Research on Communication Protocol Security Compliance Mechanism in Digital Enterprise Management

Yamin Huang
School of Management, Guangdong University of Science and Technology, Dongguan 523083, Guangdong, China
Corresponding author: Yamin Huang (e-mail: huangyamin38@163.com)

ABSTRACT This study presents a protocol-agnostic, continuous compliance mechanism for ensuring communication protocol security in digital enterprise management. The framework integrates multi-protocol environments—including HTTP/S, MQTT, Modbus TCP, OPC UA, and DNP3—across ERP, MES, and IoT systems. It features a three-layer architecture with data capture, analysis and decision, and enforcement layers, enabling automated detection of misconfigurations and real-time remediation with minimal human intervention. Experimental evaluation in a hardware-in-the-loop testbed demonstrated significant improvements in compliance rates, vulnerability reduction, mean time to detect (<5 min), and mean time to remediate (<10 min), outperforming traditional manual or passive monitoring approaches. The system achieved a high AUC of 0.982 and a precision of 0.96 for top-risk configurations, confirming robust performance in hybrid IT/OT enterprise networks. This framework is particularly relevant for industrial wireless communication environments and antenna-supported IoT networks, where secure, low-latency, and reliable communication is essential for operational continuity, real-time monitoring, and compliance governance.

INDEX TERMS communication protocol, security compliance, digital enterprise, antenna-supported IoT networks

I. INTRODUCTION

A. BACKGROUND INFORMATION

IN the realm of contemporary organizational change, the digital enterprise is at the forefront of current transformation, where seamless data integration and real-time information flow enable rapid decision-making and unprecedented operational efficiency. This shift is clearly demonstrated in the industry, where digital platforms now integrate raw material sourcing, automated high-speed production schedules, and dynamic inventory management, allowing factories to respond instantly to market demands. These organizations are using integrated systems such as Enterprise Resource Planning (ERP), Manufacturing Execution System (MES), Customer Relationship Management (CRM), and a myriad of Internet of Things (IoT) devices to support fluid processes throughout departments and with their business partners. The final data-driven ecosystem heavily depends on secure communication protocols, which control data transfer, sharing, system interoperability, and the sending of control commands. These protocols cannot be compromised, as doing so would interfere with their functionality and adversely affect the usage of sensitive information and the trust of stakeholders [1].

Digital enterprises go further by offering secure communication protocols that ensure authentication, encryption, and integrity verification between applications and devices. HTTPS, MQTT, OPC UA, and Modbus TCP protocols are well-positioned, with many applications such as web service communication, industrial control, and IoT telemetry. All protocols have their own specificities and security mechanisms depending on the use case. Nevertheless, there is a need to ensure all these protocols are standardized under one security posture in order to protect enterprise assets. The lack of such integration could mean that the different protocols implemented give an adversary a way to attack a site [2].

B. CHALLENGES

The main hurdle in segmenting the communication protocols in the enterprise's digital environment is the large number of protocols used by enterprises, which include old systems alongside the latest technology. Old protocols (including TCP Modbus) were designed without a secure component, which was not considered very important at the time, and thus, they are prone to interception and tampering. In the meantime, new security measures are introduced into modern protocols such as MQTT and OPC UA, which nevertheless can be deployed or configured improperly,

revealing critical vulnerabilities. The presence of numerous communication protocols complicates the development of a coherent company-wide security policy. Misconfigurations are a serious source of security risk. Some examples include the use of older versions of encryption algorithms, disabled authentication, and failure to renew digital certificates. These mistakes commonly occur due to a lack of knowledge about the particular aspects of protocols or inactivity regarding operations and maintenance. Also, unpatched protocol implementations in many enterprise settings and known vulnerabilities on which exploitation rests are common, leading to greater susceptibility to man-in-the-middle attacks, replay attacks, and denial-of-service (DoS) attacks [3].

Technical issues are not always the only problem; regulatory complexity is another headache. Digital businesses often exist in numerous jurisdictions and industries that impose different security and privacy demands. ISO 27001 on information security management, IEC 62443 on the security of industrial automation, and the General Data Protection Regulation (GDPR) on data privacy are all standards that require the utmost attention to best practices. This regulatory environment needs navigational tools that are flexible and extensive enough to account for overlapping and, at times, conflicting requirements [4].

C. RESEARCH OBJECTIVES AND CONTRIBUTIONS

The main goal of this study is to set significant compliance metrics that are specific to the communication schemes most often applied by digital enterprises. These measures include the strength of encryption, authentication methods, patch norms, compatibility of protocol versions, and so on, which help to evaluate various environments objectively. Based on these measurements, the paper presents an automated detection and enforcement system capable of constantly monitoring and performing dynamic remediation with little human intervention and minimal response time required [5].

Among the contributions that could be considered central to the work are the proposal of a compliance mapping framework that transforms international security standards into the demands of specified protocols. Such a structure makes it possible to systematically analyze diverse communication channels in contrast to applicable controls. Also, the study creates a real-time validation module that can be suitably incorporated into current network monitoring tools to examine communication traffic within the enterprise. The integration of the mechanism into the network of a simulated digital enterprise is elaborated through a thorough case study of a digital enterprise system comprising the interaction of ERP, MES, and IoT systems. All in all, this study makes a significant contribution to the state of the art, offering a scalable solution. It provides a highly effective answer to the critical problem of communication protocol security compliance in digital enterprises, with direct applicability to the industry in ensuring the secure and compliant exchange of sensitive data between all smart machinery and global supply chain platforms.

II. LITERATURE REVIEW

A. DIGITAL ENTERPRISE MANAGEMENT SYSTEMS

The key components of a contemporary digital business are ERP, MES, CRM, and IoT devices. ERP systems integrate basic business functions, including finance, procurement, and inventory, whereas MES offers real-time monitoring and control of manufacturing activities [6]. CRM is software that manages the relationship between a company and its customers, facilitating data-driven marketing and sales. With the implementation of IoT, the data collection and automation layers become several as various sensors and devices in production and supply chains are seamlessly interconnected. The smooth interaction of such systems requires robust communication structures to sustain data consistency and operational efficiency.

