HAI 1.0: HIL-based Augmented ICS Security Dataset

Hyeok-Ki Shin Woomyo Lee Jeong-Han Yun HyoungChun Kim

The Affiliated Institute of ETRI
Republic of Korea
{hkshin721, wmlee, dolgam, khche}@nsr.re.kr

Abstract

Datasets are paramount to the development of AI-based technologies. However, the available cyber-physical system (CPS) datasets are insufficient. In this paper, we introduce the HIL-based augmented ICS security (HAI) dataset 1.0 (https://github.com/icsdataset/hai), the first CPS dataset collected using the HAI testbed. The HAI testbed comprises three physical control systems, namely GE turbine, Emerson boiler, and FESTO water treatment systems, combined through a dSPACE hardware-in-the-loop (HIL) simulator. We built an environment to remotely and automatically manipulate all components of a feedback control loop. Using this environment, we collected the HAI dataset 1.0 while repeatedly running a large number of benign and malicious scenarios for a long period with minimal human effort. We will continue to improve the HAI testbed and release new versions of the HAI dataset.

1 Introduction

Potential security threats to industrial control systems (ICSs) are gradually increasing with increasing connectivity through digital transformation and IIoT technology. As control processes become more complex, along with a greater number of hidden interaction chains [2], it is difficult to design safety protection systems or intrusion detection systems that can respond to all types of threats to the process.

Artificial intelligence (AI) technologies developed for image, voice, imagery, and natural language processing have enabled behavioral analysis for such large-scale systems. As the stability and security of an ICS are important factors determining the competitiveness of the system, security research is being conducted in the analysis of large-scale operation information of the control system using AI [3, 4, 6, 9]. A high-quality and rich dataset is required for AI-based techniques. We surveyed ICS security datasets and found that only a few datasets are suitable for security research, among which the SWaT dataset [5] is the most widely used.

In this study, we built a HAI testbed [10] to develop a control system security dataset. The HAI testbed comprises GE turbine, Emerson boiler, and FESTO water treatment systems centered on a hardware-in-the-loop (HIL) simulator, making it easy to change the process correlation between each process, with an advantage of simulating various complex processes. In addition, a more sophisticated attack can be executed repeatedly with a new attack tool for a long period of time.

In this paper, we introduce the HAI 1.0 dataset, the first dataset collected on an ICS testbed that can simulate a hybrid power generation system. The HAI 1.0 dataset contains training data for 10 days (normal data only) and test data including 38 attacks for 5.5 days. Prior to the analysis of the existing control system dataset [1], the following points were focused on for the development of HAI 1.0:

- **Stealth attack:** Attacks targeting the sensors and actuators are often deceived by arbitrary values to conceal the state change due to the attack. None of the datasets proposed in previous studies have included stealth attacks. Our HAI dataset contains several stealth attacks that conceal changes in the sensor response due to the attack.

- **Deep attack analysis:** In dataset-based studies, there is a limitation on attack analyses when describing the abnormality and simple attack behavior only. HAI 1.0 provides actual values even in situations where the control variable values are maliciously manipulated by an attack. This enables an accurate evaluation of the predicted value of the anomaly detection model.

- **Reliable data label:** Data label is the only reference that determines the performance of the anomaly detection model. Manual data labeling deteriorates the accuracy and reliability of the data.The HAI dataset improves labeling reliability because the data labeling is done by synthesizing the information generated during the attack process using an automated attack tool.

The rest of this paper is organized as follows. Section 2 introduces an enhanced ICS testbed for generating HAI 1.0 dataset. Section 3 describes the details of the dataset. Finally, Section 4 presents the conclusions of this study and future directions for newer versions of the HAI 1.0 dataset.

2 Enhanced ICS Security Testbed

To develop a diverse and rich dataset on laboratory-scale ICS testbeds, the controlled processes of three independent testbeds are interconnected and augmented using an HIL simulator, as shown in Fig.1. Moreover the supervisory control and data acquisition (SCADA) operations were automated to minimize the human intervention and increase operation reliability under normal conditions for a long time.

2.1 HIL-Based Process Augmentation

The testbed has four primary processes: a boiler process (P1), a turbine process (P2), a water-treatment process (P3), and an HIL simulator (P4). The HIL simulator enhances the correlation between the three real-world processes at the signal level by simulating thermal power generation and a pumped-storage hydropower generator.

