Preparing Industry for IIoT: Separate Sensor Networks for Industrial Automation Security

Ricky E. Green Jr, Ryan Benton and Michael Black
University of South Alabama, Mobile USA

rgreen@southalabama.edu
rbenton@southalabama.edu
mblack@southalabama.edu

Abstract: Industrial automation has brought innovations in efficiency and safety to modern manufacturing. The emergence of big industrial data, AI and Internet-connected industrial automation systems promises many advantages in manufacturing, maintenance, health & safety, product customization, logistics, and reporting. The Industrial Internet of Things (IIoT) seeks to attach industrial systems to the Internet for data collection and processing. The Internet is not a secure place for many of the industrial systems in use today. Industrial networks are designed for speed, accuracy, and availability at the expense of security. In our experiments, we demonstrate how easy it would be for hostile actors to gather data that could be used to compromise the industrial automation systems if actors (a) had physical access to the internal network or (b) access to data streams that feed to the external network. To resolve some of these issues, we propose a hybrid solution of sensor data acquisition in which a secondary network with additional sensors and controllers are installed; the secondary network, sensors and controllers are isolated from the internal control system. The results show that our solution enables useful data collection without impacting the security, speed, accuracy, and availability of industrial control system. It also has the potential to permit anomaly detection using independent sensors on different networks.

Keywords: Sensor networks, Manufacturing, Industry 4.0, Cyber-physical systems, Industrial automation

1. Introduction

Modern industrial manufacturing systems consist of many physical components that perform work to create products. Since the first industrial revolution began replacing people's labour with machines, technological innovations have slowly made their way into manufacturing (Lukač, 2015). The third industrial revolution introduced computing for machine automation, enabling machines to run continuously without human interaction (Adeyeri, 2018). Programmable Logic Controllers (PLCs), considered the heart of these systems, are configured to automatically respond to sensors on the machines and make adjustments to control the process (E. A. Lee & Seshia, 2016). Sensors themselves are read-only devices that measure an analogue or digital value. These values are converted to an electrical signal, amplified, and sent to a controller. Supervisory Control and Data Acquisition nodes (SCADA), collect data from the controllers and save them for trending analysis while providing supervisory control to the processes being controlled by the PLCs (Daneels & Salter, 1999).

Recent research in Industry 4.0 and the Industrial Internet of Things (IIoT) proposes connecting smart automation system data collection devices, called cyber-physical systems, to the Internet (Jazdi, 2014; Lennvall et al., 2017; Sengupta et al., 2020). Research in the proposed Industry 5.0 revolution extends the previous revolution but with more of a focus on sustainability and enhancing human interaction (Demir et al., 2019; Nahavandi, 2019). Traditional automation systems in manufacturing plants were built for dependability, speed, and accuracy at the expense of security, so these devices historically remained attached to an isolated network with minimal services to limit exposure to external threats (Hong & Lee, 2008; Risley et al., 2003). Additionally, automation systems require a specifically designed low-latency network infrastructure for performance and stability (Yan et al., 2021). Adding sensors to this network adds cost and latency, so data collected from them is prioritized for operations and business needs based on value (I. Lee & Lee, 2015).

An automation system's reliability and safety depend on its components' availability. Availability in manufacturing automation systems requires feedback and feedforward methods, including performance monitoring, corrective maintenance, and lifecycle management (Ton et al., 2008). These methods require data, including data collected from the manufacturing system sensors that report the condition of the components.

While many sensors in the manufacturing system are required to give feedback to the manufacturing system for automatic control, some sensors are not used for input into the manufacturing system. Rather than feedback control functions, these sensors collect valuable information for higher-level functions of the

organization, such as predictive maintenance, quality reporting, logistics, system health and performance, environmental compliance, and auditing (I. Lee & Lee, 2015). Since these sensors do not provide feedback control, they need not be connected to the expensive low-latency infrastructure, PLC or SCADA for management. Hence, many of these additional sensors can use more cost-effective non-industrial-grade networking infrastructure, and their systems can be physically isolated from the rest of the manufacturing control systems for security and scalability.

Sensors are read-only measuring devices, but components of the automation control network that rely on the values from these sensors are not. PLCs use sensor values to influence the output of a device in the system. The controller part of a PLC sends signals to energize or make controlled changes to mechanical devices based on the values received from sensors and the logic programmed into them. SCADA systems collect and store data from these PLCs and can write values to the memory of the PLCs to affect changes in the manufacturing process. This process of continuous sensor reading and adjustment is called a feedback loop. The PLC and SCADA systems, which are not read-only devices, are vulnerable to cyber-attacks and must be protected. Advancements in IIoT and cloud connectivity will require changes in manufacturing system configuration to minimize the risk of compromising system security.

