an *Attack*. In the above example, if a prior mitigation technique was implemented to detect discrepancies in the water threshold, e.g., through physical models [3], then the Defense node would monitor the TankWaterLevel to detect the PipeBurst Attack[3].

3.3 Cross-domain Threat Inference

This section formally represents our approach to infer threats across different domains. Specifically, we formalize the threat cross-domain inference enabled by our OTThreat ontology, as depicted in Figure 4, assuming that the other KG pipeline components are implemented. For instance, we assume there is a method to extract information from threat reports and operational manuals and map it to our proposed representation. Given two KG representations and one known KG database, we envision that our framework would query the database to infer threats for the other one. From these representations, our algorithm enables entity linking between two knowledge graphs, identifying corresponding entities based on label similarity and structural context. As one potential implementation for cross-domain threat inference, we present an approach that combines string-matching techniques with graph-based context analysis. This method produces a set of entity alignments with confidence scores, serving as a proof-of-concept for how our ontology can be leveraged for cross-system threat analysis. In developing this approach, we hypothesize that lower-level abstractions, such as specific hardware components, are less likely to match across different systems due to hardware variations, even when they sense the same observable properties. However, we anticipate that higher-level software abstractions, such as the results of observations or process states, are more likely to align. This hypothesis informs our entity-linking strategy, focusing on identifying similarities at a more abstract level rather than relying solely on component-level matching. It is important to note that this is just one of many possible applications of our ontological framework, and alternative approaches could be developed to suit specific use cases or domains. Below, we formally describe our algorithm as an illustrative example of how our ontology can be utilized in practice.

Input: The algorithm takes as input two knowledge graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, where V represents the set of vertices (entities) and E represents the set of edges (relations). Each vertex $v \in V$ has attributes: label and type. Each edge $e \in E$ has an attribute: relation.

Output: The algorithm generates a set of entity alignments $A = (v_1, v_2, s) | v_1 \in V_1, v_2 \in V_2, s \in [0, 1]$, where s represents the confidence score of each alignment. A higher confidence score indicates greater similarity between nodes. This alignment set facilitates threat inference across systems: if an Attack A is known to compromise a component C1, and C1 is determined to be highly similar to a component C2 in another system (as indicated by a high confidence score), then it can be inferred that C2 is potentially vulnerable to Attack A.

Algorithm 1 Entity Linking for Cross-domain Threat Inference

```
Input: Two knowledge graphs $G_1$ and $G_2$
Output: Set of entity alignments $A$
Initialize candidate set $C$ for each entity in $G_1$
for all entities $v_1$ in $G_1$ do
    for all entities $v_2$ in $G_2$ do
        Compute string similarity between $v_1$ and $v_2$
        if similarity exceeds threshold then
            Add $v_2$ to candidate set of $v_1$
        end if
    end for
end for
for all entities in $G_1$ and $G_2$ do
    Compute context information (neighboring nodes, types, relations)
end for
Initialize alignment set $A$
for all entities $v_1$ in $G_1$ do
    Find best matching candidate based on:
      - String similarity
      - Context similarity (type, neighbors, relations)
    if best match score exceeds linking threshold then
        Add alignment to $A$
    end if
end for
return Alignment set $A$
```

Complexity: The time complexity of the algorithm is dominated by the candidate generation step, resulting in an overall complexity of $O(n^2)$. The context analysis step has a complexity of $O(n \times d)$, where d is the average degree of nodes, and the entity linking step has a complexity of $O(n \times c)$, where c is the average number of candidates per entity.

4 Implementation and Evaluation

We study the efficacy of our approach using three scenarios. We show that if we have a knowledge graph representation of two ICSs and we know the risks that are associated with one of them, we can infer the threats that are associated with the second testbed.

4.1 Experimental Setup

We implemented our approach using Python. The ontology for each selected process was represented using a consistent schema, defining classes such as Sensor, Actuator, Process, and Attack, along with their relationships. The implementation included functions

[3] While threat mitigations are orthogonal to our primary contributions, we provide an example threat mitigation adaptation in Section 4.

