SECURE ICS TESTBED:

ZERO TRUST & ENCRYPTED TRAFFIC

Ahmed Said Al-Senaidi, Al-Salt Fahad Al-Naabi, Ahmed Naser Al-Khanbashi, Mohamed Yousef Al-Harrasi

Final Year Project – Spring 2025

B.Sc. in Computer Science

Department of Computer Science
College of Science
Sultan Qaboos University

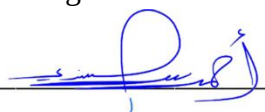
A report submitted in partial fulfillment of the requirements for the B.Sc. in Computer Science

Supervisor: Dr.Shdha ALAmri

Examiner: prof Abderezak Touzene

8 May 2025

Declaration We hereby certify that this material, which we now submit for assessment of the report in partial fulfillment of the requirements for COMP5590, is entirely our own work and completed by members of our group except for information obtained in a legitimate way from literature, company or university sources. All information from these other sources has been cited and acknowledged within the text of our work.

Signed:  Date: 08/05/2025

Signed:  Date: 08/05/2025

Signed:  Date: 08/05/2025

Signed:  Date: 08/05/2025

ABSTRACT

Industrial Control Systems (ICS) are essential for managing critical infrastructure but remain highly vulnerable due to outdated communication protocols and inadequate cybersecurity measures. This project introduces the design and implementation of a virtual ICS cybersecurity testbed using open-source tools and contemporary defense frameworks. Leveraging the Graphical Realism Framework for Industrial Control Simulation (GRFICSv2), we simulated a realistic ICS environment comprising supervisory control and data acquisition (SCADA) systems, programmable logic controllers (PLCs), and a virtual plant. The environment was segmented into IT, demilitarized zone (DMZ), and operational technology (OT) zones in accordance with the Purdue Model.

To enforce Zero Trust Architecture, we deployed pfSense firewalls and a Jumpserver, while network defense and monitoring were provided through Suricata intrusion detection system (IDS), Wazuh Security Information and Event Management (SIEM), and TLS-encrypted Modbus communications. Controlled cyberattacks using the Metasploit framework validated the presence of vulnerabilities and tested the efficacy of defensive mechanisms. Results demonstrated that network segmentation and encryption effectively blocked intrusions and enhanced detection capabilities.

This testbed confirms that robust ICS security is achievable with minimal financial investment and provides a valuable platform for cybersecurity training and research.

GLOSSARY

Term	Definition
Industrial Control System (ICS)	A system used to monitor and control industrial processes in sectors like energy, manufacturing, and water treatment.
Operational Technology (OT)	Hardware and software are used to monitor and control physical devices, processes, and infrastructure in industrial settings.
Zero Trust Architecture (ZTA)	A cybersecurity model that eliminates implicit trust, enforcing strict access control and continuous verification for all users and devices.
Supervisory Control and Data Acquisition (SCADA)	A control system used in industrial environments for remote monitoring and control of assets such as pipelines, power plants, and factories.
Distributed Control System (DCS)	A type of control system in industrial environments where controllers are distributed across the system rather than being centralized.
Programmable Logic Controller (PLC)	An industrial digital computer used for automation and real-time process control.
Intrusion Detection System (IDS)	A security system that monitors network traffic and detects suspicious activities or potential cyber threats.
Suricata IDS	An open-source intrusion detection and prevention system that analyzes network traffic to detect malicious activity.

Security Information and Event Management (SIEM)	A cybersecurity solution that collects, analyzes, and responds to security alerts generated from an organization's infrastructure.
Wazuh SIEM	An open-source SIEM and endpoint security platform used for real-time monitoring and threat detection.
Modbus/TCP	A communication protocol widely used in ICS for data exchange between industrial devices.
TLS Encryption (stunnel)	A cryptographic protocol used to secure communication by encrypting data between ICS components.
MITRE ATT&CK for ICS	A globally accessible framework that provides a structured way to analyze and classify cyber threats against industrial environments.
Man-in-the-Middle (MITM) Attack	A type of cyberattack where an attacker intercepts and alters communication between two parties without their knowledge.
Replay Attack	A cyberattack where an attacker captures and replays legitimate data transmissions to disrupt or manipulate an ICS system.
Purdue Model for ICS Security	A hierarchical model that defines security layers within an ICS network to segment and control access between IT and OT environments.
IEC 62443	An international standard for cybersecurity in Industrial Automation and Control Systems (IACS).

NIST SP 800-82	A cybersecurity guideline from the National Institute of Standards and Technology (NIST) that provides recommendations for securing ICS environments.
Firewall (pfSense)	A network security tool that filters traffic, enforces security policies, and prevents unauthorized access in ICS networks.
VirtualBox	A virtualization software used to create and run virtual machines for ICS testbed simulations.

PREFACE

This Final Year Project report is submitted in partial fulfillment of the requirements for the Bachelor of Science in Computer Science.The project focuses on the cybersecurity of Industrial Control Systems (ICS), an area that is critical yet often neglected in terms of security practices. Through the implementation of a simulated ICS environment using GRFICSv2 and the application of open-source security tools such as pfSense, Wazuh, and Suricata, we aim to highlight the vulnerabilities present in these systems and demonstrate practical methods for securing them.The experience gained throughout the project has been invaluable, allowing us to apply theoretical knowledge in a practical setting, develop hands-on skills in cybersecurity, and better understand the challenges faced by critical infrastructure operators in today's threat landscape.We hope this report serves as a useful reference for students, researchers, and practitioners who are interested in the field of industrial cybersecurity.

ACKNOWLEDGMENT

We sincerely appreciate Dr.Shdha AlAmri for her support and guidance during this project. Special thanks to our professionals in the subject matter, Dr. Yahya Al-Hadhrami, for their expertise, input, and suggestions. Lastly, we want to thank our families and colleagues for their support and encouragement during this process.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Industrial Control Systems (ICS) are essential components in the management of critical infrastructure sectors such as power generation, manufacturing, and water treatment. However, the growing integration of Operational Technology (OT) with Information Technology (IT) networks has significantly increased the exposure of ICS environments to cyber threats. Traditional ICS were not designed with security in mind and often lack fundamental protections such as encryption, authentication, and network segmentation.

Recent cyberattacks—including Stuxnet and Industroyer—have demonstrated the ability of sophisticated malware to cause physical disruption by compromising ICS environments. These incidents highlight the urgent need for modern cybersecurity strategies specifically tailored to the unique demands of industrial systems.

This project addresses these challenges by designing and implementing a secure ICS testbed based on Zero Trust Architecture (ZTA) principles. The environment simulates a realistic industrial network using GRFICSv2, segmented into IT, DMZ, and OT zones protected by pfSense firewalls, monitored by Wazuh SIEM and Suricata IDS, and enhanced with encrypted Modbus TCP communication via stunnel.

By conducting controlled cyberattack simulations and applying real-time monitoring, this project aims to improve the resilience of ICS environments against contemporary threats, while demonstrating cost-effective and practical defense strategies to protect critical industrial infrastructure.

1.1 Problem Statement

Industrial Control Systems (ICS) have been helping to run critical infrastructures like power plants, water treatment facilities, and manufacturing firms. Converging Operating Technology (OT) with contemporary IT networks has widened the attack surface and exposed ICS environments to new, highly advanced cyber threats. By tradition, Industrial Control Systems (ICS) are isolated; however, the growth in interconnectivity has made them susceptible to numerous threats, such as ransomware, advanced persistent threats (APTs), and bespoke malware like Stuxnet and Industroyer. The natural design of ICS parts typically lacks encryption, secure communication protocols, and access controls. Also, many industrial

networks have relatively flat architectures, which facilitate lateral movement across the network once a breach has been achieved [1].

The 2010 Stuxnet attack on the Natanz nuclear facility of Iran demonstrated that malware aimed at ICS could cause physical destruction without detection. Stuxnet was aimed explicitly at the Siemens PLC and caused mechanical disruption to the operating centrifuge by subtly manipulating their speeds while displaying credible operational data to system monitoring personnel. This record-breaking cyber-physical attack was capable of disrupting uranium enrichment processes and established a foundation for follow-on attacks that compromised industrial control systems, demonstrating that industrial malware could bypass air-gapped configurations [2].

The above requirements are fulfilled by applying pro-cybersecurity design based on NIST SP 800-82 and Zero Trust Architecture. Network segmentation, IDS, and SIEM solutions can protect operations for threat detection and response activities with defense-in-depth measures. A new security architecture for ICS recently offered evidence for the layers of defense approach to security architecture founded on monitoring management and network hardening for enhanced resilience against reconnaissance and intrusion activities [3]. An IDS that is tuned for ICS is capable of identifying anomalies in process control traffic, and SIEM offers real-time security event monitoring as well as correlation events [4]. Nevertheless, Zero Trust concepts facilitate ongoing authentication of users and devices and thereby reduce the possibility of unauthorized access into crucial infrastructure [5].

Apart from the cyber threat, financial loss from ICS intrusions is potential downtime in production, supply chain disruption, and millions of economic losses. Awareness of OT security must be raised; most organizations lack proper training and security suitable for industrial environments. Literature suggests the application of Zero Trust principles in ICS infrastructures dramatically improves security through continuous authentication of all access and minimizes the attack surface possibility [5]. Hardening ICS environments is not just about reducing cyber threats but also ensuring economic stability and life safety as well [3]. This program is a step forward in closing the knowledge gaps through promoting applied applications of cutting-edge ICS security objectives so that critical infrastructure has enhanced resilience against an evolving threat landscape [4].

1.2 Objectives

The aim of this project is to design and evaluate a secure, modular Industrial Control System (ICS) testbed using open-source tools and modern cybersecurity principles. The testbed simulates a realistic industrial network segmented by the Purdue Model and protected through Zero Trust Architecture (ZTA). The following objectives were established to guide the project:

- **Conduct a comprehensive literature review** to analyze ICS vulnerabilities, attack vectors, and cybersecurity frameworks, including NIST SP 800-207 (Zero Trust) and ISA/IEC 62443.
- **Define the security and functional requirements** of the testbed, including network segmentation, secure remote access, centralized logging, and encrypted communication protocols.
- **Design the ICS network architecture** using the Purdue Model with micro-segmentation across IT, DMZ, and OT zones, enforced by pfSense firewall policies.
- **Develop and configure the testbed environment** using GRFICSv2 components—such as SCADA servers, PLCs, and engineering workstations—deployed in VirtualBox.
- **Integrate cybersecurity mechanisms** including pfSense for network control, Suricata IDS for traffic monitoring, Wazuh SIEM for event correlation, and stunnel for TLS-based encryption of Modbus TCP traffic.
- **Simulate cyberattacks using Metasploit** to test the system's resilience against command injection, unauthorized access, and unencrypted protocol vulnerabilities.
- **Document the results and evaluate the testbed's performance**, identifying strengths and weaknesses, and proposing improvements for future development and application.

(Note: While the simulation of Man-in-the-Middle (MITM) attacks was originally planned, it was not executed due to resource and time constraints. This remains part of the proposed future work.)

1.3 Methodology

1.3.1 Research Methodology

The research methodology revolves around an experimental and simulation-driven technique whereby cybersecurity controls are created, installed, and tested in a virtualized Industrial Control System test bed. The methodological framework unfolds in four core stages, which allow for a logical and systematic approach that adheres to the established best practices in security research.

The first phase includes a systematic literature review of ICS security architectures, zero-trust principles, and encryption mechanisms. This phase was followed by a requirements analysis, testbed deployment, and a security evaluation phase. These four stages—literature review, requirements analysis, deployment, and evaluation—formed the basis for our experimental approach. Existing standards, such as NIST SP 800-207 for Zero Trust[6], ISAIEC 62443 for the security of ICS [7] and Purdue model segmentation are reviewed [8] to create a security baseline.

In the second phase, an analysis of requirements is carried out. Major security controls and system functions are defined. Encryption mechanisms to be provided to protect Modbus/TCP and other ICS protocols [9], an intrusion detection system for real-time network monitoring through Suricata IDS [10]. Zero trust segmentation and access control enforcement are done in the third phase. Specifically, we configured stunnel to apply TLS encryption on Modbus TCP traffic and deployed two Suricata IDS instances (one in the DMZ and one in the OT zone) to monitor traffic. While the initial plan included man-in-the-middle (MITM) attack simulations, these were deferred due to resource constraints.

Implementation of establishing and deploying the virtualized ICS testbed utilizing industrial-grade security tools. Testbed establishment of pfSense firewall for segmentation and access policy implementation, Suricata IDS for ICS traffic anomaly inspection, Stunnel for TLS encryption to protect Modbus/TCP communication, Wazuh SIEM for security event monitoring and alerting [10]. The testbed simulates real-world security settings with controlled conditions for security testing.

The fourth phase is security evaluation, where controlled attacks are launched to ascertain the detection and mitigation powers of the ICS testbed. The evaluation employs replay attacks to test the anomaly detection powers of the Suricata IDS [8], man-in-the-middle attacks to assess the strength of encryption applied on Modbus/TCP traffic [9], and invalid access attempts to

validate the enforcement of firewall rules and authentication [7]This orderly approach ensures a simulative reality of ICS security, where quantitative security improvement is possible without risk of nonconformity to industry standards.

1.3.2 Tools and Technologies

These tools include pfSense for firewall and network segmentation; two instances of Suricata IDS (one in the DMZ and one in the OT zone) for rule-based and anomaly-based detection; stunnel for providing TLS encryption to secure Modbus/TCP communication. Wazuh SIEM is used for monitoring security events and logging real-time alerts, while VirtualBox is utilized for hosting and simulating ICS network environments[6]. The Zero Trust model is applied to enforce least-privilege access and micro-segmentation [12]. These tools were selected based on efficiency, compatibility with ICS environments, and effectiveness in real-time security monitoring and network segmentation.

1.3.3 System Architecture and Testbed Setup

In alignment with the Purdue Model, the testbed adopts a multi-layered security approach. A Zero Trust solution further restricts the lateral movement of attackers and hardens access control policies. The solution architecture consists of five different levels of controls. Level three, representing the enterprise network, utilizes Wazuh SIEM and pfSense firewall logs to conduct security monitoring and log analysis. Level two, representing the process and operations management perspective, is monitored by a Suricata IDS deployed in the DMZ. In addition, a second Suricata IDS is installed at Level one (within the OT zone) to inspect internal Modbus communication. Level one represents the control network, which preserves a secure ICS communication through encrypted Modbus.TCP via stunnel. The last level, zero, represents the physical process that utilizes simulated PLCs, HMIs, and sensors inside the VirtualBox[6]. Furthermore, the testbed has further enabled the Zero Trust security with additional security measures: The least-privilege access control constrains ICS access only to authorized users and devices [12]. Micro-segmentation isolates different ICS components and blocks lateral movement in the event of an attack[8]. Continuous security monitoring is provided through the SIEM, which detects and analyzes security events in real time. A detailed network architecture diagram will be included in the final report.

1.3.4 Security Testing and Data Collection

Security testing is the operations that aim to assess the efficiency of encryption, intrusion detection, and firewall mechanisms under controlled conditions.

Three types of attack simulations were primarily performed:

- Replay attacks to test Suricata's anomaly detection capabilities
- Unauthorized access attempts to validate firewall rule enforcement and authentication policies

Although man-in-the-middle (MITM) attacks were originally planned to assess TLS resistance, they were not executed due to time constraints and the complexity of simulating them in an encrypted environment. This remains a topic for future investigation.

During the security testing phase, verification guarantees that cybersecurity controls resist and can reduce real-world ICS threats.

1.3.5 Expected Challenges and Mitigation Strategies

Some ensuing challenges shall adversely affect operational efficiency and reliability of system performance and security tests during implementation within the project. One of these challenges is encryption overhead, which may negatively impact real-time ICS communication performance. This was managed by selectively enabling encryption only in specific test scenarios. Encryption settings were fine-tuned to maintain a balance between system performance and communication security [7]. False positives were observed in Suricata, which caused alert noise and reduced detection accuracy[6]. This was partially addressed through custom rule tuning and traffic baseline analysis. The last one is about some virtualization limits due to the rare computing resources for running several virtual ICS devices [11]. Therefore, VM configurations will be optimized for lesser resource consumption. The project team further ensures the documentation of modifications according to the results based on security tests to, as such, empower proper operating procedures on the system.

1.4 Scope

This project focuses on the design, implementation, and evaluation of a secure, virtualized Industrial Control System (ICS) environment using open-source cybersecurity tools. The scope is intentionally limited to simulation-based testing in order to prioritize feasibility, accessibility, and safety while accurately modeling real-world ICS conditions.

1.4.1 Inclusions

The following components and activities are within the scope of this project:

- Simulation of ICS Components: The system includes SCADA servers, Programmable Logic Controllers (PLCs), an engineering workstation, and a plant simulation. These components are deployed using GRFICSv2 in a virtual environment.

- Network Segmentation: The testbed architecture follows the Purdue Model, implementing three network zones—IT, DMZ, and OT—using pfSense firewalls to enforce strict access policies.

- Cybersecurity Integration: Security mechanisms include Suricata IDS for real-time threat detection, Wazuh SIEM for centralized log management and alerting, and stunnel for encrypted Modbus TCP communication.

- Zero Trust Enforcement: A Jumpserver is implemented in the DMZ to control and monitor access between the IT and OT networks, following least-privilege principles.

- Attack Simulations: Cyberattacks are simulated using the Metasploit Framework to assess system vulnerabilities, including Modbus TCP command injection and unauthorized access scenarios.

1.4.2 Exclusions

Certain elements are intentionally excluded from the project to maintain focus and feasibility:

- Physical Hardware Deployment: The project is fully virtualized; no physical PLCs, SCADA equipment, or real industrial devices are used.

- Advanced Persistent Threat (APT) Simulation: Multi-stage, long-term threat simulations such as APTs are beyond the scope due to their complexity and time requirements.

- Production Environment Integration: This testbed is intended for academic and training purposes only. It is not designed for deployment in a live industrial environment.

1.5 Significance

Industrial Control Systems (ICS) form the backbone of critical infrastructure, yet they often lag behind in cybersecurity when compared to traditional IT systems. Many ICS environments continue to rely on legacy systems that lack essential protections such as network segmentation,

encrypted communication protocols, and real-time monitoring. The vulnerabilities inherent in these systems, especially when exposed to the broader internet or integrated with IT networks, pose significant risks not only to industrial operations but also to public safety and economic stability.

This project is significant in that it directly addresses these cybersecurity gaps through the development of a modular, low-cost, and realistic ICS cybersecurity testbed. By simulating SCADA environments and industrial components using GRFICSv2 and securing them with open-source tools such as pfSense, Suricata IDS, Wazuh SIEM, and stunnel, the project demonstrates how modern defense mechanisms can be practically applied in ICS networks. The testbed's architecture—based on the Purdue Model and Zero Trust principles—provides strong access control, anomaly detection, and encrypted communication.

Furthermore, the use of realistic cyberattack scenarios, including command injection and unauthorized access, provides insight into both vulnerabilities and effective countermeasures. This enables a deeper understanding of how contemporary threats behave in ICS contexts and how security tools can be configured to detect and prevent such attacks.

Beyond technical implementation, the testbed serves as an educational platform. It allows cybersecurity students and industrial professionals to gain hands-on experience with ICS security concepts, threat analysis, and response strategies. Its modular design also supports future research on topics such as intrusion detection tuning, ICS-specific encryption, and Zero Trust policy enforcement.

Ultimately, this project contributes to the broader field of ICS cybersecurity by offering a replicable model that can improve resilience in critical infrastructure systems and reduce the risks posed by emerging cyber threats. It supports both academic inquiry and practical training, thereby benefiting individuals, organizations, and society at large.

1.6 Overview

The ICS cybersecurity testbed developed in this project replicates a realistic industrial network segmented into three security zones in accordance with the Purdue Enterprise Reference Architecture. These zones—IT, DMZ, and OT—are isolated and protected through layered defenses, enabling detailed monitoring and control over access flows and communications. The

system architecture integrates both network security tools and industrial simulation components, all hosted within a virtualized environment.

1.6.1 Network Segmentation

The IT zone (LAN1) hosts the Wazuh server and a monitoring workstation. This segment is responsible for centralized log collection, real-time alerting, and administrative oversight. The DMZ (LAN2) serves as a transitional buffer between IT and OT networks. It hosts the Jumpserver and Suricata IDS, which together enforce restricted access and monitor all cross-zone traffic. The OT zone (LAN3) houses the GRFICSv2 components, including SCADA servers, PLCs, an engineering workstation, and a simulated plant. This network segment represents the core industrial environment and is strictly isolated from direct IT access.

1.6.2 Security Controls

Several security layers are implemented to protect the virtual ICS environment. pfSense is used to define and enforce strict firewall rules that prohibit direct communication between the IT and OT zones. All access to the OT network must pass through the Jumpserver in the DMZ, ensuring centralized monitoring and access logging. Suricata IDS monitors all network traffic across the zones for signs of malicious activity or policy violations. Wazuh SIEM provides real-time log analysis and alert generation based on predefined security rules. Additionally, stunnel is configured to provide TLS encryption for Modbus TCP communications, though in some test scenarios encryption was intentionally disabled to replicate realistic ICS vulnerabilities and observe detection responses.

1.6.3 Access Control

Access to OT resources is mediated through a Jumpserver deployed in the DMZ. IT administrators and security analysts must first authenticate into the Jumpserver, from which they can initiate secure, logged sessions to access the OT components. This model enforces Zero Trust principles by limiting direct trust relationships, enforcing least-privilege access, and enabling detailed access auditing.

1.6.4 Attack Simulation and Evaluation

The Metasploit Framework was used to simulate real-world cyberattacks, particularly focusing on Modbus TCP vulnerabilities. Attacks included unauthorized command injections and access attempts directed at OT devices. These simulations enabled evaluation of the system's intrusion

detection, firewall enforcement, and overall resilience. Logging and alert data were analyzed to assess whether the defensive mechanisms responded appropriately to each simulated threat scenario.

1.7 Report Overview

This report is organized into nine chapters that collectively document the design, implementation, testing, and evaluation of a secure Industrial Control System (ICS) testbed. Each chapter builds upon the previous one to provide a comprehensive view of the project's development process, technical components, and cybersecurity contributions.

Chapter 1 introduces the motivation, problem statement, objectives, scope, and significance of the project. It also outlines the research methodology and provides an overview of the system's architecture and testing approach.

Chapter 2 presents a detailed literature review, examining existing ICS vulnerabilities, Zero Trust Architecture (ZTA), the Purdue Model, industrial communication protocols, and current best practices in ICS security. It also identifies key gaps in the literature that this project aims to address.

Chapter 3 covers project planning and management, including task scheduling, team roles, Gantt chart planning, and iterative development stages. It also discusses challenges faced during the process and the strategies used to adapt the project plan.

Chapter 4 focuses on system requirements, including both functional and non-functional aspects, and provides a structured analysis of the problem space.

Chapter 5 details the system design, including architectural diagrams, component relationships, and the rationale behind using specific security tools.

Chapter 6 describes the implementation phase, outlining the configuration of pfSense, Suricata IDS, Wazuh SIEM, and stunnel encryption, along with the deployment of GRFICSv2 components in VirtualBox.

Chapter 7 presents the results of the security testing phase. It outlines the attack simulation methodology and evaluates the system's resilience based on key performance indicators such as detection rates, alert accuracy, and access control enforcement.

Finally, Chapter 8 concludes the report by summarizing key achievements, acknowledging limitations, and proposing future work to enhance the effectiveness and scalability of the testbed. The chapter also reflects on the broader impact of the project in advancing ICS cybersecurity practices.

CHAPTER 2: BACKGROUND

In this chapter, background information is provided for placing the different security challenges that Industrial Control Systems (ICS) face in the context of existing research about cybersecurity in these environments. The key themes are discussed in several subsections: general background on ICS security, Zero Trust architectures, uses of encrypted communication, and some relevant related work related to industrial cybersecurity frameworks.

2.1 General Background Information

2.1.1 Introduction to ICS and OT Security

Industrial Control Systems (ICS) and Operational Technology (OT) form integral parts across many industrial industries, ranging from energy production to manufacturing and water resource management [13]. Most often, ICS consists of a mix of different control systems, such as Supervisory Control and Data Acquisition (SCADA), Distributed Control Systems (DCS), and Programmable Logic Controllers (PLCs). These systems enable monitoring and automation of industrial processes, often in real-time, to govern physical infrastructures[14]. Traditionally, ICS and OT networks were isolated from corporate IT networks. However, with advancements in digital transformation—symbolized by cloud-based data analysis, remote support, and sophisticated Internet of Things (IoT) devices—integration with corporate IT infrastructures has increased significantly. While such integration improves operational effectiveness—demonstrated by real-time transfer of operational information to enterprise planning systems- it simultaneously raises security risks[15].

2.1.2 ICS Security Challenges

Numerous Industrial Control System environments face a variety of security challenges, which are common to both established limitations and new demands for connectivity.

