A Study of Cybersecurity Vulnerabilities in a Programmable Logic Controller

Dong-Yang Lee
Graduate Institute of Intelligent Manufacturing Technology
National Taiwan University of Science and Technology
Taipei, Taiwan
M11251025@mail.ntust.edu.tw

Wei-chen Lee
Department of Mechanical Engineering
National Taiwan University of Science and Technology
Taipei, Taiwan
wclee@mail.ntust.edu.tw

***Abstract*—With the advent of Industry 4.0 and operational technologies, industrial control systems are shifting from closed architectures to open, networked environments that enable remote monitoring, cloud integration, and cross-site connectivity. Although these advances drive smart manufacturing, they also expose critical cybersecurity risks. This study investigates vulnerabilities in a programmable logic controller, which supports multiple industrial communication protocols. We conducted vulnerability scanning and developed a cyberphysical platform that combines Factory I/O with physical controllers to simulate the effects of cyberattacks on factory operations. The implementation procedures and vulnerability findings were systematically documented. The results provide practical insight into the cybersecurity of the industrial controller.**

Keywords—programmable logic controller, cybersecurity, vulnerability

I. INTRODUCTION

With the increasing digitalization of manufacturing and critical infrastructure, Operational Technology (OT) and Industrial Control Systems (ICS) have become the core foundations that support the operation of modern factories and public facilities. Traditionally, these systems were designed for closed environments, resulting in relatively weak cybersecurity protection. However, as ICS progressively integrates with Information Technology (IT) and becomes interconnected with external networks, potential cybersecurity risks have increased substantially. Since vulnerabilities are closely tied to ICS, one of the most common and critical devices—the Programmable Logic Controller (PLC)—has attracted significant attention.

The central role of PLCs in ICS makes them attractive targets for adversaries. Well-known incidents, such as the Stuxnet attack, have demonstrated that PLCs are vulnerable once exposed to insecure or cross-linked networks. Nevertheless, many PLCs still rely on legacy or weak protection mechanisms, and their resilience against modern cyberattacks remains underexplored, particularly for certain vendors. This gap underscores the need for empirical studies that focus on PLC security.

Previous research has investigated PLC cybersecurity from several complementary perspectives.

- *Protocol-level vulnerabilities.* Thomas et al. [1] highlighted the authentication, integrity, and encryption weaknesses in Modbus TCP, demonstrating that PROFINET provides stronger guarantees in both synchronization and security. Similarly, Luswata et al. [2] showed through penetration testing with the Smod tool that Modbus TCP is vulnerable to Denial-of-Service (DoS) attacks, while also evaluating the effectiveness of an intrusion detection systems (IDS) and a firewall.
- *Detection and protection mechanisms.* Tian et al. [3] proposed a Siemens S7-based intrusion detection model (BPID), combining deep packet inspection with a self-learning whitelist to detect abnormal traffic. Huang et al. [4] introduced a dynamic watermarking approach that embeds hidden signals into control commands to detect man-in-the-middle attacks in real time. In addition, Zhang et al. [5] presented a record-and-replay strategy that leverages redundant PLCs to assume control when anomalies are detected. In addition to these works, Lanotte et al. [6] developed a runtime enforcement framework that synthesizes monitors capable of suppressing or correcting malicious actions in controller networks.
- *Empirical attack studies.* Ocaka et al. [7] assessed the impact of three cyberattacks—Code Injection, Man-in-the-Middle, and DoS—on a Siemens LOGO!8 PLC, reporting that DoS caused the most severe disruption. Wardak et al. [8] examined password-based access control and demonstrated that such mechanisms are easily compromised, leaving PLCs vulnerable to unreported security threats. Bonney et al. [9] analyzed Beckhoff's CX5020 PLC, revealing that attackers could gain control over both the program and the operating system with limited expertise, raising concerns about the adoption of standard platforms and TCP/IP encapsulated protocols. Moreover, Cui et al. [10] provided a taxonomy of PLC attacks—control logic injection, firmware modification, protocol exploitation, and memory attacks—alongside countermeasures such as firmware integrity checks, detection techniques, and encryption.
- *Testbeds and experimental platforms.* Low et al. [11] established an ICS cybersecurity testbed based on VMware virtualization, integrating Windows 10, Ubuntu, and Kali environments to facilitate penetration testing and defensive training. Likewise, Aslam et al. [12] analyzed real-world threat cases at the system level, stressing the increased cybersecurity requirements under OT/IT convergence.