The boiler process is a water-to-water heat-transfer process, controlled using four controllers: a water-level controller (LC), pressure controller (PC), temperature controller (TC), and flow-rate controller (FC). The water in the main water tank is pumped and supplied to a heat exchanger, after it is sent to the return water tank at a constant temperature and pressure. The water temperature and pressure values are then converted to the current steam temperature and pressure values in the steam-turbine power generator of the HIL simulator. Finally, the water is returned to the main water tank, thus ensuring a constant water level in the return water tank.

The turbine process is composed of a rotor kit that closely simulates the behavior of an actual rotating machine. It consists of a motor system with a direct motor speed controller (SC) and a rotor system, including a rotor shaft and two balance wheels, that enables coupling. Two pairs of vibration-monitoring proximity probes are used for the trip control. The motor speed remains synchronous with the rotating speed of the thermal power generator model of the HIL simulator.

The water-treatment process is controlled using a level controller (LC) that controls a level control pump (LCP) to pump water to the upper reservoir and a level control valve (LCV) to release water back into the lower reservoir. Information pertaining to the hydraulic pressure, flow rate, and water level of the upper water tank is sent to the HIL simulator in real time to determine the amount of power generated and consumed.

The HIL simulation model consists of two synchronous generator models and a power grid model with an electrical load. The HIL simulator adjusts the set points of P1.TC, P1.PC, and P3.LC to ensure the desired electrical load.

2.2 Unmanned Normal Operation

We developed a tool for an unmanned SCADA operation, which can be operated in a heterogeneous environment based on OPC. This is due to the fact that long-term human intervention in SCADA operation increases costs and makes it difficult to maintain consistency for replicates.

By performing the operation scheduling, it automatically checks whether the feedback controller is stable at the scheduled time and transmits a new SP command within the specified operation range. Here it was necessary to work experimentally to determine an operation range for safe use.

Five controllers (P1.PC, P1.LC, P1.FC, P1.TC, and P3.LC) were automatically operated in the HAI testbed. Five times a day, they start with a random delay, and the SP value is pro-

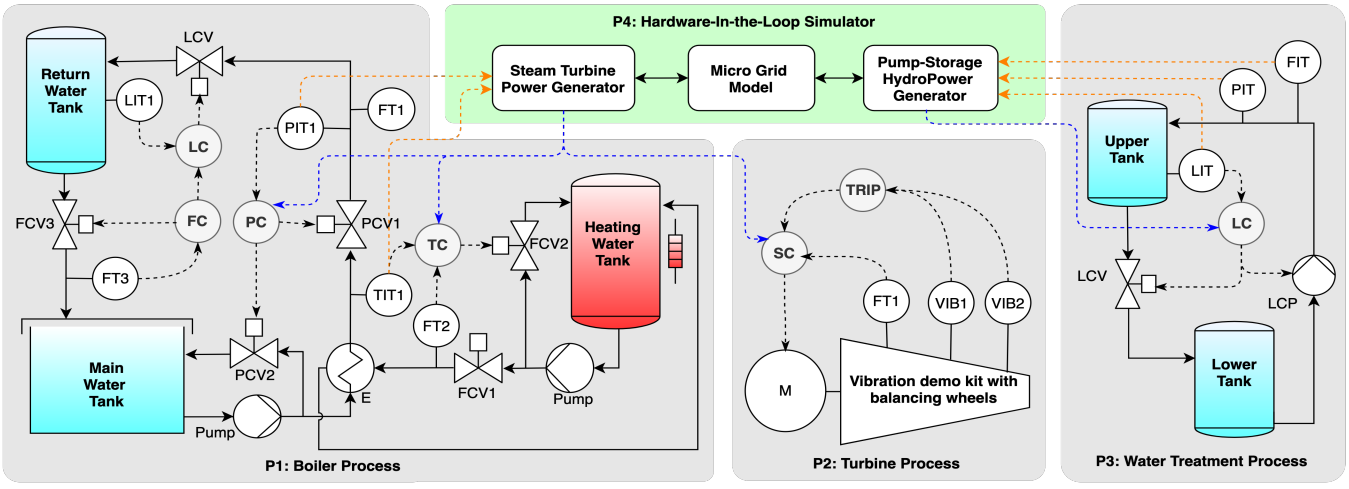

Figure 1: Overall Process Architecture of HAI Testbed

grammed to reach a random value within the operation range. All the set point values are included in the data collection to learn the system response patterns for set point changes.