Legacy System Vulnerabilities

Some industrial plants still rely on older operating systems and hardware that do not support encryption or modern authentication [13]. Because ICS often needs continuous operation, updating or patching these systems can be risky [14]. Attackers can exploit default credentials, unpatched firmware, and inadequate security measures inherent in these outdated environments.

Expanded Attack Surface

The convergence of OT and IT has expanded the overall attack surface. ICS devices increasingly connect to corporate networks, the internet, and external cloud platforms, creating more entry points for adversaries [13]. Meanwhile, Industrial IoT (IIoT) sensors and controllers—if not secured—provide additional routes for unauthorized access[15].

Notable Cyberattacks on ICS

Real-world incidents highlight the damage that can result from ICS breaches:

- Stuxnet (2010): Sophisticated computer worm that manipulated PLC logic to destroy Iranian nuclear centrifuges' operation [15].
- Triton (2017): A targeted attack against safety instrumented systems in a petrochemical plant, with a goal to disable critical fail-safes [14].
- Colonial Pipeline (2021): A ransomware attack that shut down a major U.S. fuel pipeline, showing how an IT breach can disrupt OT operations [13].

2.1.3 Cybersecurity Frameworks for ICS

NIST SP 800-82: ICS Security Guidelines

NIST SP 800-82 outlines best practices for industrial environments with a focus on risk management, network segmentation, secure remote access, and incident response to ensure minimum downtime [13].

ISA/IEC 62443: Industrial Cybersecurity Standards

The ISA/IEC 62443 standard provides a common Industrial Automation and Control system framework (IACS). It includes elements like system design, access, network segmentation, and life-cycle management, and employs security levels (SL1–SL4) to enable continual improvements [14].

MITRE ATT&CK for ICS

MITRE ATT&CK for ICS describes adversary tactics and techniques seen in real-world industrial breaches. By linking identified threats with specific TTPs, it supports defenders in unifying security strategies and in giving priority to identifying threats [15].

NIST SP 800-207 (Zero Trust Architecture)

Zero Trust Architecture eliminates implicit trust in network perimeters by requiring constant identity verification and strict access controls. In ICS, this means micro-segmentation and continuous monitoring to combat risks like credential theft or insider threats[16].

2.1.4 The Purdue Model for ICS Network Segmentation

The Purdue Enterprise Reference Architecture (PERA) is the total design framework for Industrial Control System (ICS) networks.

It classifies environments into different levels:

- Level 0–1: Field devices (e.g., sensors, actuators, PLCs)
- Level 2: Control systems and SCADA workstations
- Level 3: Operation management (for example, historians, engineering stations)
- Level 4–5: Enterprise-level Information Technology networks

By restricting unnecessary inter-layer traffic, the Purdue model limits lateral attacker movement[14]. However, with cloud integration and remote IIoT devices growing, organizations often introduce secure gateways or adapt Zero Trust principles to preserve both connectivity and security.

2.1.5 Encryption in ICS and Its Challenges

Significance of Secure Protocols

Legacy ICS protocols such as Modbus/TCP and DNP3 lack encryption, making them vulnerable to eavesdropping and manipulation[14]. Newer alternatives, including OPC UA

(with built-in encryption), Modbus Secure, and DNP3 Secure Authentication, address these weaknesses [13]. Other options include TLS/IPsec secure traffic at the network layer.

Challenges in Overseeing Encrypted Data Communication

Encryption enhances confidentiality, while it complicates network-based intrusion detection. Traffic encryption limits the effectiveness of deep packet inspection unless organizations deploy secure gateways or use host-based intrusion detection systems[17]. It's a recurring challenge to balance user privacy, performance, and security monitoring[18].

2.1.6 ICS Cybersecurity Testbed (FYP Implementation)

To evaluate these security factors in a real-world environment, this project has a complete ICS testbed:

GRFICSv2: Industrial System Emulation

GRFICSv2 simulates realistic ICS operations, allowing the creation of virtual PLCs, sensors, and network topologies. Modeling a mock industrial process enables safe experimentation with cyberattacks and countermeasures.

Wazuh: Security Monitoring Platform

Wazuh operates as an open-source Security Information and Event Management (SIEM) system, monitoring logs and events for signals of possibly malicious activity. It sends alerts when it discovers irregular activity, like unusual login or config modifications.

pfSense Firewall: Network Segmentation

pfSense implements firewall rules and segmentation following Purdue model principles. It blocks unauthorized traffic between ICS layers and the IT network, reducing lateral movement.

Jump Server: Remote Connection Security

A committed jump server provides secure external access to ICS devices. This approach conforms to the principles of Zero Trust [16] by ensuring strong authentication, session logging, and restricting direct access to HMIs or PLCs.

Suricata IDS: Detecting ICS Cyber Threats

Suricata is designed to scan industrial protocols like Modbus and DNP3. It uses both signature-based and anomaly-based detection methods to determine malicious commands or scan activity, thus giving early warning for possible intrusions [18].

In summary, modern ICS and OT environments balance efficiency gains from connectivity and the increased risk posed by cyber threats [13][14]. Leading frameworks such as MITRE ATT&CK for ICS and Zero Trust Architecture illustrate how adversaries operate and how defenders can respond. Although the Purdue model remains essential for network segmentation, ongoing adaptations—like encryption and secure remote gateways—are vital to address current demands for cloud integration [15][16].

By building a testbed with GRFICSv2, Wazuh, pfSense, a jump server, and Suricata, this project demonstrates a layered approach to ICS security. The goal is to evaluate defense-in-depth methods under realistic scenarios, ultimately guiding industrial organizations toward robust cybersecurity without compromising operational reliability.

2.2 Related Work

Industrial Control Systems (ICS) are becoming more integrated with Information Technology (IT) networks, resulting in emerging cybersecurity threats. The conventional perimeter-based security models do not meet the demands of new cyber threats aimed at ICS environments. Zero Trust Architecture (ZTA), encrypted traffic inspection, and intrusion detection systems (IDS) have been areas of research as significant methods to improve ICS security. This section reviews ongoing research in these areas. It compares the limitations and gaps that this project aims to address by integrating ZTA, encrypted traffic analysis, and IDS within an ICS security testbed.

2.2.1 Zero Trust in ICS Security

Zero Trust Architecture (ZTA) has been studied as a cybersecurity framework for ICS networks. Zanasi et al.[19]propose an identity-based ZTA model that removes implicit trust and continuously authenticates access.

From their work, ZTA is seen to improve security through the blockade of lateral movement within the ICS network but lacks large-scale real-world deployments.

Similarly, Fang et al.[20] propose a Cyber-Physical ZTA that integrates adaptive access controls and behavioral analytics for monitoring system behavior. Their research illustrates that this approach enhances security in cyber-physical systems but is tested mainly in controlled laboratory simulations and not in production ICS environments.

Formby et al.[21] Analyze the impact of ZTA policies on ICS security by assessing system resilience to cyberattacks. They analyze different attack scenarios and determine the need for practical ZTA deployment in ICS networks. However, their work covers network segmentation and authentication but does not monitor encrypted traffic.

2.2.2 Encrypted Traffic Security and Intrusion Detection

As ICS communications become increasingly encrypted, network intrusion detection faces new challenges. Papadogiannaki et al. [22]discuss the impact of encryption on ICS monitoring. They demonstrate that encrypted Modbus/TCP traffic impedes traditional signature-based IDS solutions and requires alternative methods such as metadata-based anomaly detection.

Katulić et al. [14]propose Modbus/TCP be secured using Message Authentication Codes (MACs) to prevent unauthorized changes and replay attacks. MAC-based authentication in their work enhances data integrity but does not address intrusion detection within encrypted messages.

Martins and Oliveira [23] extend this work by adding role-based authentication to Modbus/TCP to strengthen access control. Their paper stresses the necessity for layered security in ICS but does not address the effect of encryption on IDS performance.

Oyewumi et al. [24] test IDS performance within a substation environment with rule-based and anomaly-based detection methods. They conclude that hybrid IDS solutions achieve better accuracy detecting ICS-specific threats but require fine-tuning to avoid false positives.

Almashhadani et al. [25] introduce an Industrial Control System Anomaly Detection Dataset (ICS-ADD) for real-time monitoring. Their dataset enables better evaluation of IDS models but is not designed explicitly for encrypted traffic analysis.

2.2.3 ICS Cybersecurity Testbeds

Testbeds are crucial in the validation of the ICS security framework. Oyewumi et al. [24] Describe the ISAAC Idaho CPS Smart Grid Cybersecurity Testbed, which simulates cyberattacks on SCADA systems to validate security mechanisms.

Mathur and Tippenhauer [26] Presents an advanced water treatment facility for cyber-physical security research: the SWaT testbed. While this allows for the simulation of realistic attacks, it lacks zero-trust implementation.

WADI testbed was introduced by Ahmed et al. [27], which expands on SWaT. The testbed models are water distribution networks. The focus is on intrusion detection, while encrypted traffic monitoring is absent in the testbed.

VICSORT is a virtualized ICS research testbed proposed by Ekisa et al. [28]. This testbed allows scalable cybersecurity experiments. It lowers the existing barriers to ICS security research but is confined to purely virtual environments.

2.2.4 Comparison of Related Studies

To better understand existing work and its limitations, Table 2.1 summarizes key studies and highlights research gaps.

Table 2. 1: Comparison of Related Studies.

Study	Focus Area	Security Approach	Testbed Used?	Research Gap
Zanasi et al. [19]	Zero Trust in ICS	Identity-based access control	No	No real-world implementation
Fang et al.[20]	Cyber-Physical ZTA	Adaptive access control	No	No application in operational ICS
Formby et al. [21]	Zero Trust in ICS	Attack impact analysis	Simulated	No encrypted traffic analysis

Papadogiannaki et al.[22]	Encrypted Traffic in ICS	Metadata-based IDS	No	No real-world testbed evaluation
Katulić et al. [14]	Modbus/TCP Security	Message Authentication Codes (MACs)	Yes	No IDS integration
Martins & Oliveira [23]	Modbus/TCP Security	Role-based authentication	Simulated	No focus on encrypted traffic
Oyewumi et al. [24]	SCADA Power Grid Security	Hybrid IDS	ISAAC	No Zero Trust implementation
Mathur & Tippenhauer [26]	ICS Water Treatment	Cyber-attack detection	SWaT	No ZTA, no encrypted traffic
Ahmed et al. [27]	ICS Water Distribution	Intrusion detection	WADI	No encrypted traffic monitoring
Ekisa et al. [28]	Virtual ICS Testbed	Open-source testbed	VICSORT	Limited to virtual environments

Research into ICS security has focused on applications of the Zero Trust model, encrypted traffic monitoring, and intrusion detection systems. Various approaches include identity-based ZTA, anomaly-based IDS, and secure communication protocols in industrial networks. Several testbeds, such as SWaT, ISAAC, and VICSORT, have been applied to testing cybersecurity solutions under controlled conditions.

Despite this, studies are typically limited to theoretical models, small-scale simulations, or isolated security mechanisms and rarely speak to the integration of various layers of security in a true-to-life ICS setting. Thus, few efforts exist to combine Zero Trust principles into an IDS for encrypted ICS traffic. Even fewer investigations have focused on bridging hybrid security solutions enveloping Suricata, Wazuh, and pfSense embedded in operational industrial environments using various testbeds.

This project will fill that gap by providing a testbed for ICS security that integrates ZTA, encrypted traffic monitoring, and IDS solutions, giving a lot more real-life and comprehensive results about modern ICS security challenges when applied in the field.

CHAPTER 3: PROJECT MANAGEMENT

3.1 Approach

The project was approached systematically by dividing the work into multiple phases, each addressing a specific aspect of building and securing the Industrial Control System (ICS) environment.

The major phases included:

- **Research and Planning**: Understanding ICS security challenges, selecting appropriate simulation and security tools.
- **Environment Setup**: Deploying GRFICSv2 for ICS simulation.
- **Network Design and Segmentation**: Creating a multi-zoned network using pfSense firewall.
- **Security Implementation**: Installing and configuring Wazuh SIEM and Suricata IDS.
- **Attack Simulation**: Performing controlled cyberattacks to validate the environment's security.
- **Testing and Evaluation**: Monitoring results, analyzing alerts, and identifying system improvements.

A phased approach ensured that each task was manageable and that risks were minimized through careful planning and incremental testing.

3.2 Initial Project Plan

3.2.1 Project Timeline & Task Allocation

Table 3.1 presents the key phases, task distribution, and dependencies.

Table 3. 1: Project Phases, Tasks, and Timeline.

Phase	Key Activities	Duration	Assigned Team Member(s)
Phase 1: Research & Literature Review	Study ICS security frameworks, Zero Trust, encryption techniques, and security gaps.	5 weeks (Feb 15 – Mar 21)	All Members
Phase 2: System Architecture & Security Design	Develop testbed architecture, define security controls, and document requirements.	2 weeks (Mar 22 – Apr 5)	Member 1
Phase 3: Implementation	Configure firewall (pfSense), IDS (Suricata), SIEM (Wazuh), and encrypted communication (stunnel).	3 weeks (Apr 6 – Apr 21)	Member 2, Member 3, Member 4
Phase 4: Security Testing & Evaluation	Conduct cyberattack simulations (Replay, Unauthorized Access). While MITM attacks were initially planned, they were excluded from this phase due to complexity and time limitations.	1 week (Apr 26 – May 3)	Member 2, Member 4
Phase 5: Documentation & Presentation	Finalize the report, incorporate feedback, and prepare the presentation.	3 weeks (May 4 – May 22)	All Members

A Gantt chart **Figure 3.1** is included to represent task dependencies and critical milestones visually.

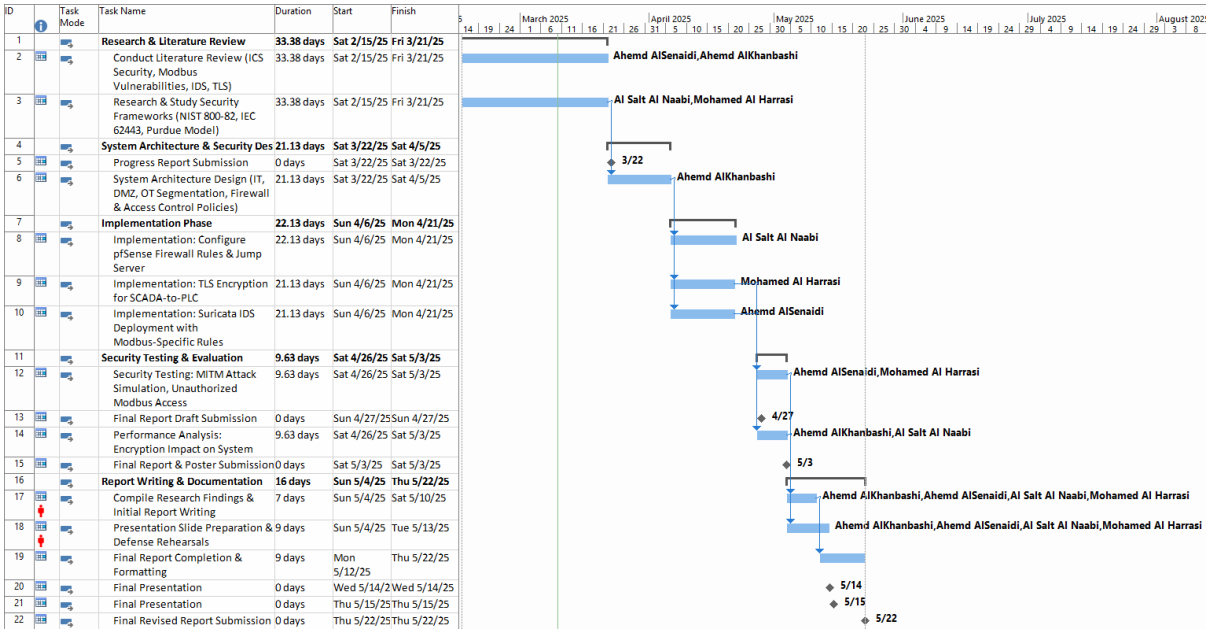


Figure 3. 1: Gantt Chart Representation of Project Timeline.

3.2.2 Risk Management & Mitigation Strategies

Table 3.2 outlines the identified risks and mitigation strategies to ensure uninterrupted progress.

Table 3. 2: Risk Assessment and Contingency Strategies.

Potential Risk	Impact	Mitigation Strategy
Software Configuration Issues	Delays in tool deployment	Early system testing and debugging
Network Failures	Disruptions in secure communication testing	Use a controlled virtual environment
Time Constraints	Risk of missing deadlines	Parallel documentation to ensure continuous progress
False Positives in IDS	Excessive alerts reducing detection accuracy	Fine-tuning Suricata IDS rules

Regular system audits and configuration testing will help reduce setbacks during implementation.

3.2.3 Evaluation & University Deadline Alignment

The project's success will be evaluated based on security effectiveness, system performance, and compliance. **Table 3.3** summarizes the evaluation metrics.

Table 3. 3: Evaluation Strategy & Criteria.

Metric	Evaluation Criteria
Security Effectiveness	IDS detection accuracy against MITRE ATT&CK-based attack simulations
Performance Analysis	Qualitative impact of TLS encryption on Modbus/TCP behavior; formal latency measurement deferred to future work
Compliance Check	Validation of security implementations against IEC 62443 & NIST 800-82 standards

The project schedule aligns with university deadlines, ensuring timely submission and presentation preparation.

Table 3. 4: University Submission Deadlines & Alignment.

Submission	Deadline	Plan Alignment
Progress Report	March 22, 2025	Literature review and system design completed.
Final Report Draft	April 27, 2025	Implementation is nearing completion.
Final Report & Poster	May 10-11, 2025	Security testing concludes before submission.
Presentation	May 14-15, 2025	1.5 weeks allocated for preparation.
Final Revised Report	May 22, 2025	Feedback incorporated.

3.3 Problems and Changes to the Plan

During the project execution, some challenges and necessary adjustments were encountered:

- **Wazuh Log Collection Issue:**

  The pfSense default syslog package did not integrate well with Wazuh. Solution: Syslog-ng package was installed and configured for correct log forwarding.

- **Modbus TCP Encryption:**

  TLS encryption was successfully configured using stunnel to protect Modbus TCP traffic between the SCADA server and PLCs. However, during the initial phase of testing, encryption was deliberately disabled to simulate real-world industrial environments where legacy systems often communicate in plain text. This allowed us to observe how attacks could succeed in unprotected systems.

  In the later testing phase, TLS encryption was re-enabled to evaluate its impact on blocking replay attacks and command injections. This staged approach allowed us to compare the system's performance and security posture under both insecure and secure communication conditions.

  The decision to toggle encryption was intentional and part of the test design—not due to a technical failure. This clarification was necessary to accurately assess the effectiveness of encryption within an ICS context.

- **Firewall Configuration Complexity:**

  Additional firewall rules were required to allow Wazuh agents to communicate correctly from all network zones, particularly from the DMZ and OT zones.

- **Multi-Factor Authentication Deployment:**

  MFA using Google Authenticator was implemented for SSH access to the Jumpserver, enhancing remote access security.

• **IDS Tuning:**

  Custom Suricata rules were written to detect specific Modbus TCP function codes, including read coils, write coils, and unauthorized register writes.

- **Alert Detection Limitations:**

  Wazuh and Suricata detected baseline network anomalies, but default rules did not provide specific Modbus TCP attack alerts. This was addressed by developing and applying custom Suricata detection rules for Modbus operations such as coil writes and shutdown commands.

Adjustments to the initial plan were made promptly to ensure project objectives remained achievable.

3.4 Final Project Plan

The final project plan was adapted to incorporate the adjustments and successfully completed all planned objectives.

- **Research and Planning**: Completed successfully.
- **Environment Setup**: All GRFICSv2 components deployed and operational.
- **Network Design and Segmentation**: pfSense firewall correctly segmented the networks.
- **Security Tool Deployment**: Wazuh and Suricata installed and configured.
- **Attack Simulation**: Modbus TCP command injection successfully executed from the Jumpserver.
- **Testing and Evaluation**: System monitored, logs collected, performance evaluated.
- **Documentation**: Final report, screenshots, and analysis prepared.

The flexibility in the project management approach allowed the team to handle unexpected challenges and deliver a complete, functional, and tested ICS security environment.

Chapter 4: Secure ICS Testbed Design and Security Requirements

This chapter specifies the requirements for which a modular Industrial Control System (ICS) cybersecurity testbed must be developed, with particular focus on Zero Trust principles and encrypted traffic monitoring. It outlines the problem modeling, functional and non-functional requirements, and initial design solutions that outline the system implementation.

4.1 Problem Modeling

4.1.1 Summary of ICS Security Issues

Industrial Control Systems (ICS) form the spine of critical infrastructure sectors, including energy, manufacturing, and transportation. The systems were not developed with a strong emphasis on cybersecurity capability but focused on their operation's reliability. A growing intertwining with Information Technology (IT) via Operational Technology (OT) broadened the potential attack vectors and rendered ICS environments acutely susceptible to cyber-attack [29].

4.1.2 Threat Landscape and Attack Vectors

ICS environments face multiple cyber threats, including:

- **Unauthorized Access:** Weak authentication mechanisms and lack of network segmentation allow attackers to gain control over critical components.
- **Man-in-the-Middle (MITM) Attacks:** Unencrypted communication channels expose ICS protocols, such as Modbus/TCP, to interception and manipulation.
- **Denial of Service (DoS) Attacks:** Attackers can disrupt ICS operations by overwhelming network resources.
- **Intrusion and Malware Attacks:** Sophisticated malware like Stuxnet has demonstrated the potential for ICS-targeted cyber sabotage.

4.1.3 Security Challenges in ICS Environments

ICS security concerns include:

- **Lack of Encryption**: Most ICS communication protocols do not use encryption, and thus data is vulnerable to eavesdropping [30].
- **Insecure Remote Access:** Unsecure Remote Access: Remote access systems do not have strict authentication policies [31].
- **Lack of Intrusion Detection Systems (IDS**): Conventional ICS configurations do not have IDS solutions tailored for industrial network protocols [32].

The proposed testbed seeks to overcome these shortcomings by introducing Zero Trust models, encrypted communications, and strong intrusion detection methods.

4.2 Functional Requirements

The following functional requirements specify the anticipated functions of the ICS cybersecurity testbed:

FR1: Secure Network Segmentation

- **Description:** The system shall implement firewall policies to enforce micro-segmentation and isolate OT, IT, and DMZ networks based on Zero Trust principles.
- **Measurement and Validation:** Validate the segmentation through pfSense firewall rules and VLAN isolation and verify that unauthorized IT-to-OT communication is blocked via penetration testing.

FR2: Confidentiality of ICS Traffic

- **Description:** Description: SCADA-to-PLC communication over Modbus TCP must be secured using TLS encryption to prevent eavesdropping and data manipulation.
- **Measurement & Validation:** stunnel was configured to wrap Modbus TCP traffic with TLS encryption. Wireshark packet captures were used to confirm encryption

effectiveness. Other ICS traffic was not encrypted in this phase due to resource and time constraints, and is proposed as future work.

FR3: Threat Monitoring and Intrusion Detection

- **Description:** Suricata IDS must be installed to monitor, detect, and log anomalies in ICS-specific protocols such as Modbus/TCP.
- **Measurement & Validation:** The IDS should alert unauthorized Modbus function codes, anomalous traffic patterns, and replay attacks, with logs analyzed in Wazuh SIEM.

FR4: Secure Authentication and Access Control

- **Description:** The system implements multi-factor authentication (MFA) to protect access to the OT network. All remote access is routed through a Jumpserver in the DMZ, and MFA ensures that only authorized users with verified credentials and a second authentication factor can initiate sessions with industrial devices.
- **Measurement & Validation:** MFA was configured using Google Authenticator for SSH access to the Jumpserver. Each user must provide a username, password, and a time-based one-time password (TOTP) generated by the authenticator app. All login attempts are monitored and logged in Wazuh SIEM for auditing and compliance verification.

FR5: Security Auditing and Logging

- **Description:** The system must possess detailed security logs for forensic analysis and auditing for compliance. Logs must include authentication attempts, network activity, and intrusion detection alerts.
- **Measurement & Validation:** All logs must be collected centrally in Wazuh SIEM with periodic log audits to ensure log integrity and compliance with NIST SP 800-82.

*Note: Some originally planned features, such as full protocol-wide encryption and detailed performance benchmarking, were partially implemented due to resource limitations. These areas are acknowledged in Chapter 8 as future work. *

4.3 Non-functional Requirements

The non-functional requirements define the testbed's performance, security, and compliance expectations.

NFR1: System Performance

- Encryption overhead was considered in the system design and observed qualitatively during SCADA polling with and without TLS. However, no formal latency benchmark or percentage increase was recorded. Future development should include performance profiling with packet timing.