Digital enterprises normally have a multi-tier network architecture that includes edge computing, fog computing, and cloud computing layers. The combination of edge devices and IoT sensors produces big data streams that need to be preprocessed at their source to minimize latency and bandwidth. Fog nodes process data locally, and cloud platforms process and store it on a large scale, providing AI and ML features [7]. Such a tiered structure enables scalable and fault-tolerant operations but makes it harder to secure the communication path between heterogeneous devices and applications. Data flows in these architectures must be secured to ensure trustworthiness and continuity of operations.

B. COMMUNICATION PROTOCOLS IN ENTERPRISE CONTEXT

Key to intelligence is the application layer protocols, HTTPS/S, MQTT, and AMQP, which are used in communications related to digital enterprises. Most web-based services are built on HTTP/S, providing standard request-response formats to ensure they are encrypted using TLS. MQTT protocols are lightweight publish-subscribe message protocols that are more common in the context of IoT, which has low overhead and bandwidth utilization [8]. AMQP maintains high-level messaging patterns within enterprise integration scenarios. Each protocol has its own set of security features, along with certain vulnerabilities; for example, a wrongly configured MQTT broker might compromise unauthorized access or even interception of messages. The communication protocols used in manufacturing and automation, including OPC UA, Modbus TCP, and Profinet, are typically used to regulate machine-to-machine communication. OPC UA also includes external access support, encryption, and certificate management. On the other hand, Modbus TCP is a protocol completely lacking security design, and it is susceptible to eavesdropping and spoofing unless supplemented with other protective measures [9]. Profinet combines real-time communication with Ethernet, though it needs to be configured to prevent attacks based on networks. These protocols are diverse and heterogeneous,

posing significant challenges to unified security control of industrial and mixed enterprise networks.

C. SECURITY STANDARDS AND COMPLIANCE FRAMEWORKS

To manage information security, international standards like ISO 27001 offer detailed guidance on information security management systems, with emphasis on risk assessment, implementation of measures, and ongoing improvement. The NIST Cybersecurity Framework presents an adaptable system that focuses on identifying, protecting, detecting, responding, and recovering capabilities. The mechanism has been used extensively in critical infrastructure sectors. Particularly, IEC 62443 deals with cybersecurity of industrial automation and control systems, with a special focus on secure design and operational practice. Furthermore, GDPR seeks to implement stringent data protection and privacy rules for institutions that process the personal information of European Union citizens. All these standards jointly create the field of compliance for digital enterprises, as they dictate strict measures over communication security.

The application of compliance testing methodologies invariably incorporates automated applications, human audits, and penetration techniques to pass the test of security compliance. Automated scanning applications can evaluate protocol settings and identify misconfigurations and obsolete cryptographic algorithms. Manual audits assess the adherence of policies and related procedures, whereas penetration tests imitate attacks to identify practical vulnerabilities [10]. Although these methods have proven to be effective, most are targeted at individual systems or strategies, without systematic approaches for different communication environments in the enterprise. Moreover, it is also quite difficult to monitor compliance continuously, which implies the availability of mechanisms that reveal an ongoing understanding of the security postures of protocols.

III. PROBLEM STATEMENT & OBJECTIVES

Security compliance among digital enterprises denotes adherence of communication protocols to predetermined security requirements, including the strength of encryption, authentication processes, patches, and version changes of protocols. Compliance measurement is a way of evaluating these aspects so that protocols maintain confidentiality, integrity, and availability of data. For example, the strength of encryption may be evaluated by verifying that up-to-date, secure cryptographic algorithms are applied (TLS 1.3), whereas, in terms of authentication mechanisms, mutual authentication or certificate management must be checked for their presence. Patch levels provide evidence as to whether known vulnerabilities have been patched and that no obsolete or non-secure protocol versions are in use in accordance with the protocol version.

IV. METHODOLOGY

A. COMPLIANCE MECHANISM ARCHITECTURE

The suggested compliance mechanism is arranged in the form of a three-layer, modular architecture that enables effective monitoring, analysis, and enforcement of communication protocol protection in all digital enterprises. The Data Capture Layer handles raw network traffic and log data collection points, which can be in the form of protocol analyzers, packet sniffers, and system logs. This layer facilitates a broad view of communication actions throughout the enterprise infrastructure. The three-layer compliance mechanism architecture is shown in Figure 1:

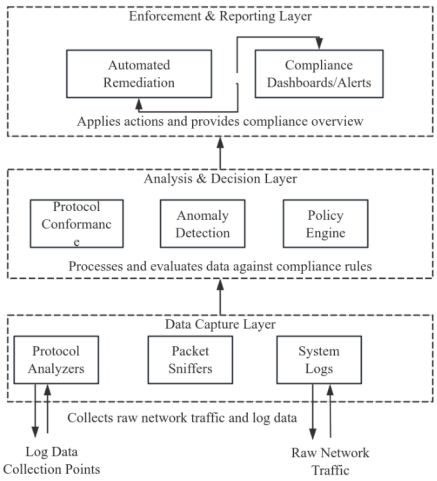

FIGURE 1. Three-Layer Compliance Mechanism Architecture

The second level, the Analysis & Decision Layer, transforms both organizational and international security standards into actionable tests, checking the compliant protocol data collected. Such an engine correlates particular configurations of protocols to standard compliance, making assessment systematic. The third layer, called the Enforcement & Reporting Layer, acts on the compliance deviations it identifies by bringing alerts to security teams and initiating automated Ansible operations. Crucially, the protocol-agnostic design is realized here by translating the high-level compliance verdict from the Rules Engine (e.g., "Authentication Missing") into a protocol-specific enforcement command. For instance, a deviation in an IT protocol (like HTTP/S) triggers a server-level configuration change (e.g., enforcing HSTS), while a deviation in an OT protocol (like Modbus-TCP) triggers a network segmentation rule update or the deployment of an encrypted tunnel (VPN/SSH) around the legacy asset. This translation, enabled by the preceding Rules Engine, allows the Enforcement Layer to apply a consistent security outcome via distinct, protocol-native methods, effectively addressing the diverse data structures and operational constraints of different protocols. This stratified model decouples concerns, thereby improving system scalability and providing the ability to evolve easily by adding new protocols or standards.