3 Dataset: HAI 1.0

The HAI dataset 1.0 was built by collecting 59 points every second from the testbed and by labeling 4 attack tags. The normal dataset was continuously collected with no attacks. The attack dataset was collected with 38 attacks combining 14 attack primitives.

3.1 Attack Primitives Based on PCL

In a previous work [10], a new attack approach based on a process control loop (PCL) was proposed by forcing four variables: set point (SP), process variable (PV), control output (CO), and control parameters (CP).

We generalize the attack method for a single PCL as described in Algorithm 1. First, the PV response prevention function can be used to prevent the operator from finding an abnormal response on the HMI during an attack. This function is achieved to replay the recent snapshot of the PV recorded in steady-state condition. In order to change the actuator output, the control algorithm output CO can be directly forced or the SP value can be forced to indirectly change the CO value through the control algorithm such as the PID controller. In the former method, the system output can be easily controlled to the desired value using the SP value. In the latter method, a quick operation can be realized by directly controlling the output of the valve, pump, etc.

Algorithm 1: Pseudo-code for a PCL attack primitive

Input: target control loop, forced variables and values, PV response prevention, time limits
Output: attack logs
Data: the latest PV snapshot at steady state

```
Store the current SP, PV, and CO recovery values
Generate an attack sequence that linearly increases to
  the forced value and decreases to the recovery value
  over time limits for each forced variable
Wait until the target control loop is at steady state
while any attack sequence remains do
    if the PV response prevention is activated then
        Replay the PV using the snapshot data
    if the SP attack sequence remains then
        Force the SP value from the attack sequence
    if the CO attack sequence remains then
        Force the CO value from the attack sequence
    Record all the attack sequences
```

Table 1: PCL attack primitives for HAI testbed

No	Attack ID	Target	SP	CO	PV-RP
1	AP-P1PC-SP	P1.PC	✓		
2	AP-P1PC-SPRP		✓		✓
3	AP-P1PC-CO			✓	
4	AP-P1PC-CORP			✓	✓
5	AP-P1FC-SP	P1.FC	✓		
6	AP-P1FC-SPRP		✓		✓
7	AP-P1LC-SP	P1.LC	✓		
8	AP-P1LC-SPRP		✓		✓
9	AP-P1LC-CO			✓	
10	AP-P1LC-CORP			✓	✓
11	AP-P2SC-SP	P2.SC	✓		
12	AP-P2SC-SPRP		✓		✓
13	AP-P3LC-SP1CO1	P3.LC	$✓_{LH}$	$✓_{LCV}$	
14	AP-P3LC-SP2CO2		$✓_{LL}$	$✓_{LCP}$	

This method is more effective and scalable to realize stealthy ICS attacks for any ICS through the combination of PCL attack primitives for its PCLs. By combining the 14 attack primitives (Table 1) identified for the HAI testbed, we can create an attack situation for a wider range of cases. In fact, it was possible to combine two attacks into a pair to create 19 new attacks in the development of the HAI dataset.

3.2 Attack Scenario

The attack scenarios (Table 2) can be classified based on the following attack characteristics.

- **Number of attack targets:** An attacker can be divided into 18 single attacks (1∼13, 16, 22, 25, 27, 33) that only perform one attack primitive (Table 1) and 21 multiple attacks (14, 15, 17∼21, 23, 24, 26, 28∼32, 32∼36, 38) with two or more combinations.

- **PV response prevention:** An attacker can hide their attack by covering up the PV response because the PV is the fundamental measurement to monitor the current operating condition. We conducted a total of 15 stealthy attacks (1, 3, 4, 7, 9, 10, 14, 16, 17, 20, 23, 24, 26∼28).

- **SP attack:** An attacker can change the SP and then naturally manipulate the PV as desired. The controller automatically adjusts the CO until the relevant PV reaches the SP when an operator changes set-point. 31 SP attacks (1, 5∼7, 10∼36) were conducted.