This work was financially supported by the National Science and Technology Council [Grant number NSTC 113-2221-E-011-077-MY2], Taiwan, R.O.C.

Despite these valuable contributions, most previous studies have focused on Siemens, Allen-Bradley, or Beckhoff PLCs, or emphasized protocol-specific weaknesses such as Modbus TCP. In particular, the cybersecurity characteristics of PLCs other than those mentioned above have not been systematically investigated. To address this issue, the present study conducts a cybersecurity assessment of the Omron NX102-9000 PLC. Specifically, vulnerability scanning, penetration testing, and abnormal communication simulations were performed to evaluate its security features. Moreover, to approximate realistic threat scenarios in smart manufacturing, a cyberphysical framework is introduced, enabling a cyber-physical integrated simulation platform for attack testing and security behavior observation between the physical PLC and the connected virtual warehouse. It is important to note that all PLCs exhibit their own strengths and limitations with respect to cybersecurity. In this study, the PLC was selected simply because it is the controller deployed in our smart manufacturing facility. We used this device as a representative case study for evaluating PLC security characteristics in practice.

II. Materials and Methods

This study adopted a standardized vulnerability assessment workflow to evaluate the PLC and communication protocols commonly used in Industrial Control Systems (ICS). The testing procedure consisted of five sequential phases and utilized multiple cybersecurity tools within a Kali Linux environment. The objective was to establish a repeatable and practice-oriented ICS security testing framework. The methodology is outlined below.

A. *Preparation:*

- Define the precise scope of the testing.
- Update the Kali Linux system and relevant toolsets (e.g., Nmap, Wireshark).
- Collect technical documentation of the target controller, including supported communication protocols, open ports, and authentication methods.

B. *Network Discovery and Mapping*

- Use Nmap to scan the target IP range, identify live hosts, and enumerate exposed services.
- Capture network traffic with Wireshark and analyze packet traces to verify whether the controller responds to specific industrial protocols.

C. *Vulnerability Assessment*

- Evaluate the controller for risks such as unauthorized access, weak or default credentials, unencrypted data transmission, and other common misconfigurations.

D. *Penetration Testing*

- Perform active exploitation attempts and functional tests, including issuing read/write commands to the PLC via relevant protocols (e.g., Omron FINS, EtherNet/IP) to validate the impact of identified weaknesses.

E. *Post-Test Analysis and Reporting*

- Generate a comprehensive test report documenting the findings and providing remediation recommendations and security hardening measures.
- Remove any temporary accounts and artifacts introduced during testing to restore the system to its pre-test state.

III. Vulnerability Analysis

The vulnerability scanning and analysis of the ICS network were performed using Kali Linux. Network discovery was conducted with Nmap to probe common industrial service ports (see Fig. 1). The scan revealed that port 443 (HTTPS) and port 44818 (EtherNet/IP) were open on the target device, with a comparative summary provided in Table 1.

```
┌──(kali㉿kali)-[~]
└─$ nmap -Pn -sT --scan-delay 2s --max-parallelism 1 \
-p 22,80,102,443,502,530,593,631,789,1089-1091,1880,1911,1962,2222,2404,4000,4840,4843,4911,5900,9600,1
9999,20000,20547,34962-34964,34980,44818,46823,46824,55000-55003 \
192.168.250.1
Warning: --min-parallelism and --max-parallelism are ignored with --scan-delay.
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-16 04:15 EDT
Nmap scan report for 192.168.250.1
Host is up (0.0021s latency).
Not shown: 35 filtered tcp ports (no-response)
PORT      STATE SERVICE
443/tcp   open  https
44818/tcp open  EtherNetIP-2
```

Fig. 1. Scanning the ports of the PLC using Nmap.

TABLE I. Comparison of Ports 443 and 44818

	Port 443	**Port 44818**
Protocol	HTTPS	EtherNet/IP
Transport Layer	TCP	TCP/UDP
Typical Use in ICS	Web-based HMI	Industrial communication

The PLC was further scanned using WhatWeb and Nikto. As shown in Fig. 2, both tools returned error responses indicating an SSL/TLS handshake failure (ssl/tls alert handshake failure). This prevented the establishment of connections, making it impossible to retrieve HTTP headers or server-related information.