In this paper, we conduct an experiment that adds additional sensors to an existing industrial control system for data collection, isolating these sensors from the current feedback control sensors. The data collected from the additional sensors is recorded into a separate system, which is then analysed to gain alternative insights that were unobtainable without the additional sensor data. The data was then compared with the internal sensor data to gain insights into the operation of the industrial control system devices. The experiment concludes by validating that the any malfunction with the additional sensors will not reduce the availability of the existing control system.

The remainder of this paper is organized as follows. Section two presents the background in industrial automation and introduces the concepts in IIoT. Section three presents our methodology. Section four presents our experiment of modifying an isolated cyber-physical network by adding a separate sensor network developed for this research. The results of the experiments are presented in Section five. Section six discusses and concludes this research.

2. Background

2.1 Adding Automation in the Industrial Revolution

Beginning in the early 1970s, the third industrial revolution introduced electronic automation systems to the mechanized processes provided by the first two industrial revolutions (Adeyeri, 2018). Automation was designed to make decisions and monitor processes with little human intervention. These systems increased efficiency by removing the human element from repetitive tasks and replacing them with autonomous machines that could run continuously. At the most basic level, an automation system consists of 4 physical components (Abbate, 2000; Roberts, 1978; Ton et al., 2008):

- A device that causes something to happen, such as a motor that drives movement, an actuator that opens a valve, or a heating element that causes a change in temperature
- A sensor that measures the changes that are happening to the device, such as a flow meter, temperature sensor, or limit switch
- A controller device that collects the sensor information, calculates a measured value and generates a response that loops back to the device.
- Infrastructure to connect the physical components

Combined, these physical elements create a process control loop at the heart of industrial automation.

2.2 Industry 5.0

A fifth industrial revolution promises to add additional intelligence to the processes and increase the synergy between humans and machines. Where Industry 4.0 sought to automate the process to eliminate as much human interaction as possible, Industry 5.0 aims to have tasks requiring high levels of creativity be performed by a human, leaving the remaining tasks to automation systems better suited for them (Sehrawat & Gill, 2019; Weis, 2010). The automation systems of Industry 5.0 also seek to increase reliance on AI, big data, and cloud computing to support new functions, such as additive manufacturing, predictive maintenance, and intelligent product customization. These additions will require an increase in the number and type of sensors collecting

data, contributing to a greater volume, velocity, and variety of data (Lennvall et al., 2017; Sehrawat & Gill, 2019).

2.3 Industrializing the Internet of Things

Traditional sensors in industrial automation systems generate data to provide feedback to the automation system and record data for trending and quality control purposes (Boyes et al., 2018). These sensors are expensive to implement because each sensor requires an input channel on a PLC, infrastructure for the connection to the PLC, engineering design and development of the PLC program to implement the feedback logic, a SCADA license for control and data collection, and supporting computer hardware and software. A cost-effective industrial control system collects only the data required to run the industrial processes and produces the needed quality analysis and regulatory information.

Industrial automation control systems were designed for low-overhead, low-latency operations. To reduce the risk of external influence on internal control systems, protection of the control network is implemented at the perimeter, which typically follows a standard security model such as the popular Purdue Enterprise Reference Architecture (Boyes et al., 2018; Williams, 1994). This model segments networks into zones separated by restrictive firewalls to create an isolated control network that carefully controls access to this network. While this protects the network from outside influence, it complicates expansion and serviceability when new technology replaces failed or outdated systems.

Where the Internet of Things sought to use sensors to detect the movement of objects, the Industrial Internet of Things (IIoT) aims to implement the scalability and logistical benefits into the industrialized business setting. Since these types of sensors do not need to be industrialized, connected to a PLC, or be required to provide feedback for automation and control, they are a candidate for separation from the industrial control system. This separate sensor and data collection system can be configured to eliminate interference with the existing feedback control sensors and their associated data collection systems.

3. Methodology

For our research, an experiment was conducted that analyses the security issues when exposing an industrial control system network to an external, uncontrolled network. The results from this experiment were used to conduct a second experiment that uses a separate, read-only sensor network attached to the external, uncontrolled network, leaving the industrial control system network secured in its closed network.

In the first experiment, a second computer on the control network was used to conduct a series of network scans and packet captures to view the raw traffic on the control network. This computer will represent a threat actor that has gained access to the control system network. With this experiment, the scans were used to map devices and discover points of weakness in the system.