- **CO attack:** An attacker can directly control the actuators by changing the CO values. This attack can cause malfunction in actuators and disrupt process production. 21 attacks (2∼4, 8, 9, 11, 12, 15, 17, 19∼22, 24, 28, 30∼32, 34, 37, 38) were carried out.

Table 2: Attack scenarios combining the 14 attack primitives.

No	Attack IDs	Start Time	Duration
1	AP-P1LC-SPRP	19-10-29 13:40	370 secs
2	AP-P1LC-CO	19-10-29 14:35	312 secs
3	AP-P1LC-CORP	19-10-29 15:45	868 secs
4	AP-P1FC-CORP	19-10-29 16:30	262 secs
5	AP-P1LC-SP	19-10-30 08:50	371 secs
6	AP-P1PC-SP	19-10-30 09:40	334 secs
7	AP-P1PC-SPRP	19-10-30 10:35	504 secs
8	AP-P1PC-CO	19-10-30 11:37	268 secs
9	AP-P1PC-CORP	19-10-30 12:30	518 secs
10	AP-P2SC-SPRP	19-10-30 14:30	370 secs
11	AP-P3LC-SP2CO2	19-10-30 15:35	180 secs
12	AP-P3LC-SP1CO1	19-10-30 16:33	154 secs
13	AP-P2SC-SP	19-10-31 08:42	348 secs
14	AP-(P1PC-SPRP, P2SC-SPRP)	19-10-31 10:30	518 secs
15	AP-(P1PC-CO, P2SC-SP)	19-10-31 11:33	346 secs
16	AP-P2SC-SPRP [Repeat No.11]	19-10-31 13:25	368 secs
17	AP-(P1LC-CORP, P2SC-SPRP)	19-10-31 14:30	396 secs
18	AP-(P1FC-SP, P2SC-SP)	19-10-31 15:41	348 secs
19	AP-(P1PC-SP, P3LC-SP1CO1)	19-10-31 16:30	398 secs
20	AP-(P1LC-SPRP, P3LC-SP1CO1)	19-11-01 09:29	560 secs
21	AP-(P1LC-CO, P3LC-SP1CO1)	19-11-01 10:41	310 secs
22	AP-P3LC-SP1CO1 [Repeat No.12]	19-11-01 11:23	180 secs
23	AP-(P1FC-SPRP, P1LC-SP)	19-11-01 12:31	506 secs
24	AP-(P1PC-CO, P1FC-SPRP)	19-11-01 13:41	580 secs
25	AP-P1PC-SP [Repeat No.6]	19-11-01 14:23	310 secs
26	AP-(P1FC-SP, P1PC-SPRP)	19-11-01 15:31	520 secs
27	AP-P1FC-SPRP	19-11-01 16:18	560 secs
28	AP-(P1PC-SPRP, P3LC-SP2CO2)	19-11-01 17:20	520 secs
29	AP-(P1FC-SP, P1PC-SP)	19-11-04 15:31	410 secs
30	AP-(P1PC-SP, P3LC-SP2CO2)	19-11-04 17:20	520 secs
31	AP-(P1LC-CO, P3LC-SP2CO2)	19-11-05 09:30	380 secs
32	AP-(P1FC-SP, P3LC-SP2CO2)	19-11-05 10:20	290 secs
33	AP-P2SC-SP [Repeat No.13]	19-11-05 11:23	340 secs
34	AP-(P2SC-SP, P3LC-SP2CO2)	19-11-05 12:30	340 secs
35	AP-(P1LC-SP, P2SC-SP)	19-11-05 14:45	2880 secs
36	AP-(P1PC-SP, P1LC-SP)	19-11-05 16:20	330 secs
37	AP-P1LC-CO [Repeat No.2]	19-11-05 17:23	310 secs
38	AP-(P1PC-CO, P1LC-CO)	19-11-06 08:58	310 secs

3.3 Data Collection

The HAI 1.0 dataset was collected over time in the order of normal (approximately 7 days), attack (approximately 4 days, 28 attacks), normal (approximately 3 days), and attack (approximately 1.5 days, 10 attacks) situations. Table 2 lists the execution times of all the attack scenarios. The datasets are presented in four CVS files separately for two sets of the normal and attack situations.