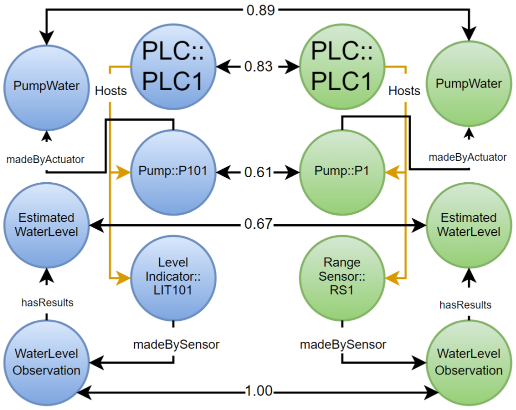

Figure 5: Preliminary results of entity linking mapping the SWaT and MiniSWaT KGs using the OTThreat ontology.

to map ontology classes and properties to graph nodes and edges, ensuring semantic consistency across different ICS representations. We applied our entity linking algorithm to three pairs of testbeds (MiniSWaT-SWaT, MiniSWaT-Oil Treatment, SWaT-Oil Treatment), identifying corresponding entities based on label similarity and structural context.

4.2 Threat Mapping Across Physical Domains

4.2.1 SWaT-MiniSWaT. We evaluated two types of attacks for this specific set of systems. The first attack is an Underflow attack, which targets the Level Indicator::LIT101. The second attack is a PipeBurst Attack, which targets the Pump::P102. Our analysis revealed that the algorithm failed to identify any similarities between the pairs Level Indicator::LIT101(SWaT)-Range Sensor::RS1(MiniSWaT), and Pump::P102(SWaT)-Pump::P1(MiniSWaT). It is worth noting that our approach uses string similarity as a simple initial method. However, we can establish mappings without relying on string similarity when semantic structures can be inferred (such as identifying components as pumps). This semantic-based comparison can be more effective in cases where component names differ, but their functions are equivalent. Interestingly, we identified similarities between Pump::P101(SWaT)-Pump::P1(MiniSWaT). However, our domain expertise tells us that Pump::P102 and Pump::P101 in SWaT likely have similar roles and vulnerabilities. With this in mind, we can reasonably conclude that a threat capable of compromising Pump::P102 in SWaT could also pose a risk to Pump::P1 in MiniSWaT. The entity linking results are depicted in Figure 5. Furthermore, despite our inability to identify similarities between the sensor pair, we were able to identify similarities in their associated observable properties. The result is consistent with the hypothesis we stated in Section 3.3. We simulated the attacks in the MiniSWaT by compromising the relevant nodes. The physical impact of MiniSWaT was similar to that of SWaT.

4.2.2 MiniSWaT-Oil Treatment. For this pair of ICSs, our entity linkers were not able to link the node pairs: Range Sensor::RS1 (MiniSWaT) - TANK LEVEL SENSOR(Oil Treatment) and Pump::P1 (MiniSWaT) - FEED PUMP, but correctly identified the observable properties associated with the sensors. In this case, since the component pairs were very similar, we inferred similar kinds of threats. We did not simulate any attack in the Oil Treatment Plant. As a result, we cannot say for sure about the physical impact. We could be confident about the physical impact if we had a physical ICS or data for the physical or the simulated ICS.

4.2.3 SWaT-Oil Treatment. We observed a similar result as the previous one. Our entity linkers were not able to link the node pairs: Level Indicator::LIT101(SWaT)-TANK LEVEL SENSOR(Oil Treatment) and Pump:P102(SWaT)-FEED PUMP, but correctly identified the observable properties associated with the sensors.

4.3 Defense Adaptation

We performed a preliminary analysis to understand how known mitigation for the inferred threats can be automatically adapted and suggested as a candidate mitigation for a target ICS. We leveraged mitigations for both attacks from prior works [1, 3] that developed an anomaly detector based on the physical invariant properties. Our goal was to formalize how the state-based detectors can be automatically adapted. The SWaT anomaly detector code was written as a simple Python script with the encoded control invariant properties to analyze traces for the target sensor values and raise a flag when there was a discrepancy. Specifically, the control invariant properties were defined based on observable properties (i.e., the results produced from the sensor input scans). Thus, adaptation was a simple matter of adapting the parameters of the threshold values. In real-world settings, we envision candidate mitigation can be parametrized either by the domain expert, known safety thresholds for the target ICS (e.g., if the associated process already had a safety property and we map the associated state abstractions) or can be inferred through physics-based modeling or data-driven approaches.