A second experiment was conducted that isolates the control network from the uncontrolled network. Existing industrial devices were configured with a secondary read-only sensor connected to a separate network that can be safely exposed to an external, uncontrolled network for data collection. This experiment consisted of four parts.

In part one of the second experiment, data was collected using the existing sensors, PLCs, and SCADA systems on the isolated network already in place. A pre-set task was executed on the control system to generate feedback data from the PLC, and the data was stored on the SCADA computer. This data was used as control data and compared with data collected by the sensors on the external network in part three.

In part two of the second experiment, additional sensors were installed on the controlled devices to collect information from the same device. These additional sensors were then connected to a separate computer to read their signals. This computer does not provide any output control to any devices. The pre-set task was again executed on the control system to generate feedback data from the PLC and stored on the SCADA. The data collected from this task was compared to the control data from part one.

In part three of the second experiment, data collected from the separate sensor system was examined and compared to the control data. Although the sensors are designed to collect the same information as the remote sensors, the sensor types were different so the values did not match exactly. The parameters for the new data set were adjusted to compensate for these differences, then compared with the data collected from part one.

A threat actor was introduced to the uncontrolled network in part four of the second experiment. The same threat actor computer used in experiment one was attached to the uncontrolled network with the secondary, read-only sensors. The data was then analysed and compared the results to the packets collected in experiment one.

4. Experiment

For our experiments, we used a power plant simulator training system that contains an Allen Bradley CompactLogix PLC with input/output cards in a rolling cabinet, shown in Figure 1. On top of the cabinet is a diorama of a small town designed to emulate a power plant and distribution of substations, shown in Figure 2 (next page).