The data are listed in 63 columns. The first column represents the local time, whereas the remaining 59 columns show the recordings of the SCADA points representing the variables measured or controlled by the control system. The last four columns are the attack labels, where a nonzero value indicates an attack. Here, the first attack label indicates that an attack has occurred for the entire process, whereas the remaining columns are for the corresponding processes.

3.4 Performance Metric for Anomaly Detection in Time-Series data

We recommend the eTaPR [7] to evaluate the detection performance with HAI 1.0. The conventional precision and recall are instance based metrics, widely employed to evaluate anomaly detection methods applied to time-series data. However, the instance based metrics tend to overlook these characteristics, thus suffering from the problem of producing a high recall to a method that detects long anomalies. To overcome this issue, the time-series aware precision and recall (TaPR)[1] [8] has been proposed for precision and recall metrics tailored for time-series and anomaly detection. Recently, an enhanced version of TaPR (eTaPR) has been disclosed [7].

4 Conclusion and Future Work

HAI 1.0 is the first ICS dataset developed using the HAI testbed. Our future plans are as follows.

- Improving the data labeling: The labels of time-series data must include all the effects of attack on the physical system. If the controller output is forced directly or indirectly by an arbitrary attack, the system response should stabilize to a steady state through a transient state condition. Therefore, if it was in a steady state before being attacked, it should be considered an attack until the transient state is restored to the steady-state. The transient-state section can be automatically extracted by analyzing the tracking error for each PCL to provide additional attack effects on the physical system.

- HAI dataset: We will launch a new dataset by changing the HIL logic, sensors, and actuators to simulate another system. The HAI testbed can easily reconstruct another complex process of actual controllers using the HIL. HAI datasets can be used to check how quickly detection methods that are optimized to other site.

References

[1] Seungoh Choi, Jeong-Han Yun, and Sin-Kyu Kim. A comparison of ics datasets for security research based on attack paths. In *The 13th International Conference on Critical Information Infrastructures Security (CRITIS)*, 2018.

[2] Wenbo Ding and Hongxin Hu. On the safety of iot device physical interaction control. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, pages 832–846, 2018.

[1] https://github.com/saurf4ng/TaPR

[3] Pavel Filonov, Fedor Kitashov, and Andrey Lavrentyev. RNN-based Early Cyber-Attack Detection for the Tennessee Eastman Process. *CoRR*, September 2017.

[4] Jairo Giraldo, David Urbina, Alvaro Cardenas, Junia Valente, Mustafa Faisal, Justin Ruths, Nils Ole Tippenhauer, Henrik Sandberg, and Richard Candell. A survey of physics-based attack detection in cyber-physical systems. *ACM Computing Surveys (CSUR)*, 51(4):76, 2018.

[5] Jonathan Goh, Sridhar Adepu, Khurum Nazir Junejo, and Aditya Mathur. A dataset to support research in the design of secure water treatment systems. In *The 11th International Conference on Critical Information Infrastructures Security (CRITIS)*, 2016.

[6] Jonathan Goh, Sridhar Adepu, Marcus Tan, and Zi Shan Lee. Anomaly detection in cyber physical systems using recurrent neural networks. In *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*, pages 140–145. IEEE, 2017.

[7] Won-Seok Hwang, Jeong-Han Yun, Jonguk Kim, and Hyoung Chun Kim. Time-series aware precision and recall for anomaly detection: Enhanced metrics addressing the antinomy, obscurity, and inflexibility. In *The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD), submitted*.

[8] Won-Seok Hwang, Jeong-Han Yun, Jonguk Kim, and Hyoung Chun Kim. Time-series aware precision and recall for anomaly detection: Considering variety of detection result and addressing ambiguous labeling. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2241–2244. ACM, 2019.

[9] Jonguk Kim, Jeong-Han Yun, and Hyoung Chun Kim. Anomaly detection for industrial control systems using sequence-to-sequence neural networks. 5th ESORICS Workshop on the Security of Industrial Control Systems of Cyber-Physical Systems (CyberICPS), 2019.

[10] Hyeok-Ki Shin, Woomyo Lee, Jeong-Han Yun, and HyoungChun Kim. Implementation of programmable CPS testbed for anomaly detection. In *12th USENIX Workshop on Cyber Security Experimentation and Test (CSET 19)*, Santa Clara, CA, August 2019. USENIX Association.