International Journal of Innovative Computing, Information and Control
Volume **22**, Number **3**, June **2026**

ICIC International ©2026 ISSN 1349-4198
pp. **723–736**

AN OPEN-SOURCE INTRUSION DETECTION AND NOTIFICATION FRAMEWORK FOR OT ATTACKS TARGETING S7-SERIES PLCS

Krit Smerpitak[1], Chayangkun Saleethong[1], Tanawat Suphapod[1]
Siripong Wongkharn[2], Amphawan Julsereewong[1,*] and Jirasak Sittigorn[1]

[1]School of Engineering
King Mongkut's Institute of Technology Ladkrabang
1 Chalong Krung, 1 Alley, Lat Krabang, Bangkok 10520, Thailand
{ krit.sm; 64011087; 64010331; jirasak.si }@kmitl.ac.th
*Corresponding author: amphawan.ju@kmitl.ac.th

[2]Mahanakorn Institute of Innovation
Mahanakorn University of Technology
140 Chueam Samphan Road, Krathum Rai, Nong Chok, Bangkok 10530, Thailand
siripong@mut.ac.th

Received August 2025; revised November 2025

Abstract. *This article presents an open-source intrusion detection and notification (IDN) framework designed to detect attacks targeting Siemens S7-series programmable logic controllers (PLCs) in operational technology (OT) environments. The framework is validated using a simulated weight-based sorting system developed in Factory I/O, with TIA Portal control programs deployed on S7-300, S7-1200, and S7-1500 PLCs. Suricata serves as the core intrusion detection engine, while the Elasticsearch, Logstash, and Kibana (ELK) stack and LINE Notify are integrated to visualize alerts and provide real-time operator notifications. Simulated attacks are carried out using Snap7 to interact with each PLC during live process operations. The experimental results show that the framework reliably detects intrusions across all tested S7 PLC models and delivers timely alerts, demonstrating its effectiveness for real-time monitoring and incident response. In contrast to previous studies that focus primarily on protocol analysis or on individual PLC types, this work offers a practical and scalable intrusion detection solution validated on real hardware and designed to accommodate the coexistence of legacy and modern controllers within OT systems.*

Keywords: ELK stack, Factory I/O, Intrusion detection and notification (IDN), Operational technology (OT), Siemens S7 PLCs, Suricata, Weight-based sorting system

1. **Introduction.** Operational technology (OT) systems form the foundation of modern industrial automation by enabling real-time monitoring and control of physical processes across sectors such as manufacturing, energy, and logistics. Programmable logic controllers (PLCs) play a central role in these systems by providing deterministic execution for mission-critical tasks and interacting with sensors, actuators, and supervisory interfaces. As interoperability and Industry 4.0 integration continue to expand, PLC platforms have evolved to support richer programmability and advanced network communication capabilities [1-3]. However, this increasing connectivity between PLCs, enterprise networks, and cloud-based services has significantly widened the OT cyber-attack surface. As noted in [4], connecting PLC-based systems to external networks introduces considerable security risks, while the use of IoT-enabled sensing devices can bypass traditional

DOI: 10.24507/ijicic.22.03.723

defense-in-depth architectures and create new attack pathways within industrial environments [5]. Despite these technological advancements, many OT installations, including legacy PLCs, were originally designed with a strong emphasis on operational continuity and safety rather than cybersecurity. A major challenge highlighted in [6] is the limited digital forensic readiness of industrial control systems, which is often due to inadequate data acquisition tools and inconsistent vendor support. At the same time, Internet-wide scanning studies show that many PLCs remain directly reachable online with weak or default security settings, making them vulnerable to unauthorized access and manipulation [7]. As a result, contemporary ICS-focused cyberattacks increasingly exploit control-logic modification techniques, weaknesses in communication protocols, and network-level interception methods. For instance, SHADOWPLCs [8] demonstrates real-time detection of unauthorized logic manipulation in S7-300 PLCs. Similarly, [9] reports that man-in-the-middle and denial-of-service attacks remain effective across different PLC brands, while [10] demonstrates how simple packet-injection attacks targeting Modbus TCP can disrupt industrial processes. The Stuxnet incident [11] remains a prominent example of malware engineered specifically for covert logic manipulation and physical process sabotage.

Siemens S7-series PLCs are widely deployed in industrial environments and communicate using proprietary S7 protocols. Earlier versions of these protocols lack strong authentication and encryption mechanisms, which allow replay attacks and logic-modification attempts on S7-300 controllers [12,13]. Even newer S7-1500 devices that use cryptographically protected S7CommPlus can still be exposed to security weaknesses. The findings in [14] show that malicious interrupt-based logic patches can remain hidden within S7-1500 PLC memory for long periods, and the results in [15] demonstrate that attackers can manipulate hash-generation processes in S7CommPlusV3 to craft valid encrypted packets and execute replay attacks. These limitations make direct inspection of encrypted payloads increasingly difficult and reinforce the need for protocol-aware intrusion detection approaches that can identify unauthorized memory operations, abnormal S7 communication patterns, and other behaviors targeting PLCs. In practical attack scenarios, open-source tools such as Snap7 can also be used to perform unauthenticated read and write operations over the S7 protocol, which further illustrates the risks present in OT environments.

To address these challenges, this paper presents an open-source intrusion detection and notification (IDN) framework designed specifically for Siemens S7 PLC environments. The framework integrates Suricata for protocol-aware intrusion detection, LINE Notify for real-time alerting, and the ELK stack for log processing and visualization. Validation is conducted using a weight-based sorting system simulated in Factory I/O, with TIA Portal control programs deployed on S7-300, S7-1200, and S7-1500 hardware. Simulated attacks are performed using Snap7 to interact with each PLC during live process operation, representing realistic misuse of the S7 communication protocol. The study contributes a replicable and accessible open-source IDN framework tailored to Siemens S7-series PLC cybersecurity, provides improved situational awareness and forensic readiness through integrated dashboards and real-time notifications, and delivers S7-focused detection rules that enhance the identification of anomalous PLC behaviors compared with conventional IT-centric IDS configurations. Overall, the proposed framework offers a practical, cost-effective, and scalable approach to safeguarding mixed-generation OT environments in Industry 4.0 settings.