Results. We observed that we were able to detect both the underflow and overflow attacks that were mapped to the MiniSWaT testbed after adapting the SWaT attack mitigation code using the MiniSWaT safety properties. However, we were not able to automatically adapt the pipe burst attack. Intuitively, the pipe burst attack targets the observable property associated with internal pipe pressure–which we cannot directly observe. Such mitigation would require manually encoding attacks that, e.g., detect that the valve cannot be closed while an inflow pump is on. Nonetheless, our framework would still suggest the candidate mitigation with a property that is currently unobserved–implying that the developers should implement a solution to observe such a property. This approach is aligned with the industry trend to provide resiliency through redundancy.

5 Discussion

We demonstrated the feasibility of mapping and analyzing threats across different ICS domains using our proposed ontological framework. While these preliminary results are promising, they highlight several challenges and limitations that demand further discussion. One significant challenge we encountered was interpreting results, particularly in cases where the physical impact of inferred threats could not be directly realized due to the absence of a physical ICS or comprehensive simulation data(Oil Treatment Plant). This shows how important it is to have access to a wide range of datasets when conducting cross-domain threat inference. Finally, The defense adaptation entailed adapting the parameters of existing mitigation scripts. The mitigation was based on physical models of the control processes. In practice, data-driven approaches have proven to be more robust at anomaly detection [7]. However, end-to-end data-driven approaches are difficult to adapt, given that the model is finely tuned to the target ICS. Thus, future work can explore the use of neurosymbolic programming [6] that leverages the power of deep learning models while maintaining the benefits of symbolic interfaces, e.g., to adapt parameters in a data-efficient manner easily.

There are several critical areas for future work. End-to-end automation of the threat inference process and improving the scalability of our approach are crucial next steps. Additionally, incorporating provenance analysis within the alignment module could enhance inferred threats' reliability and traceability. We also recognize the need to expand our representation to include other critical ICS components, such as Human-Machine Interface (HMI) and Supervisory Control and Data Acquisition (SCADA).

Although our current approach does not adhere to a specific language for defining safety properties, future research could explore how a grounded representation of safety properties in terms of observable properties could enable more sophisticated reasoning and improved adaptation of threats and defenses across different domains. This could lead to more nuanced and context-aware threat and defense inference mechanisms, further enhancing the practical applicability of our framework in diverse ICS environments.

6 Conclusion

In this paper, we introduced OTTHREAT, a cyber-physical threat ontology designed to map and infer ICS threats across physical domains. We created a framework that enhances the detection and mitigation of cross-physical-domain threats by integrating cyber attack abstractions and safety properties into the Semantic Sensor Network (SSN) ontology. We provided a proof-of-concept evaluation of three different ICS systems with common sensor and actuator abstractions targeted by known attacks. We discussed future work focusing on the automation of information extraction and the formalization of safety properties to facilitate cross-domain threat mitigation adaptation.

7 Acknowledgment

We would like to thank the anonymous reviewers for their constructive comments. This work was supported in part by NSF FMitF-2425711, NSF CNS-2435642, NSF OAC-2115167, DARPA HR00112120009 Cooperative Agreement, a Cisco gift, and a USHE Deep Technology Initiative Grant. Any opinions, findings, and conclusions made in this material are those of the authors and do not necessarily reflect the views of the funding agency.

References

[1] Sridhar Adepu, Ferdinand Brasser, Luis Garcia, Michael Rodler, Lucas Davi, Ahmad-Reza Sadeghi, and Saman Zonouz. 2020. Control behavior integrity for distributed cyber-physical systems. In *2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 30–40.

[2] Sridhar Adepu and Aditya Mathur. 2016. Using process invariants to detect cyber attacks on a water treatment system. In *ICT Systems Security and Privacy Protection: 31st IFIP TC 11 International Conference, SEC 2016, Ghent, Belgium, May 30-June 1, 2016, Proceedings 31*. Springer, 91–104.