NFR2: Security Standards Compliance

- The system will adhere to NIST SP 800-82 and IEC 62443 standards to facilitate secure industrial control system practices. Security compliance is based on network segmentation (pfSense firewalls), role-based access control (RBAC), multi-factor authentication (MFA), encrypted communication (TLS over Modbus), and intrusion detection (Suricata IDS). A minimum of five security controls are to be applied as specified by IEC 62443-3-3 (SR 3.1, 3.2). Compliance will be validated through security audits, penetration testing, and log analysis. As per IEC 62443-3-3, the security architecture of ICS must implement multi-layered defense mechanisms such as segmentation and anomaly detection.

NFR3: Scalability

- The testbed must be scalable to accommodate additional security mechanisms, such as additional IDS rules, honeypots, machine learning-based anomaly detection, and SIEM solutions. Scalability relies on a modular network architecture, virtualization (Proxmox/EVE-NG), and pfSense-based network segmentation, which will enable the incorporation of additional security elements without extensive reconfigurations. Testing will verify that new deployments do not result in system instability or significant redesigns.

IEC 62443-3-3 (SR 7.1) mandates that ICS environments must be built for future scalability and modular security upgrades.

NFR4: Reliability and Availability

- 99.9% uptime in Industrial Control Systems (ICS) is a critical requirement meeting IEC 62443-3-3 (SR 7.6) and NIST SP 800-82 (Section 3.2.4) to minimize operational downtime. The industry standards provide that 99.9% uptime—providing 8.76 hours of downtime in a year—is necessary for mission-critical ICS systems, with greater levels such as 99.99% (providing 52.56 minutes of downtime) and 99.999% (providing 5.26 minutes of downtime) being preferable for SCADA and telecommunication networks. This 99.9% threshold is a compromise between reliability, redundancy, and security that aligns with best practices. Cisco's IEC 62443 whitepaper and NIST's OT Security Guide provide methodologies for high availability via redundancy, failover capabilities, and monitoring in advance.

NFR5: User Training and Research Support

- The testbed should be a cybersecurity research education platform, providing hands-on training in ICS security, network monitoring, and incident response. Training effectiveness is based on detailed documentation, pre-configured security tools, simulated cyberattack environments, and interactive laboratory exercises. The testbed will provide step-by-step tutorials, interactive laboratories, and research experiment documentation to enable researchers and students to use them. NIST SP 800-82 encourages the development of ICS security training environments to enhance cybersecurity skills in critical infrastructure sectors.

4.4 Preliminary Design Solutions

The proposed ICS cybersecurity testbed integrates network segmentation, encrypted communication, and intrusion detection to enhance security and resilience against cyber threats.

4.4.1 Network Architecture

The testbed follows the Purdue Model for ICS Security, implementing multiple security layers to enforce strict network segmentation and limit lateral movement between IT and OT environments.

4.4.2 Security Components

- **Firewall (pfSense):** Filters network traffic and enforces micro-segmentation to isolate IT and ICS environments.
- **Suricata IDS:** Monitors and detects anomalies in industrial protocols (e.g., Modbus/TCP) to identify potential cyber threats.
- **TLS Encryption (stunnel**): Encrypts SCADA-to-PLC communication to ensure data confidentiality and prevent eavesdropping or tampering.
- **SIEM (Wazuh):** Collects, analyzes, and correlates security logs for real-time threat detection and forensic investigations.
- **Jump Server:** This server functions as a secure access gateway between IT and OT networks, enforcing MFA, access control, logging, and encrypted communication to restrict unauthorized access.

4.4.3Attack Simulation and Evaluation

To evaluate the robustness of the implemented ICS cybersecurity architecture, a set of controlled cyberattacks was designed and executed within the testbed. These simulations focused on common and high-impact threats to industrial protocols and network zones.

Attack scenarios were primarily based on **Modbus TCP vulnerabilities**, including:

- Unauthorized coil write commands
- Modbus function abuse (e.g., write to register)
- Payload replay and command injection attempts

The attacks were executed using the **Metasploit framework** and custom traffic injection tools. Suricata IDS and Wazuh SIEM were used to detect, log, and correlate these malicious activities in both **unencrypted and encrypted traffic conditions**.

Although **MITM attacks and lateral movement** were included in the initial test plan, they were not fully executed due to time and tooling limitations. These scenarios were deferred in favor of deeper testing of encrypted vs. unencrypted Modbus behavior. However, the current

testbed design is modular and supports the future addition of MITM simulations using tools such as Ettercap or Bettercap.

Replay attacks were partially simulated through repeated Modbus command injections before and after enabling TLS encryption. Further development would be needed to simulate synchronized replay attacks across multiple stages.

Chapter 5: Project Design

5.1 Product Features

The primary outcome of this project is a virtualized, secure **Industrial Control System (ICS) testbed**, designed to replicate real-world industrial environments. The system allows simulation of industrial processes and incorporates modern cybersecurity mechanisms to detect and respond to potential threats. The testbed is structured to enable secure monitoring, access control, and attack simulation in a controlled setting.

Key product features include:

- **ICS Simulation:** Utilization of GRFICSv2 to emulate SCADA systems, programmable logic controllers (PLCs), and plant processes.
- **Network Segmentation:** Logical separation of IT, DMZ, and OT zones through pfSense firewall configuration.
- **Controlled Access via Jump Server:** A centrally placed Jump Server enforces restricted access to the OT network.
- **Security Monitoring:** Integration of Wazuh SIEM and Suricata IDS for real-time threat detection and centralized log management.
- **Attack Simulation:** Execution of Modbus TCP-based attacks to test and evaluate the testbed's defensive capabilities.

5.2 User Interface

The testbed is primarily accessed through web-based dashboards and secure terminal sessions. Interfaces are provided for different components to enable system monitoring, management, and interaction by users such as students and cybersecurity researchers.

Main interfaces include:

- GRFICSv2 Web Interface: Displays industrial process simulations, SCADA dashboard, and plant status.

- pfSense Web GUI: Used for managing firewall rules, interface assignments, and traffic monitoring.
- Wazuh SIEM Dashboard: Central platform for viewing logs, alerts, and network behavior analytics.
- Suricata Console or Web Integration: Provides visibility into IDS alerts and network anomalies.

All interfaces are accessed via standard web browsers or secure shell (SSH) terminals from the IT Zone.

5.3 Interfaces to External Hardware and Software

The system operates entirely within a virtualized environment and does not interface with physical industrial hardware. All components are implemented using open-source software and virtual machines hosted on a local hypervisor.

Software components integrated:

- GRFICSv2: For simulating plant operations, SCADA systems, and PLC communication.
- pfSense: Deployed as a firewall and router to implement network segmentation.
- Wazuh: Serves as the Security Information and Event Management (SIEM) platform.
- Suricata: Implements network intrusion detection capabilities.
- Metasploit Framework: Used to conduct ethical attack simulations on the testbed.

5.4 Functional Requirements

The testbed was designed to meet the following core functional requirements:

- **ICS Simulation:** Accurate emulation of industrial network traffic and devices using GRFICSv2.

- **Zone-Based Segmentation:** Creation of isolated zones (IT, DMZ, OT) with firewall rules to enforce access boundaries.
- **Controlled Access:** Enforcement of indirect access to OT systems exclusively through the Jump Server.
- **Multi-Factor Authentication:** SSH access to the Jump Server is protected using Google Authenticator for time-based one-time password (TOTP) verification.
- **Security Monitoring:** Real-time logging and detection of anomalies across all zones using Suricata and Wazuh.
- **Attack Execution Capability:** Support for launching simulated cyberattacks without impacting real systems.

5.5 Non-Functional Requirements

Non-functional aspects were also considered to ensure usability and future extensibility:

- **Reliability:** The system must remain stable during simulation and testing without crashes.
- **Security:** All inter-zone communications are strictly regulated to prevent unauthorized access.
- **Scalability:** The architecture is designed to support the future addition of components, attack scenarios, or defenses.
- **Usability:** Interfaces are intuitive and accessible from standard web browsers and terminals, ensuring ease of use for educational purposes.

5.6 Data Storage

Log and alert data generated by Suricata and Wazuh are stored securely on the Wazuh server. The data supports forensic analysis, auditing, and threat detection.

Stored data includes:

- **Event Logs:** Records of network activity, user authentication, and system events.

- **Alert Logs:** Intrusion detection results highlighting suspicious behavior.
- **Firewall Logs:** pfSense sends its logs via syslog-ng to the Wazuh server for centralized visibility.

5.7 High-Level Design

The testbed follows a multi-zone architecture modeled on the Purdue Model and Zero Trust principles. Each zone is logically segmented and assigned specific roles:

- **WAN Zone:** External internet access.
- **LAN1 – IT Zone:** Hosts Wazuh server, monitoring workstation, and attacker simulation machine.
- **LAN2 – DMZ Zone:** Contains the Jump Server and Suricata IDS.
- **LAN3 – OT Zone:** Hosts SCADA systems, PLCs, plant simulation (GRFICSv2), and engineering workstation.

The architectural layout is illustrated in **Figure 5.1** and **Figure 5.2**, emphasizing the isolation of critical assets and the flow of monitored traffic across zones.

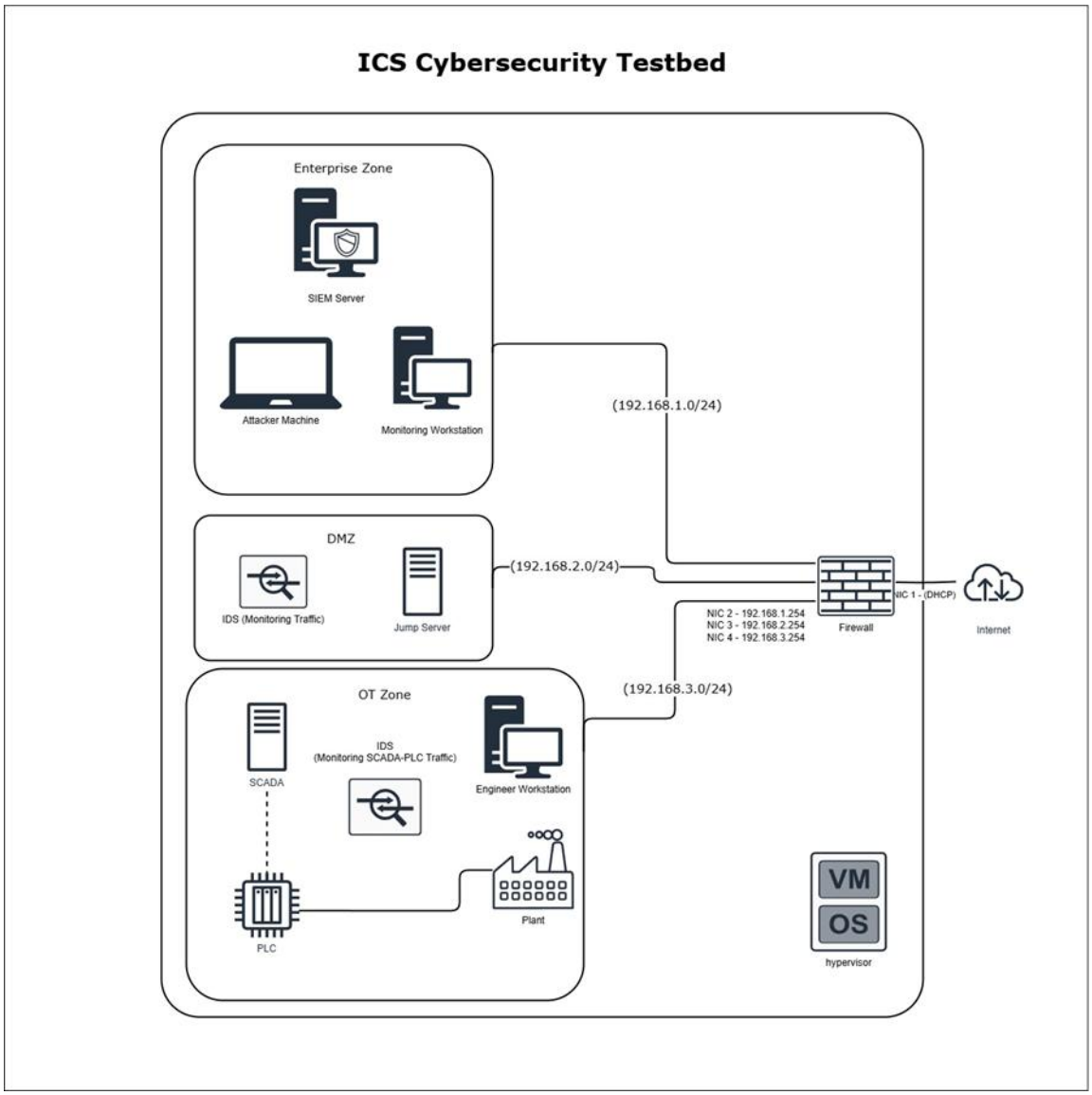


Figure 5. 1: High-Level ICS Testbed Network Architecture.

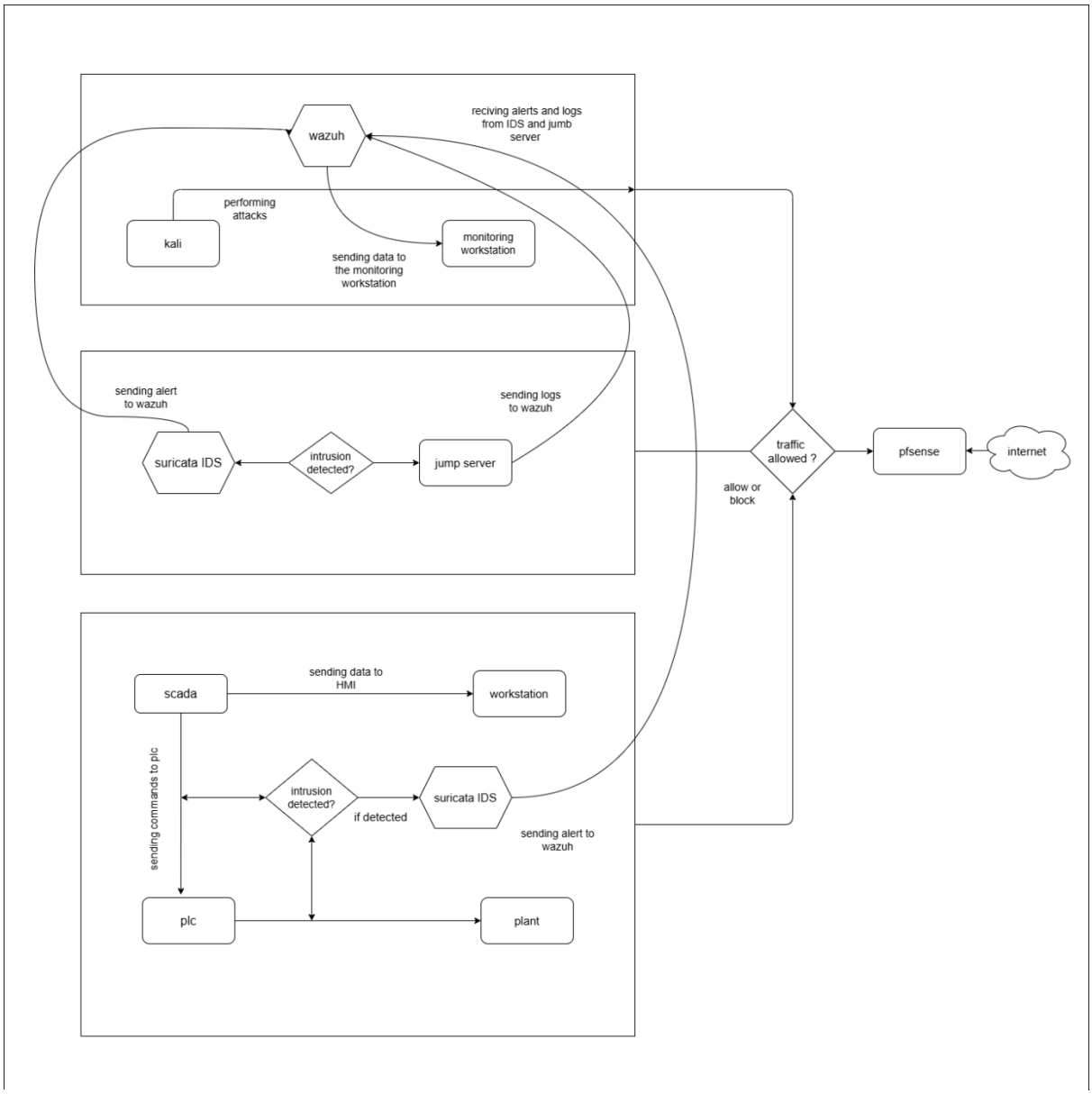


Figure 5. 2: Data Flow Diagram of the ICS Cybersecurity Testbed.

5.8 Detailed Design

The detailed system design outlines the specific configurations and security mechanisms used to enforce segmentation and monitoring.

- **pfSense Firewall:** Configured with rule sets that permit only authorized traffic. Examples include:
    - Allow SSH from the IT zone to the Jump Server (DMZ).
    - Allow SCADA-to-PLC encrypted communication via stunnel.
    - Deny all direct traffic between IT and OT networks.
- **Jump Server:** Acts as an intermediary with logging and session control. All SSH access to OT systems is routed through it and protected using MFA. Google Authenticator was configured for TOTP verification during SSH login, enhancing remote access security.
- **Suricata IDS:** Monitors mirrored traffic from all zones and forwards alerts to Wazuh.
- **Stunnel:** Configured for TLS encryption of Modbus TCP traffic. During attack testing, encryption is disabled to simulate realistic threat exposure.

5.9 Design Verification

The design was validated through structured testing of functionality, segmentation, and response to simulated attacks.

- **Firewall Testing:** Confirmed that unauthorized direct traffic between zones was blocked by pfSense.
- **Access Control Testing:** Verified that OT systems could only be reached through the Jump Server.
- **Attack Simulation:** Executed Modbus TCP shutdown and command injection attacks to test resilience.

- **Monitoring Validation:** Confirmed that Suricata detected attacks and Wazuh recorded and displayed alert events.

Each subsystem was tested individually and in combination to confirm the reliability, functionality, and security of the full architecture.

CHAPTER 6: PROJECT IMPLEMENTATION

This chapter outlines the detailed implementation process of the ICS cybersecurity testbed, which was based on Zero Trust Architecture and best practices from NIST SP 800-82. The testbed integrates open-source tools to simulate a realistic, segmented industrial network with layered defenses. Implementation tasks involved configuring firewalls, deploying security monitoring systems, establishing encrypted communication channels, and validating access control policies through simulated attacks. The entire system was deployed virtually using VirtualBox and involved components aligned with the Purdue Model, such as IT (Level 4), DMZ (Level 3.5), and OT (Levels 0–2) zones.

6.1 System Configuration and Deployment

The system configuration phase primarily involved deploying and fine-tuning open-source components to simulate a secure ICS environment. While no traditional coding was required, the work involved precise configuration of firewalls, logging mechanisms, intrusion detection systems, and encryption tools.

6.1.1 Firewall Configuration and Network Segmentation

To enhance security and enforce network isolation in the ICS cybersecurity testbed, **pfSense**, an open-source firewall, was deployed as a core control mechanism. The firewall was configured to segment the network into four logically separated zones, each assigned a specific subnet and **VirtualBox network adapter**:

- **Adapter 1 – WAN Zone (Internet-facing):** Dynamic IP via NAT
- **Adapter 2 – LAN1 (Enterprise IT Network):** 192.168.1.0/24
- **Adapter 3 – LAN2 (Demilitarized Zone - DMZ):** 192.168.2.0/24
- **Adapter 4 – LAN3 (Operational Technology - OT Network):** 192.168.3.0/24

These interfaces were configured using VirtualBox's **internal network mode**, providing strict segmentation for IT, DMZ, and OT layers. Screenshots of each adapter configuration are included in **Appendix B**.

Segmentation was applied following the Zero Trust principle of "never trust, always verify." A default-deny policy was enforced on all interfaces, ensuring that no inter-zone traffic could occur without explicit permission. This approach was critical in protecting the sensitive OT systems from potential threats originating from the less-secure IT environment or external networks.

Firewall Rules Implementation

Specific firewall rules were created for each zone to strictly control communication:

Enterprise IT (LAN1) Rules:

- HTTPS access (port 443) was allowed for firewall administration through the pfSense Web GUI.
- IT administrators were permitted access to the Wazuh SIEM Server for centralized security monitoring.
- SSH access to the Jump Server in the DMZ was allowed for authorized personnel, enabling indirect management of OT assets.
- Wazuh Agents on IT systems were allowed to send logs to the Wazuh Server.
- All other traffic from IT to DMZ and OT zones was explicitly blocked.

DMZ (LAN2) Rules:

- SSH traffic from the Jump Server to the SCADA system was permitted for operational management.
- Jump Server and IDS devices were allowed to forward security logs to the Wazuh Server.
- Direct access from the Jump Server to PLCs was blocked to prevent unauthorized control or monitoring of OT equipment.
- All non-specified traffic in the DMZ was denied by default.

Operational Technology (LAN3) Rules:

- TLS-encrypted Modbus communications were allowed between SCADA and PLC devices through a stunnel proxy.
- Controlled SSH access was permitted from the Jump Server to the SCADA system only.
- Wazuh Agents in the OT environment were permitted to send logs to the Wazuh Server for centralized monitoring.
- All other access attempts within the OT Zone were denied.

Firewall Architecture and Security Control

The firewall enforced a strict access hierarchy in the testbed environment.

Direct communication from the IT Network to the OT Zone was not permitted; instead, connections had to be established through the Jump Server residing in the DMZ.

This design ensured that operational systems remained isolated from external threats and unauthorized IT-originated attacks.

All firewall events, including blocked or allowed connections, were logged and forwarded to the Wazuh SIEM Server, providing full visibility and traceability of network activities.

As shown in **Figure 6.1**, pfSense was configured with explicit allow and deny rules within the IT Zone (LAN1) to enforce controlled access to security services such as Wazuh and the Jump Server while blocking unauthorized communications.

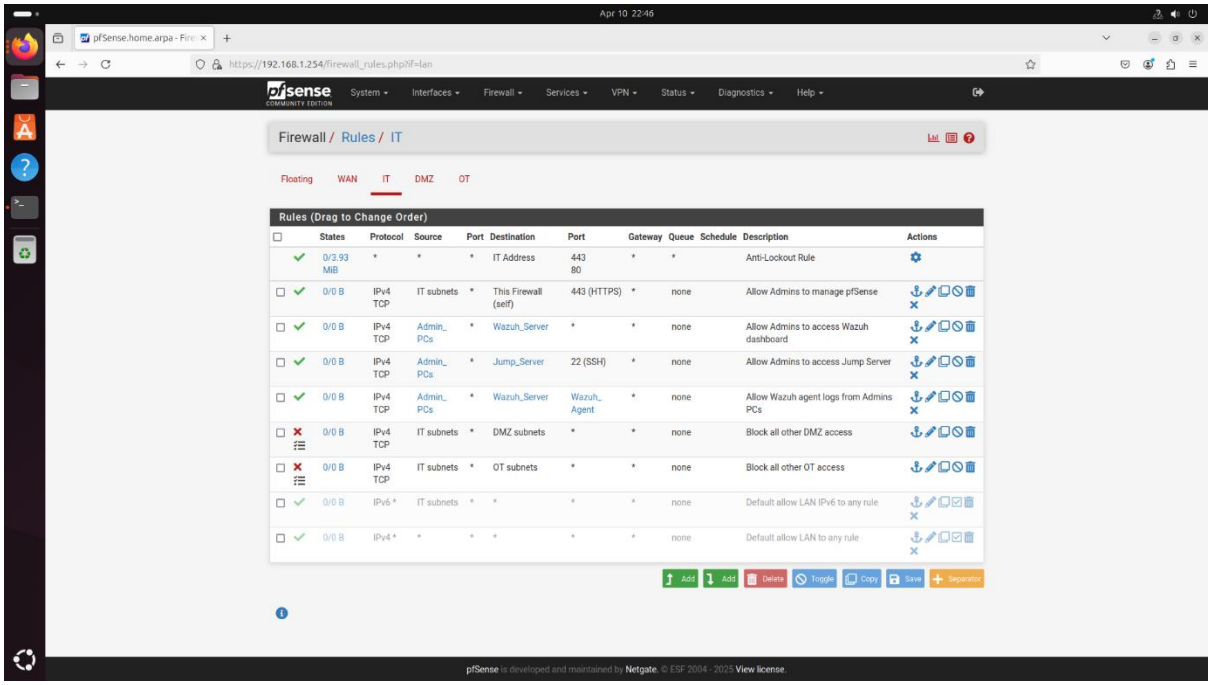


Figure 6. 1: pfSense Firewall Rules Configured for Network Segmentation.

For complete firewall rule sets, interface configurations, and additional technical settings, refer to **Appendix B**.