The remainder of this article is organized as follows. Section 2 introduces the overall IDN framework concept. Sections 3 and 4 describe the penetration testing using Snap7 and the intrusion detection and notification mechanisms, respectively. Section 5 presents

experimental results and comparative analysis. Finally, Section 6 provides the conclusions and outlines potential directions for future work.

2. **Proposed Framework Concept.** Figure 1 illustrates the concept of the proposed IDN framework for the OT environment, demonstrated through a case study involving a weight-based sorting system simulated using Factory I/O (see Figure 2(a)). Laptop A, functioning at the supervisory layer, acts as the workstation where TIA Portal V18 is installed to configure the PLCs at the control layer and to develop the human-machine interface (HMI) for real-time monitoring of key parameters, such as item weight and current sorting status (see Figure 2(b)). The connection between Factory I/O and TIA Portal is established via the S7 protocol, enabling the PLC configured in TIA Portal to receive data from the simulated Factory I/O environment and issue control commands for sorting items based on their weight. The intrusion detection system, powered by Suricata, monitors and detects suspicious network packets. These packets are forwarded to Laptop B, which operates at the enterprise level and hosts the ELK stack, comprising Elasticsearch for data storage, Logstash for data processing, and Kibana for data visualization, as shown in Figure 3. Figure 4(a) presents the flowchart of the PLC control logic programmed in TIA

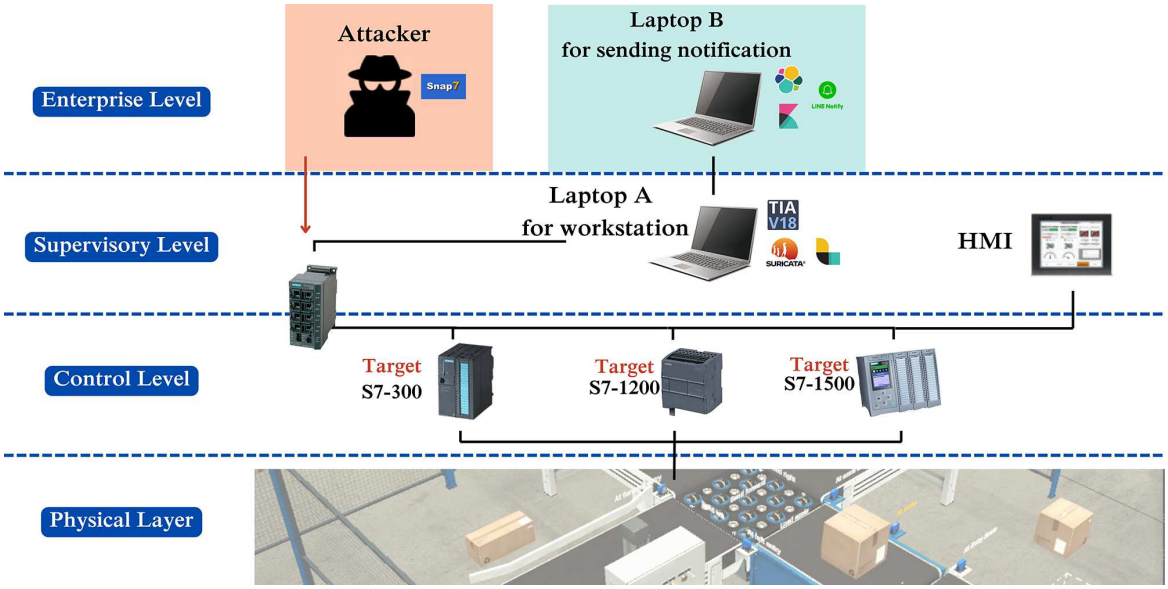
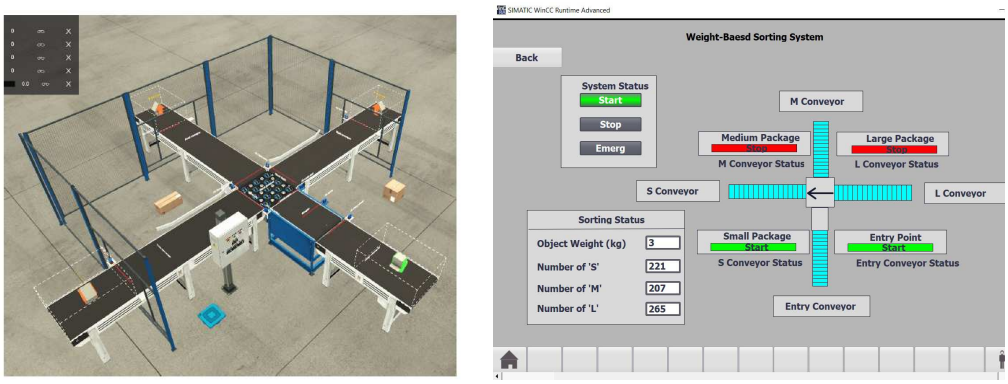


FIGURE 1. System architecture of the proposed framework



(a) Simulated system (physical layer) (b) HMI (supervisory layer)