These results suggest that although TCP port 443 is exposed, its communication mechanism does not conform to conventional web protocols or to TLS versions and cipher parameters typically expected by browsers or scanning tools. The port is likely reserved for encapsulated protocols such as CIP over TLS or FINS over TLS, rather than for a web-based management interface. Such closed-form HTTPS communication is relatively common in industrial control devices, reflecting a design choice that prioritizes proprietary secure channels over general-purpose web access.

```
┌──(kali㉿kali)-[~]
└─$ whatweb https://192.168.250.1
nikto -host https://192.168.250.1

ERROR Opening: https://192.168.250.1 - SSL_connect returned=1 errno=0 peeraddr=192.168.250.1:443 state=
error: ssl/tls alert handshake failure
- Nikto v2.5.0
---------------------------------------------------------------------------
---------------------------------------------------------------------------
+ 0 host(s) tested
```

Fig. 2. Error message obtained using WhatWeb and Nikto to scan the PLC.

To further investigate the function of TCP port 443 on the controller, packet capture and protocol analysis were conducted using Wireshark (see Fig. 3). The captured traces revealed that the device employs the TLSv1.2 cryptographic protocol. During the handshake process, the controller successfully executed standard TLS procedures, including Server Hello, Certificate, Client Key Exchange, Change Cipher Spec, and Encrypted Handshake Message. Subsequent communications showed the sustained transfer of large volumes of Application Data, indicating that port 443 is primarily used as an internal control communication interface rather than for delivering conventional web content.

Although TLSv1.2 remains a widely adopted encryption standard, the captured traffic exhibited no evidence of HTTP headers or any common web requests such as GET or POST. This strongly suggests that the transmitted data correspond to a proprietary industrial protocol encapsulated within TLS, rather than standard HTTPS traffic. These findings corroborate the earlier results from the WhatWeb and Nikto scans, which failed to successfully interrogate port 443 because such tools assume the presence of standard HTTPS (i.e., HTTP-over-TLS) connections, and are therefore incapable of parsing proprietary protocol exchanges.

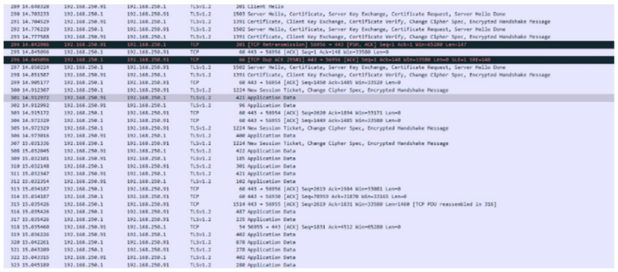

Fig. 3. Communication packets obtained using Wireshark.

To obtain a more detailed view of the encryption mechanisms, the openssl s_client utility was subsequently employed. This tool enabled the retrieval of the complete controller's certificate information as well as the cryptographic protocol parameters negotiated during the TLS handshake. The response obtained from this test is summarized in Table 2.

TABLE II. OPENSSL RESPONSE

Item	Content
Version of the TLS protocol	TLSv1.2
Cipher Suite	ECDHE-ECDSA-AES256-GCM-SHA384
Server Certificate	Self-signed certificate generated by the PLC
Distinguished Name	O = OMRON Corporation, OU = Controller Development
Validity Period	2022-07-11 to 2047-07-11 (25 years)
Handshake Result	TLS handshake failure with server Alert 40 (handshake_failure)
Failure Reason	The server requires a client certificate. The handshake failed because no certificate was provided or the provided certificate was not accepted (i.e., not signed by OMRON Root CA 1/3/4/5).
Supported Client CA	OMRON Control Development Root CA 1 / 3 / 4 / 5
Supported Signature Algorithms	RSA, DSA, ECDSA with SHA-512 / 384 / 256 / 224