6.1.2 Log Forwarding to Wazuh SIEM

Centralized log management is a critical component of any secure ICS architecture. In this project, **pfSense** was configured to forward its firewall and system logs to a centralized **Wazuh SIEM** server for real-time monitoring, threat detection, and compliance tracking.

Wazuh is an open-source Security Information and Event Management (SIEM) solution that offers powerful capabilities such as log aggregation, correlation analysis, alert generation, and compliance reporting. Wazuh was selected for this testbed due to its strong support for ICS environments, customizable detection rules, and open-source flexibility.

Log forwarding from pfSense to Wazuh was implemented using **Syslog-ng**, a secure and scalable logging agent. Syslog-ng was configured on pfSense to forward logs over the internal network to the Wazuh server located in the Enterprise Zone.

The types of logs forwarded included:

- Firewall rule events (allowed and blocked traffic)
- System events (reboots, configuration changes)
- Authentication events (login attempts, SSH access)
- Networking alerts (interface status changes)

As illustrated in **Figure 6.2**, the firewall events from pfSense were successfully ingested into the Wazuh dashboard, enabling comprehensive visualization, analysis, and correlation with other network activities.

The centralized SIEM setup provided several advantages:

- Real-time alerting for suspicious or unauthorized activities
- Forensic analysis capabilities for incident response
- Continuous compliance monitoring with industry security standards
- Full audit trails for network communications and firewall activities

This centralized visibility is critical for enforcing Zero Trust policies and maintaining a secure and resilient ICS environment.

Full log forwarding setup details and Syslog-ng configuration files are included in **Appendix B**.

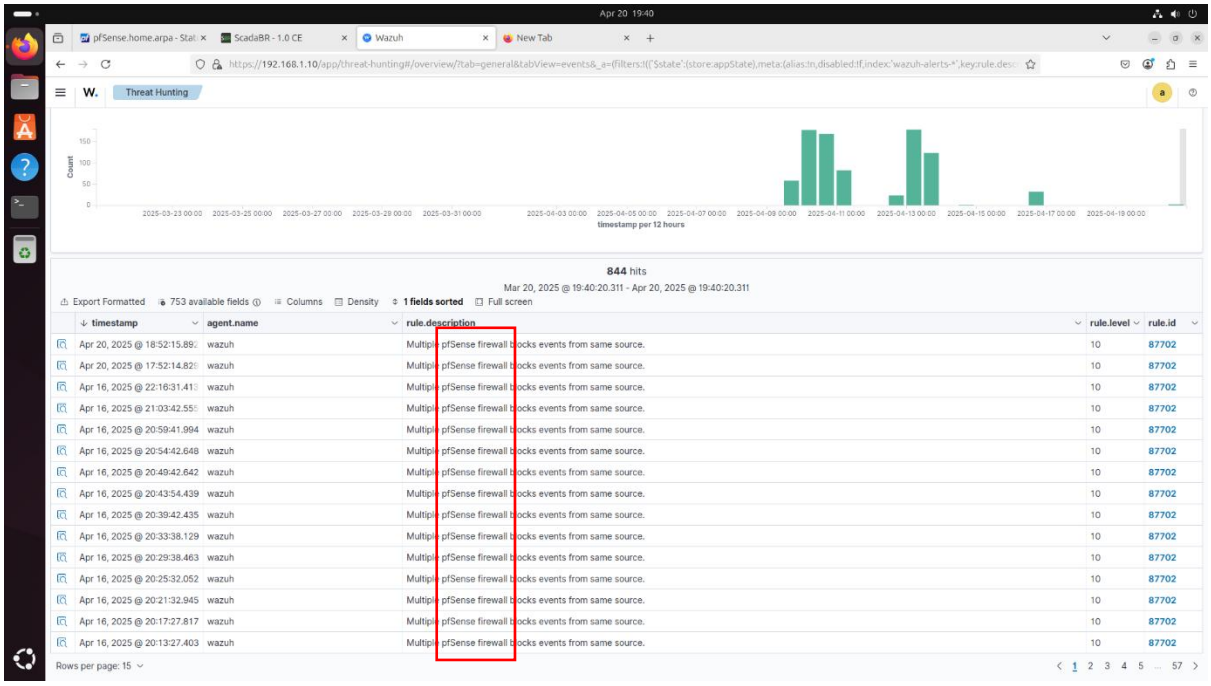


Figure 6. 2: Wazuh Dashboard Displaying Collected Firewall Logs.

6.1.3 Intrusion Detection System (IDS) Setup

To enhance monitoring and detection across the ICS network, two instances of the Suricata Intrusion Detection System (IDS) were deployed: one on the **Jump Server in the DMZ** and another inside the **OT zone**. This dual-deployment architecture provided layered visibility into both IT-to-OT traffic and local OT communications, aligning with the defense-in-depth principle of industrial cybersecurity.

Suricata in the DMZ monitored mirrored traffic from the IT zone (LAN1), the DMZ itself (LAN2), and inbound traffic into the OT zone (LAN3). Meanwhile, the OT-based Suricata instance focused on internal OT traffic, including SCADA-to-PLC communication, allowing for more granular inspection of Modbus commands at the process layer.

Traffic mirroring was configured using SPAN ports on virtual switches and the pfSense firewall. Both Suricata nodes operated in passive mode and did not interfere with normal network operations.

A set of **custom ICS-specific rules** was developed to detect Modbus TCP anomalies, including:

- Function Code 0x01 – Read Coils
- Function Code 0x05 – Write Single Coil
- Function Code 0x10 – Write Multiple Registers
- Function Code 0x06 – Preset Single Register (unauthorized access attempt)
- Coil write to address 0x0000 – used for emergency shutdown simulation

Sample rule:

alert tcp any any -> any 502 (msg:"Modbus TCP Shutdown Coil Write (0x0000)"; content:"|00 00|"; offset:2; depth:2; content:"|05 00 00 00 00|"; offset:7; depth:5; classtype:attempted-dos; sid:1000005; rev:1;)

Alerts from both IDS nodes were forwarded to the centralized **Wazuh SIEM** for aggregation and correlation. This provided unified visibility through the Wazuh dashboard, where Suricata alerts were displayed alongside firewall and system event logs **Figure 6.3**.

Limitations included reduced visibility when TLS encryption was enabled (e.g., in Scenario 6 and 7), since encrypted payloads could not be inspected by Suricata. To address this, future work could explore encrypted traffic metadata analysis, endpoint detection, or Suricata's Modbus protocol parser.

The rules were manually tested during unencrypted attack simulations and adjusted for accuracy to reduce false positives while maintaining effective Modbus threat detection.

All custom rules and real alert examples are included in **Appendix D**.

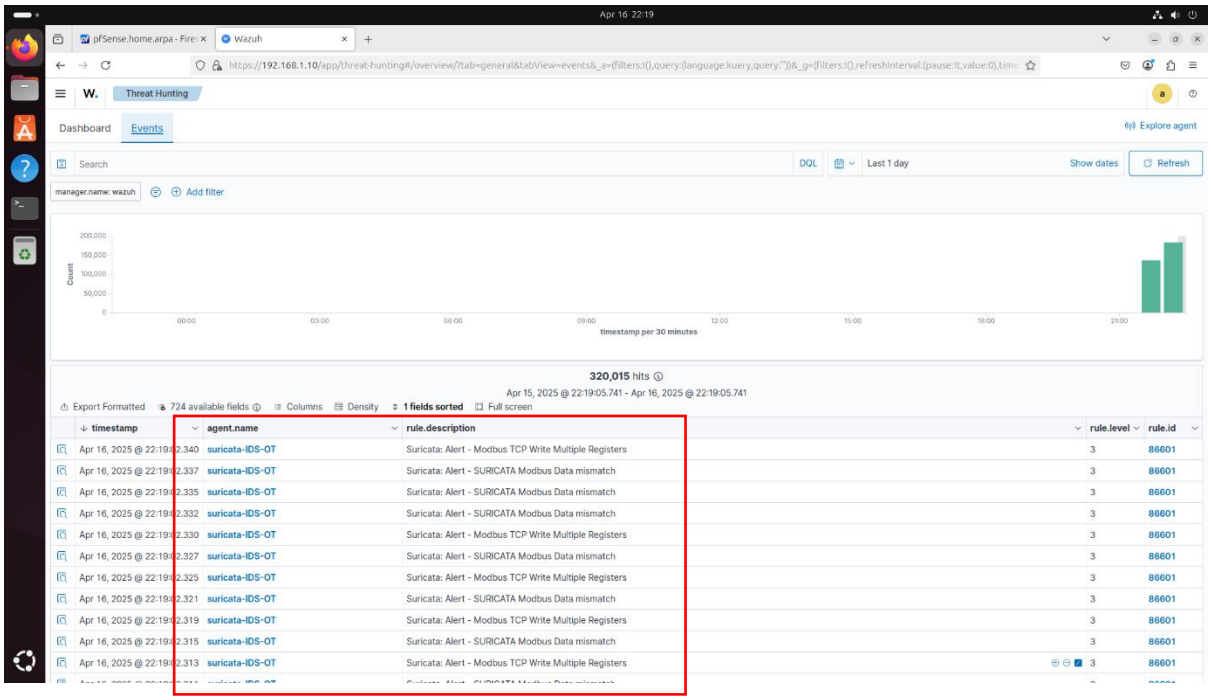


Figure 6. 3: Wazuh Dashboard Displaying Suricata-Detected Anomalies.

6.1.4 Jumpserver Deployment

To enforce strict access control between the Enterprise (IT) and Operational Technology (OT) zones, a Jumpserver was deployed in the DMZ as the only authorized intermediary for accessing operational systems. This deployment followed Zero Trust principles, ensuring that no direct communication from the IT network to critical OT assets such as the SCADA server or PLCs was possible.

The Jumpserver served as a secure gateway through which IT administrators and engineers could initiate controlled and monitored SSH sessions to OT systems. Multi-factor authentication (MFA) was enabled on the Jumpserver using Google Authenticator to ensure

enhanced identity verification. Users were required to enter their SSH credentials and a time-based one-time password (TOTP) generated by the authenticator app. MFA was integrated using the Pluggable Authentication Module (PAM) on a Linux-based system.

Access control policies on the Jumpserver and pfSense firewall were configured to:

- Allow SSH connections only from approved IT Zone workstations
- Limit access to selected OT devices (e.g., SCADA server) while blocking direct access to PLCs
- Enforce session timeout and inactivity lock policies
- Log all authentication attempts and session activities for auditing

All access logs and SSH authentication events were forwarded to Wazuh SIEM, enabling centralized visibility and detection of suspicious login behavior. The presence of the Jumpserver significantly reduced the risk of lateral movement by potential attackers. Even in the case of credential theft, attackers would be forced to authenticate using MFA, and all activity would be logged for forensic analysis.

As shown in **Figure 6.4**, a successful SSH session was established from an IT workstation to the Jumpserver, followed by controlled access to the SCADA server. All activities were securely logged to support compliance and incident response procedures.

Additional session logs, PAM configuration details, and SSH security settings are provided in **Appendix E**.

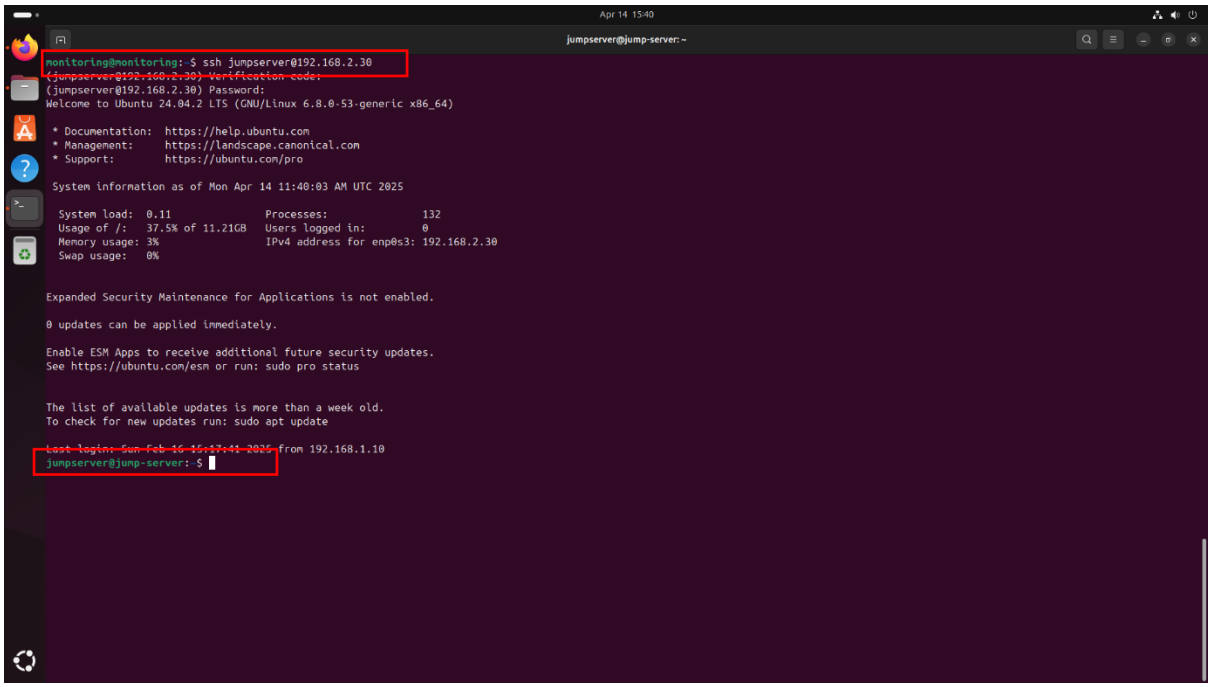


Figure 6. 4: SSH Connection to Jumpserver Demonstrating Controlled Access.

6.1.5 Encrypted Modbus Communication with stunnel

In the default architecture of ICS environments, Modbus TCP traffic between SCADA systems and PLCs is typically unencrypted, exposing critical industrial operations to risks such as interception, manipulation, and replay attacks. This plaintext communication poses a serious threat to the integrity and confidentiality of control commands and sensor data within industrial systems.

To address this vulnerability, stunnel was deployed as a lightweight TLS tunneling proxy. Stunnel wraps unencrypted Modbus TCP traffic inside a secure TLS (Transport Layer Security) tunnel, ensuring that all communications between the SCADA system and PLCs are encrypted using certificates generated with OpenSSL.

The stunnel service acts as an intermediary, listening on a secure port, encrypting outgoing traffic, and decrypting incoming traffic transparently. This ensures that legacy Modbus TCP applications, which are not natively TLS-aware, can still benefit from modern encryption mechanisms without requiring modifications to the SCADA or PLC systems.

As shown in **Figure 6.5**, the stunnel configuration files were set up to define secure client-server communication endpoints. The client side of stunnel (running near the SCADA system) accepts local connections on port 5502 and forwards them encrypted to the PLC stunnel server side listening on port 1502.

The key configuration parameters include:

- accept: Specifies the local port to accept connections.
- connect: Defines the remote server and port to forward encrypted traffic.
- cert and key: SSL/TLS certificates and private keys used to establish trust and encrypt communications.

Proper certificate generation and exchange were done beforehand to ensure mutual authentication between both ends of the tunnel, increasing the trustworthiness of the channel.

After deploying stunnel, Wireshark was used to capture network traffic for verification. As seen in **Figure 6.6**, the traffic formerly visible as plaintext Modbus TCP packets (such as read/write coil commands) is now encapsulated within TLS records. The encrypted payload prevents an attacker from reading, altering, or replaying sensitive Modbus commands.

As part of the planned test design, TLS encryption was temporarily disabled during early attack simulations to replicate common real-world ICS configurations where legacy systems still use Modbus TCP in plaintext. This allowed us to observe the system's exposure to command injection and replay attacks under unprotected conditions. This step simulated realistic ICS conditions where legacy industrial systems often lack encryption, allowing us to assess the relative security benefit gained when TLS is later applied. In later scenarios, encryption was re-enabled to directly compare the impact of TLS on attack success rates and IDS visibility. This staged approach provided a clearer understanding of how encryption improves both confidentiality and operational security in industrial environments.

This significant security enhancement protects the ICS network from passive eavesdropping and active man-in-the-middle attacks (MITM), dramatically improving the confidentiality and integrity of industrial control traffic.

Configuration examples and certificate generation steps can be found in **Appendix F**.

```
  GNU nano 2.7.4            File: /etc/stunnel/stunnel.conf

[modbus-secure]
client = yes
accept = 127.0.0.1:5502
connect = 192.168.3.2:1502
CAfile = /etc/stunnel/stunnel.pem

  GNU nano 2.5.3            File: /etc/stunnel/stunnel.conf

[modbus_secure]
accept = 1502
connect = 127.0.0.1:502
cert = /etc/stunnel/stunnel.pem
key = /etc/stunnel/stunnel.key
```

Figure 6. 5: stunnel Configuration File Securing Modbus Communications.

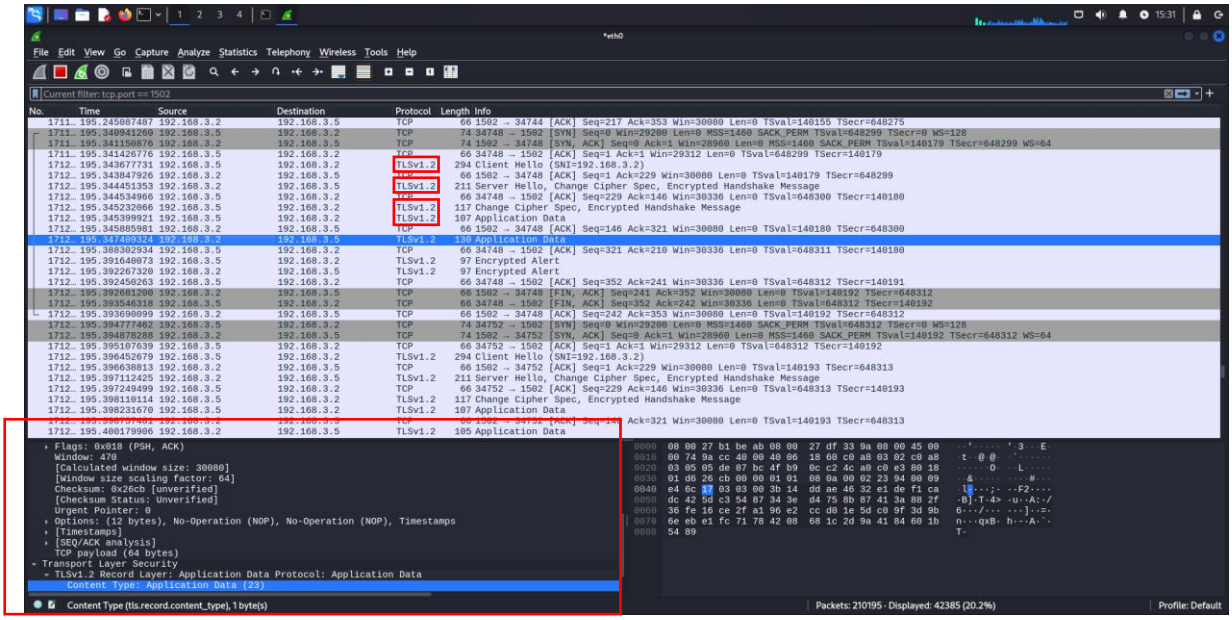


Figure 6. 6: Wireshark Capture Confirming Encrypted Modbus Traffic.

6.2 Verification

At each implementation phase, systematic verification tasks were performed to ensure correct functionality. Firewall segmentation was tested by attempting direct SSH and Ping access from IT to OT — these were blocked by pfSense, confirming Zero Trust enforcement. Access through the Jumpserver was allowed, confirming rule precision.

The Wazuh dashboard verified receipt of logs from pfSense, and Suricata was observed capturing network anomalies. A Metasploit test using Modbus command injection was successful prior to TLS deployment and failed after stunnel was enabled, verifying the encryption's effectiveness.

Each component performed as expected, validating the correctness of the configured architecture.

6.3 Validation

System validation focused on confirming that all functional and non-functional requirements were fulfilled. The firewall correctly segmented the network, and Wazuh and Suricata provided centralized and detailed monitoring. Modbus attacks were successfully detected or mitigated, depending on the encryption status.

Jumpserver logs verified that access controls were enforced, and all activity between IT and OT zones passed through a monitored, authenticated gateway. The simulated attacks demonstrated that unencrypted ICS protocols are highly vulnerable, and that proper architecture and monitoring can prevent real damage.

This simulation achieved its objective of building a Zero Trust-compliant ICS testbed that highlights cybersecurity gaps and offers practical defense mechanisms.

CHAPTER 7: PROJECT TESTING (EVALUATION)

This chapter provides a comprehensive evaluation of the Secure ICS Testbed by simulating realistic attack scenarios and validating the effectiveness of the implemented Zero Trust Architecture. Each scenario is presented as a hands-on experiment that highlights the design objectives, testing activities, outcomes, and contributions to the overall system integrity. The goal is to show the rigorous testing conducted, the challenges encountered, and the cybersecurity improvements achieved. The detailed procedures and results are supported by visual evidence and additional materials provided in the Appendices.

7.1 Testing Metrics and Goals

The testing strategy was designed to assess multiple dimensions of cybersecurity resilience in the ICS testbed. The primary goals were to measure the attack success rate, confirm the enforcement of network segmentation, verify monitoring efficiency using Wazuh and Suricata, validate the strength of TLS encryption through stunnel, and ensure system stability under cyber-attack conditions. These criteria align with real-world ICS security requirements and are benchmarked against current standards.

7.2 Experimental Setup

To better illustrate the structure of the virtualized environment, **Table 7.1** outlines the full list of virtual machines (VMs) used in the testbed, their purpose, and their assigned network zone. This architecture is aligned with the Purdue Model segmentation and Zero Trust principles. **Figure 7.1** shows the visual grouping of these VMs within the VirtualBox Manager.

Table 7. 1: Virtual Machine Allocation Overview.

VM Name	**Function**	**Zone**
ChemicalPlant	GRFICSv2 – Simulated plant process	OT
ScadaBR	SCADA system (unencrypted Modbus TCP)	OT
workstation	Engineering/Operator workstation	OT

plc_2	Simulated PLC (unencrypted)	OT
ScadaBR Encryption	SCADA system with TLS encryption	OT
plc_2 Encryption 2	PLC with TLS encryption	OT
SIEM-WAZUH	Wazuh SIEM for log collection and visualization	IT
ubuntu-monitoring	Monitoring node and admin interface	IT
kali	Attack platform (Metasploit, Wireshark)	IT
jumpServer	SSH gateway for controlled access with MFA	DMZ
pfsense	Firewall and router with policy enforcement	DMZ
IDS-DMZ	Suricata IDS monitoring cross-zone traffic	DMZ

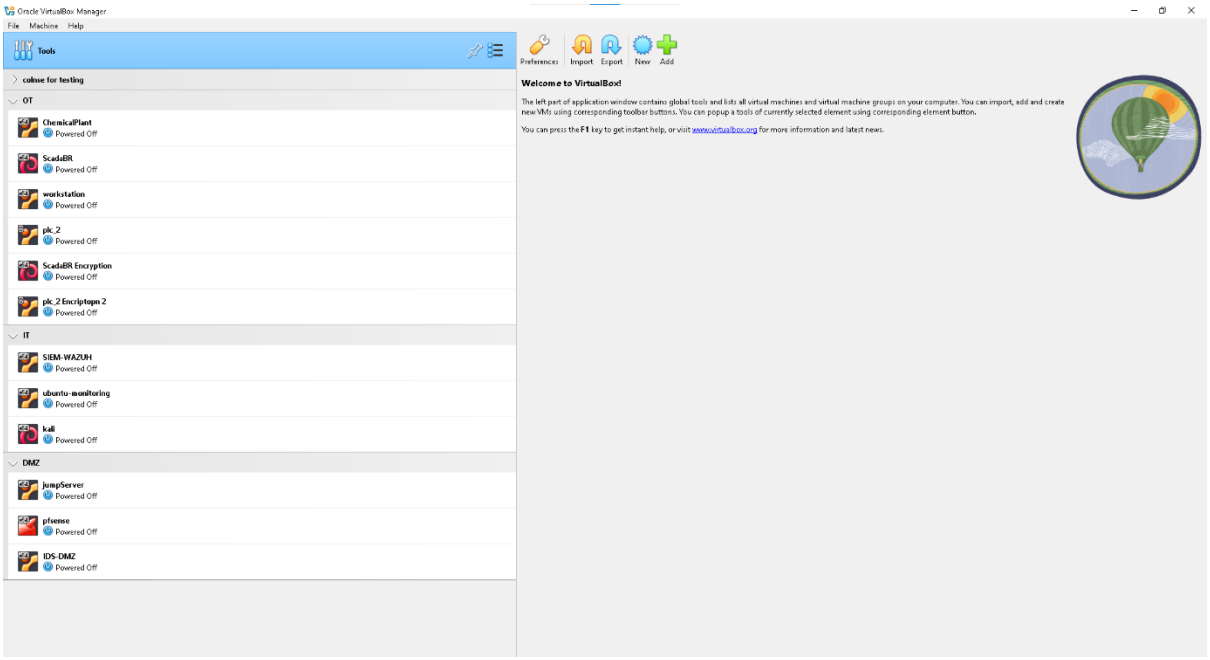


Figure 7. 1: Virtual Machine Grouping in VirtualBox.