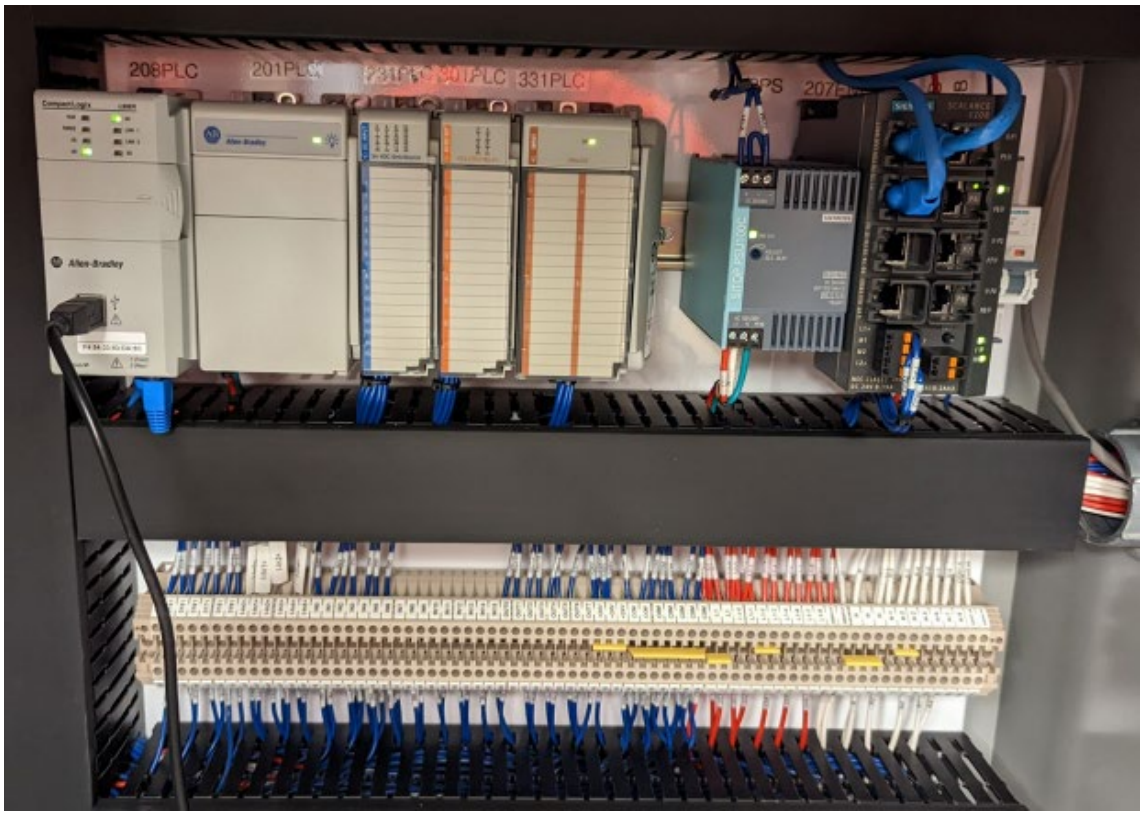

Figure 1: Allen Bradley CompactLogix PLC chassis

Next to each substation is a red and green light emitting diode (LED) and an analogue meter. The LEDs represent the status of the substation as either energized or not energized, and the analogue meters represent the power load at each substation. A variable resistor next to the power plant is used for input and represents the power load placed on the power generation plant. The LEDs are attached to the PLC digital output card. The analogue meters are connected to the PLC analogue output cards to simulate a change in the load of each substation. The variable resistor is attached to the PLC input card.

The PLC is connected via Ethernet to a Seimens SCALANCE X208 industrial network switch connected to a Windows 10 PC. This PC runs Allen Bradley ControlLogix software for programming the PLC and includes a Human-Machine Interface (HMI) program and a SCADA development and runtime environment. The PLC and the Windows PC are the only devices attached to this network switch. The Windows PC is not connected to any other network, and all devices on the industrial switch remained isolated on their network during the experiment.

A relay was manually activated in the control system software running on the SCADA computer to execute the task on the power plant simulator. The power plant simulator represents this by turning on the green LED light next to the substation. The circuits are designed to turn off the red LED when the green LED is turned on. This indicates the simulated substation has been activated.

Figure 2: Power plant simulator

Experiment one, illustrated in Figure 3, used a second computer attached to the Seimens network switch to act as the threat actor that has gained access to an internal computer on the control network. A standard laptop running Kali Linux was used. This laptop was connected to the Seimens network switch and was not attached to any other network during this experiment. A packet capture of the control network switch was performed using Wireshark to examine the communication packets between the PLC and the SCADA computer. The substation switchover task on the power plant simulator was executed and recorded the data on the SCADA machine and the packet capture from Wireshark.

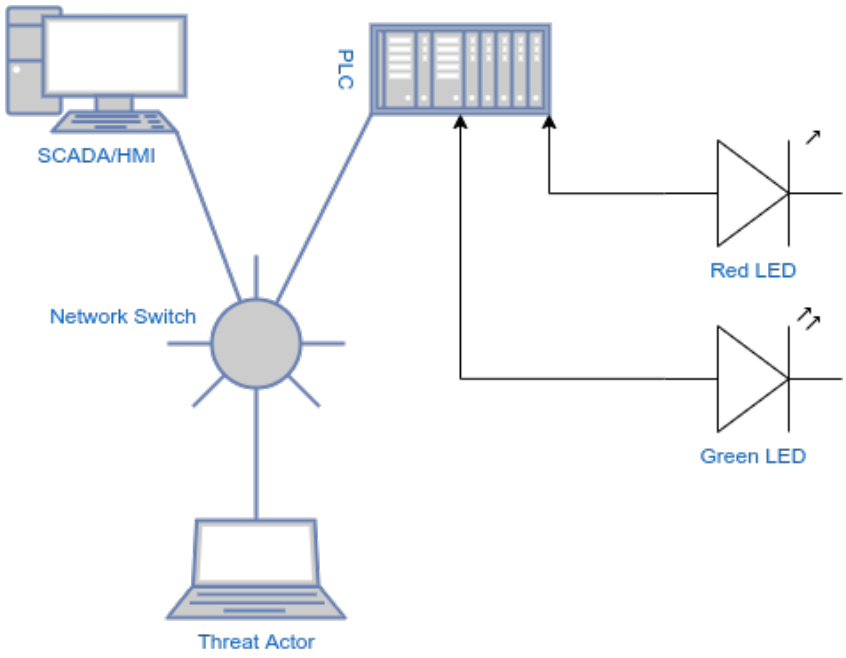

Figure 3: Configuration for experiment one

The packet capture data of the conversation between the SCADA computer and the PLC was analysed. The data included packets that carried information from the PLC to be written to the SCADA database, as well as

commands from the HMI to the PLC to activate the relay to trigger the execution of the substation switchover task on the power plant simulator. This concluded experiment one.

A new data sample was first taken from the PLC and SCADA system for experiment two. The substation switchover task was executed, and the time was manually recorded while the data was collected on the SCADA. The data collected was verified and matched to the manually recorded timeframe. This concluded part one of experiment two.