FIGURE 2. OT environment under study

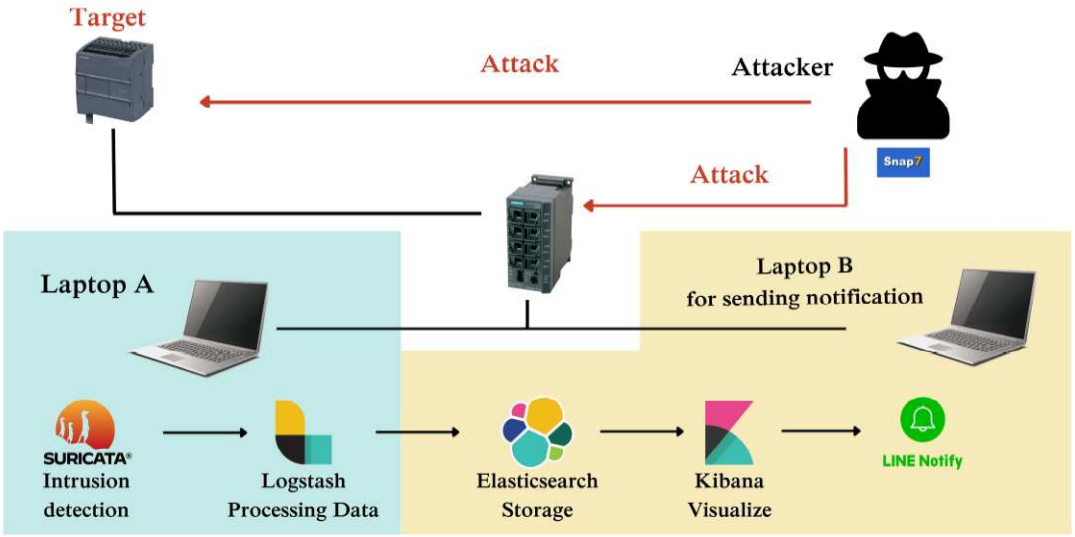


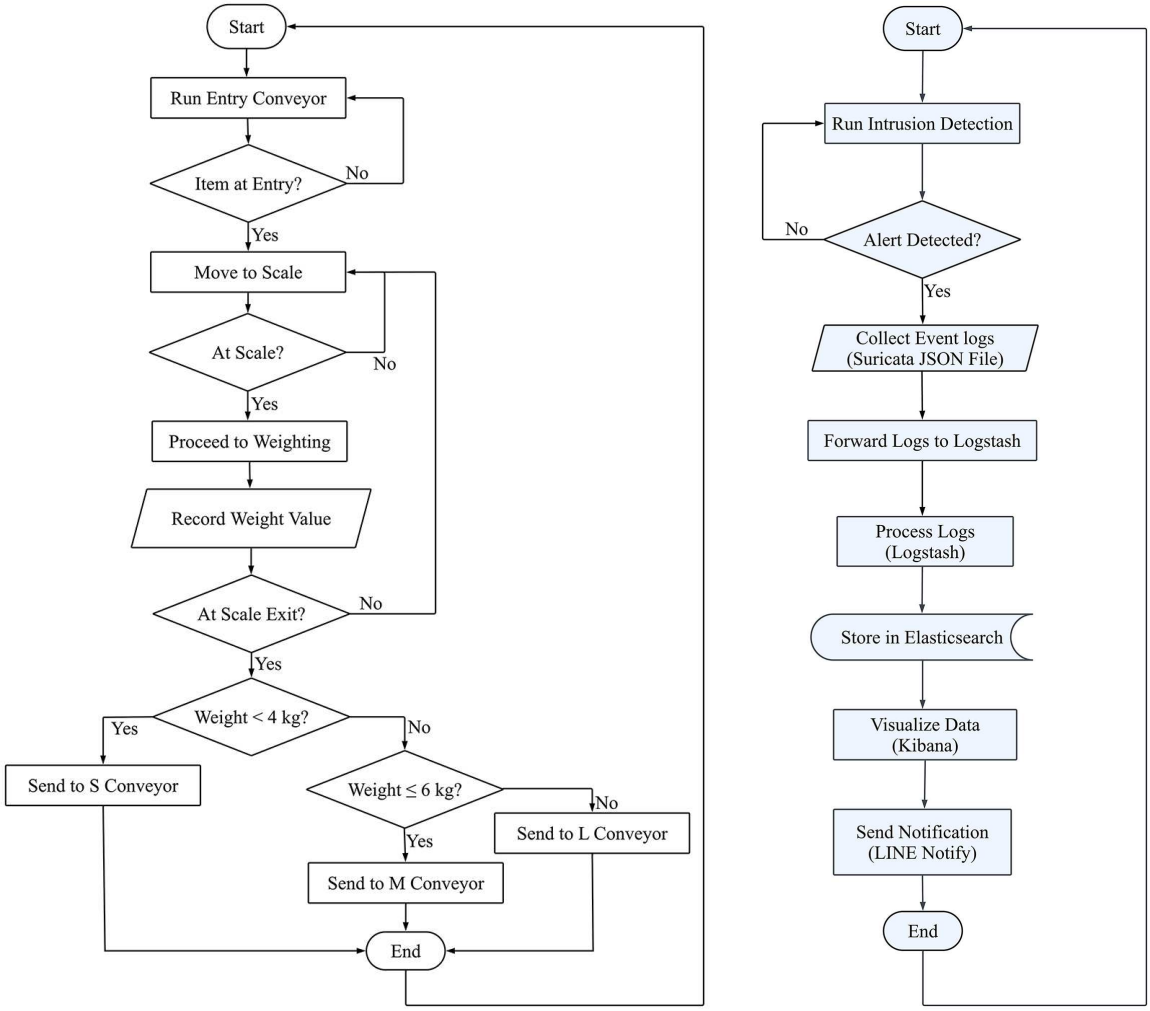
FIGURE 3. Data flow and software architecture of the proposed framework

Portal. The system processes weight data from the simulated environment and activates actuators to sort items accordingly: items weighing less than 4 kilograms (size 'S') are sent left, 4-6 kilograms (size 'M') are sent forward, and those exceeding 6 kilograms (size 'L') are sent right. After each operation, the PLC resets the system to prepare for the next item. The IDN alerting mechanism is illustrated in Figure 4(b). Suricata acts as the first line of defense, analyzing network traffic in real time to detect anomalies or signs of intrusion. When no threat is detected, the system continues monitoring. If an intrusion is identified, Suricata generates an event log in JSON format. This log is processed by Logstash, indexed by Elasticsearch, and visualized in real-time dashboards using Kibana for operator review. Simultaneously, LINE Notify pushes alerts to administrators, ensuring immediate awareness and enabling timely responses to potential security incidents.

3. **Penetration Testing Using Snap7.**

3.1. **Vulnerabilities in PLC memory.** The memory structure of Siemens S7 PLCs is divided into two main areas: the main memory, which stores the control logic, and the register memory, which is temporarily used for processing instructions. However, these memory areas lack basic security protections such as authentication and encryption, making them directly accessible over TCP/IP networks through port 102. In this study, a simulated attack was performed on the PLC's register memory using Snap7 by issuing unauthorized 'read' and 'write' commands to manipulate Boolean values and memory data. Such actions can result in system malfunctions or unauthorized control, posing significant operational risks. This type of attack underscores the importance of implementing robust security measures, including disabling unused ports, deploying intrusion detection systems, and configuring access control policies to mitigate cybersecurity threats in OT environments.