This screenshot displays the full set of VMs used in the testbed as configured in Oracle VirtualBox. VMs are organized by their security zones (IT, DMZ, OT) to replicate the Purdue Model structure. This design supports segmentation, secure monitoring, and attack simulation across a controlled virtual infrastructure.

The experimental environment was fully virtualized using VirtualBox and included all core components of an industrial control network. The pfSense firewall enforced strict segmentation between IT, DMZ, and OT layers. GRFICSv2 was used to emulate SCADA and PLC operations, providing a realistic industrial backdrop. Wazuh served as the SIEM platform to collect logs and trigger alerts, while Suricata acted as an IDS to inspect network traffic. The Jumpserver enabled secure cross-zone access, and Metasploit was used for launching Modbus TCP-based cyberattacks. Wireshark provided packet-level visibility for protocol analysis and verification. A complete diagram of this setup is provided in **Figure 7.2**.

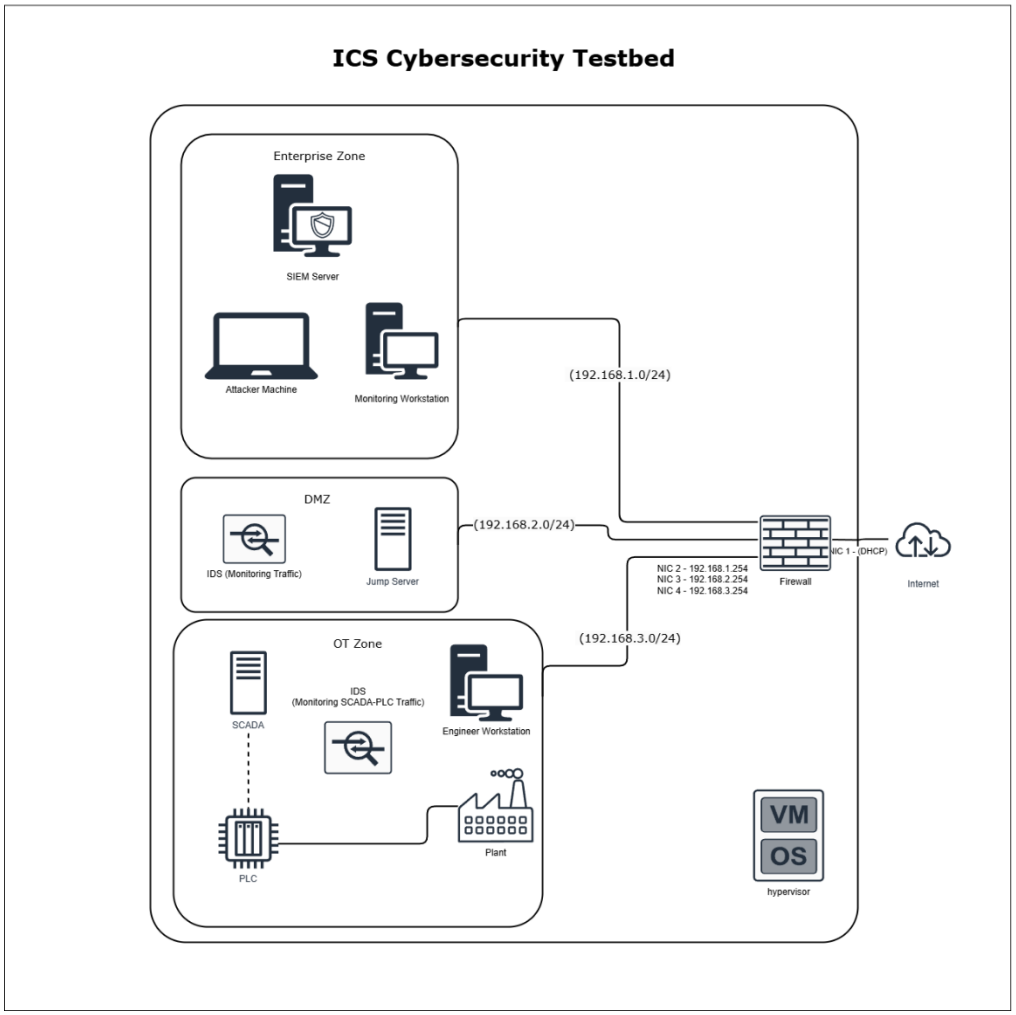


Figure 7. 2: Virtualized ICS Cybersecurity Testbed Architecture.

7.3 The Experiments

Scenario 1: Direct IT-to-OT Access Blocked (Zero Trust Validation)

To validate Zero Trust enforcement and network segmentation, a ping and SSH attempt were launched from the IT zone to the OT zone. As expected, these connection requests were blocked by pfSense firewall rules. This confirmed that the testbed effectively enforces Zero Trust principles, preventing unauthorized lateral access. The screenshot showing blocked requests is displayed in **Figure 7.3**, while detailed firewall configuration rules are presented in **Appendix B**.

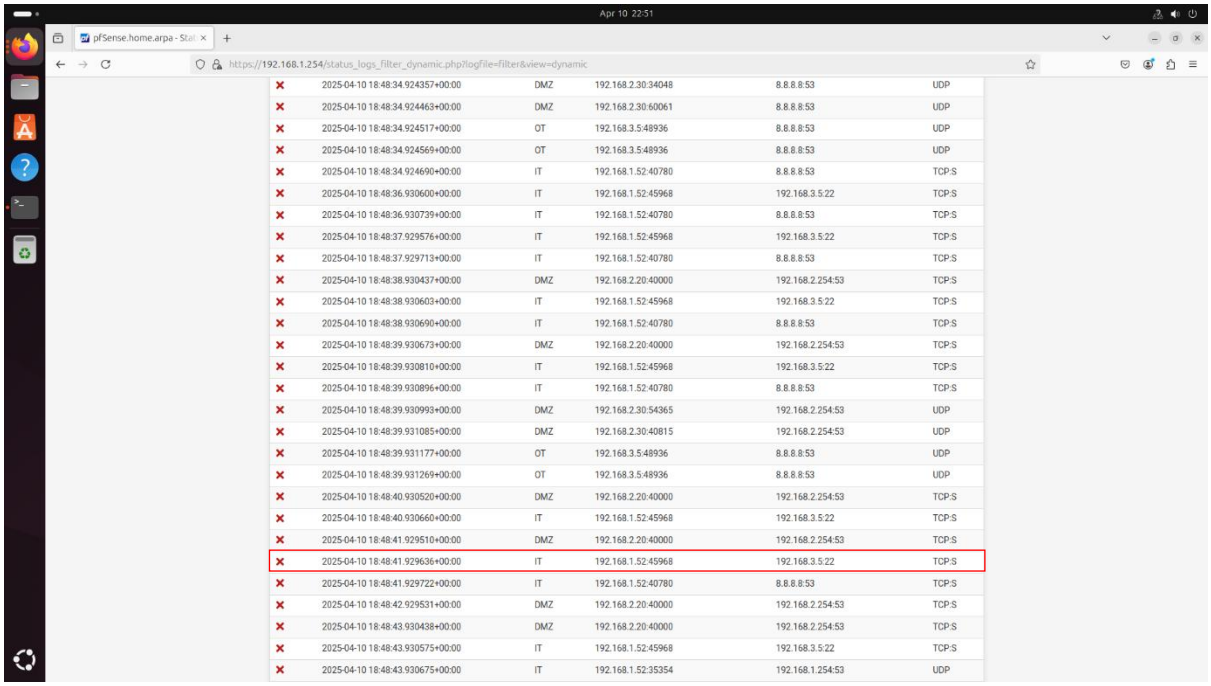


Figure 7. 3: pfSense Firewall Blocking Unauthorized IT-to-OT Access.

Scenario 2: Controlled Access via Jumpserver

This scenario demonstrated secure communication routing using a jumpserver. An SSH connection from IT to the Jumpserver in the DMZ succeeded, followed by another SSH session from the Jumpserver to the SCADA system in the OT. This confirmed that controlled access could be achieved under a Zero Trust model, with all traffic routed through a monitored gateway. Screenshots showing successful multi-hop access and Jumpserver authentication are provided in **Figure 7.4** and elaborated further in **Appendix E**.

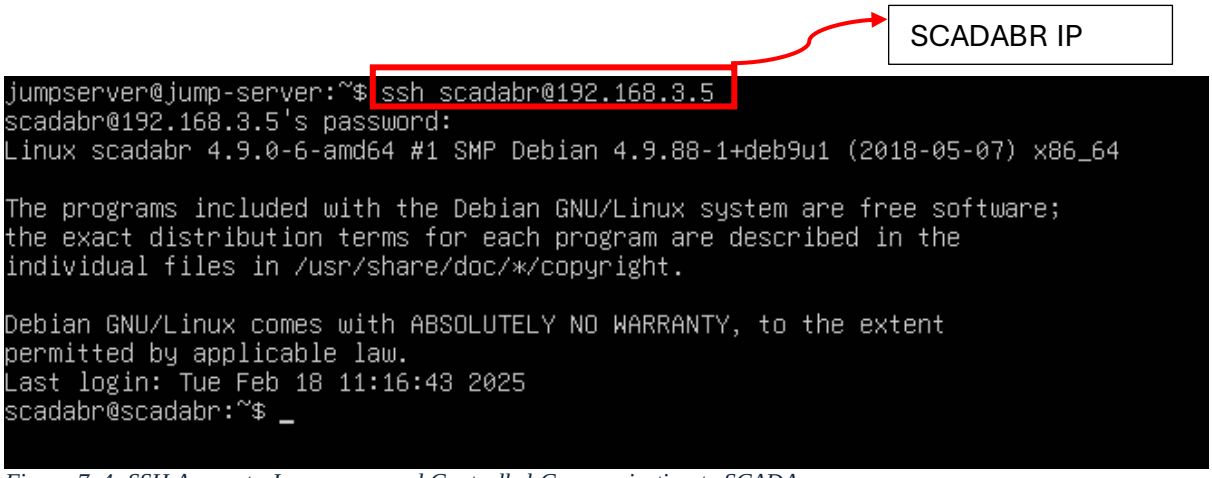


Figure 7. 4: SSH Access to Jumpserver and Controlled Communication to SCADA.

Scenario 3: Modbus TCP Traffic Capture (Unencrypted – Encryption Disabled)

Prior to applying encryption, we used Wireshark to capture and analyze Modbus TCP traffic between SCADA and PLC. The captured packets revealed critical operational commands and device states in plaintext. This validated the vulnerability of unprotected ICS protocols. The unencrypted data is clearly shown in **Figure 7.5**, with additional packet captures archived in **Appendix G**.

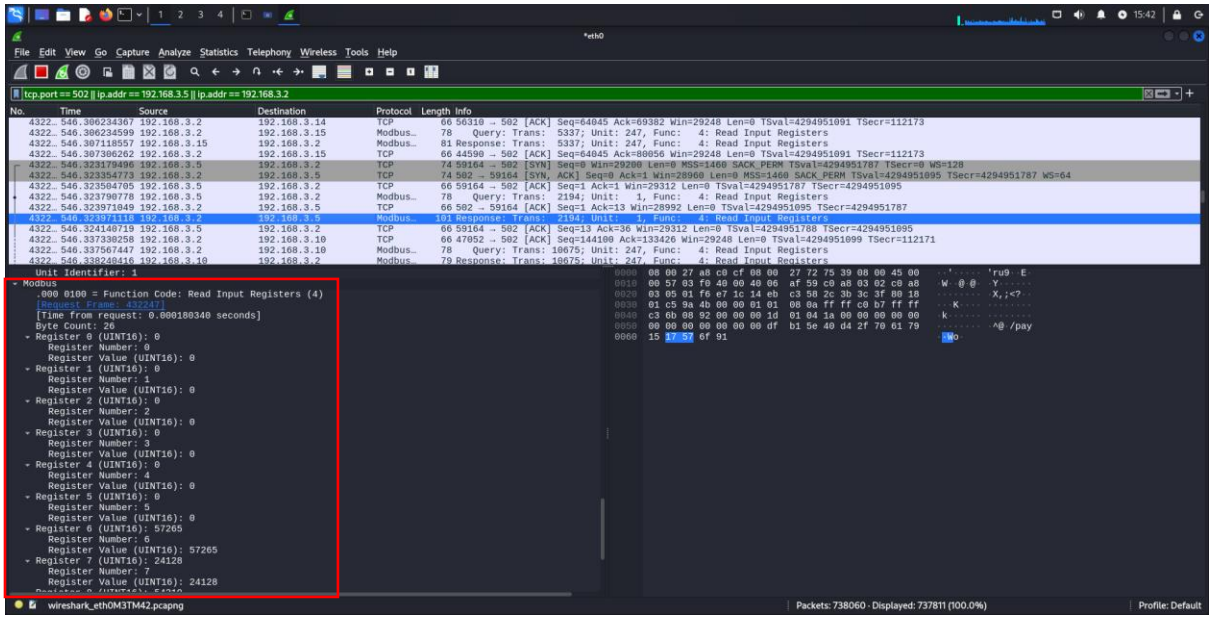


Figure 7. 5: Wireshark Capture Showing Unencrypted Modbus TCP Traffic.

Scenario 4: Modbus Attack Simulation Using Metasploit

With TLS encryption still disabled, we launched an attack using Metasploit's Modbus client module to send coil write commands to the PLC. This resulted in unauthorized state changes, illustrating the testbed's susceptibility to command injection when encryption is not in place. The attack process and effects are documented in **Figure 7.6**, with simulation steps and attack configurations listed in **Appendix G**.

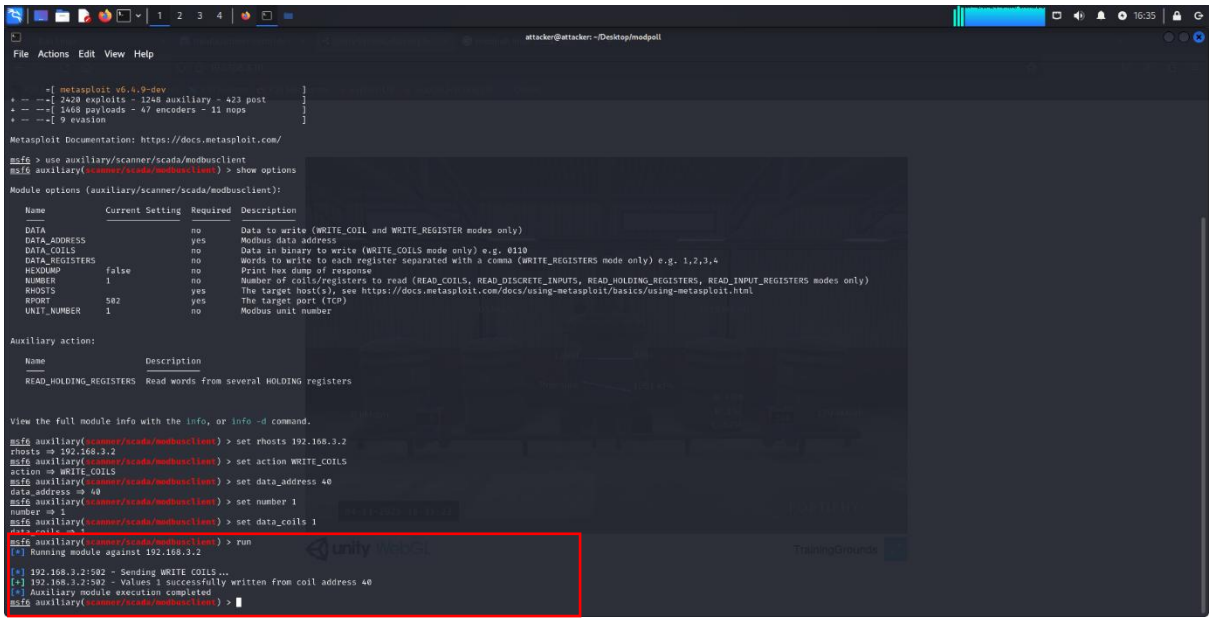


Figure 7. 6: Successful Modbus Coil Write Attack Using Metasploit.

Scenario 5: SCADA Emergency Shutdown

Following the command injection in Scenario 4, the PLC state change triggered an emergency shutdown event on the SCADA HMI. This demonstrated the real-world implications of protocol abuse, where a malicious write command could disrupt plant operations. A screenshot of the HMI shutdown alert is presented in **Figure 7.7**, showcasing the severity of a successful attack.

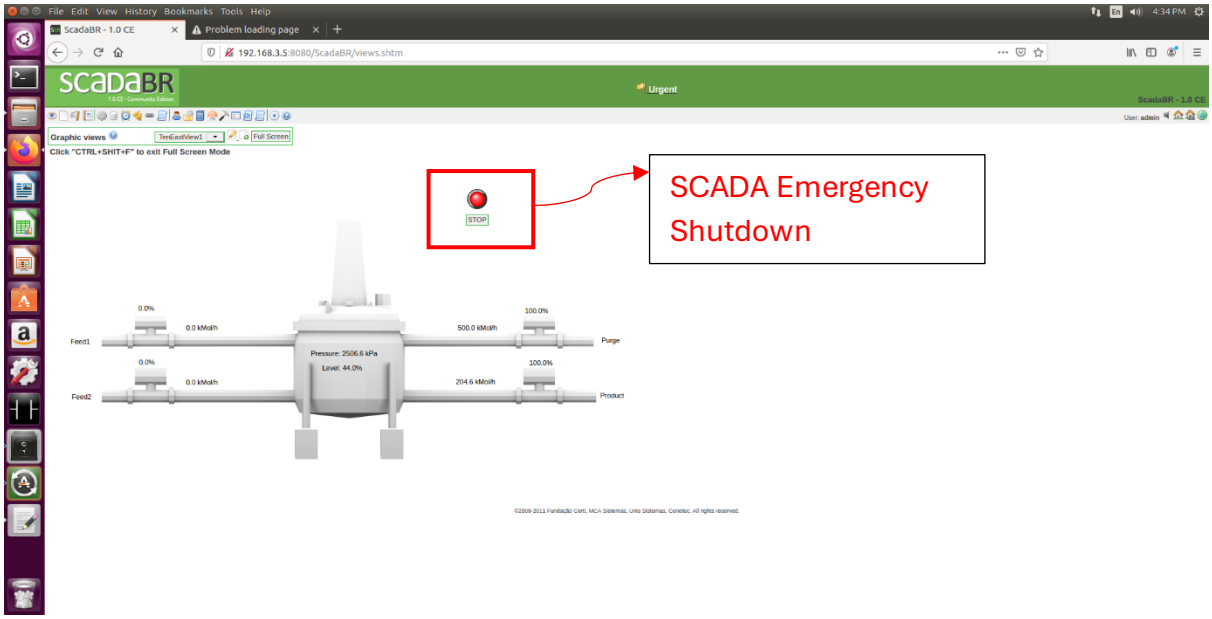


Figure 7. 7: SCADA HMI Displaying Emergency Shutdown after Attack.

Scenario 6: Encryption of Modbus Traffic with stunnel

TLS encryption was now introduced via stunnel to secure Modbus TCP communication, marking the start of the **encrypted testing phase**. After configuration, Wireshark was used again to analyze the traffic, which now showed encrypted payloads over port 1502. This encryption rendered Modbus messages unreadable to unauthorized sniffers. The encrypted capture is shown in **Figure 7.8**.

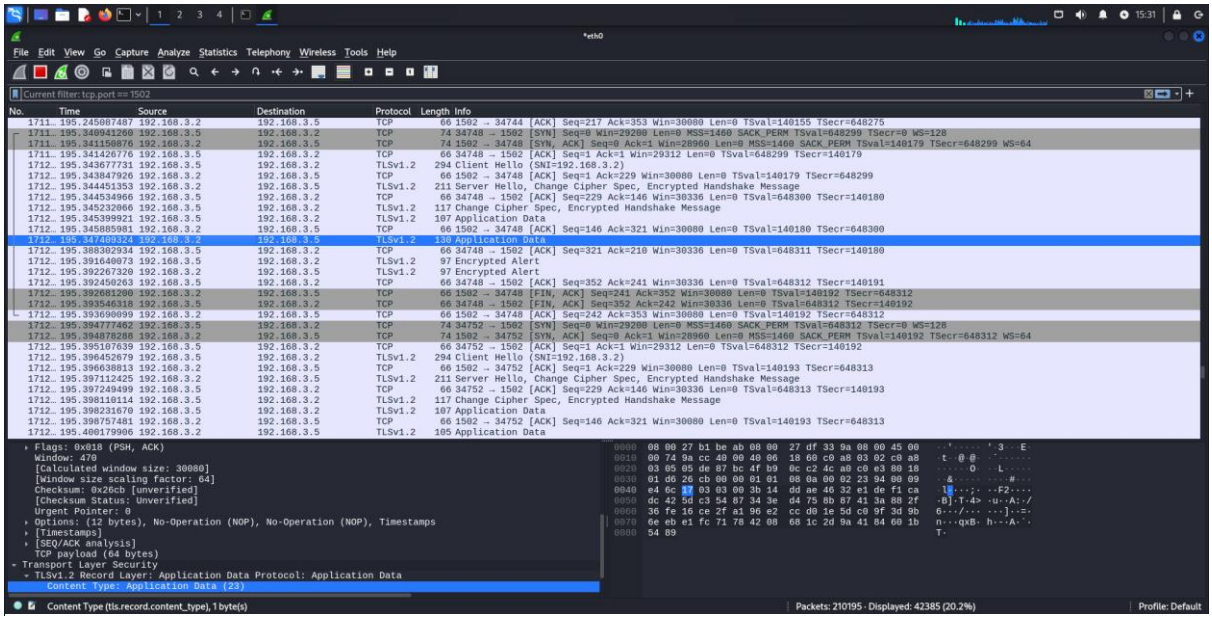


Figure 7. 8: Encrypted Modbus TCP Traffic Secured by TLS.

Scenario 7: Attack Attempt After Encryption

To test encryption resilience, another attack using Metasploit was performed **while TLS encryption was active**. Unlike the previous attempt, this time the attack failed to manipulate the PLC. Wireshark confirmed that injected commands were encapsulated and unreadable, demonstrating encryption effectiveness. The outcome is depicted in **Figure 7.9**.

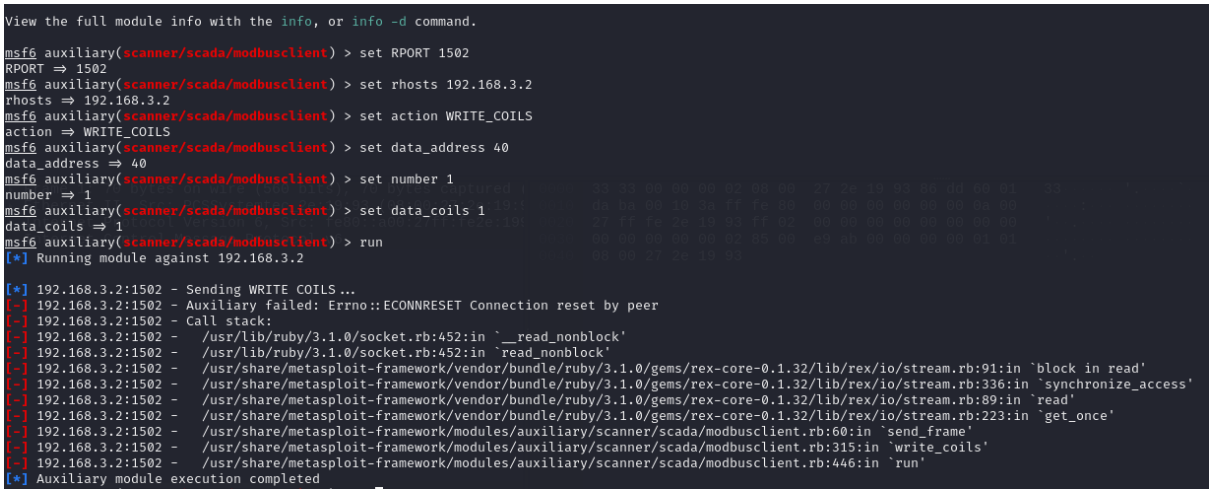


Figure 7. 9: Failed Modbus TCP Attack Attempt on TLS-Protected Communication.