The OpenSSL response revealed that the device provides a self-signed long-lived (25 years) certificate issued by OMRON. Notably, the certificate is not signed by a trusted third-party Certificate Authority (CA). The supported cipher suite, ECDHE-ECDSA-AES256-GCM-SHA384, aligns with strong encryption standards commonly adopted in industry practice. However, the session negotiated TLS 1.2 (not TLS 1.3) and reports "Extended master secret: no." as shown in Fig. 4. While TLS 1.2 with AEAD (Authenticated Encryption with Associated Data) is still acceptable, omitting EMS reduces protection against known handshake-transcript attacks and leaves a larger legacy attack surface than TLS 1.3.

```
SSL handshake has read 1455 bytes and written 458 bytes
Verification error: self-signed certificate
---
New, TLSv1.2, Cipher is ECDHE-ECDSA-AES256-GCM-SHA384
Server public key is 256 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
SSL-Session:
    Protocol  : TLSv1.2
    Cipher    : ECDHE-ECDSA-AES256-GCM-SHA384
    Session-ID:
    Session-ID-ctx:
    Master-Key: DA92F01D873F958650EA956F3F03AF0940922253C78CA78F82DFF992C5910A7D
BABE341E4
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    Start Time: 1747384880
    Timeout   : 7200 (sec)
    Verify return code: 18 (self-signed certificate)
    Extended master secret: no
```

Fig. 4. The openssl s_client output of the PLC.

IV. CYBER ATTACK ON A VIRTUAL FACTORY CONTROLLED BY PHYSICAL CONTROLLERS

A. The Test Platform

A cybersecurity test platform was constructed combining the Factory I/O warehouse scenario (see Fig. 5) with multiple physical controllers. The scenario included automated conveyors, a stacker crane, and multi-tier racks to emulate a logistics system.

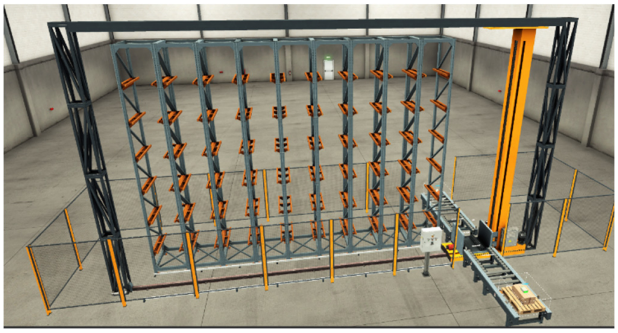

Fig. 5. The warehouse system provided by Factory I/O is used as the virtual environment for our industrial controllers.

The platform integrated a Raspberry Pi single board computer, an Omron NX102-9000 PLC, a Mitsubishi FX5U PLC, and a Siemens S7-1200 PLC, forming a multi-protocol communication environment. The Siemens S7-1200 served as the primary controller, handling the core logic and workflow management, while the other controllers interacted through their respective protocols. As illustrated in Fig. 6, the architecture combined Modbus TCP, Omron FINS, and S7 protocols. The completed platform is shown in Fig. 7.

The control workflow was managed through an HMI interface (see Fig. 8) developed in Node-RED on the Raspberry Pi. Users could input commands such as position, material infeed/outfeed, and start/stop operations. Commands were transmitted through FINS to the NX102-9000 PLC, relayed through Modbus TCP to the FX5U, and synchronized with a secondary FX5U over RS-485. The S7-1200 aggregated control data and established a cyberphysical connection with Factory I/O through the S7 protocol, thus driving the virtual warehouse.

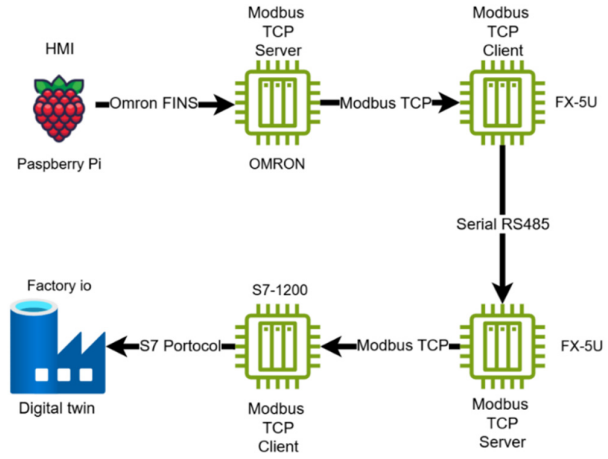

Fig. 6. The network structure of the control platform for the virtual factory.