To demonstrate this vulnerability, a test scenario was defined in which the S7 protocol was targeted using Snap7 and Python. The goal was to perform abnormal 'read' and 'write' operations on the PLC controlling the simulated system. The attack tests were carried out using two primary methods: reading and writing Boolean values, and reading and writing memory data within the PLC.



(a) PLC control (b) IDN process

FIGURE 4. Flowcharts of the proposed framework

3.2. **Connection between Snap7 and the PLC.** A connection to Siemens S7 PLCs can be established using a Snap7 client, which communicates over TCP/IP without requiring authentication. As a result, memory 'read' and 'write' operations can be executed directly. In this test, Python and Snap7 were used to establish the connection by calling the 'snap7.client.Client()' function to create a client object. The connection to the target PLC was then configured using the command 'plc.connect('192.168.0.102', 0, 1)', which specifies the IP address, rack, and slot of the PLC, as illustrated in Figure 5. Once the connection is established, commands can be sent to the PLC to read from or write to process control variables without any security protections in place.

```
plc = snap7.client.Client()
plc.connect('192.168.0.102', 0, 1)  # IP address, rack, slot
```

FIGURE 5. Connection configuration between the PLC and Snap7

3.3. **Reading and writing Boolean values in the PLC data block.** A data block (DB) in the PLC stores critical variables that control the sorting process, as displayed in Figure 6 (highlighted with dashed red rectangles). In this type of attack, Snap7 is used to

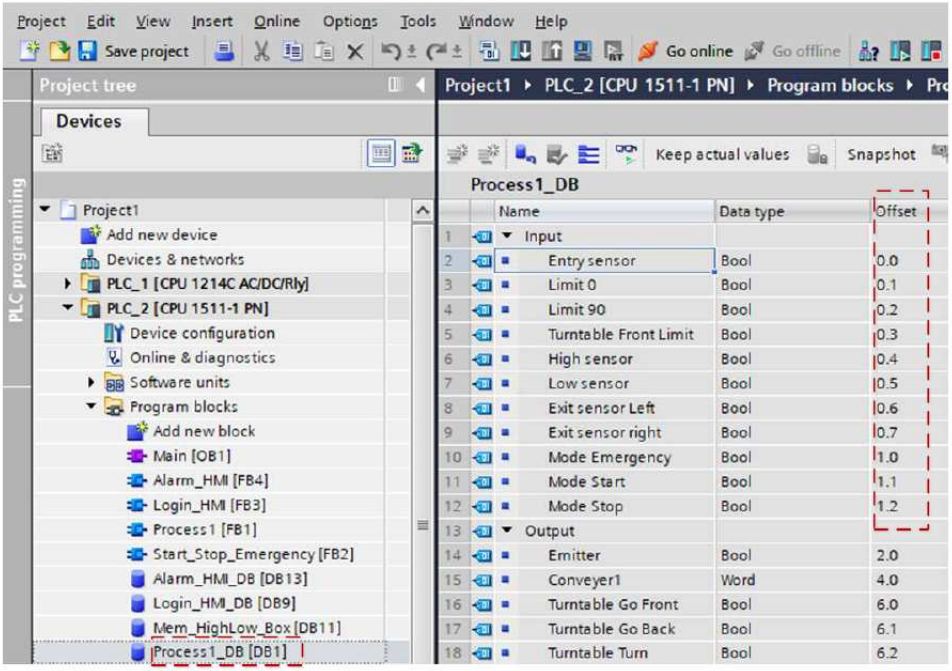


FIGURE 6. Example of PLC address for Boolean data

```
db_number = 1
start_offset = 4
bit_offset = 0
value = 0

def writeBool(db_number, start_offset, bit_offset, value):
    reading = plc.db.read(db_number, start_offset, 1)
    snap7.util.set_bool(reading, 0, bit_offset, value)
    plc.db_write(db_number, start_offset, reading)
    return None

def readBool(db_number, start_offset, bit_offset):
    reading = plc.db_read(db_number, start_offset, 1)
    a = snap7.util.get_bool(reading, 0, bit_offset)
    print('DB Number: ' + str(db_number) + ' Bit ' + str(start_offset) + '.' + str(bit_offset))
    return None
```

(a) Boolean values

```
start_address = 500
lenght = 4

def readMemory(start_address, lenght):
    reading = plc.read_area(snap7.type.Areas.MK, 0, start_address, lenght)
    value = struct.unpack('>f', reading)[0]

    print("\n")
    print(f'Start Address: %W{start_address}')
    print(f'Raw Value: {value} kg')
    print("\n")

def writeMemory(start_address, lenght, value):

    plc.mb_write(start_address, lenght, bytearray(struct.pack('>f', value)))
    print("\n")
    print(f'Start Address: %W{start_address}')
    print(f'Weight Input: {value} kg')
    print("\n")
```

(b) PLC memory

FIGURE 7. Functions for reading and writing Boolean values and PLC memory

read Boolean values from the DB and overwrite them to alter the system's behavior. The reading process begins with the command 'plc.db_read(db_number, start_offset, 1)', which retrieves data from a specific address in the DB. The retrieved data are then converted into a Boolean value using 'snap7.util.get_bool(reading, 0, bit_offset)'. To write a new Boolean value, the command 'snap7.util.set_bool()' is used to modify the data, followed by 'plc.db_write()' to apply the changes to the PLC, as shown in Figure 7(a). Writing the modified value back to the PLC can cause the system to misinterpret sensor readings or lead to unintended motor and actuator operations. These actions can result in safety hazards and disrupt the sorting process.