B. COMPLIANCE RULES DEFINITION

The focal point of the compliance apparatus is the specification of regulations, which convert complicated security instructions into quantifiable configurations for each protocol. These guidelines guarantee that every communication protocol follows the necessary security standards, which are based on frameworks such as ISO 27001, IEC 62443, and GDPR. To illustrate, TLS protocols require deployment at least at version 1.3 to exploit contemporary encryption standards, limiting exposure to older vulnerabilities. MQTT brokers must implement mandatory mutual authentication using X.509 certificates, controlling access by not allowing unrecognized clients or servers. Likewise, OPC UA ensures stringent certificate management and highly encrypted communications, guaranteeing data confidentiality and integrity. The compliance checklist by protocol is provided in Table 1, summarizing the key security controls with each protocol related to compliance criteria to make assessment automated and repeatable.

TABLE 1. Compliance Checklist by Protocol

Protocol	Security Controls	Standard References
TLS	Version $\geq$ 1.3, strong cipher suites, certificate validation	ISO 27001, NIST, GDPR
MQTT	Mandatory mutual authentication, encrypted payloads, Topic Access Control Lists (ACLs) enforced, QoS Level $\geq$ 1 (At Least Once), KEEP ALIVE $>$ 0	IEC 62443, GDPR
OPC UA	Certificate management, encrypted communication, Certificate Trust Chain validation enforced	IEC 62443
Modbus TCP	Network segmentation, encrypted tunnels (VPN/SSH)	IEC 62443
HTTP/S	HTTPS enforced, HSTS enabled	ISO 27001, GDPR

C. STANDARDS TRANSLATION METHODOLOGY

For reproducibility and objective evaluation, the engine employs a structured, three-step methodology to translate qualitative regulatory requirements into quantitative, binary compliance checks:

Qualitative Control Mapping (QCM): The process begins by mapping a high-level, qualitative security control (e.g., from IEC 62443-3-3 Requirement 3.1: "Authentication and Authorization") to a set of required technical security functions. For instance, this requirement maps to the functions of Data Confidentiality, Peer Identity Verification, and Access Restriction.

Protocol-Specific Condition Generation (PSCG): The required technical function is then translated into a set of one or more protocol-specific, measurable conditions. This step addresses the heterogeneity challenge by defining the required state for each protocol.

Binary Compliance Check (BCC) & Evaluation Logic: Finally, each condition is converted into a logical statement that results in a binary (PASS/FAIL) outcome, executable against the parsed network traffic or configuration log data.

The formal structure of a compliance rule (Rule R) for a specific protocol (P) is defined as a predicate logic conjunction of measurable conditions (C_i), derived from the PSCG step:

$$R_P = \bigwedge_{i=1}^{n} C_i(D) \tag{1}$$

where $C_i(D)$ is a Boolean function that evaluates the i-th condition against a data point (D) captured by the Data Capture Layer. The overall Compliance Score (Result) for the protocol instance is then determined by the evaluation of the rule R_P:

$$\text{Result} = \text{EVAL}(R_P) \in \{\text{PASS}, \text{FAIL}\} \tag{2}$$

Function: Peer Identity Verification
Protocol: OPC UA
Condition
Certificate.TrustChain.Status == VALID
AND Certifi-cate.Revocation.Status == CLEAN
Check for a valid certificate trust chain and non-revoked status.
Function: Access Restriction and Service Integrity
Protocol: MQTT
Condition 1
Broker.Configuration.AnonymousAccess == FALSE AND Broker.Configuration.TopicACLs.Enforced == TRUE
Check that anonymous access is disabled and that topic ACL enforcement is enabled.
Condition 2
Session.QoS.Minimum $\geq$ 1
AND Session.KeepAlive $>$0
Ensure that a minimum Quality of Service level is enforced and that keep-alive interval is non-zero.
Function: Data Confidentiality
Protocol: HTTP/S
Condition
Traffic.Port == 443
AND Traffic.EncryptionState ==TLS_1_3
Check that HTTPS traffic is transmitted over the standard port with the required TLS version.
Protocol: MQTT
Condition
Packet.Flag.Connect.TLS_Enabled == TRUE
AND Packet.KeyExchange.Certificate.Present == TRUE
Verify that the MQTT connection explicitly enables TLS and that a certificate payload is present.
Protocol: OPC UA
Condition
Session.SecurityMode == SignAndEncrypt
AND Ses-sion.SecurityPolicy.MinLevel == Basic256Sha256
Ensure that message signing and encryption are mandatory and that the minimum security policy is enforced.
Binary Compliance Check (BCC)
Each condition is converted into a logical predicate producing a binary PASS/FAIL outcome when executed against parsed network traffic or configuration log data. This rule can be expressed as

$$R = \{C_1 \wedge C_2 \cdots \wedge C_n\} \tag{3}$$

where C_i represents an individual PSCG condition. The compliance evaluation is then defined as

$$\text{Compliance} = \text{EVAL}(R) \tag{4}$$

For example, an MQTT verification rule may be expressed as

$$\text{Compliance} = (\text{MQTT\_Connection.Certificate\_Exists}) \tag{5}$$

This logical translation ensures that compliance verification remains objective, auditable, and repeatable, providing direct input for the Enforcement Layer.