[3] Sridhar Adepu and Aditya Mathur. 2017. From design to invariants: Detecting attacks on cyber physical systems. In *2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*. IEEE, 533–540.

[4] Otis Alexander, Misha Belisle, and Jacob Steele. 2020. MITRE ATT&CK for industrial control systems: Design and philosophy. *The MITRE Corporation: Bedford, MA, USA* 29 (2020).

[5] Harold Booth, Doug Rike, and Gregory A Witte. 2013. The national vulnerability database (nvd): Overview. (2013).

[6] Swarat Chaudhuri, Kevin Ellis, Oleksandr Polozov, Rishabh Singh, Armando Solar-Lezama, Yisong Yue, et al. 2021. Neurosymbolic programming. *Foundations and Trends® in Programming Languages* 7, 3 (2021), 158–243.

[7] Yuqi Chen, Christopher M Poskitt, and Jun Sun. 2018. Learning from mutants: Using code mutation to learn and monitor invariants of a cyber-physical system. In *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 648–660.

[8] Katherine R Davis, Charles M Davis, Saman A Zonouz, Rakesh B Bobba, Robin Berthier, Luis Garcia, and Peter W Sauer. 2015. A cyber-physical modeling and assessment framework for power grid infrastructures. *IEEE Transactions on smart grid* 6, 5 (2015), 2464–2475.

[9] Alessandro Di Pinto, Younes Dragoni, and Andrea Carcano. 2018. TRITON: The first ICS cyber attack on safety instrument systems. *Proc. Black Hat USA* 2018 (2018), 1–26.

[10] Nicolas Falliere, Liam O Murchu, and Eric Chien. 2011. W32.Stuxnet Dossier. https://www.symantec.com/content/en/us/enterprise/media/security_response-/whitepapers/w32_stuxnet_dossier.pdf.

[11] Nicolas Falliere, Liam O Murchu, Eric Chien, et al. 2011. W32. stuxnet dossier. *White paper, symantec corp., security response* 5, 6 (2011), 29.

[12] Luis Garcia, Ferdinand Brasser, Mehmet Hazar Cintuglu, Ahmad-Reza Sadeghi, Osama A Mohammed, and Saman A Zonouz. 2017. Hey, My Malware Knows Physics! Attacking PLCs with Physical Model Aware Rootkit.. In *NDSS*. 1–15.

[13] Luis Garcia, Ferdinand Brasser, Mehmet Hazar Cintuglu, Ahmad-Reza Sadeghi, Osama A Mohammed, and Saman A Zonouz. 2017. Hey, My Malware Knows Physics! Attacking PLCs with Physical Model Aware Rootkit.. In *NDSS*. 1–15.

[14] Irlán Grangel-González, Lavdim Halilaj, Maria-Esther Vidal, Omar Rana, Steffen Lohmann, Sören Auer, and Andreas W. Müller. 2018. Knowledge Graphs for Semantically Integrating Cyber-Physical Systems. In *Database and Expert Systems Applications: 29th International Conference, DEXA 2018, Regensburg, Germany, September 3–6, 2018, Proceedings, Part I* (Regensburg, Germany). Springer-Verlag, Berlin, Heidelberg, 184–199. https://doi.org/10.1007/978-3-319-98809-2_12

[15] Moses Ike, Kandy Phan, Anwesh Badapanda, Matthew Landen, Keaton Sadoski, Wanda Guo, Asfahan Shah, Saman Zonouz, and Wenke Lee. 2023. Bridging Both Worlds in Semantics and Time: Domain Knowledge Based Analysis and Correlation of Industrial Process Attacks. *arXiv preprint arXiv:2311.18539* (2023).

[16] Utkarshani Jaimini, Tongtao Zhang, Georgia Olympia Brikis, and Amit Sheth. 2022. imetaversekg: Industrial metaverse knowledge graph to promote interoperability in design and engineering applications. *IEEE Internet Computing* 26, 6 (2022), 59–67.