In preparation for part two of the second experiment, the threat actor was removed from the internal network. The separate sensor network was prepared by adding external sensors to the power plant simulator. The external sensors consist of photoresistors, which will detect the switchover of a substation by registering a change in the light intensity of the LEDs on the diorama. Since the LEDs are acting as a digital device on the PLC to represent on and off, and the photoresistors are an analogue device, it will be necessary to adjust the parameters of the output data of the photoresistors to compensate for the differences in the digital data of the LED's when comparing the on and off values to analogue integer values.

Two photoresistors were placed in proximity to the LEDs of a substation, as shown in Figure 4. The photoresistors were attached to the analogue input ports of an Arduino microcontroller, as shown in Figure 5. The Arduino was programmed to detect variations in light intensity and write them as an integer value to the serial buffer every half second. The Arduino was able to detect the brightness and record it as an integer value.

The Arduino controller collects data but cannot store it, so it was connected to a Raspberry Pi 400 via USB to store the collected data. A simple Python program was developed for the Raspberry Pi to collect the data from the Arduino every second, adding a timestamp field and appending the data to a CSV file on the micro-SD card inserted into the Raspberry Pi. The Arduino performs a function similar to the PLC. The Raspberry Pi in this case performs a function similar to a SCADA Historian database.

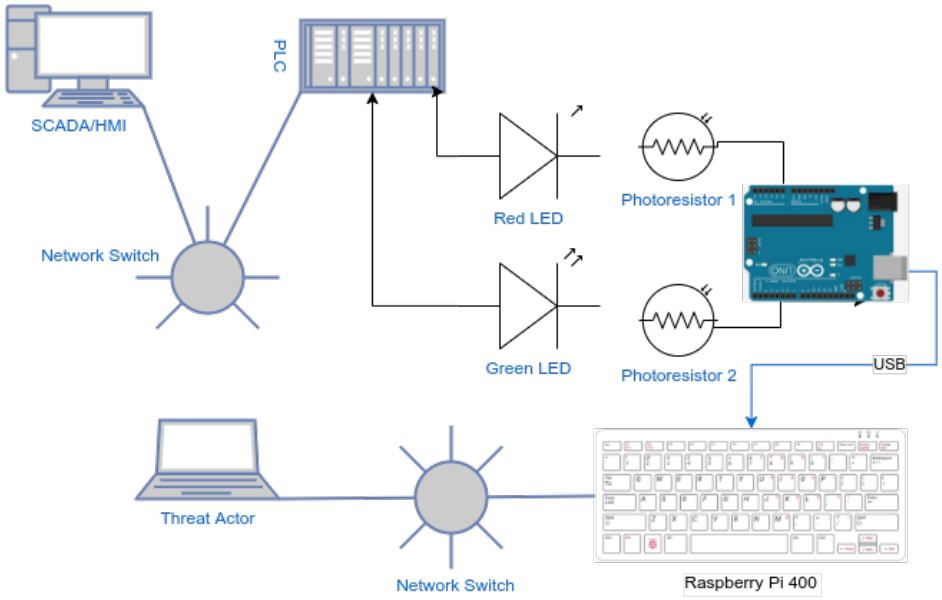

Figure 4: Configuration for experiment two

The Arduino was powered on, and the Raspberry Pi was booted up in preparation for part two of the experiment. Once the Python application was started and connected to the Arduino, the same substation switchover task from experiment one was executed and the time was manually recorded. The still visible LEDs made their change to indicate a successful switchover. The Python program was stopped after 300 seconds, concluding the test. The data collected from the Arduino was saved to a CSV file on the Raspberry Pi, and a quick check was performed to verify there was data in the file. This concluded in part two of the second experiment.

Part three of the second experiment required the data set collected from the SCADA computer in experiment one and the data set collected from the Raspberry Pi in experiment two. The data sets were exported as CSV

files and saved to a flash drive for analysis on a separate machine. The Results section presents the findings of the comparison of the data sets. This concluded part three of the second experiment.

In part four of the second experiment, we reintroduced the threat actor to the additional sensor network to determine if the threat actor would be able to carry out the same kind of attacks on both networks. The same laptop used in experiment one was connected to the same network as the Raspberry Pi and began capturing packets with Wireshark. The Python program was executed to collect the external sensor data, and the same substation switchover task was executed. The results section presents the findings of the network data that was collected. This concluded part four of the second experiment.