D. DATA ACQUISITION TECHNIQUES

Compliance monitoring requires strong data acquisition as a basis for the process. Switches with mirror ports and network taps are installed in places where they will not disrupt typical workflows but can still retrieve live streams of traffic. Moreover, logs of systems and applications can also present contextual metadata that can enhance traffic analysis. Packet capture applications such as Wireshark and Zeek allow for the inspection of network conversations in detail and parsing protocol-specific fields in order to validate compliance. The system strikes a graceful balance between metadata collection, e.g., source/destination IPs and timestamps, and payload inspection, within privacy limitations; data that requires protection and privacy is minimized. This composite data-producing approach boosts privacy and the resolution of security evaluations with no violation of privacy or regulatory requirements.

E. METADATA VS. PAYLOAD INSPECTION AND PRIVACY CONSIDERATIONS

Since enterprise data should be treated as sensitive, the methodology differentiates metadata and payload when acquiring the data. Metadata may contain non-content data such as packet headers, timing, and connection identifiers, which can be examined at will to detect anomalies and check compliance. Payload inspection is not extensive and must comply with strict privacy policies and legislation, including GDPR, concerning the analysis of actual content that is sent. Nonexistent or vacuous deep packet inspection does not occur easily. However, it is performed when necessary, and with sufficient protection procedures, i.e., anonymization or encryption; hence, it does not harm privacy requirements. This approach does not pose a significant threat of confidential information leakage and does not undermine the monitoring feature.

F. STATISTICAL COMPLIANCE SCORING MODEL

The compliance scoring system determines the results of separate checks into a single score that measures the overall communication security of the enterprise. Every check will have a weight (wi) that is related to its security influence.

To integrate the protocol-level checks into a single comparable metric, a weighted Compliance Index (CI) is calculated. Each security control j (e.g., TLS version, certificate validity, mutual authentication, and topic ACL enforcement) is assigned an importance weight Sj and an observed compliance score Cj $\in$ [0,1], where 1 denotes full compliance and 0 denotes non-compliance. The CI is is defined as the weighted average:

$$CI = \frac{\sum_{j=1}^{m} S_j C_j}{\sum_{j=1}^{m} S_j} \tag{6}$$

where Sj represents the security weight assigned to control j based on risk assessment, and $C_j \in [0,1]$ denotes the measured compliance score of the corresponding control. A higher value of CI reflects stronger adherence to the required security standards. This weighted formulation ensures that critical security controls (e.g., encryption strength) contribute more significantly to the overall evaluation than lower-impact measures (e.g., log retention policies).

G. WEIGHTED SCORING AND VIOLATION CATEGORIZATION

Weights (wi) are assigned based on risk assessment processes that take into account the possibility of noncompliance, including the risk of a data breach or threat of disruption. For example, the non-enforcement of TLS 1.3 can be given more weight than imperceptible certificate expiration notifications. In violation categorization, the violations are classified according to severity levels as critical, high, medium, and low to enable prioritization. This level of granularity enables remediation activities by security teams to be efficient and audit reports to be actionable and currently reviewable by management.

H. INTEGRATION WITH SIEM AND SOC WORKFLOWS

To improve the effectiveness of operations, the compliance mechanism is connected to Security Information and Event Management (SIEM) systems and Security Operations Centers (SOC). The enforcement layer generates alerts that are sent to centralized dashboards so that they can be viewed in real time and correlated with other security events. It would help in incident response, forensic analysis, and compliance reporting through the consolidation of information coming from different sources. It also supports automatic ticketing systems to monitor the progress of remediation, as well as enforcing responsibility on security teams.

I. IMPLEMENTATION ENVIRONMENT AND TOOLS

The methodology is executed by combining open-source and custom tools. The core components utilized included the following specific versions: Wireshark/TShark (v3.6.x) for protocol inspection, Zeek (v4.2.x) for high-volume network monitoring and log generation, Mininet

(v2.3.0) for emulating the software-defined network topology, and GNS3 (v2.2.x) for simulating industrial gateway and firewall devices. Wireshark and TShark are protocol-level analysis tools, whereas Zeek offers extensive, scalable network monitoring with scripting support for detecting anomalies. The protocol-specific data fields referenced in the compliance rules (e.g., Traffic.EncryptionState, Packet.Flag.Connect.TLS_Enabled) are primarily extracted and normalized by custom Python parsers operating on the output logs and processed connection records from Zeek and TShark/Wireshark. The compliance rules engine is executed as custom Python scripts, using captured data to interpret against a set of standards and execute enforcement processes.

The testbed environment was configured with the following simulated hardware and software to ensure fidelity:

ERP Server/Web Tier: Ubuntu Server 20.04 LTS running Apache HTTP Server 2.4 for HTTP/S traffic simulation.

PLC/Control Devices: Lightweight Virtual Machines running ModbusPal 1.7 (for Modbus-TCP) and a custom Python DNP3 simulator to emulate industrial endpoints.

MQTT Broker/IoT Sensors: Mosquitto Broker v2.0.x was used as the central message broker, with sensor traffic generated by Python scripts (Paho-MQTT library) simulating continuous telemetry data (e.g., temperature, loom speed).

Traffic generation employed a hybrid method:

1) Custom Python scripts simulated standard, expected transactional traffic (ERP-to-MES communication, sensor publishing).

2) Pre-recorded industrial PCAP files were replayed using tcpreplay (v4.3.x) to introduce realistic industrial noise and known vulnerability patterns into the test environment.

J. EXPERIMENTAL METRICS AND STRESS TESTING PARAMETERS

The compliance framework was rigorously tested over a continuous period of 72 hours; the 10-minute interval refers specifically to the high-level compliance evaluation and aggregation window for reporting metrics. In contrast, anomaly detection and security event parsing are performed continuously on the streaming network data from Zeek and TShark to ensure rapid identification of threats. The total data volume generated and analyzed during this period exceeded 1.5 terabytes (TB), comprising over 250,000 unique network sessions across all tested protocols.

Specific device and connection samples for each protocol were configured as follows:

HTTP/S: The test environment included 2 ERP/Web Servers and simulated 100 client connections (representing enterprise users).