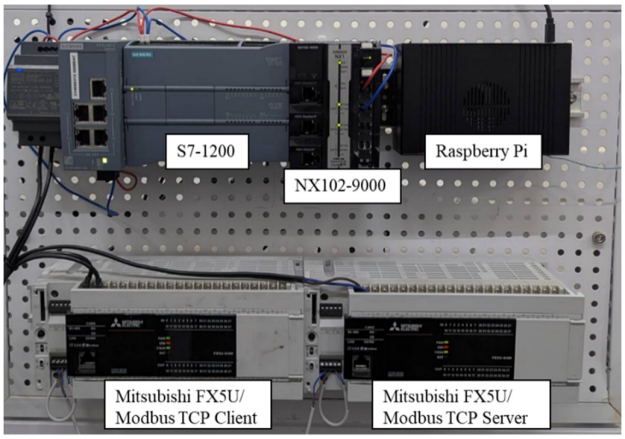

Fig. 7. The completed control system consisted of PLCs and a Raspberry Pi.

Within the HMI, the operator must first input the target storage position and then select the corresponding operation type (In/Out) from a dropdown menu, as shown in Fig. 8. The In operation indicates placing goods into the designated position, whereas the Out operation represents retrieving goods from that position. Based on the position entered and the selected operation, the system automatically issues and executes the corresponding control commands.

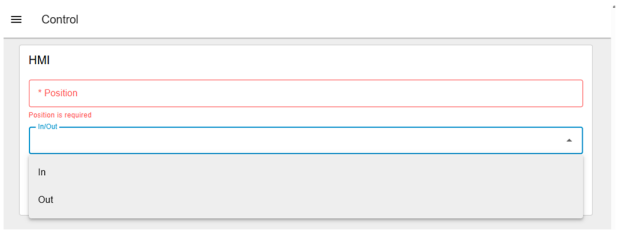

Fig. 8. The HMI of the Raspberry Pi.

After the HMI input is provided, the data are written into the designated registers (W1 and W2) of the PLC via the Omron FINS protocol (see Fig. 9). The process is described below.

- Form Node: The user inputs the target storage position and operation type through the HMI.
- Function Node ("Split input into two outputs"): The input value is parsed into two independent parameters, which are mapped to W1 and W2, respectively.
- Write W1 and Write W2: Using the FINS protocol, the parsed data are written into the PLC's W1 and W2 registers.
- Clear and 1-s Delay: Upon successful writing, the form is cleared, and the action is reset to prevent repeated triggering.
- FINS Write Nodes Connected: All FINS write nodes are shown in Fig. 9 as "connected," indicating that communication with the PLC is working properly.

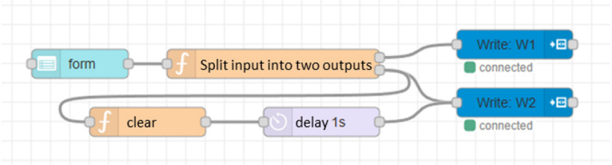

Fig. 9. The Node-Red program was executed on the Raspberry Pi.

B. Cyberattack Simulation

This study conducted a simulated cybersecurity assessment of the NX102-9000 PLC using the FINS communication protocol and validated the real-world impact within a Factory I/O simulation environment. The threat model assumes an attacker located on the same local network who can eavesdrop on and intercept packets exchanged between the PLC and the supervisory HMI. Because the FINS protocol transmits in plaintext without authentication or encryption, the attacker can perform packet replay and spoofing to inject unauthorized commands. The test procedure proceeded as follows.

1) ARP Spoofing and Packet Interception:

All controllers and devices were placed within the same subnet. An ARP scan (see Fig. 10) was performed, which identified the HMI built on Raspberry Pi at IP 192.168.2.7.

```
┌──(root㉿kali)-[~]
└─# sudo arp-scan --interface=eth1 --localnet

Interface: eth1, type: EN10MB, MAC: 00:0c:29:69:4d:3b, IPv4: 192.168.2.87
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.2.6     10:4b:46:65:7a:f7       Mitsubishi Electric Corporation
192.168.2.7     7e:0d:c6:89:04:21       (Unknown: locally administered)
192.168.2.5     e0:dc:a0:5a:55:1a       Siemens Industrial Automation Products Ltd., Chengdu
192.168.2.8     3c:f7:d1:30:6d:4b       OMRON Corporation
192.168.2.10    10:4b:46:65:89:53       Mitsubishi Electric Corporation
192.168.2.91    60:cf:84:5d:94:42       (Unknown)
```

Fig. 10. The ARP scan results.