3.4. **Reading and writing PLC memory values.** The memory of Siemens S7 PLCs stores process data such as sensor readings and motor speed settings. This type of attack targets the PLC's memory values to extract control-related information or to modify operational parameters. Memory reading can be performed using the command 'plc.read_area (snap7.types.Areas.MK, 0, start_address, length)', which retrieves data from the specified address. The retrieved data are then converted into a floating-point number using 'struct.unpack('>f', reading)[0]'. To write memory values, the desired value must first be

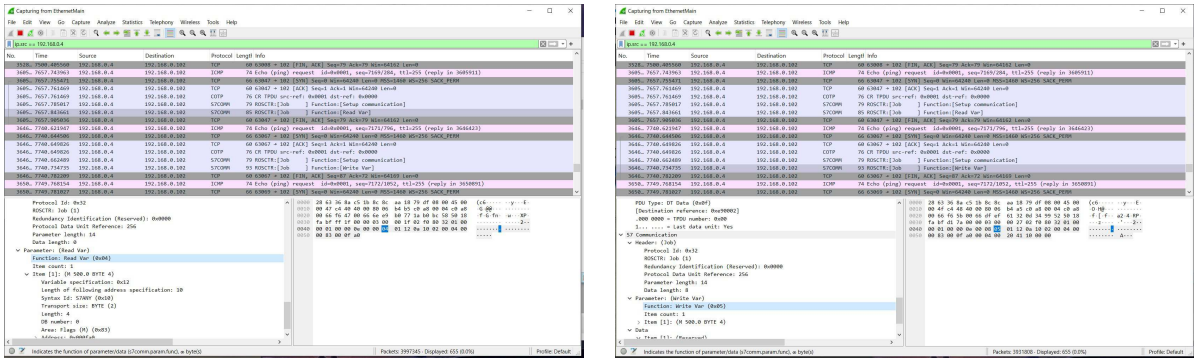
converted into byte format using 'struct.pack('>f', value)'. This byte array is then written to the target memory using 'plc.mb_write(start_address, length, bytearray(struct.pack('>f', value)))', as illustrated in Figure 7(b). In this article, the memory address targeted is %MW500, which corresponds to the measured weight value. Such an attack can cause the PLC to behave abnormally, for instance by changing the sorting direction. This may lead to significant safety risks for both machinery and personnel.

4. **Intrusion Detection and Notification.** This article employs a network-based IDS, with Suricata serving as the primary tool for identifying intrusions within the network. The detection focuses on analyzing network packets associated with the S7 protocol, which is the primary communication protocol used by Siemens S7 PLCs. Suricata functions as the IDS by monitoring suspicious activities using signature-based detection rules. It specifically targets 'read' and 'write' operations to PLC memory transmitted via port 102, which is the default communication port for the S7 protocol. This detection mechanism enables real-time alerting when abnormal behaviors are identified.

4.1. **Packet analysis using Wireshark.** Before defining detection rules in Suricata, it is essential to understand the packet structure used in S7 protocol communications between S7 PLCs. To this end, Wireshark was employed to capture and analyze relevant network packets.

For memory 'read' operations, Wireshark captured packets with a source IP address of 192.168.0.4 and a destination IP address of 192.168.0.102. These packets utilized the S7COMM protocol with the function code 'Read Var (0x04)', indicating a memory read request to the PLC. Key components in the payload included the S7 communication header (|32|), the 'read' memory command (|04|), and the memory access structure (|00 04|), as shown in Figure 8(a). These components serve as the foundation for writing effective detection rules in Suricata.

For memory 'write' operations, the captured packets also originated from 192.168.0.4 and were destined for 192.168.0.102. These packets similarly employed the S7COMM protocol, but with the function code 'Write Var (0x05)', indicating a request to write data into the PLC's memory. The payload contained the S7 communication header (|32|), the 'write' memory command (|05|), and the relevant memory access structure, as illustrated in Figure 8(b). These fields provide the necessary conditions for defining signature-based rules in Suricata to detect unauthorized memory manipulation.



(a) Memory 'read' packet (b) Memory 'write' packet

FIGURE 8. Memory 'read' and 'write' packets of the S7 protocol

4.2. **Detection rules in Suricata.** Suricata uses custom detection rules, such as those illustrated in Figure 9, to analyze network packets in real time. In this article, the rules are specifically designed to identify behaviors associated with the S7 protocol, which is commonly used by Siemens PLCs. The detection is based on two primary conditions: the use of TCP port 102 and the presence of specific byte patterns within the packet payload. These criteria allow Suricata to monitor for 'read' and 'write' commands issued through the S7 protocol. When a packet meets the defined conditions, the system generates an alert in the activity log. The rules in this study are constructed to detect Boolean 'read' and 'write' operations from the PLC by identifying packets that contain the byte sequences '32 04' for read operations and '32 05' for write operations. These sequences correspond to the S7 protocol function codes for accessing Boolean data. To improve detection precision, the rules also apply the distance and within keywords to specifying the expected byte offset between matching patterns within the payload, as shown in Figure 10. For memory 'read' and 'write' operations, the same byte patterns ('32 04' and '32 05') are used. However, an additional condition, content:"|00 04|", is included to indicate access to the PLC's memory area. This condition allows the system to distinguish between general protocol communication and direct memory access attempts. The combination of byte patterns and offset constraints forms the basis of effective detection rules, as illustrated in Figure 11.

```
<rule action> <protocol> <source IP> <source port> -> <destination IP> <destination port>
(<rule options>)
```

FIGURE 9. Structure of a Suricata rule

```
alert tcp 192.168.0.4 any -> 192.168.0.102 102 (msg:"Snap7 Read Bool Detected"; content:"|32|";
content:"|04|"; distance:0; within:2; flowbits:set,snap7_read_bool; sid:1000052; rev:6;)
alert tcp 192.168.0.4 any -> 192.168.0.102 102 (msg:"Snap7 Write Bool Detected"; content:"|32|";
content:"|05|"; distance:0; within:2; flowbits:set,snap7_write_bool; sid:1000056; rev:6;)
```

FIGURE 10. Detection rules for reading and writing Boolean values

```
alert tcp 192.168.0.4 any -> 192.168.0.102 102 (msg:"Snap7 Read Memory Detected"; content:"|32|";
content:"|04|"; content:"|00 04|"; flowbits: isnotset,snap7_write_detected; sid:1000054; rev:6;)
alert tcp 192.168.0.4 any -> 192.168.0.102 102 (msg:"Snap7 Write Memory Detected"; content:"|32|";
content:"|05|"; content:"|00 04|"; flowbits:set,snap7_write_detected; sid:1000058; rev:6;)
```