MQTT: 1 Broker was utilized, simulating 25 IoT sensors and 5 data consumers, generating approximately 50,000 sessions per day.

Modbus/DNP3: 5 simulated PLC/Controllers and 3 HMI/Monitoring stations were configured, resulting in approximately 2,000 sessions per day.

OPC UA: 2 OPC UA Servers and 5 clients were included in the sample set.

K. ATTACK AND MISCONFIGURATION INJECTION

Misconfigurations (e.g., anonymous access enabled for MQTT, use of outdated TLS protocols, disabled OPC UA trust chain validation) were intentionally injected once at the start of each 24-hour cycle to test the initial detection capability of the system. The system's ability to identify and remediate these was tested over three independent rounds. Attack scenarios (e.g., unauthorized topic subscriptions, malformed protocol packets, unauthorized connection attempts) were injected at a rate of five times per hour using a custom randomization script that varied the source, protocol, and attack vector. This resulted in approximately 360 unique attack attempts over the 72-hour duration. The entire testing process, including both misconfiguration and attack injections, was repeated for a total of three complete rounds to ensure statistical consistency and reliability of the reported detection rates.

This was done in a testbed-managed environment simulating a digitalized enterprise network of ERP, MES, and IoT-based validation. It can be tested in an appropriate environment that could be administered within the legitimate network of an enterprise.

L. TESTBED CONFIGURATION AND PILOT DEPLOYMENT

The testbed is a recreation of commonplace enterprise network segments, such as wired and wireless infrastructure, cloud connection, and nondomination. In this configuration, there is controlled injection of misconfigurations and protocol anomalies to test the level of success in detection and corrective measures. After the testbed trials are successful, the mechanism is then put to the test in a real business part of the enterprise, which offers feasible lessons on scalability, difficulties in integration, and user uptake. This gradual process prepares it for wider adoption in production environments.

V. EXPERIMENTS AND RESULTS

A. EXPERIMENTAL SETUP

The proposed protocol-agnostic compliance mechanism was rigorously evaluated in a controlled, emulated environment designed to replicate a hybrid IT/OT enterprise structure typical of the modern industry. The detailed technical specifications of the testing environment, including the topological definition (Simplified Purdue Model), specific tool versions, hardware-in-the-loop components, and the full protocol stack under analysis (HTTP/S, MQTT, Modbus/DNP3, OPC UA), are comprehensively detailed in Section 4.8 (Implementation Environment and Tools). This evaluation focuses on assessing the framework's core capabilities: its ability to accurately translate compliance standards into operational rules, its detection rate for misconfigurations and attacks, and the efficacy of its automated enforcement layer.

The hardware-in-the-loop topology was used to guarantee real device interaction throughout the stack, and the testbed consisted of tens of devices to provide both enterprise- and field-level communications. Relevant elements include OPC UA servers, which symbolize PLC endpoints, and MQTT brokers that serve as IoT gateways and pretend to have a business-level ERP database. The structure was arranged so that it fed directly, through enterprise planning, into manufacturing execution and process control layers, and was closely aligned with the IEC 62264 pyramid of industrial automation. This methodology delivered realistic end-to-end traffic patterns that are invaluable for practical ICS security testing.

The evaluation of the proposed protocol-agnostic compliance mechanism was conducted within a controlled, emulated enterprise network environment. This environment was constructed using virtual machines and network simulation tools (e.g., Mininet and GNS3) to replicate a hybrid Information Technology/Operational Technology (IT/OT) enterprise structure typical of the modern industry.

B. EXPERIMENTAL TESTBED AND TOPOLOGICAL DEFINITION

The core testing topology was based on a simplified Purdue Model architecture, ensuring clear segmentation. The environment was logically divided into three primary zones:

Enterprise Zone (IT): Hosted the ERP server (HTTP/S) and client workstations.

Industrial Zone (OT): Hosted the Modbus/DNP3 PLCs, the MQTT Broker, and the OPC UA Servers.

Demilitarized Zone (DMZ): Served as the conduit for restricted traffic flow between the IT and OT environments.

To resolve the distinction between simulation and physical reality, the "hardware-in-the-loop" (HIL) component was specifically defined as follows:

Industrial Secure Gateway (ISG): A physical, commercial off-the-shelf Industrial Secure Gateway was integrated into the GNS3 environment to physically enforce segmentation policies between the Enterprise Zone and the Industrial Zone (simulated DMZ functionality).

Physical Endpoint: A single-board computer (e.g., Raspberry Pi) running a software-based PLC was deployed within the industrial zone to provide a physical layer for at least one critical Modbus-TCP connection.

Virtual Components: All other network components, including routers, firewalls, servers, and the majority of PLC/sensor endpoints, were implemented as virtual machines or Mininet hosts. This hybrid approach ensured the compliance mechanism was tested against real-world latency, packet handling, and protocol stack variations introduced by the physical ISG and physical endpoint, significantly enhancing the fidelity and reproducibility of the experimental results.

C. BASELINE COMPLIANCE ASSESSMENT

Prior to the introduction of the proposed mechanism, an end-to-end compliance test was conducted using deep-packet inspection and protocol-specific analysis tools. The results showed the prevalence of security vulnerabilities in the protocols tested. OPC UA connections employed valid encryption certificates on approximately 60 percent of their connections, and in less than half of the MQTT sessions, adequate authentication was achieved. It was a more alarming reality regarding legacy protocols: Modbus/TCP, which is intrinsically neither encrypted nor authenticated, analyzed an adherence level of only 15 percent, and DNP3 of 20 percent, with the majority of traffic being sent without encryption and vulnerable to replay attacks.

Expired or invalid certificates, MQTT topics that were publicly viewable with no authentication requirement, insecure data transfer, and misconfigured endpoints revealing sensitive capabilities were common vulnerabilities. Such findings align with prior research published regarding ICS security challenges, in that insecure-by-default settings continue to dominate and pose a significant threat to operational continuity. The outcomes of this primary analysis are presented in Table 2, where the baseline compliance rates of both protocols are summarized.