Scenario 8: Detection by Wazuh and Suricata

Monitoring effectiveness was tested by observing how Wazuh and Suricata responded to the attack attempts. Suricata generated alerts based on unusual Modbus behavior, and Wazuh detected unauthorized access attempts. These tools provided real-time visibility and reinforced incident response capability. Alert dashboards are displayed in **Figures 7.10 and 7.11**.

While baseline rules detected general anomalies, more specific alerts were triggered only after applying custom Modbus TCP rules, highlighting the importance of tailored detection in ICS networks.

Figure 7. 10: Suricata Detection of Suspicious Modbus Traffic.

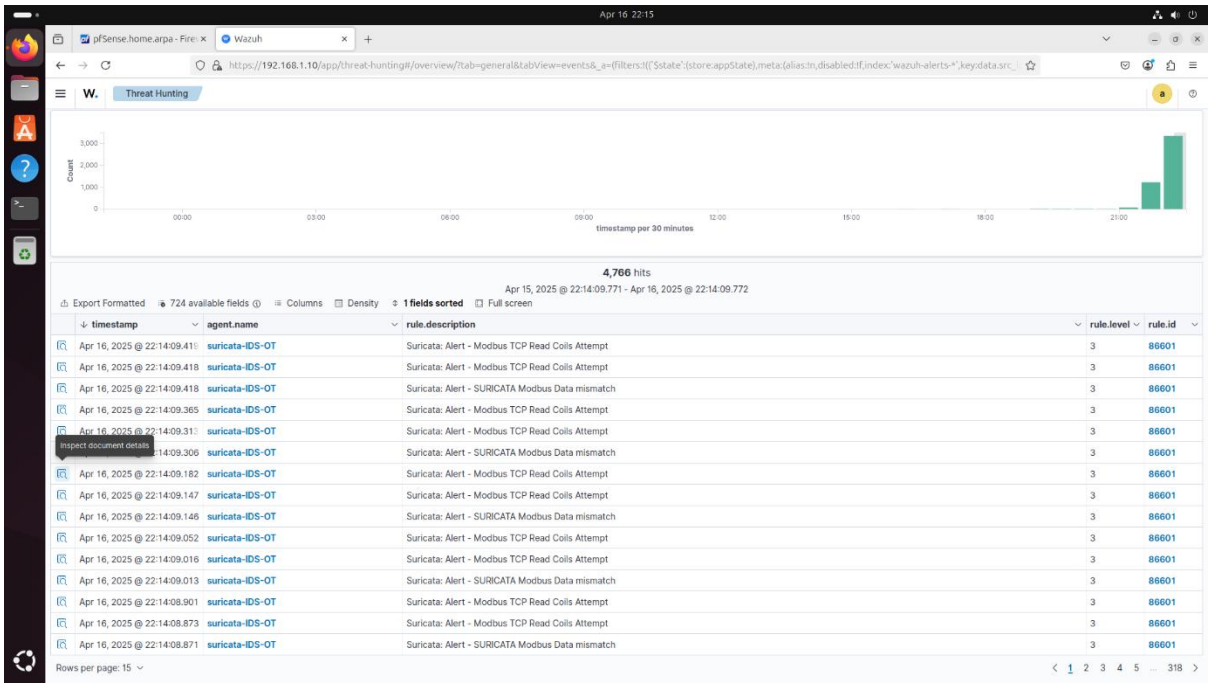


Figure 7. 11: Suricata Alerts Visualized in Wazuh Dashboard.

7.4 Results Summary

The results validated the success of our cybersecurity architecture. All unauthorized IT-to-OT access was blocked, jumpserver routing worked correctly, and encryption shielded ICS data. Attack attempts were detected and logged, and SCADA integrity was preserved post-encryption. The outcomes aligned closely with expected results and are summarized visually in **Table 7.2**.

Table 7. 2:Results Summary.

Test	Expected Outcome	Actual Result	Screenshot
Direct IT → OT Access	Blocked by firewall	Blocked	Figure 7.3
Controlled Jumpserver Access	Allowed	Allowed	Figure 7.4
Modbus Unencrypted Capture	Visible traffic	Visible	Figure 7.5
Attack Simulation (Unencrypted)	Successful	Successful	Figure 7.6
SCADA Shutdown	Emergency shutdown	Shutdown	Figure 7.7
Encryption Capture	Encrypted traffic	Encrypted	Figure 7.8
Attack Simulation (Encrypted)	Attack fails	Blocked	Figure 7.9
Wazuh/Suricata Detection	Capture anomalies	Detected	Figures 7.10, 7.11

7.5 Validity

The testbed emulated real-world industrial configurations and threats. All tests were executed in an isolated virtual environment, allowing safe, controlled simulations. Additionally, multi-factor authentication on the Jumpserver enhanced access control by preventing unauthorized

logins even in the case of credential theft. While general-purpose rulesets in Suricata and Wazuh detected baseline anomalies, advanced tuning for Modbus-specific behavior was limited and is acknowledged as a point for future enhancement.

7.6 Summary of Findings (Discussion)

The experiments confirmed that combining Zero Trust principles with encryption, segmentation, and detection tools leads to a highly resilient ICS environment. Access was strictly controlled, encrypted traffic remained secure, and all attack attempts were detected. The firewall and Jumpserver were instrumental in limiting lateral movement, and encryption neutralized protocol-level vulnerabilities. The key limitation involved the need for more advanced, ICS-specific alert rules. Nevertheless, the project demonstrates a comprehensive and scalable approach to ICS security, representing a significant effort in design, implementation, and evaluation.

CHAPTER 8: CONCLUSION

8.1 Summary

This project successfully designed, implemented, and evaluated a secure Industrial Control System (ICS) environment using a fully virtualized setup and open-source tools. Through a comprehensive approach, we developed a modular cybersecurity testbed that simulates real-world ICS operations and integrates essential security mechanisms such as network segmentation, centralized monitoring, encrypted communication, and access control enforcement.

The testbed was constructed using GRFICSv2 components to replicate SCADA systems, programmable logic controllers (PLCs), and plant simulations. A multi-zoned network topology was deployed using pfSense, segmenting the system into IT, DMZ, and OT zones. Access to the OT environment was controlled via a Jumpserver, in line with Zero Trust Architecture principles. Multi-factor authentication (MFA) was enabled on the Jumpserver using Google Authenticator to enhance identity verification, ensuring that even compromised credentials could not grant unauthorized access.

Security monitoring was implemented using Wazuh SIEM and Suricata IDS. These tools enabled real-time detection and analysis of network traffic. The environment was further hardened through the configuration of stunnel to enable TLS encryption for Modbus TCP communications. TLS encryption was successfully applied using stunnel during the final attack scenarios. Unauthorized Modbus TCP commands were blocked, and encrypted traffic was confirmed via Wireshark analysis, validating the effectiveness of the secure communication layer.

Controlled attack simulations using the Metasploit framework revealed vulnerabilities in unsecured protocols and confirmed the effectiveness of segmentation and access control in mitigating threats.

Overall, the project met its goals of building a safe, functional ICS cybersecurity testbed that demonstrates how open-source technologies and Zero Trust principles can secure critical infrastructure systems. The testbed provides a valuable platform for future research, learning, and professional development.

8.2 Limitations

While the project accomplished its core objectives, several limitations were observed during development and testing.

First, the use of a virtualized environment, while beneficial for safety and cost, limited the realism of operational scenarios. Real-world industrial hardware behaviors such as device latency, timing dependencies, or mechanical failures could not be simulated.

Second, although Suricata and Wazuh successfully monitored general anomalies, but default rulesets lacked the specificity needed to detect fine-grained Modbus TCP attacks. Custom rules were added for better detection, but full coverage of all Modbus scenarios was limited. This highlighted the need for custom detection rules tailored to industrial protocols.

Additionally, encryption was not evaluated across all attack types or extended to multiple layers of the testbed due to time and system constraints. Although TLS was successfully demonstrated, future work should explore broader encrypted communication coverage and certificate management.

Attack simulations were also limited to Modbus command injections using Metasploit, without simulating more advanced threats like lateral movement, malware propagation, or insider compromise.

Lastly, the limited computational resources available for virtualization restricted the number of concurrent simulations and devices that could be deployed.

Despite these constraints, the system still achieved meaningful results and demonstrated critical concepts in ICS security.

8.3 Future Work

The current project lays the groundwork for future improvements and enhancements to the ICS cybersecurity testbed.

One important area is the integration of custom Suricata rules specifically designed for industrial protocols like Modbus TCP. This would significantly increase the detection accuracy for protocol-specific threats. Additionally, full deployment and evaluation of TLS encryption

using stunnel across all ICS communication channels would further improve data confidentiality and protect against packet injection or interception attacks.

The inclusion of more sophisticated attack scenarios, such as Man-in-the-Middle (MITM), ransomware, and privilege escalation simulations, Man-in-the-Middle (MITM) attacks were originally planned for this project but postponed due to technical constraints. Future implementations should include MITM testing using tools like Bettercap or Ettercap to evaluate encryption resilience. Transitioning from a virtualized to a hybrid testbed that includes real industrial hardware such as physical PLCs and Human-Machine Interfaces (HMIs) would also provide deeper insight into practical deployment challenges and vulnerabilities.

Moreover, activating Intrusion Prevention System (IPS) features in Suricata would allow the system not only to detect threats but also to take proactive mitigation actions. Finally, integrating external threat intelligence feeds and automated alert-based response mechanisms would enhance the testbed's realism and relevance to production ICS deployments.

8.4 Impact of the Solution on Individuals, Organizations, and Society

The design and deployment of this ICS cybersecurity testbed offer significant contributions at the individual, organizational, and societal levels.

At an individual level, this project provides students, researchers, and early-career professionals with an experiential platform to learn about ICS architectures and cybersecurity practices. It bridges the gap between theoretical learning and real-world application, offering hands-on experience with technologies such as pfSense, Wazuh, Suricata, and stunnel. The inclusion of real-time attack simulations and the use of multi-factor authentication further enrich the training experience, enabling learners to interact with realistic security controls in a risk-free setting. These skills are essential for careers in OT cybersecurity and critical infrastructure protection.

At the organizational level, the testbed demonstrates how implementing layered security controls such as network segmentation, access management, and centralized monitoring can significantly reduce exposure to cyber threats. It also shows how Zero Trust principles can be realistically applied in ICS environments, enhancing resilience while maintaining operational continuity.

On a broader societal scale, securing ICS environments contributes to the protection of critical infrastructure sectors such as energy, water, manufacturing, and transportation. The research supports public safety, economic stability, and national security by exploring ways to reduce the risk of cyber-physical attacks. The testbed itself can be reused or expanded for professional training, academic curricula, and the validation of next-generation security technologies.

In conclusion, this project not only fulfilled its educational and technical objectives but also delivered meaningful impact and insight into the future of industrial cybersecurity.

REFERENCES

[1] S. Maesschalck, "Next-Generation Industrial Control System (ICS) Security Towards ICS Honeypots for Defence-in-Depth Security," 2023.

[2] "STUXNET AND STRATEGY A Special Operation in Cyberspace?," *JFQ / issue*, vol. 63, 2011.

[3] H. Ramachandran, R. Smith, K. A. David, T. Al-Hadhrami, and P. Acharya, "Towards Net Zero Resilience: A Futuristic Architectural Strategy for Cyber-Attack Defence in Industrial Control Systems (ICS) and Operational Technology (OT)," *Computers, Materials & Continua*, vol. 82, no. 2, pp. 3619–3641, Feb. 2025, doi: 10.32604/CMC.2024.054802.

[4] M. Nankya, R. Chataut, and R. Akl, "Securing Industrial Control Systems: Components, Cyber Threats, and Machine Learning-Driven Defense Strategies," *Sensors 2023, Vol. 23, Page 8840*, vol. 23, no. 21, p. 8840, Oct. 2023, doi: 10.3390/S23218840.

[5] M. Pallavi, P. Kumar, T. Ali, and S. B. Shenoy, "Modeling of a Negative Refractive Index Metamaterial Unit-Cell and Array for Aircraft Surveillance Applications," *IEEE Access*, vol. 10, pp. 99790–99812, 2022, doi: 10.1109/ACCESS.2022.3206358.

[6] C. Ekisa, D. Briain, Y. K.-2021 32nd I. S. and, and undefined 2021, "An open-source testbed to visualise ics cybersecurity weaknesses and remediation strategies–a research agenda proposal," *ieeexplore.ieee.org*, Accessed: Mar. 08, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9467852/?casa_token=-6P38zvDw4AAAAAA:qJimK-PoFRz7li78q5aGaB5icn2heQ63ii47cZk4DMYNm1x36_4XPuy3XwmCtKhNTeyHgGVWulx_

[7] N. Gholizadeh, "Iec 61850 standard and its capabilities in protection systems," 2015, Accessed: Mar. 08, 2025. [Online]. Available: https://www.politesi.polimi.it/bitstream/10589/123532/1/thesis.pdf

[8] M. Conti, D. Donadel, & F. T.-I. C. S., and undefined 2021, "A survey on industrial control system testbeds and datasets for security research," *ieeexplore.ieee.org*, Accessed: Mar. 08, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9471765/?casa_token=LKdafTRHFQEAAAAA:8NNThXB_04Fl6E7gDjy2lam7ooA8n2rL3rkZIzZERqpivgUaK33G4040Tz_udfclM3PmAr4zGe0D

[9] C. Zanasi, S. Russo, M. C.-A. H. Networks, and undefined 2024, "Flexible zero trust architecture for the cybersecurity of industrial IoT infrastructures," *Elsevier*, Accessed: Mar. 08, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870524000258

[10] A. Elmasry, “Developing and Evaluating Simulation Testbeds for IEC61850 GOOSE Protocol: Enhancing Cybersecurity in Substations and Smart Grids,” 2024, Accessed: Mar. 08, 2025. [Online]. Available: https://search.proquest.com/openview/0bd320fa8e93c8f2e1dd4b74561cb649/1?pq-origsite=gscholar&cbl=2026366&diss=y&casa_token=VNitOoATFRcAAAAA:LntlMS97aBGppWElp_g8YbXLET4hc7N6LDqioHuHUmbpXxqAAaPZNuFW-zHHUMIubBHn2WUQsIZf

[11] S. Russo, I. Giorgio, and V. Santangelo, “Industrial Demilitarized Zone and Zero Trust cybersecurity models for Industrial Control Systems,” *amslaurea.unibo.it*, Accessed: Mar. 08, 2025. [Online]. Available: https://amslaurea.unibo.it/id/eprint/26737/

[12] G. Ahmadi-Assalemi and S. Ahmadi-Assalemi……, “Anomalous behaviour detection for cyber defence in modern industrial control systems,” 2022, Accessed: Mar. 08, 2025. [Online]. Available: https://wlv.openrepository.com/handle/2436/625131

[13] K. Stouffer *et al.*, “NIST Special Publication NIST SP 800-82r3 Guide to Operational Technology (OT) Security”, doi: 10.6028/NIST.SP.800-82r3.

[14] F. Katulic, D. Sumina, S. Gros, and I. Erceg, “Protecting Modbus/TCP-Based Industrial Automation and Control Systems Using Message Authentication Codes,” *IEEE Access*, vol. 11, pp. 47007–47023, 2023, doi: 10.1109/ACCESS.2023.3275443.

[15] “Matrix - ICS | MITRE ATT&CK®.” Accessed: Mar. 06, 2025. [Online]. Available: https://attack.mitre.org/matrices/ics/

[16] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, “NIST Special Publication 800-207 Zero Trust Architecture”, doi: 10.6028/NIST.SP.800-207.

[17] “Secure Remote Access for Industrial Networks Design Guide THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS DESCRIBED IN THIS DOCUMENT ARE SUBJECT TOCHANGE WITHOUT NOTICE. THIS DOCUMENT IS PROVIDED ",” 2023, Accessed: Mar. 01, 2025. [Online]. Available: www.cisco.com/go/offices.

[18] P. Radoglou-Grammatikis, I. Siniosoglou, T. Liatifis, A. Kourouniadis, K. Rompolos, and P. Sarigiannidis, “Implementation and Detection of Modbus Cyberattacks”, doi: 10.1109/MOCAST49295.2020.9200287.

[19] C. Zanasi, F. Magnanini, S. Russo, and M. Colajanni, “A Zero Trust approach for the cybersecurity of Industrial Control Systems,” *NCA 2022 - 2022 IEEE 21st International Symposium on Network Computing and Applications*, pp. 1–7, 2022, doi: 10.1109/NCA57778.2022.10013559.

[20] X. Feng and S. Hu, “Cyber-Physical Zero Trust Architecture for Industrial Cyber-Physical Systems,” *IEEE Transactions on Industrial Cyber-Physical Systems*, vol. 1, pp. 394–405, Nov. 2023, doi: 10.1109/TICPS.2023.3333850.

[21] L. S. Cruz and I. E. Fonseca, “Industrial Control Systems in Environments with Zero Trust Architecture: Analysis of Responses to Various Attack Types,” *2023 Workshop on Communication Networks and Power Systems, WCNPS 2023*, 2023, doi: 10.1109/WCNPS60622.2023.10344788.

[22] E. Papadogiannaki, G. Tsirantonakis, and S. Ioannidis, “Network Intrusion Detection in Encrypted Traffic,” *5th IEEE Conference on Dependable and Secure Computing, DSC 2022 and SECSOC 2022 Workshop, PASS4IoT 2022 Workshop SICSA International Paper/Poster Competition in Cybersecurity*, 2022, doi: 10.1109/DSC54232.2022.9888942.

[23] T. Martins and S. V. G. Oliveira, “Enhanced Modbus/TCP Security Protocol: Authentication and Authorization Functions Supported,” *Sensors 2022, Vol. 22, Page 8024*, vol. 22, no. 20, p. 8024, Oct. 2022, doi: 10.3390/S22208024.

[24] I. A. Oyewumi *et al.*, “ISAAC: The Idaho CPS smart grid cybersecurity testbed,” *2019 IEEE Texas Power and Energy Conference, TPEC 2019*, Mar. 2019, doi: 10.1109/TPEC.2019.8662189.

[25] G. B. Gaggero, A. Armellin, G. Portomauro, and M. Marchese, “Industrial Control System-Anomaly Detection Dataset (ICS-ADD) for Cyber-Physical Security Monitoring in Smart Industry Environments,” *IEEE Access*, vol. 12, pp. 64140–64149, 2024, doi: 10.1109/ACCESS.2024.3395991.

[26] A. P. Mathur and N. O. Tippenhauer, “SWaT: A water treatment testbed for research and training on ICS security,” *2016 International Workshop on Cyber-physical Systems for Smart Water Networks, CySWater 2016*, pp. 31–36, May 2016, doi: 10.1109/CYSWATER.2016.7469060.

[27] C. M. Ahmed, V. R. Palleti, and A. P. Mathur, “WADI: A water distribution testbed for research in the design of secure cyber physical systems,” *Proceedings - 2017 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks, CySWATER 2017*, pp. 25–28, Apr. 2017, doi: 10.1145/3055366.3055375.

[28] C. Ekisa, D. O. Briain, and Y. Kavanagh, “VICSORT-A Virtualised ICS Open-source Research Testbed,” *2022 Cyber Research Conference - Ireland, Cyber-RCI 2022*, 2022, doi: 10.1109/CYBER-RCI55324.2022.10032670.

[29] K. Stouffer *et al.*, “Guide to Industrial Control Systems (ICS) Security,” Jun. 2015, doi: 10.6028/NIST.SP.800-82R2.

[30] Y. Li, S. Wu, and Q. Pan, “Network Security in the Industrial Control System: A Survey”, Accessed: Mar. 08, 2025. [Online]. Available: http://en.Wikipedia.org/wiki/High-Level

[31] T. Sasaki, A. Fujita, C. H. Ganan, M. Van Eeten, K. Yoshioka, and T. Matsumoto, “Exposed Infrastructures: Discovery, Attacks and Remediation of Insecure ICS

Remote Management Devices," *Proc IEEE Symp Secur Priv*, vol. 2022-May, pp. 2379–2396, 2022, doi: 10.1109/SP46214.2022.9833730.

[32] Y. Hu, A. Yang, H. Li, Y. Sun, and L. Sun, "A survey of intrusion detection on industrial control systems," *Int J Distrib Sens Netw*, vol. 14, no. 8, Aug. 2018, doi: 10.1177/1550147718794615.

[33] D. Fauri, B. De Wijs, J. Den Hartog, E. Costante, E. Zambon, and S. Etalle, "Encryption in ICS networks: A blessing or a curse?," *2017 IEEE International Conference on Smart Grid Communications, SmartGridComm 2017*, vol. 2018-January, pp. 289–294, Jul. 2017, doi: 10.1109/SMARTGRIDCOMM.2017.8340732.

APPENDICES

Appendix A: Project Plan and Timeline

Table A. 1: Initial Project Plan Timeline.

Phase	Key Activities	Duration	Assigned Team Member(s)
Phase 1: Research & Literature Review	**Study ICS security frameworks, Zero Trust, encryption techniques, and security gaps.**	**5 weeks (Feb 15 – Mar 21)**	**All Members**
Phase 2: System Architecture & Security Design	**Develop testbed architecture, define security controls, and document requirements.**	**2 weeks (Mar 22 – Apr 5)**	**Member 1**
Phase 3: Implementation	**Configure firewall (pfSense), IDS (Suricata), SIEM (Wazuh), and encrypted communication (stunnel).**	**3 weeks (Apr 6 – Apr 21)**	**Member 2, Member 3, Member 4**
Phase 4: Security Testing & Evaluation	**Conduct cyberattacks simulations (MITM, Replay, Unauthorized Access) and measure performance.**	**1 week (Apr 26 – May 3)**	**Member 2, Member 4**
Phase 5: Documentation & Presentation	**Finalize the report, incorporate feedback, and prepare the presentation.**	**3 weeks (May 4 – May 22)**	**All Members**

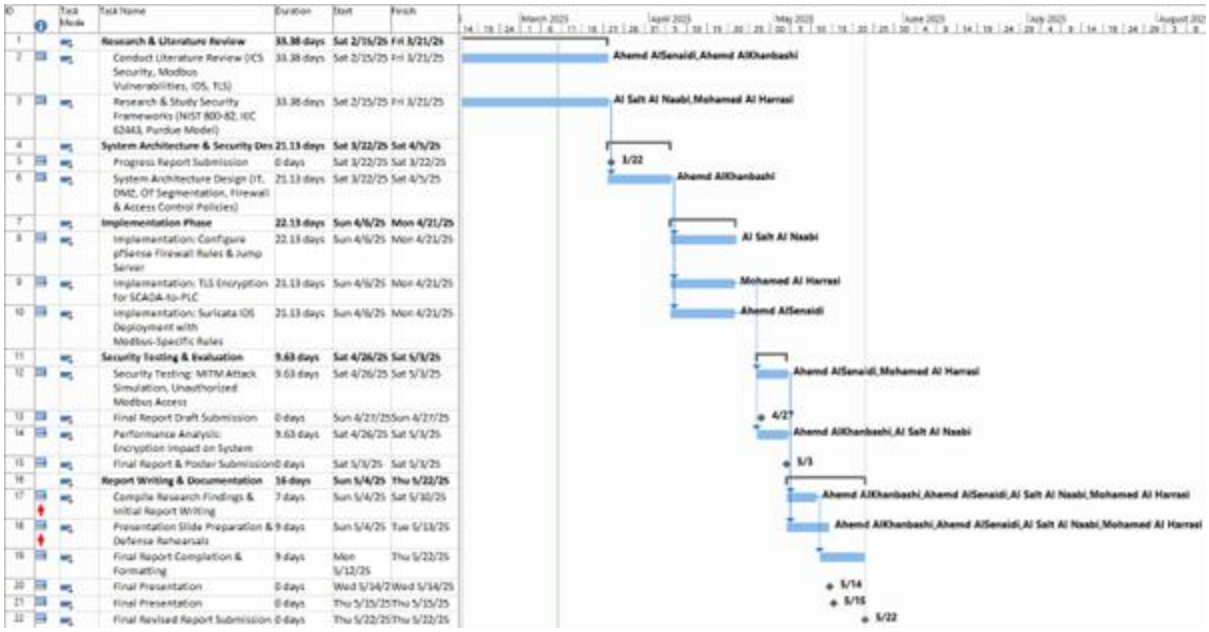


Figure A. 1: Gantt Chart for Project Phases.