Using arpspoof on Kali Linux, the attacker inserted itself as a man-in-the-middle so that traffic between the HMI (Raspberry Pi) and the PLC at IP 192.168.2.8 would be routed through the attacker's host. The commands used were:

arpspoof -i eth1 -t 192.168.2.7 192.168.2.8
arpspoof -i eth1 -t 192.168.2.8 192.168.2.7

This forced bidirectional traffic through the attacker, enabling the capture of FINS messages.

2) Packet Capture and Command Analysis:

Wireshark filtering on udp.port == 9600 revealed that the HMI-to-PLC communication is approximately every 0.2 s (see Fig. 11).

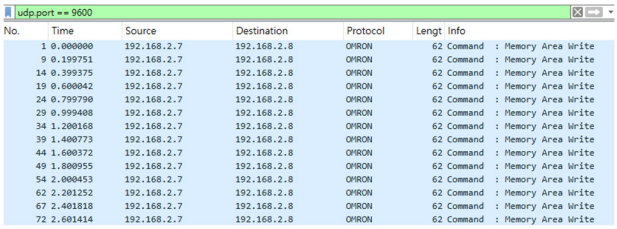

Fig. 11. The packets after filtering using udp.port == 9600.

A packet was decoded as a Memory Area Write command to address W1 (0001) with value 0006 (see Fig. 12).

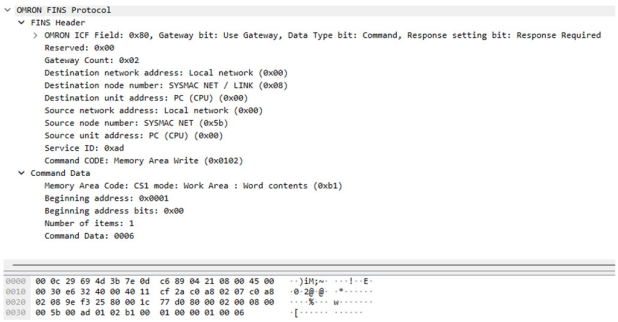

Fig. 12. Packet content captured by Wireshark.

3) Replay Attack Execution:

The attacker reconstructed the flow in Node-RED (see Fig. 13) with an inject node writing 6 into the W1 register every 0.2 s. Because the attacker continuously overwrote W1, any alternative values entered by the user via the HMI were immediately superseded by the injected value, effectively nullifying legitimate HMI commands.

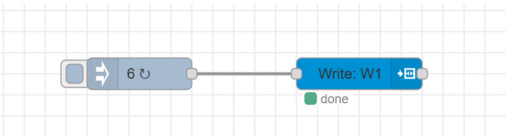

Fig. 13. Node-RED code to write the value 6 in the W1 register.

From the attack results, the following can be observed:

- Forced Data Overwrite on the PLC: Regardless of HMI operations, the value in the W1 register remained fixed at 6, indicating that control of the PLC had been hijacked by an external entity.
- Equipment Malfunction: When a storage rack slot was already occupied and the operator attempted to change the target position, the attack continuously overwrote the W1 register with the fixed value. Consequently, the controller repeatedly executed incorrect push commands, preventing the system from updating operations according to the actual requirements. Since the control logic failed to recognize that the slot was already in use, the system persistently issued loading commands, resulting in duplicate loading at the same position. This ultimately caused stored items to be displaced or dropped, triggering equipment anomalies.
- Operator Unawareness: The HMI interface displayed normal values with no error messages, preventing the operator from detecting the abnormal source of the data.

These attack results further validate that the FINS protocol lacks packet verification and access control mechanisms. Without adequate protective measures, external nodes can sustain long-term manipulation of the controller, posing substantial threats to production safety and operational stability.

C. Proposed Countermeasures

For protocols such as Omron FINS, which inherently lack authentication and encryption mechanisms, the protocol design itself cannot be directly modified. Nevertheless, several approaches can be adopted to strengthen the overall security of the transmission. The following recommendations are proposed:

- Application-Layer Proxy with Signature Verification: A forwarding proxy or gateway can be deployed in front of communication endpoints. The proxy may append fields such as a Hash-based Message Authentication Code (HMAC), timestamp, and sequence number to each packet. These fields can then be verified at the receiving end to prevent packet replay, spoofing, and command injection attacks.
- Protocol Substitution or Upgrade: Where the system architecture allows, communication protocols with built-in authentication and encryption features may be adopted. For example, OPC UA, which is supported by the NX102-9000 PLC, can serve as a more secure alternative to the FINS protocol.
- Access Control Lists (ACLs) and Whitelisting: Even if the protocol itself is insecure, firewalls and device-level whitelisting strategies can restrict communication, ensuring that only authorized hosts are permitted to send protocol-specific packets, thereby reducing the risk of forged traffic from external sources.