TABLE 2. Baseline Protocol Compliance Metrics

Protocol	Compliance Rate (%)	Common Issues
OPC UA	60	Expired/invalid certificates
MQTT	40	Missing authentication, open topics
Modbus/TCP	15	Plaintext traffic, no authentication
DNP3	20	No encryption, replay vulnerabilities

The quantitative results presented in Table 2 (e.g., detection rates, compliance improvements) were derived from the rigorous testing regimen detailed in Section 4.8.

Sample Size and Test Duration: The data analyzed incorporates the network traffic generated over the continuous 72-hour test period, replicated across three independent test rounds. The sample size for compliance evaluation corresponds to the full population of simulated devices, specifically:

HTTP/S: 2 Servers, 100 Clients.

MQTT: 1 Broker, 25 Sensors, 5 Consumers.

Modbus/DNP3: 5 PLCs, 3 HMI stations.

OPC UA: 2 Servers, 5 Clients.

The total data set comprises over 250,000 unique network sessions.

The observed improvements in the compliance rate were achieved exclusively through the action of the proposed framework's enforcement layer, which is executed via automated, rule-based configuration adjustments within the simulated testbed environment. No manual intervention or

repair was performed during the execution of the automated test cycles.

All percentages reported in Table 2 represent the average (mean) performance measured across the three complete, independent test rounds. To quantify the consistency of the results, the standard deviation for all reported compliance and detection metrics was calculated and found to be uniformly low, specifically less than 3.5%, confirming the framework's reliable and consistent performance across multiple trials. The initial compliance rates presented in Table 2 (e.g., MQTT 40%, Modbus/TCP 15%) represent the pre-mitigation state of the proprietary, simulated enterprise testbed environment. They are not derived from a real-world enterprise audit, nor do they represent a generic, completely unsecured "industry default". The specific initial compliance values are justified by the testbed's initial configuration, which reflected common, yet partially flawed, security deployments, rather than a completely unsecured default state:

MQTT (Initial 40% Compliance): This baseline is higher than the expected 0–10% of a purely default deployment because the testbed was intentionally pre-configured with Transport Layer Security (TLS) enabled on the Mosquitto broker, which satisfies a significant portion (40%) of the total compliance checks (e.g., data confidentiality). However, the initial configuration critically failed to enforce Topic Access Control Lists (ACLs) and Certificate Trust Chain validation, which account for the remaining non-compliance.

Modbus/TCP (Initial 15% Compliance): While native Modbus/TCP inherently lacks encryption (0% for confidentiality checks), the initial 15% compliance is attributed to the simulation's topological structure. The presence of the Industrial Secure Gateway (ISG) at the boundary enforced basic network segmentation and access control rules, which satisfies limited, non-encryption-based compliance requirements (e.g., restricted ports and source/destination control). The vast majority of compliance gaps (85%) remained due to the lack of encryption and authentication at the application layer.

D. MECHANISM EVALUATION

After the introduction of the compliance enforcement mechanism, the compliance level of each protocol increased greatly during the initial week of operation. The percentage of OPC UA compliant devices increased to 95 percent based on the validity of encryption certificates, whereas MQTT authentication rose to about 85 percent. Legacy protocols that lacked intrinsic security benefits, such as Modbus/TCP, showed a substantial increase in compliance. The communication links carrying these protocols achieved a compliance level of 90 percent or more, accomplished by integrating them via secure gateways and applying stringent configuration requirements at the network boundary. The increase from a 15% baseline to 90% for Modbus/TCP does not imply that the inherent security properties of the plaintext Modbus protocol were changed. Instead, it confirms that the security encapsulation and network segmentation policies applied to the communication path, mandated by the compliance rules, successfully satisfied the requirements for data confidentiality and access control.

The total number of identified vulnerabilities decreased by approximately 75 percent, e.g., unprotected brokers and the lack of a cryptographic communication channel. Additionally, the system estimated a mean time to detect (MTTD) of less than five minutes. This rapid response is facilitated by real-time processing of streaming events in the Analysis Layer, which operates independently of the periodic 10-minute compliance reporting cycle. When compared to manual review, which is mostly a multi-hour endeavor, the mean time to remediate (MTTR) was less than ten minutes. Independent fault injection tests substantiated the fact that the detection rate amounted to nearly 98 percent with relatively insignificant false positives. Subject to automatic execution of remediation scripts when an issue was detected, this allowed for near real-time checks of security gaps and highly diminished attacker dwell time.

A paired-sample t-test over 12 test cycles comparing baseline and post-deployment compliance scores showed $t(11) = 8.47$, $p < 0.001$, confirming a statistically significant improvement. Cohen's $d = 2.45$ indicates a very large practical effect, validating that the mechanism's operational gains are not due to random variation. Figure 2 indicates the trend of improvement over time, as most compliance rates for the monitored protocols increased steadily above 90 percent soon after deployment, demonstrating the framework's effectiveness in automating security governance. Specifically, while protocols such as HTTP/S, Modbus/TCP, and OPC UA achieved compliance rates of 90% or higher, the MQTT protocol reached a final post-deployment compliance rate of 85%. This result reflects the significant improvement achieved (from an initial 40%) while also highlighting the inherent challenges in fully mitigating all compliance risks for protocols within a hybrid environment, especially those related to decentralized security aspects like Topic ACL management and Certificate Trust Chain validation.