Appendix B: Firewall Configuration Screenshots

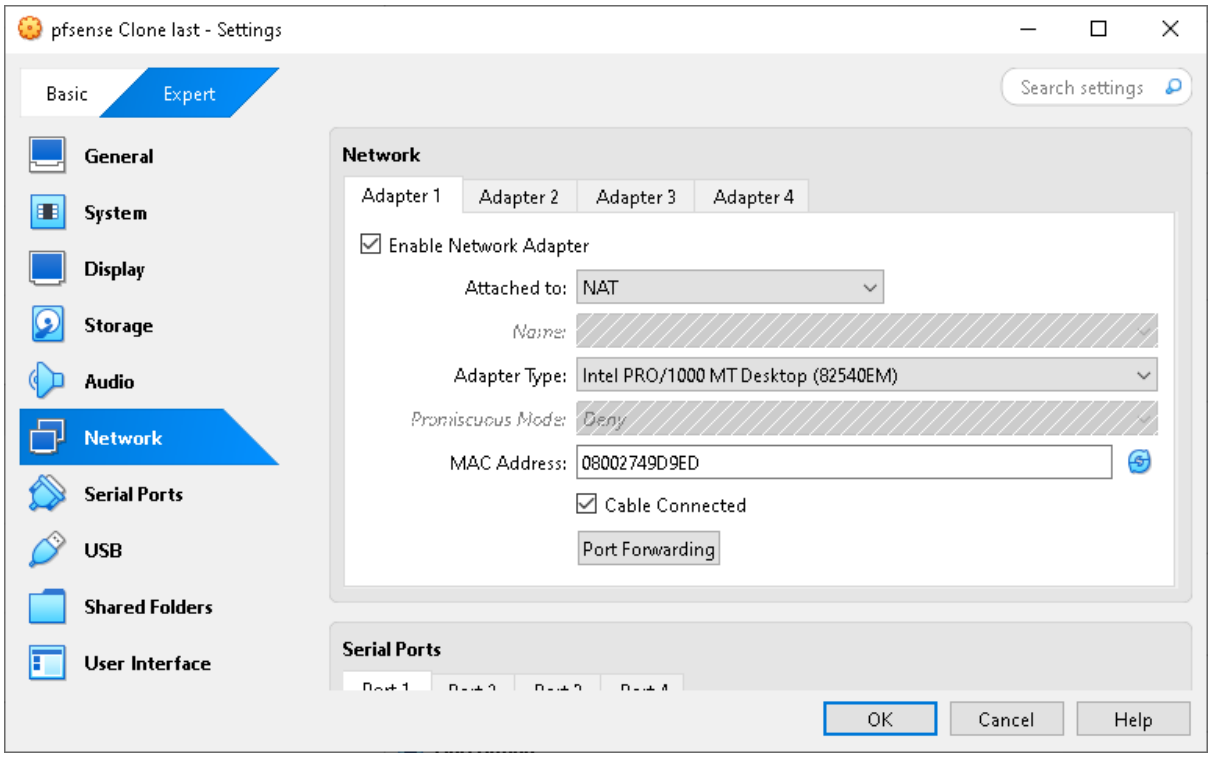


Figure B. 1: Adapter 1 – Connected to NAT network for outbound internet connectivity.

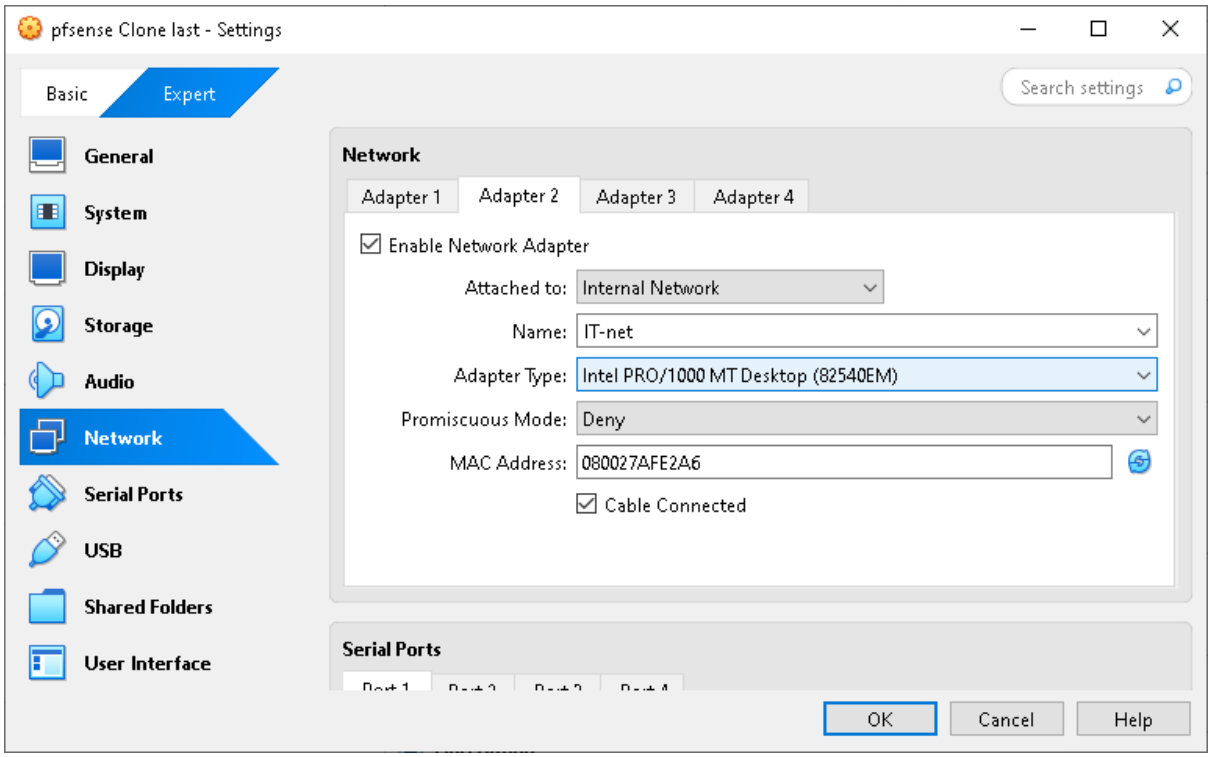


Figure B. 2: Adapter 2 – Connected to IT-net internal network (Enterprise Zone).

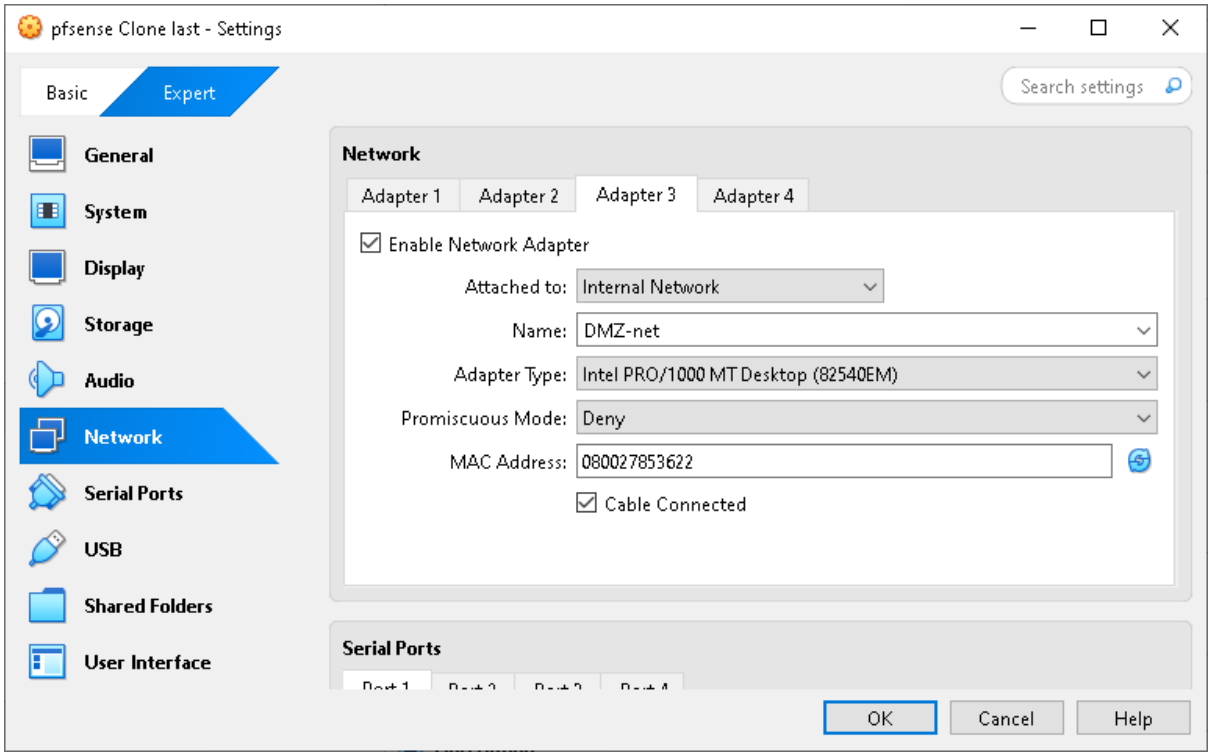


Figure B. 3: Adapter 3 – Connected to DMZ-net internal network (Demilitarized Zone).

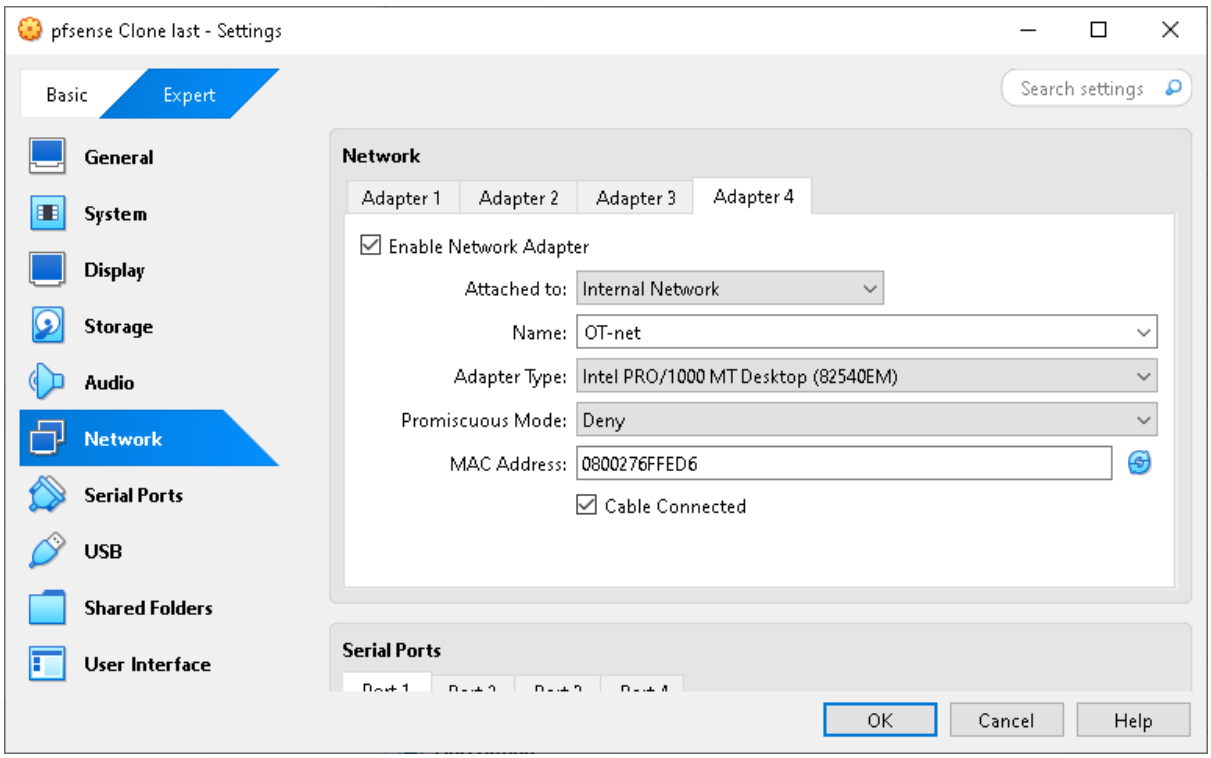


Figure B. 4: Adapter 4 – Connected to OT-net internal network (Operational Technology).

Each adapter is configured with “Intel PRO/1000 MT” and “Cable Connected” enabled, ensuring reliable virtual connectivity between the testbed zones.

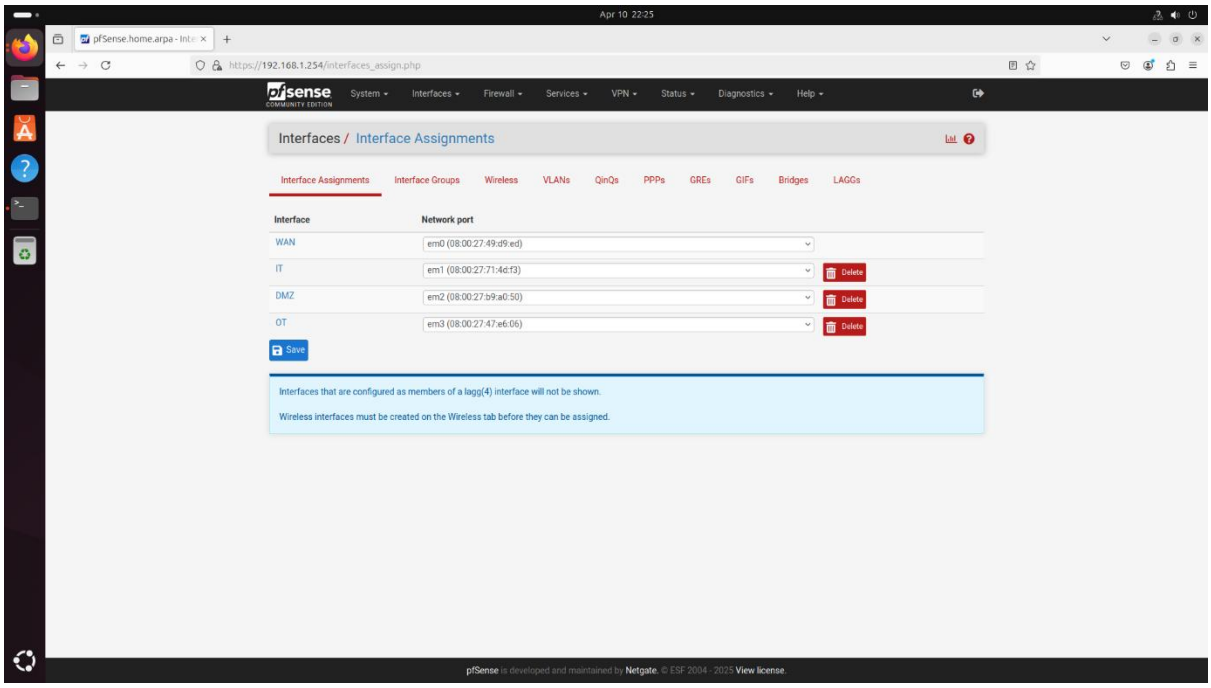
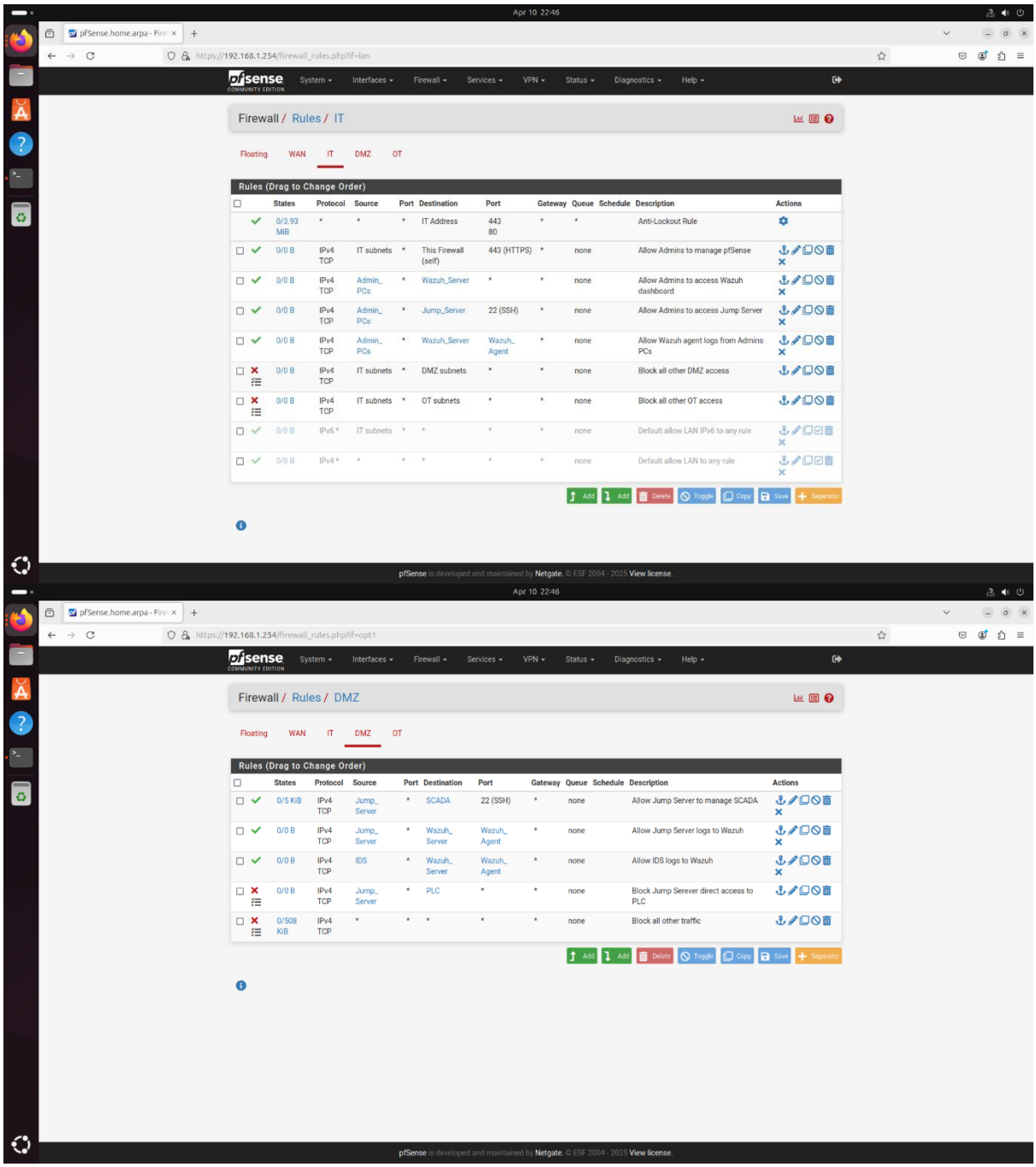


Figure B. 5: pfSense Interface Assignments (WAN, LAN1 - IT, LAN2 - DMZ, LAN3 - OT).


Apr 10 22:46
pfSense.home.arpa - Firewall: Rules: IT
https://192.168.1.254/firewall_rules.php?if=lan
pfsense COMMUNITY EDITION
System Interfaces Firewall Services VPN Status Diagnostics Help
Firewall / Rules / IT
Floating WAN IT DMZ OT
Rules (Drag to Change Order)
States Protocol Source Port Destination Port Gateway Queue Schedule Description Actions
0/3.93 MiB * * * IT Address 443 80 * * Anti-Lockout Rule
0/0 B IPv4 TCP IT subnets * This Firewall (self) 443 (HTTPS) * none Allow Admins to manage pfSense
0/0 B IPv4 TCP Admin_PCs * Wazuh_Server * * none Allow Admins to access Wazuh dashboard
0/0 B IPv4 TCP Admin_PCs * Jump_Server 22 (SSH) * none Allow Admins to access Jump Server
0/0 B IPv4 TCP Admin_PCs * Wazuh_Server Wazuh_Agent * none Allow Wazuh agent logs from Admins PCs
0/0 B IPv4 TCP IT subnets * DMZ subnets * * none Block all other DMZ access
0/0 B IPv4 TCP IT subnets * OT subnets * * none Block all other OT access
0/0 B IPv6 * IT subnets * * * * none Default allow LAN IPv6 to any rule
0/0 B IPv4 * * * * * * none Default allow LAN to any rule
Add Add Delete Toggle Copy Save Separator
pfSense is developed and maintained by Netgate. © ESF 2004 - 2025 View license.
Apr 10 22:46
pfSense.home.arpa - Firewall: Rules: DMZ
https://192.168.1.254/firewall_rules.php?if=opt1
pfsense COMMUNITY EDITION
System Interfaces Firewall Services VPN Status Diagnostics Help
Firewall / Rules / DMZ
Floating WAN IT DMZ OT
Rules (Drag to Change Order)
States Protocol Source Port Destination Port Gateway Queue Schedule Description Actions
0/5 KiB IPv4 TCP Jump_Server * SCADA 22 (SSH) * none Allow Jump Server to manage SCADA
0/0 B IPv4 TCP Jump_Server * Wazuh_Server Wazuh_Agent * none Allow Jump Server logs to Wazuh
0/0 B IPv4 TCP IDS * Wazuh_Server Wazuh_Agent * none Allow IDS logs to Wazuh
0/0 B IPv4 TCP Jump_Server * PLC * * none Block Jump Serever direct access to PLC
0/508 KiB IPv4 TCP * * * * * * none Block all other traffic
Add Add Delete Toggle Copy Save Separator
pfSense is developed and maintained by Netgate. © ESF 2004 - 2025 View license.

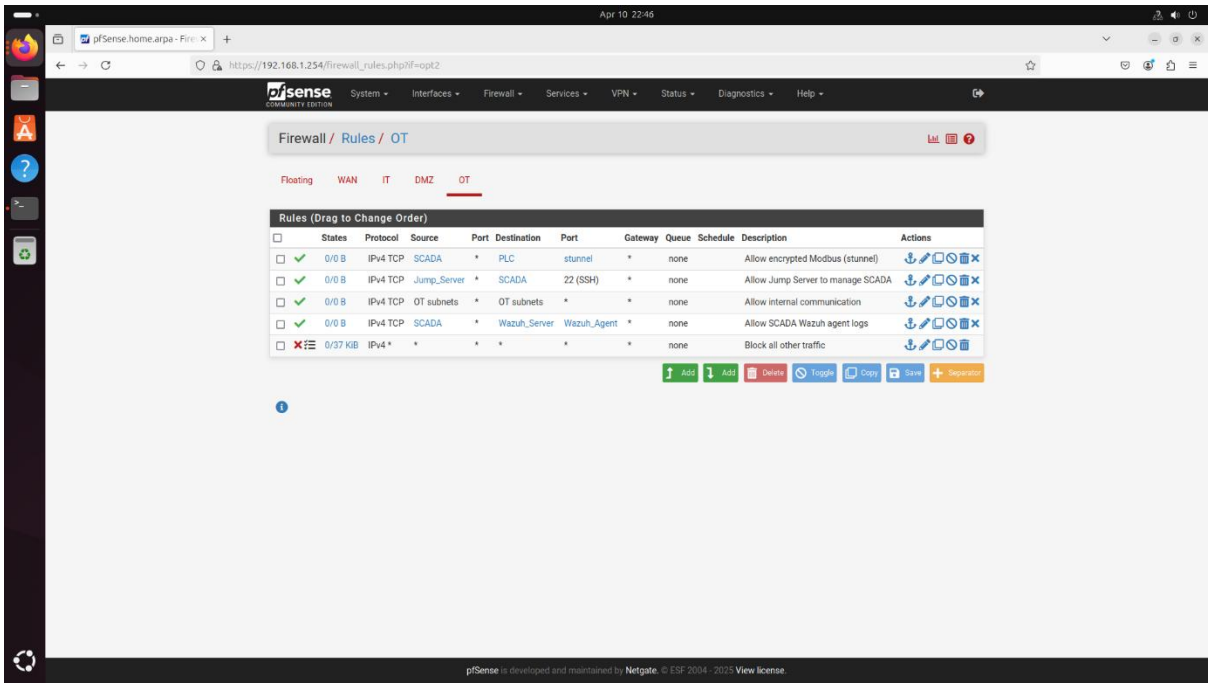


Figure B. 6: pfSense Firewall Rules Applied to IT, DMZ, and OT Interfaces.