[17] Krzysztof Janowicz, Armin Haller, Simon J.D. Cox, Danh Le Phuoc, and Maxime Lefrançois. 2019. SOSA: A lightweight ontology for sensors, observations, samples, and actuators. *Journal of Web Semantics* 56 (2019), 1–10. https://doi.org/10.1016/j.websem.2018.06.003

[18] Robert Lee, Michael Assante, and Tim Conway. 2016. Analysis of the Cyber Attack on the Ukrainian Power Grid. https://www.nerc.com/pa/CI/ESISAC/Documents/E-ISAC_SANS_Ukraine_DUC_18Mar2016.pdf.

[19] Zhenyuan Li, Jun Zeng, Yan Chen, and Zhenkai Liang. 2022. AttacKG: Constructing technique knowledge graph from cyber threat intelligence reports. In *European Symposium on Research in Computer Security*. Springer, 589–609.

[20] Sean Lyngaas. 2023. Federal investigators confirm multiple US water utilities hit by hackers. https://www.cnn.com/2023/12/01/politics/us-water-utilities-hack/index.html.

[21] Aditya P Mathur and Nils Ole Tippenhauer. 2016. SWaT: A water treatment testbed for research and training on ICS security. In *2016 international workshop on cyber-physical systems for smart water networks (CySWater)*. IEEE, 31–36.

[22] Holger Neuhaus and Michael Compton. 2009. The semantic sensor network ontology. In *AGILE workshop on challenges in geospatial data harmonisation, Hannover, Germany*. 1–33.

[23] Yulu Qi, Zhaoquan Gu, Aiping Li, Xiaojuan Zhang, Muhammad Shafiq, Yangyang Mei, and Kaihan Lin. 2023. Cybersecurity knowledge graph enabled attack chain detection for cyber-physical systems. *Computers and Electrical Engineering* 108 (2023), 108660. https://doi.org/10.1016/j.compeleceng.2023.108660

[24] Frances Robles and Nicole Perlroth. 2021. 'Dangerous Stuff': Hackers Tried to Poison Water Supply of Florida Town. https://www.nytimes.com/2021/02/08/us/oldsmar-florida-water-supply-hack.html.

[25] Nada Sahlab, Simon Kamm, Timo Müller, Nasser Jazdi, and Michael Weyrich. 2021. Knowledge graphs as enhancers of intelligent digital twins. In *2021 4th*

IEEE international conference on industrial cyber-physical systems (ICPS). IEEE, 19–24.

[26] Guowei Shen, Wanling Wang, Qilin Mu, Yanhong Pu, Ya Qin, and Miao Yu. 2020. Data-Driven Cybersecurity Knowledge Graph Construction for Industrial Control System Security. *Wirel. Commun. Mob. Comput.* 2020 (2020), 8883696:1–8883696:13. https://doi.org/10.1155/2020/8883696

[27] Saleh Soltan, Prateek Mittal, and H. Vincent Poor. 2018. BlackIoT: IoT Botnet of high wattage devices can disrupt the power grid. In *Proceedings of the 27th USENIX Conference on Security Symposium* (Baltimore, MD, USA) *(SEC'18)*. USENIX Association, USA, 15–32.

[28] Peter Prjevara-Dima van de Wouw. 2018. Improving Machine Learning based Intrusion and Anomaly Detection on SCADA and DCS using Case Specific Information. (2018).

[29] W3C SSN Community Group. 2024. SSN Applications. https://www.w3.org/community/ssn-cg/wiki/SSN_Applications.html. Accessed: 2024-09-06.

[30] Shuqin Zhang, Peng Chen, Guangyao Bai, Shijie Wang, Minzhi Zhang, Shuhan Li, and Chunxia Zhao. 2022. An automatic assessment method of cyber threat intelligence combined with ATT&CK matrix. *Wireless Communications and Mobile Computing* 2022, 1 (2022), 7875910.

[31] Yipeng Zhang, Zhonghao Sun, Liqun Yang, Zhoujun Li, Qiang Zeng, Yueying He, and Xiaoming Zhang. 2020. All Your PLCs Belong to Me: ICS Ransomware Is Realistic. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. 502–509. https://doi.org/10.1109/TrustCom50675.2020.00074