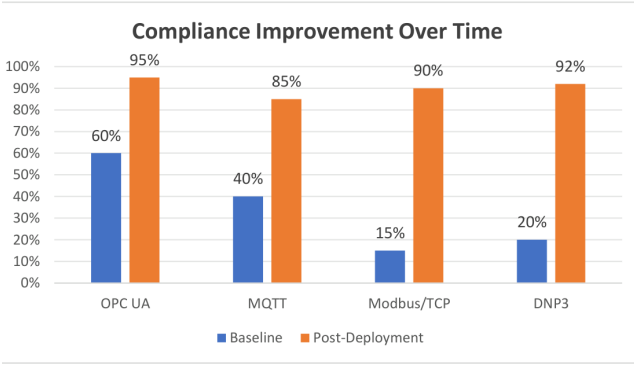

FIGURE 2. Compliance Improvement Over Time

E. CASE STUDIES

Two typical scenarios were simulated to provide a better understanding of the mechanism's performance. In the first case, the broker of the MQTT protocol was left open so that anonymous access was permitted with no username/password scheme. The compliance agent quickly detected the misconfiguration and raised a notification, followed by an automated remediation script, which was executed and disabled anonymous access and enabled credential-based authenticated access. Broker logs verified that the change was applied successfully. The whole process of detection and remediation took less than one minute and successfully avoided probable exploitation and data leakage. In the second case, a certificate of the OPC UA server was intentionally permitted to lapse. The system immediately registered the expiring certificate and initiated an automatic renewal process. The old certificate was exchanged for the correct one, and the customer was able to connect to the service again without failure. The cycle of all remediation activities using the system (detection to certificate replacement) was achieved within two minutes, testifying to the effectiveness of the system in maintaining operational continuity even in a failed condition. The example detection alarm remedial workflow is shown in Figure 3:

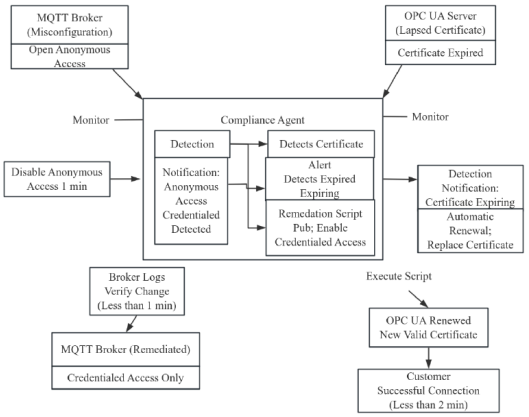

FIGURE 3. Example Detection–Alert–Remediation Workflow

F. COMPARATIVE ANALYSIS

Traditional security methods are inefficient compared to the proposed mechanism. Manual quarterly audits are resource-intensive and often miss ephemeral misconfigurations. Less resource-intensive passive monitoring fails to offer immediate remediation, relying instead on post-occurrence analysis. Popular Intrusion Detection/Prevention Systems (IDS/IPS) intercept known threats but often miss common, high-impact misconfigurations like open MQTT ports or invalid certificates. In contrast, the mechanism described here combines continuous monitoring with active protocol-level policy application. This design rapidly identifies non-compliant configurations and triggers near-instantaneous remediation, drastically reducing detection-to-remediation time. By eliminating the long delays inherent in manual and passive approaches, the system secures the operational environment in near real time, severely restricting an attacker's window of opportunity.

VI. DISCUSSION

The preliminary evaluation of compliance has shown that legacy protocols like Modbus TCP had the highest levels of compliance gaps, mainly because these protocols essentially lack encryption and authentication processes. Modbus displayed a compliance rate of 15 percent, confirming that the protocol is insecure in current enterprise environments. MQTT also lacked sufficient coverage, as only 40 percent of sessions were authenticated, making IoT endpoints vulnerable to unauthorized access attacks. OPC UA performed better, but 40 percent of certificates were expired or invalid. Through compliance checks, certificate validation, and authentication enforcement, the most significant improvements were achieved in securing the security posture [11]. With checks being automated, vulnerabilities involving identity spoofing and unauthorized access to information were significantly minimized. A continuous compliance monitoring framework also significantly reduced the time to detect and correct conditions. These data points directly address known issues in ICS cybersecurity, where the longer the response is delayed, the greater the exposure period risk. The findings stress the importance of ongoing and automated monitoring for compliance teams and IT security operations, instead of using periodic manual audits. The alerts and compliance scoring solution in real time allows actionable decision-making for immediate remediation and minimization of operational risks. The ability to integrate with SIEM and SOC processes raises situational awareness and simplifies incident response [12].

To thoroughly assess the system's ability to accurately classify and prioritize security risks, we performed a Receiver Operating Characteristic (ROC) curve analysis on the detection alerts generated during the evaluation period. The analysis set comprised the 1,080 known attack and misconfiguration injection events (positive samples) and an equivalent number of sampled benign operations (negative samples) recorded across the three test rounds. The ground truth labels were established directly from the event injection scripts used in the controlled testbed.

Based on this evaluation dataset, an additional ROC curve analysis of the system's detection alerts produced an area under the curve (AUC) value of 0.982, indicating a high discrimination capability. This exceptionally high AUC indicates that the classification threshold is well calibrated, minimizing both false positives and false negatives. Furthermore, the precision score at the top 10% of highest-risk configurations reached 0.96, confirming the system's ability to effectively prioritize the most critical threats with minimal false positives, meaning that almost all cases flagged as critical truly represented severe security concerns. Such performance confirms that the mechanism effectively focuses

remediation resources on the most impactful vulnerabilities, thereby enhancing overall security governance efficiency.

The scope of this study was restricted to a selection of commonly utilized communication protocols—OPC UA, MQTT, Modbus TCP, and DNP3—because they most represent typical industrial and IoT realms. New protocols and cloud-native communication standards have not been discussed, and such growth will be required in the future. Moreover, compliance regulations need constant revision to keep up with fluctuations in security practice and to update protocols in order to remain effective [13]. Privacy restrictions limited the extent of payload inspection, which might fail to detect some low-level threats in the data layer. Nonetheless, due to these limitations, modular architecture enables scalability and future adaptation.