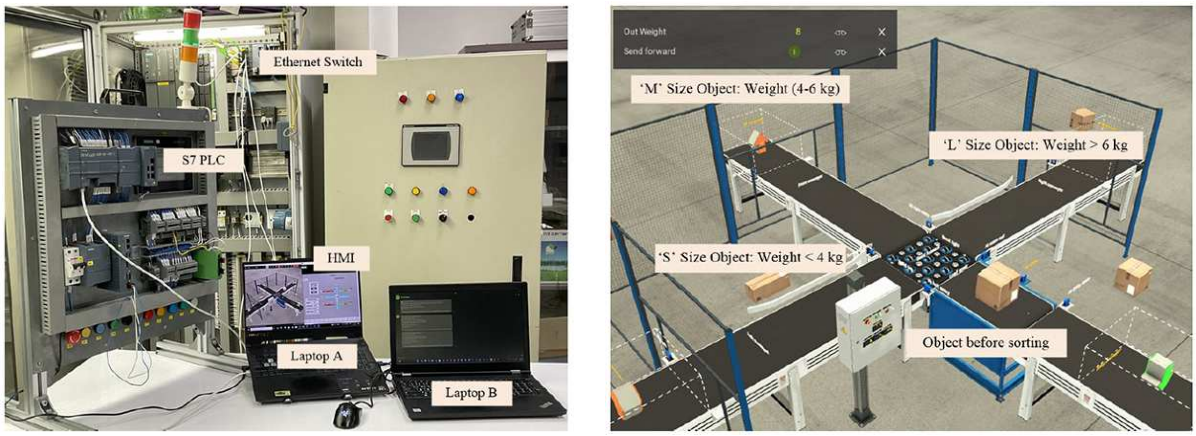
FIGURE 11. Detection rules for reading and writing memory values

4.3. **Suricata configuration and execution.** To configure Suricata, the 'suricata.yaml' file must be edited to define system settings such as network parameters and output preferences. In this study, both 'fast.log' and 'eve.log' are enabled to record detected network activities. Custom rule files are then added to the configuration to identify and log suspicious behavior. To start Suricata, the user navigates to the installation directory and executes the appropriate command to launch Suricata, specifying the network interface that will be used for packet inspection.

4.4. **Real-time alerting with LINE Notify.** The system is designed to detect and respond to suspicious activities in the OT environment in real time. Suricata monitors

the network for malicious behaviors, such as unauthorized access to the S7 protocol on Siemens PLCs, and logs these events. Logstash receives the detection data from Suricata and forwards it to Elasticsearch for storage and further analysis. Kibana is then used to visualize the data and configure alerting rules. When suspicious activity is detected, the system automatically sends a real-time notification via LINE Notify, ensuring immediate awareness and enabling prompt responses by system administrators.

5. **Experimental and Comparison Results.** Figures 12(a) and 12(b) show the hardware setup and the HMI screen of the simulated sorting system, respectively. The S7 PLCs under evaluation are connected to Laptop A via an Ethernet switch, while Laptop B is used for data processing and for generating operator alerts in the event of an intrusion. To evaluate the effectiveness of the proposed IDN system, intrusion and detection tests were conducted for each S7 PLC model. These attacks were performed using the S7 protocol, the primary communication interface for S7 PLCs. Detection results were analyzed using Suricata, and the responsiveness of the LINE Notify alerting system was also assessed.

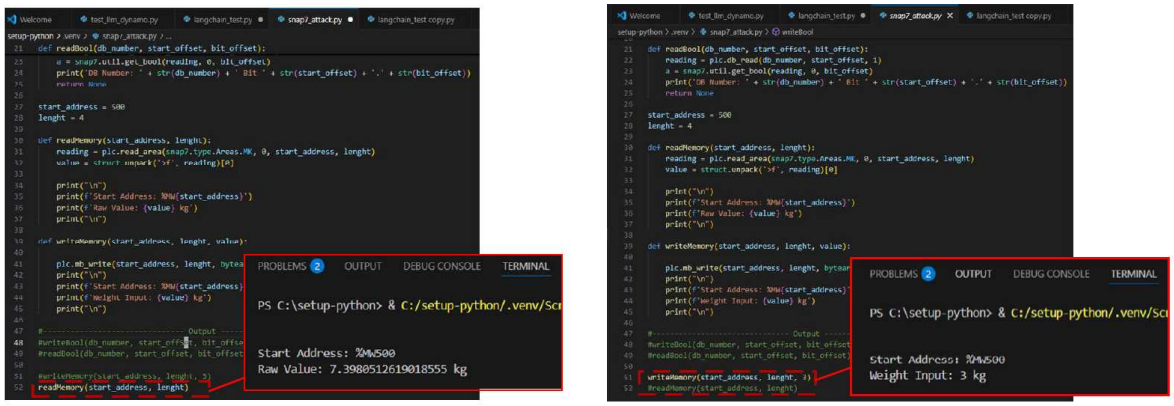


(a) Hardware installation (b) Simulated sorting process

FIGURE 12. Experimental setup to test the proposed framework workability

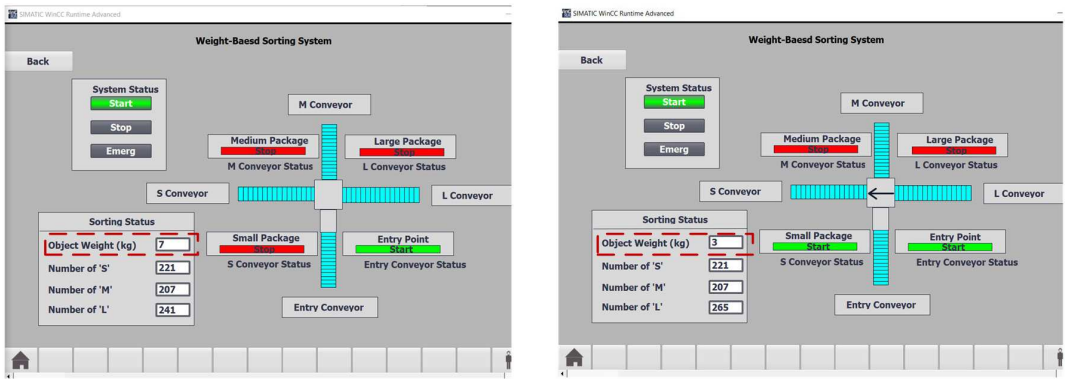
5.1. **Results of the attack testing.** Before the attack tests were conducted, the S7 PLC operated normally, with no changes made to its memory values. Using the Snap7 tool to send memory 'read' commands to the PLC via the S7 protocol (see Figure 13(a)) demonstrates that an attacker can extract data from the PLC without any form of authentication. Additionally, an attacker can use Snap7 to send write commands directly to the PLC's memory, potentially disrupting the operation of the sorting system (see Figure 13(b)). Figures 14(a) and 14(b) show the HMI screens before and after the attack, respectively, with the modified memory values highlighted by dashed red rectangles, confirming that the attack was successful. The tests were performed on S7-300, S7-1200, and S7-1500 models using Snap7 to execute 'read' and 'write' operations over the S7 protocol. The results confirm that these attacks were effective and indicate that control systems using S7 PLCs remain vulnerable to network-based intrusions.