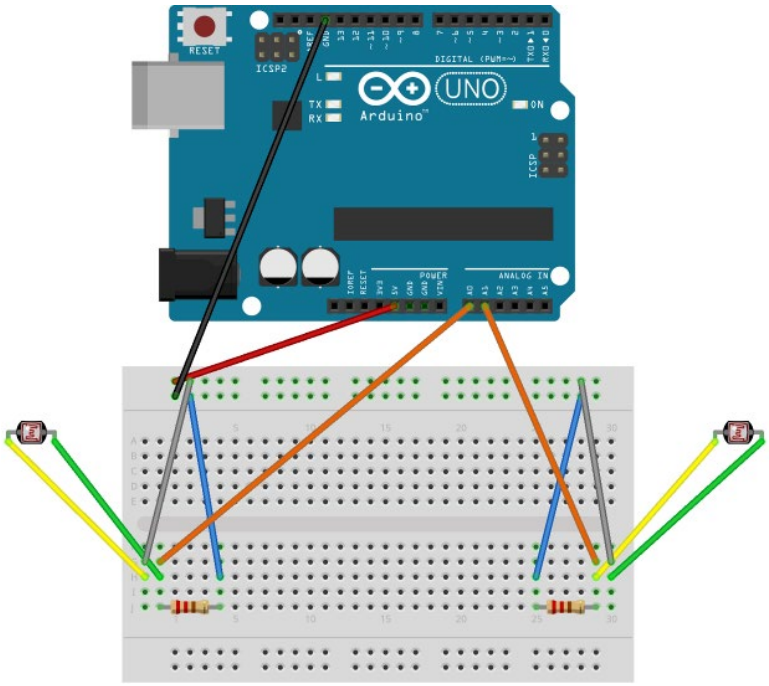

Figure 5: Arduino with photoresistors

5. Results

Experiment one was a controlled experiment to analyse the network traffic of a control system to understand the weaknesses of the system. A threat actor with access to this level of the control system network would be able to carry out attacks that could cause systems to go offline through a denial of service attack or cause substantial damage to any cyber-physical devices by intercepting packets between the PLC, HMI, and SCADA systems and modifying the commands or generating new ones (Wardak et al., 2016).

In experiment one, data was collected from the PLC during the execution of a task that caused a red LED to turn off and a Green LED to turn on. The Wireshark capture of the network packets displayed the data collected from the event. The packets sent from the SCADA/HMI computer to the PLC showed the event request to perform the switchover to the substation in the power plant simulator. The packets sent from the PLC to the SCADA/HMI computer showed the response to the request so the HMI could be updated, and the SCADA recorded the change in the database. The date and time manually recorded at the beginning of the event matched the data collected from the event and stored in the SCADA database. This created our control database for comparison.

For experiment two, the control system network remained isolated as it was designed. Additional sensors were added to the devices to create a separate sensor network. The HMI and data collection to the SCADA PC remained unaffected by the addition of the sensors. The LEDs in the diorama remained visible since the photosensors did not block their view but were only in proximity.

The Raspberry Pi and Arduino successfully collected data from the photoresistors in the power station diorama and saved the data to a local CSV file. The CSV file contains 301 timestamped records that were recorded over 300 seconds.

Since the photoresistors are analogue sensors that send an integer value based on the light intensity, the values recorded to the CSV file had to be tuned for each photoresistor to determine an LED's digital on or off event in the diorama. In Figure 6, the blue line labelled A1 is the analogue input of the photoresistor for the Green LED, and the digital output equivalent for the same Green LED is the yellow line labelled D1. Likewise, the red line labelled A2 and the green line labelled D2 are the analogue and digital outputs for the Red LED, respectively. The LEDs in the diorama were not shining at the same colour or intensity. Also, ambient light around the LEDs could change intensity based on the surroundings, such as the status of the lights in the lab or the shades on the windows being open during daylight hours. The Y-axis values of Figure 6 account for this tuning, with a 0 representing a digital "off" value and 900 representing the digital "on" value to accommodate the variability in the light coming from the analogue photo resistors. The highest analogue value of A2 peaked at 880 from the Arduino. This tuning of the data parameters was also performed manually through several lighting condition experiments. Once the values were tuned to determine the proper integer ranges for the light intensity of an on-and-off condition, a formula was added to LibreOffice Calc for two additional columns to determine the on-and-off status of the LEDs. After tuning, the data reliably showed the timeframes of the LEDs turning on and off compared to the observations made during experiment two.