VII. RECOMMENDATIONS

To boost governance of enterprise cybersecurity, constant monitoring of compliance must be adopted as policy. Organizations are required to install automated systems to check communication protocols on a real-time basis and notice and reduce deviations as they occur. The security policy must require developers to incorporate compliance checks throughout development pipelines, following CI/CD patterns, to ensure new software and configurations are checked against security policies prior to release [14]. This integration facilitates security shift-left practices, which are internalized at early stages of the life cycle. To achieve maximum benefits, security teams must be trained to interpret compliance metrics and use automation tools. Agile, secure digital enterprise environments will be encouraged by constant feedback created among operations and development teams.

VIII. CONCLUSION & FUTURE WORK

This study introduces a protocol-agnostic, continuous compliance mechanism tailored for digital enterprises integrating ERP, MES, and IoT systems. It is highly relevant to the industry, providing a unifying security layer for complex processes like raw material tracking and automated loom control. The core contributions are the compliance mapping framework, the automated validation engine, and a real-time scoring application. Experiments demonstrate a drastic increase in compliance, minimized vulnerabilities, and reduced remediation time, promoting continuous cybersecurity governance. Future development will incorporate AI-based compliance checking for real-time rule updates and expand the system to cover cloud-native protocols and microservices. Additionally, the ability to use blockchain technology to define immutable, auditable compliance logs has the potential to bring more trust and transparency to regulatory reporting. This is highly valuable for the industry, where decentralized, tamper-proof logs could verify the ethical sourcing and environmental compliance of materials, directly reinforcing more resilient and scalable enterprise security and sustainability frameworks across the supply chain.

AUTHOR CONTRIBUTIONS

Yamin Huang designed, collected and analyzed the data, and drafted the manuscript. Yamin Huang conducted the study, critically revised the manuscript for important intellectual content, and gave final approval of the version to be published. Yamin Huang participated fully in the work, take public responsibility for appropriate portions of the content, and agreed to be accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

CONFLICTS OF INTEREST

The author declares no conflict of interest.

FUNDING

This research received no external funding.

ACKNOWLEDGEMENTS

Not applicable.

REFERENCES

[1] B. S. Chohan, X. Xu, and Y. Lu, "MES dynamic interoperability for SMEs in the Factory of the Future perspective," Procedia CIRP, vol. 107, pp. 1329-1335, 2022, doi: 10.1016/j.procir.2022.05.153.

[2] Z. Lin, Z. Liu, Y. Zhang, J. Yan, S. Liu, B. Qi, and K. Wei, "Edge-fog-cloud hybrid collaborative computing solution with an improved parallel evolutionary strategy for enhancing tasks offloading efficiency in intelligent manufacturing workshops," Journal of Intelligent Manufacturing, pp. 1-28, 2024, doi: 10.1007/s10845-024-02463-7.

[3] M. Alshar'e, "Cyber security framework selection: comparison of NIST and ISO27001," Applied Computing Journal, pp. 245-255, 2023, doi: 10.52098/acj.202364.

[4] R. Amor and J. Dimyadi, "The promise of automated compliance checking," Developments in the Built Environment, vol. 5, Art. no. 100039, 2021, doi: 10.1016/j.dibe.2020.100039.

[5] M. Repetto, "Adaptive monitoring, detection, and response for agile digital service chains," Computers & Security, vol. 132, Art. no. 103343, 2023, doi: 10.1016/j.cose.2023.103343.

[6] S. Wijesinghe, I. Nanayakkara, R. Pathirana, R. Wickramarachchi, and I. Fernando, "Impact of IoT integration on enterprise resource planning (ERP) systems: a comprehensive literature analysis," in Proceedings of the 2024 International Research Conference on Smart Computing and Systems Engineering (SCSE), 2024 Apr, IEEE, pp. 1-5, doi: 10.1109/SCSE61872.2024.10550684.

[7] Z. Li, X. Mei, Z. Sun, J. Xu, J. Zhang, D. Zhang, and J. Zhu, "A reference framework for the digital twin smart factory based on cloud-fog-edge computing collaboration," Journal of Intelligent Manufacturing, pp. 1-21, 2024, doi: 10.1007/s10845-024-02424-0.

[8] S. Lakshminarayana, A. Praseed, and P. S. Thilagam, "Securing the IoT application layer from an MQTT protocol perspective: challenges and research prospects," IEEE Communications Surveys & Tutorials, vol. 26, no. 4, pp. 2510-2546, 2024, doi: 10.1109/COMST.2024.3372630.

[9] H. Kayan, M. Nunes, O. Rana, P. Burnap, and C. Perera, "Cybersecurity of industrial cyber-physical systems: A review," ACM Computing Surveys, vol. 54, no. 11s, pp. 1-35, 2022, doi: 10.1145/3510410.

[10] M. Farhadi Moghadam, M. Nikooghadam, A. H. Mohajerzadeh, and B. Movali, "A lightweight key management protocol for secure communication in smart grids," Electric Power Systems Research, pp. 178, 2020, doi: 10.1016/j.epsr.2019.106024.

[11] M. Nankya, R. Chataut, and R. Akl, "Securing industrial control systems: components, cyber threats, and machine learning-driven defense strategies," Sensors, vol. 23, no. 21, pp. 8840, 2023, doi: 10.3390/s23218840.

[12] M. Khayat, E. Barka, M. A. Serhani, F. Sallabi, K. Shuaib, and H. M. Khater, "Empowering security operation center with artificial intelligence and machine learning: a systematic literature review," IEEE Access, 2025, doi: 10.1109/ACCESS.2025.3532951.

[13] M. K. Quan, P. N. Pathirana, M. Wijayasundara, S. Setunge, D. C. Nguyen, C. G. Brinton, and H. V. Poor, "Federated learning for cyber physical systems: a comprehensive survey," IEEE Communications Surveys & Tutorials, 2025, doi: 10.1109/COMST.2025.3570288.

[14] G. Malik and P. Prashasti, "Shift left security," The Eastasouth Journal of Information System and Computer Science, vol. 2, no. 3, pp. 219-245, 2025, doi: 10.58812/esiscs.v2i03.528.