5.2. **Results of the intrusion detection.** Intrusion detection tests were performed on the selected S7 PLCs by configuring Suricata to monitor traffic related to the S7 protocol, specifically targeting 'read' and 'write' operations. The results demonstrate that Suricata successfully identified suspicious behaviors. The logged data included key information such as the timestamp of the attack (see Figure 15(a)), the source and destination IP addresses,



(a) Result from reading (b) Result from writing

FIGURE 13. Example of using Snap7 to read and write the PLC memory and their result



(a) Before the attack (b) After the attack

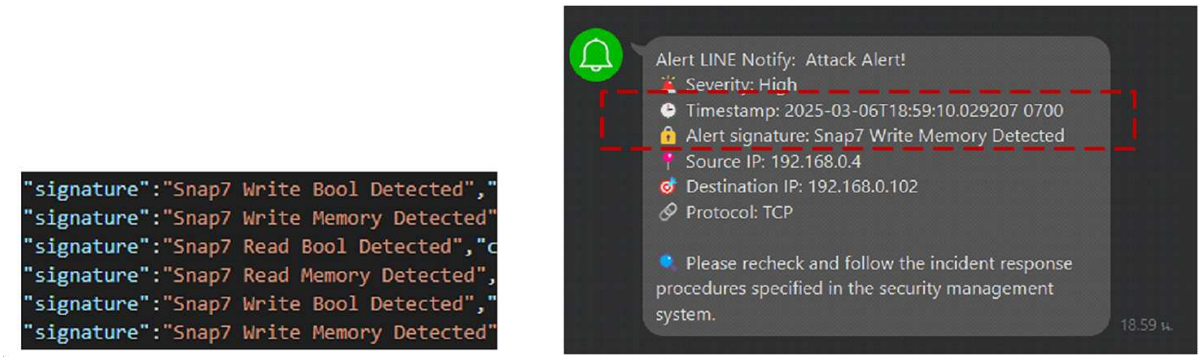
FIGURE 14. Example of HMI screens before and after the attack

(a) Attack timestamp (b) Destination port

FIGURE 15. Example of logged data from intrusion detection test

the source and destination ports (see Figure 15(b)), and the alert signatures (see Figure 16(a)). Examples of alert messages include “Snap7 Read Bool Detected”, “Snap7 Write Bool Detected”, “Snap7 Read Memory Detected”, and “Snap7 Write Memory Detected”.

5.3. **Results of the alert notification.** The test results demonstrate that LINE Notify effectively delivers real-time alerts when suspicious activity related to the S7 protocol, used by Siemens S7 PLCs, is detected in OT systems. The notifications provide key details, including the severity level of the event, the precise timestamp of the intrusion, the name



(a) Logged alert signature (b) Alert notification

FIGURE 16. Example of alert signature and notification via LINE Notify

of the triggered detection rule (such as "Snap7 Read Memory Detected"), the source and destination IP addresses involved, the associated protocol, and a message advising system administrators to investigate the incident. An example notification is shown in Figure 16(b).

5.4. **Comparison analysis.** The comparative evaluation across all PLC models shows a consistent pattern in both attack execution and intrusion detection. Each of the tested controllers (S7-300, S7-1200, and S7-1500) was fully compromised using Snap7 through unauthenticated S7 protocol read and write operations during live process operation. Suricata detected every malicious activity in all scenarios, and the resulting activity logs exhibited a uniform structure, as summarized in Table 1.

TABLE 1. Attack and IDN testing results

S7 PLC	Penetration testing	Intrusion detection	Alert notification
S7-300	Success	Success	Success
S7-1200	Success	Success	Success
S7-1500	Success	Success	Success

This consistency indicates that the proposed IDN framework provides stable and repeatable detection performance across different PLC generations. The identical attack outcomes further confirm that all three PLC models remain susceptible to unauthorized memory access when exposed to the network, regardless of architectural differences or firmware enhancements. Meanwhile, Suricata reliably identified all abnormal S7 protocol behaviors, and LINE Notify delivered timely alerts in every case.

Overall, these findings validate that the framework is suitable for deployment in mixed-generation OT environments where legacy and modern S7-series controllers coexist. The results also highlight the need for continued research on advanced detection methods capable of handling scenarios where PLC communications are fully encrypted, as this represents an increasingly common configuration in modern industrial installations.

6. **Conclusions.** A practical, open-source intrusion detection and notification framework for OT systems using S7-series PLCs has been presented in this article. The framework integrates Suricata for protocol-aware intrusion detection, the ELK stack for centralized log processing and visualization, and LINE Notify for real-time operator alerts. Validation was performed using a simulated weight-based sorting system developed in Factory I/O, with TIA Portal-based control programs deployed on physical S7-300, S7-1200, and S7-1500

hardware. The experimental results demonstrate that the framework consistently detects unauthorized S7 protocol read and write operations across all tested PLC generations and delivers actionable alerts during live process operations. These capabilities make the solution particularly suitable for industrial settings in which legacy and modern controllers coexist, offering a cost-effective and accessible method for improving OT security posture. While the framework effectively detects unencrypted S7 communications, its reliance on packet inspection limits its ability to analyze fully encrypted S7CommPlus traffic used in modern PLC deployments. Future research will focus on integrating machine-learning-based anomaly detection, incorporating behavioral and statistical models, expanding support for additional industrial protocols, and exploring automated mitigation strategies to achieve faster and more resilient OT incident response.