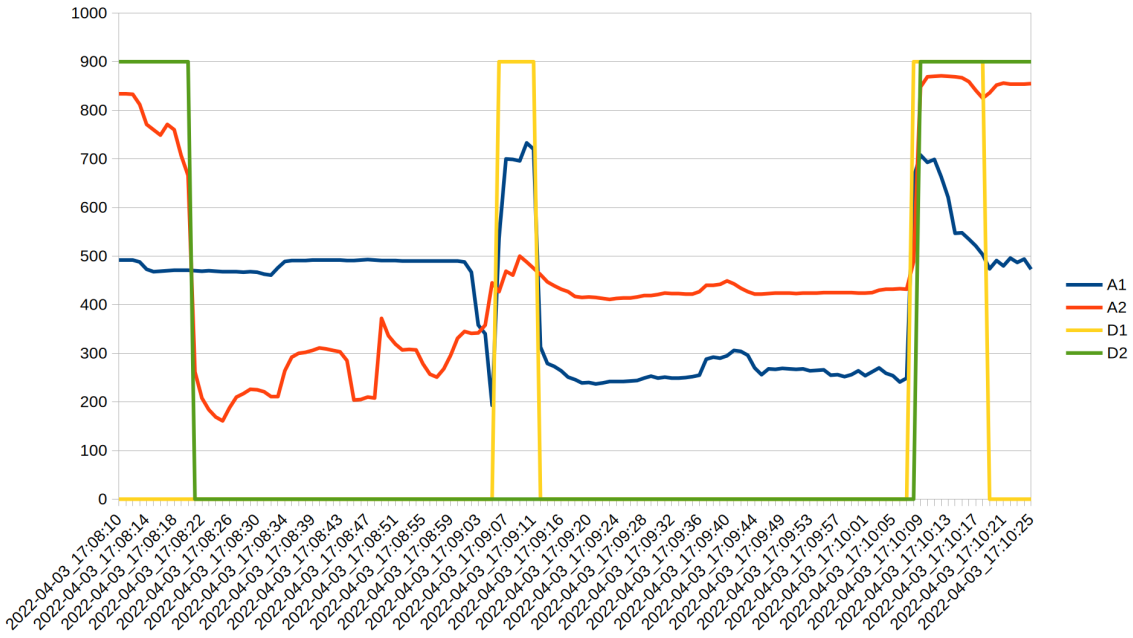

Figure 6: Data collection and results converted to digital output

After the analogue data parameters from the external sensor network were tuned, the data was compared to the binary values of the internal sensor network. The values matched and were time synchronized with the recorded date-time stamps and the manually recorded time. The results shown in Figure 6 indicate that the external sensor system can record sensor data reliably while remaining independent of the internal sensor system.

The additional step of exposing the threat actor to the secondary sensor network did not disclose any internal or external sensor data via a Wireshark capture due to the way the devices were connected and used. The threat actor at this level could modify data or cause denial of service, but this would not affect the PLC operation and would be detectable when comparing the data between the two sensors. However, this experiment was not complete since the data saved to the Raspberry Pi was not transmitted over the network. In production environments, the data would likely not remain on the edge device, such as this Raspberry Pi, but would be processed and sent to another storage location, such as a cloud server. In this case, the threat agent could capture the sensor data from the external sensor network. Although the data collected from the external sensors is a representative facsimile of the internal sensors, they remain physically isolated, which is the point of this experiment. A threat actor would not be able to affect devices or data generated from the internal sensor network.

6. Conclusion and Perspectives

While in the initial stages, this research completes one step for future research in implementing IIoT in manufacturing that current research still needs to identify. Separating sensors on the same networking equipment using Virtual Local Area Networks (VLANs) to reduce latency and group network devices is not unusual. Still, this experiment uses physically isolated networks to separate the PLC feedback data from non-critical data-collecting sensors.

Manufacturing plants that follow the Purdue Enterprise Reference Architecture model for securing the sensor networks against external attacks may find this hybrid solution an acceptable alternative to opening their sensor network for outbound Internet connectivity. Their sensitive sensor networks remain in managed isolation, and they can take advantage of the benefits of IoT sensors with no feedback capability to the automation system. Should this exposed sensor network be compromised or taken offline, it will not affect the operation of the critical internal systems.

Future research will explore using the extra sensor data to validate the data from internal sensors used in the feedback loop. The Stuxnet worm caused PLCs to perform a destructive physical operation on manufacturing devices, then lie to the SCADA system to cover up what was happening (Kriaa et al., 2012). Had a system like our proposed system been in place, data from internal sensors could be compared to data from external sensors using anomaly detection algorithms and alert someone to the discrepancy. Since the two sensor networks are physically separated and could consist of different types of components, the attacker would have to compromise both systems for the attack to be hidden successfully.