However, additional encryption or packet overhead may increase latency and load on PLCs, potentially affecting real-time operations. Therefore, any countermeasure must be carefully evaluated before deployment.

V. Conclusive Remarks

This research conducted a risk assessment and penetration test on the Omron NX102-9000 PLC, a widely used controller in ICS. Using Kali Linux tools, a vulnerability scan identified open ports, while further analysis revealed TLS-encrypted but proprietary communications. A simulation platform integrating multiple controllers (Siemens, Mitsubishi, Omron, Raspberry Pi) and protocols (Modbus TCP, RS485, S7, FINS) was built with Factory I/O virtual warehouse, enabling a cyberphysical environment. Within this framework, a packet replay attack demonstrated successful hijacking of PLC operations. The observed anomalies highlight significant risks that arise from insecure protocols. Countermeasures were proposed to mitigate these risks while emphasizing the importance of balancing security with real-time operational requirements.

References

[1] D. M. Thomas, N. Pandey, V. K. Shukla, and A. V. Singh, "Attack Vectors and Susceptibilities of the Modbus in TCP/IP Model," presented at the 2021 9th International Conference on Reliability,

Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2021.

[2] J. Luswata, P. Zavarsky, B. Swar, and D. Zvabva, "Analysis of SCADA Security Using Penetration Testing: A Case Study on Modbus TCP Protocol," in 2018 29th Biennial Symposium on Communications (BSC), 6-7 June 2018, pp. 1-5.

[3] Z. Tian, W. Wu, S. Li, X. Li, Y. Sun, and Z. Chen, "Industrial Control Intrusion Detection Model Based on S7 Protocol," in 2019 IEEE 3rd Conference on Energy Internet and Energy System Integration (EI2), 8-10 Nov. 2019, pp. 2647-265.

[4] P.-H. Huang, J. Kim, P. R. Kumar, J. Rajendran, and P. Enjeti, "Enhancing Cybersecurity for Industrial Control Systems: Innovations in Protecting PLC-Dependent Industrial Infrastructures," IEEE Internet of Things Journal, vol. 11, no. 22, pp. 36486-36493, 2024.

[5] W. Zhang et al., "Armor PLC: A Platform for Cyber Security Threats Assessments for PLCs," Procedia Manufacturing, vol. 39, pp. 270-278, 2019.

[6] R. Lanotte, M. Merro, and A. Munteanu, "Industrial Control Systems Security via Runtime Enforcement," ACM Trans. Priv. Secur., vol. 26, no. 1, pp. 1-41, 2022.

[7] A. Ocaka, D. O´ Briain, and K. Barrett, "Evaluating the Impact of Cyberattacks on PLC Performance: A Systematic Implementation and Empirical Investigation," IFAC-PapersOnLine, vol. 58, no. 3, pp. 387-392, 2024.

[8] H. Wardak, S. Zhioua, and A. Almulhem, "PLC access control: a security analysis," in 2016 World Congress on Industrial Control Systems Security (WCICSS), 12-14 Dec. 2016, pp. 1-6.

[9] G. Bonney, H. Höfken, B. Paffen, and M. Schuba, "ICS/SCADA security analysis of a Beckhoff CX5020 PLC," in 2015 International Conference on Information Systems Security and Privacy (ICISSP), 9-11 Feb. 2015, pp. 1-6.

[10] H. Cui, J. Hong, and R. Louden, "An Overview of the Security of Programmable Logic Controllers in Industrial Control Systems," Encyclopedia, vol. 4, no. 2, pp. 874-887, 2024.

[11] X. Low, D. Yang, and D. Yang, "Design and Implementation of Industrial Control Cyber Range System," presented at the 2022 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2022.

[12] M. M. Aslam, A. Tufail, R. A. A. H. M. Apong, L. C. De Silva, and M. T. Raza, "Scrutinizing Security in Industrial Control Systems: An Architectural Vulnerabilities and Communication Network Perspective," IEEE Access, vol. 12, pp. 67537-67573, 2024.