REFERENCES

[1] S. Pongswatd, P. Niyomtamarat, S. Kummool and T. Thepmanee, PLC function block creation for minimizing data link limitation on CC-link-based conveyor handling system, *ICIC Express Letters, Part B: Applications*, vol.13, no.7, pp.681-688, 2022.

[2] D.-N. Truong, H.-T. Pham, P.-N. Tran and V.-P. Ta, A fault detection and diagnosis solution for PLC-based industrial process control systems, *ICIC Express Letters, Part B: Applications*, vol.14, no.12, pp.1225-1233, 2023.

[3] S. Taimaingam and P. Pannil, Comprehensive performance evaluation of PROFIBUS and PROFINET in PLC-based control systems, *ICIC Express Letters, Part B: Applications*, vol.15, no.7, pp.687-699, 2024.

[4] W. Alsabbagh and P. Langendorfer, Security of programmable logic controllers and related systems: Today and tomorrow, *IEEE Open Journal of the Industrial Electronics Society*, vol.4, pp.659-693, 2023.

[5] T. Sauter and A. Treytl, IoT-enabled sensors in automation systems and their security challenges, *IEEE Sensors Letters*, vol.7, no.12, pp.1-4, 2023.

[6] M. M. Cook, A. K. Marnerides, C. Johnson and D. Pezaros, A survey on industrial control system digital forensics: Challenges, advances, and future directions, *IEEE Communications Survey & Tutorials*, vol.25, no.3, pp.1705-1747, 2023.

[7] T. Lee, S. Kim and K. Kim, A research on the vulnerabilities of PLC using search engine, *Proc. of the 10th International Conference on ICT Convergence*, Jeju Island, Korea, pp.184-188, 2019.

[8] J. Liu, X. Lin, X. Chen, H. Wen, H. Li, Y. Hu, J. Sun, Z. Shi and L. Sun, SHADOWPLCs: A novel scheme for remote detection of industrial process control attacks, *IEEE Transactions on Dependable and Secure Computing*, vol.19, no.3, pp.2054-2069, 2022.

[9] S. G. Tan, I.-H. Liu and J.-S. Li, Simulation and analysis of common attacks against PLCs used in dam testbed, *Proc. of 2023 International Conference on Advanced Robotics and Intelligent Systems*, Taipei, Taiwan, pp.1-6, 2023.

[10] G. Lazaridis, A. Drosou, P. Chatzimisios and D. Tzovaras, Security Modbus TCP communications in I4.0: A penetration testing approach using OpenPLC and factory IO, *Proc. of 2023 IEEE Conference on Standards for Communications and Networking*, Munich, Germany, pp.265-270, 2023.

[11] T. Miyachi, H. Narita, H. Yamada and H. Furata, Myth and reality on control system security revealed by Stuxnet, *Proc. of SICE Annual Conference*, Tokyo, Japan, pp.1-4, 2011.

[12] Y. Zhang, Z. Sun, L. Yang, Z. Li, Q. Zeng, Y. He and X. Zhang, All your PLCs belong to me: ICS ransomware is realistic, *Proc. of 2020 IEEE 19th International Conference on Trust, Security, and Privacy in Computing and Communications*, Guangzhou, China, pp.502-509, 2020.

[13] W. Alsabbagh and P. Langendorfer, A stealth program injection attack against S7-300 PLCs, *Proc. of 2021 22rd International Conference on Industrial Technology*, Valencia, Spain, pp.986-993, 2021.

[14] W. Alsabbagh and P. Langendorfer, A new injection threat on S7-1500 PLCs – Disrupting the physical process offline, *IEEE Open Journal of the Industrial Electronics Society*, vol.3, pp.146-162, 2022.

[15] W. Alsabbagh and P. Langendorfer, You are what you attack: Breaking the cryptographically protected S7 protocol, *Proc. of 2023 IEEE 19th International Conference Factory Communication Systems*, Pavia, Italy, pp.1-8, 2023.

Author Biography

Krit Smerpitak received the B.Eng. and M.Eng. degrees in Instrumentation Engineering, as well as the D.Eng. degree in Electrical Engineering, from King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand, in 2001, 2006, and 2017, respectively.

He is currently a full-time Assistant Professor at School of Engineering, KMITL, Thailand. His research interests include factory automation and smart manufacturing systems.

Chayangkun Saleethong received the B.Eng. degree in Automation Engineering from King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand, in 2025.

He currently serves as a Super Architect at XFINIT Co., Ltd., Thailand. His research interests focus on industrial automation systems and cybersecurity.

Tanawat Suphapod received the B.Eng. degree in Automation Engineering from King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand, in 2025.

He currently serves as an Electrical Engineer (Development) at Thai NOK Co., Ltd., Thailand. His areas of interest include industrial automation systems and machine development.

Siripong Wongkharn received the B.Eng. degree in Instrumentation Engineering and the M.Eng. degree in Electrical Engineering from King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand, in 1994 and 1997, respectively.

He is currently a full-time Lecturer at Mahanakorn Institute of Innovation, Mahanakorn University of Technology, Thailand. His research interests include mechatronics and smart manufacturing systems.

Amphawan Julsereewong received the B.Eng. degree in Instrumentation Engineering and the M.Eng. and D.Eng. degrees in Electrical Engineering from King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand, in 1998, 2000, and 2006, respectively.

She is currently a full-time Associate Professor at School of Engineering, KMITL. Her research interests include digital fieldbuses, process control and automation, and industrial energy efficiency.

Jirasak Sittigorn received the B.Eng. degree in Telecommunication Engineering from Suranaree University of Technology, Thailand, in 2001, and the M.Eng. degree in Telecommunication Engineering and the D.Eng. degree in Electrical Engineering from King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand, in 2004 and 2015, respectively.

He is currently a full-time Assistant Professor at School of Engineering, KMITL. His research interests include data communications and networks, as well as IoT systems.