Ethics Declaration: Ethical clearance for this research was not required.

AI Declaration: No AI of any kind was used in the creation of this paper.

References

Abbate, J. (2000). Inventing the Internet. MIT Press.

Adeyeri, M. (2018). From Industry 3.0 to Industry 4.0: Smart Predictive Maintenance System as Platform for Leveraging. Arctic, 71, 64–81.

Boyes, H., Hallaq, B., Cunningham, J., & Watson, T. (2018). The industrial Internet of things (IIoT): An analysis framework. Computers in Industry, 101, 1–12. https://doi.org/10.1016/j.compind.2018.04.015

Daneels, A., & Salter, W. (1999). What is SCADA? International Conference on Accelerator and Large Experimental Physics Control Systems. International Conference on Accelerator and Large Experimental Physics Control Systems, Trieste, Italy.

Demir, K. A., Döven, G., & Sezen, B. (2019). Industry 5.0 and Human-Robot Co-working. Procedia Computer Science, 158, 688–695. https://doi.org/10.1016/j.procs.2019.09.104

Hong, S., & Lee, S. (2008). Challenges and Perspectives in Security Measures for the SCADA System. In *Proc. 5th Myongji-Tsinghua University Joint Seminar on Protection & Automation.*

Jazdi, N. (2014). Cyber physical systems in the context of Industry 4.0. 2014 IEEE International Conference on Automation, Quality and Testing, Robotics, 1–4. https://doi.org/10.1109/AQTR.2014.6857843

Lee, E. A., & Seshia, S. A. (2016). Introduction to Embedded Systems, Second Edition: A Cyber-Physical Systems Approach. MIT Press.

Lee, I., & Lee, K. (2015). The Internet of Things (IoT): Applications, investments, and challenges for enterprises. Business Horizons, 58(4), 431–440. https://doi.org/10.1016/j.bushor.2015.03.008

Lennvall, T., Gidlund, M., & Åkerberg, J. (2017). Challenges when bringing IoT into industrial automation. 2017 IEEE AFRICON, 905–910. https://doi.org/10.1109/AFRCON.2017.8095602

Lukač, D. (2015). The fourth ICT-based industrial revolution "Industry 4.0"—HMI and the case of CAE/CAD innovation with EPLAN P8. 2015 23rd Telecommunications Forum Telfor (TELFOR), 835–838. https://doi.org/10.1109/TELFOR.2015.7377595

Nahavandi, S. (2019). Industry 5.0—A Human-Centric Solution. Sustainability, 11(16), Article 16. https://doi.org/10.3390/su11164371

Risley, A., Roberts, J., & LaDow, P. (2003, April 1). Electronic Security of Real-Time Protection and SCADA Communications. Fifth annual western power delivery automation conference, Spokane, Washington.

Roberts, L. G. (1978). The evolution of packet switching. Proceedings of the IEEE, 66(11), 1307–1313. https://doi.org/10.1109/PROC.1978.11141

Sehrawat, D., & Gill, N. S. (2019). Smart Sensors: Analysis of Different Types of IoT Sensors. 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 523–528. https://doi.org/10.1109/ICOEI.2019.8862778

Sengupta, J., Ruj, S., & Das Bit, S. (2020). A Comprehensive Survey on Attacks, Security Issues and Blockchain Solutions for IoT and IIoT. Journal of Network and Computer Applications, 149, 102481. https://doi.org/10.1016/j.jnca.2019.102481

Ton, G., Iliescu, M., & Ton, D. G. (2008). A Methodology of Availability Assessment for Complex Manufacturing Systems. 7(6), 12.

Wardak, H., Zhioua, S., & Almulhem, A. (2016). PLC access control: A security analysis. 2016 World Congress on Industrial Control Systems Security (WCICSS), 1–6. https://doi.org/10.1109/WCICSS.2016.7882935

Weis, A. H. (2010). Commercialization of the Internet. Internet Research, 20(4), 420–435. https://doi.org/10.1108/10662241011059453

Williams, T. J. (1994). The Purdue enterprise reference architecture. Computers in Industry, 24(2), 141–158. https://doi.org/10.1016/0166-3615(94)90017-5

Yan, B., Liu, Q., Shen, J., Liang, D., Zhao, B., & Ouyang, L. (2021). A survey of low-latency transmission strategies in software defined networking. Computer Science Review, 40, 100386. https://doi.org/10.1016/j.cosrev.2021.100386