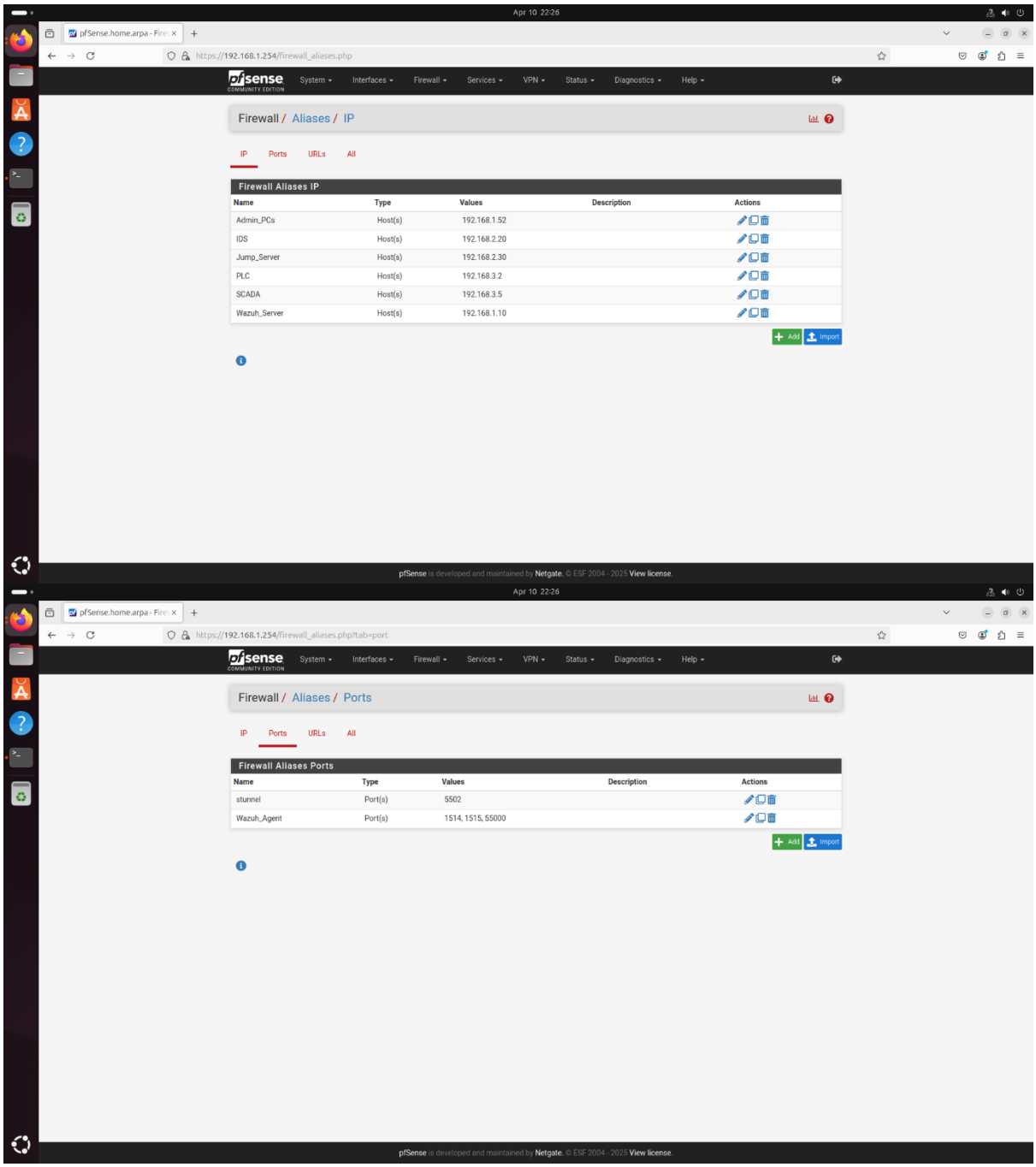


Figure B. 7: pfSense Alias Definitions for Critical Assets (SCADA, PLC, Jumpserver, Wazuh).

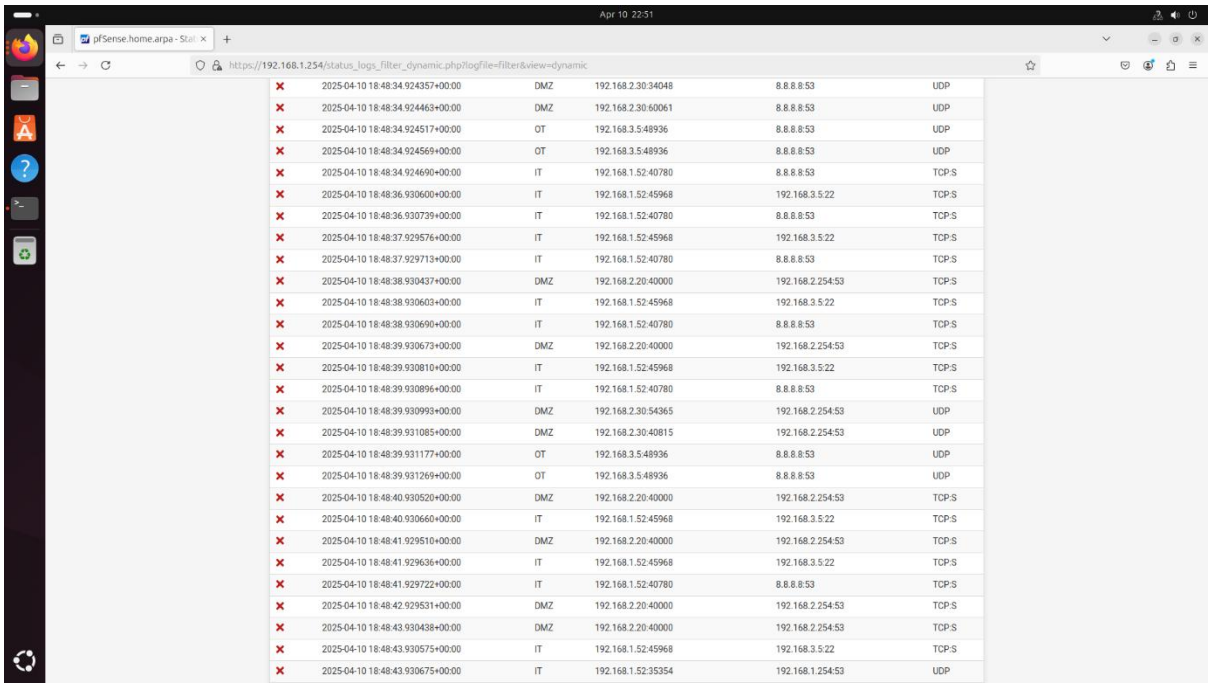
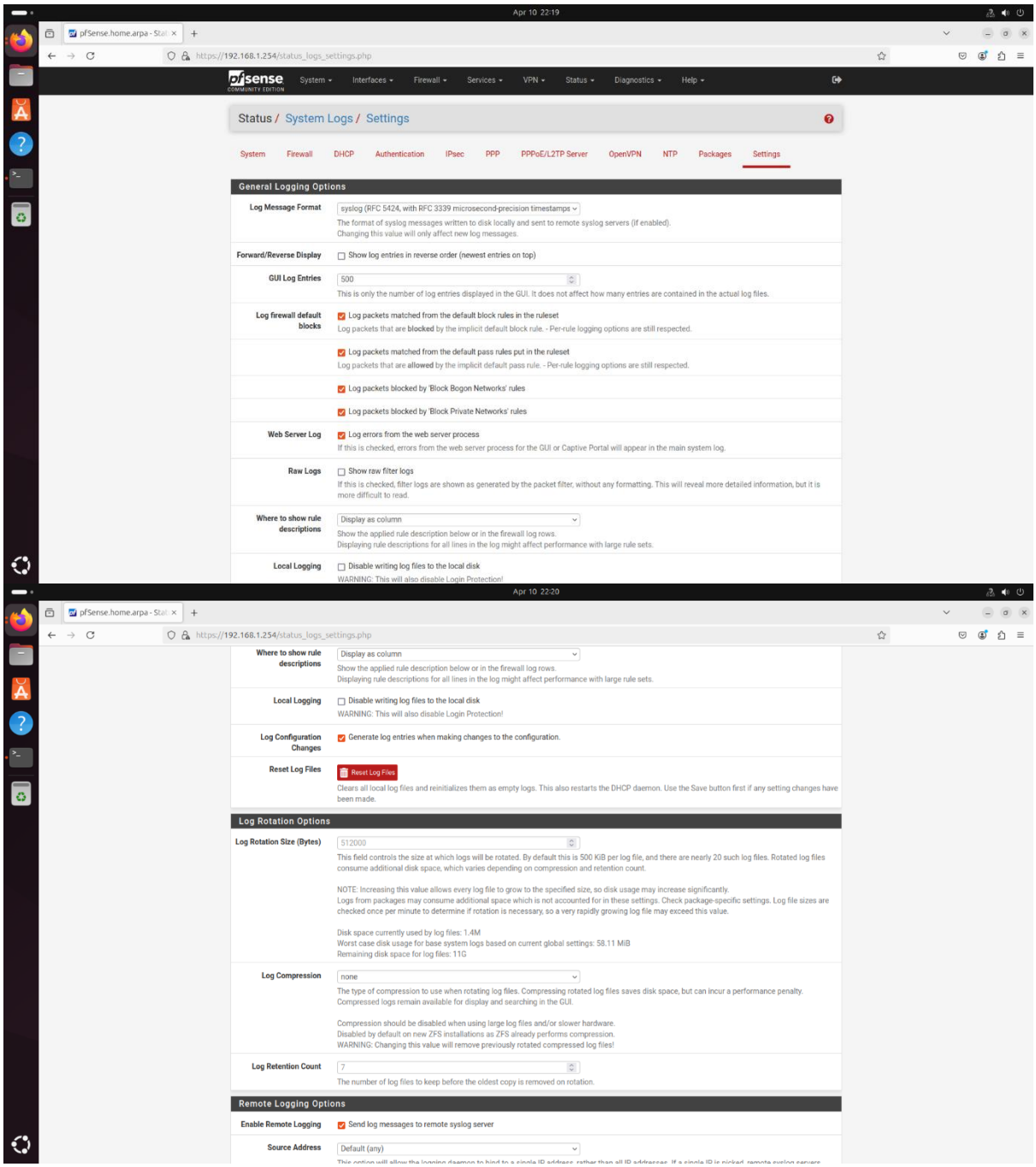
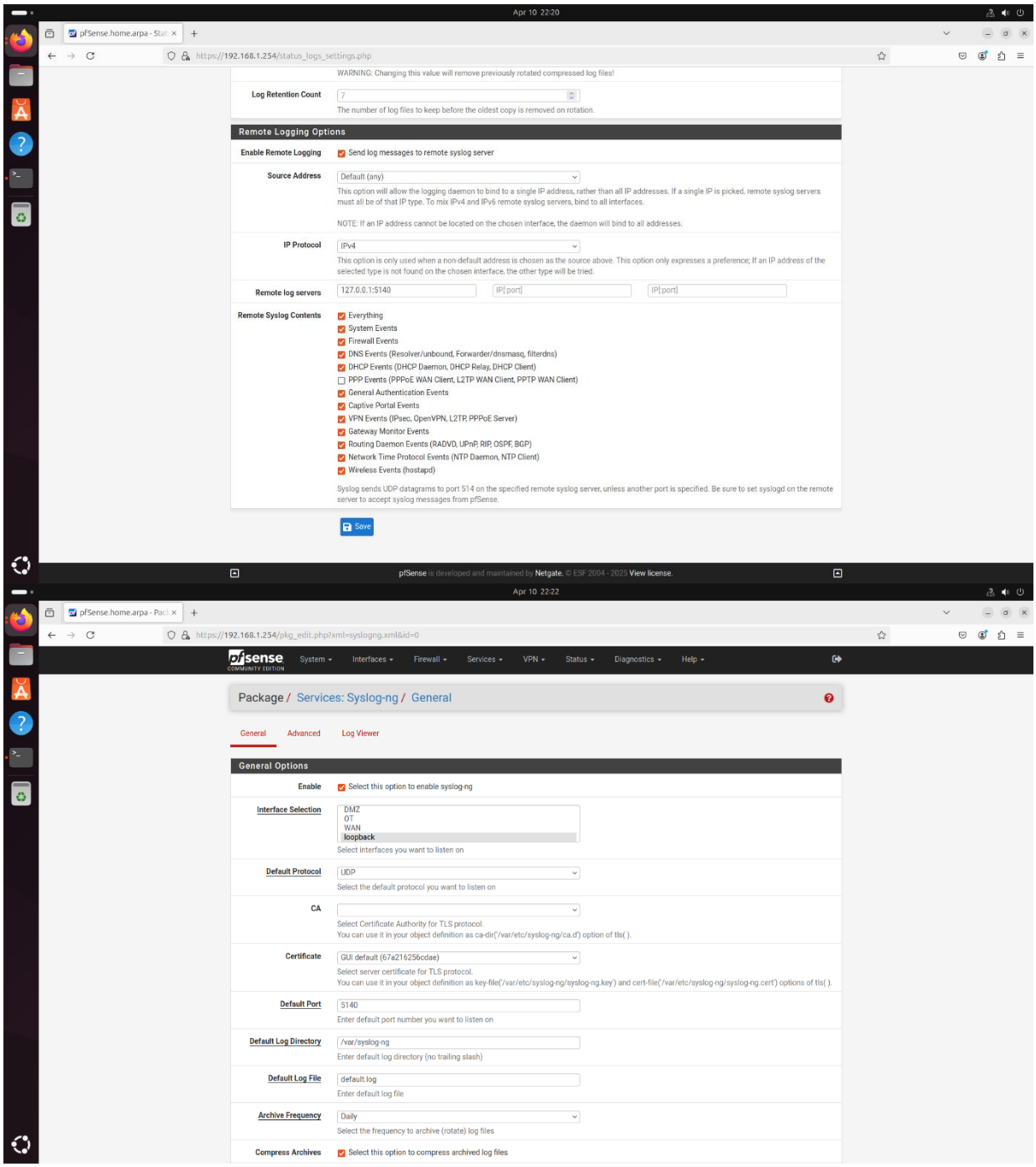
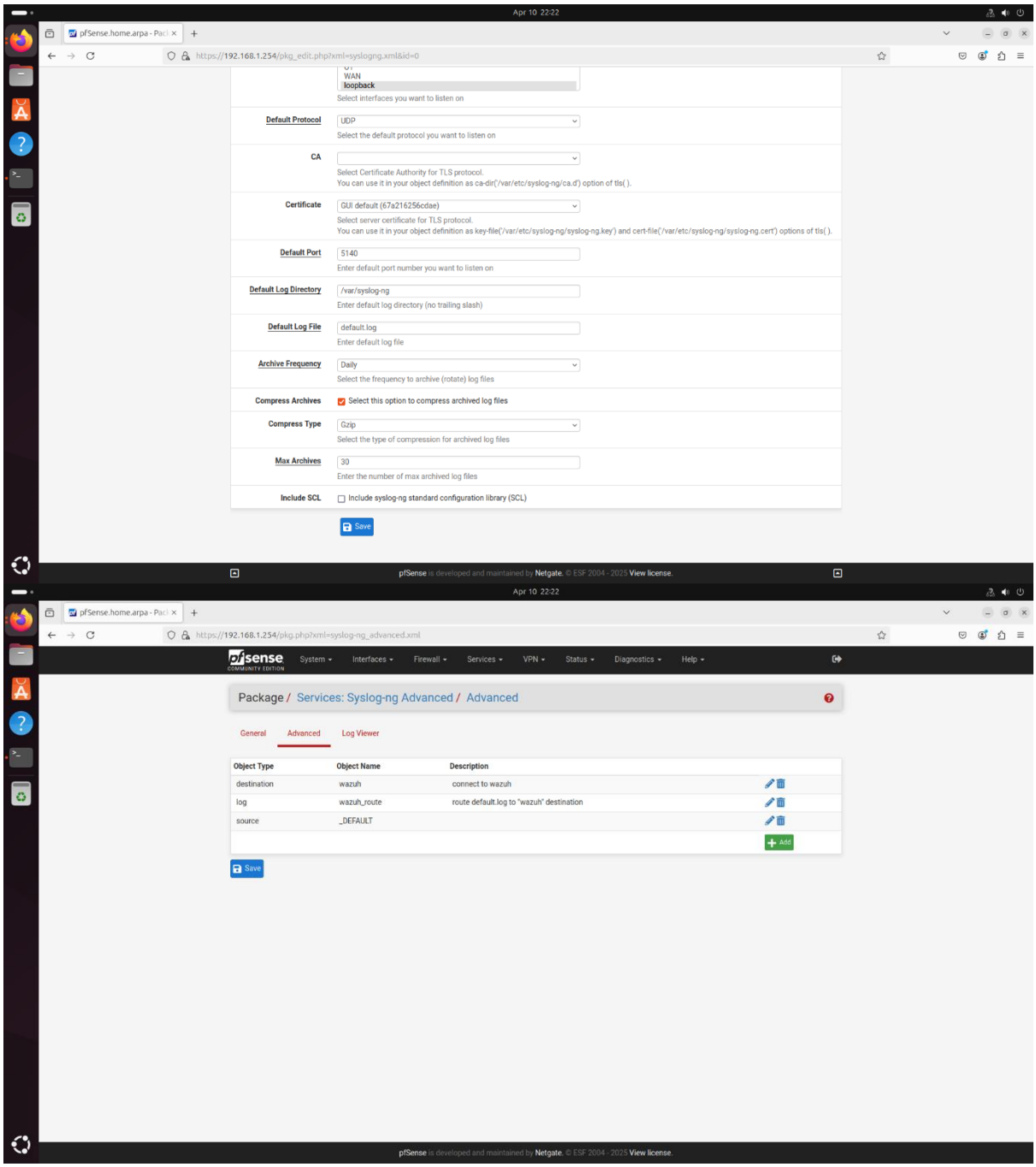
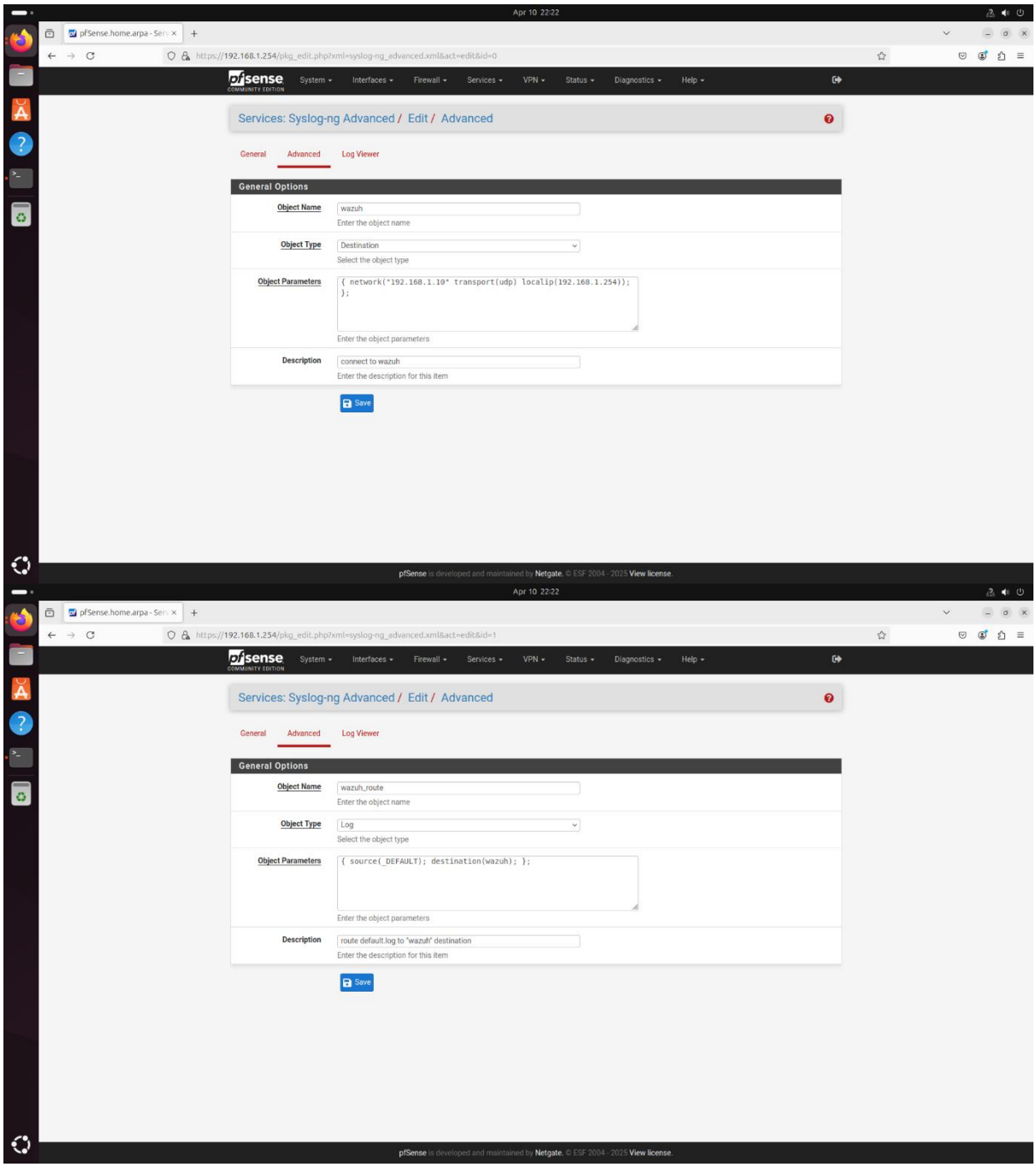


Figure B. 8: pfSense Block Rules Showing Dropped Unauthorized DNS and ICMP Traffic.


Apr 10 22:19
pfSense.home.arpa - Stat
https://192.168.1.254/status_logs_settings.php
pfsense COMMUNITY EDITION
System Interfaces Firewall Services VPN Status Diagnostics Help
Status / System Logs / Settings
System Firewall DHCP Authentication IPsec PPP PPPoE/L2TP Server OpenVPN NTP Packages Settings
General Logging Options
Log Message Format
syslog (RFC 5424, with RFC 3339 microsecond-precision timestamps)
The format of syslog messages written to disk locally and sent to remote syslog servers (if enabled).
Changing this value will only affect new log messages.
Forward/Reverse Display
Show log entries in reverse order (newest entries on top)
GUI Log Entries
500
This is only the number of log entries displayed in the GUI. It does not affect how many entries are contained in the actual log files.
Log firewall default blocks
Log packets matched from the default block rules in the ruleset
Log packets that are blocked by the implicit default block rule. - Per-rule logging options are still respected.
Log packets matched from the default pass rules put in the ruleset
Log packets that are allowed by the implicit default pass rule. - Per-rule logging options are still respected.
Log packets blocked by 'Block Bogon Networks' rules
Log packets blocked by 'Block Private Networks' rules
Web Server Log
Log errors from the web server process
If this is checked, errors from the web server process for the GUI or Captive Portal will appear in the main system log.
Raw Logs
Show raw filter logs
If this is checked, filter logs are shown as generated by the packet filter, without any formatting. This will reveal more detailed information, but it is more difficult to read.
Where to show rule descriptions
Display as column
Show the applied rule description below or in the firewall log rows.
Displaying rule descriptions for all lines in the log might affect performance with large rule sets.
Local Logging
Disable writing log files to the local disk
WARNING: This will also disable Login Protection!
Apr 10 22:20
pfSense.home.arpa - Stat
https://192.168.1.254/status_logs_settings.php
Where to show rule descriptions
Display as column
Show the applied rule description below or in the firewall log rows.
Displaying rule descriptions for all lines in the log might affect performance with large rule sets.
Local Logging
Disable writing log files to the local disk
WARNING: This will also disable Login Protection!
Log Configuration Changes
Generate log entries when making changes to the configuration.
Reset Log Files
Reset Log Files
Clears all local log files and reinitializes them as empty logs. This also restarts the DHCP daemon. Use the Save button first if any setting changes have been made.
Log Rotation Options
Log Rotation Size (Bytes)
512000
This field controls the size at which logs will be rotated. By default this is 500 KiB per log file, and there are nearly 20 such log files. Rotated log files consume additional disk space, which varies depending on compression and retention count.
NOTE: Increasing this value allows every log file to grow to the specified size, so disk usage may increase significantly.
Logs from packages may consume additional space which is not accounted for in these settings. Check package-specific settings. Log file sizes are checked once per minute to determine if rotation is necessary, so a very rapidly growing log file may exceed this value.
Disk space currently used by log files: 1.4M
Worst case disk usage for base system logs based on current global settings: 58.11 MiB
Remaining disk space for log files: 11G
Log Compression
none
The type of compression to use when rotating log files. Compressing rotated log files saves disk space, but can incur a performance penalty.
Compressed logs remain available for display and searching in the GUI.
Compression should be disabled when using large log files and/or slower hardware.
Disabled by default on new ZFS installations as ZFS already performs compression.
WARNING: Changing this value will remove previously rotated compressed log files!
Log Retention Count
7
The number of log files to keep before the oldest copy is removed on rotation.
Remote Logging Options
Enable Remote Logging
Send log messages to remote syslog server
Source Address
Default (any)


Apr 10 22:20
pfSense.home.arpa - Stat
https://192.168.1.254/status_logs_settings.php
WARNING: Changing this value will remove previously rotated compressed log files!
Log Retention Count
7
The number of log files to keep before the oldest copy is removed on rotation.
Remote Logging Options
Enable Remote Logging
Send log messages to remote syslog server
Source Address
Default (any)
This option will allow the logging daemon to bind to a single IP address, rather than all IP addresses. If a single IP is picked, remote syslog servers must all be of that IP type. To mix IPv4 and IPv6 remote syslog servers, bind to all interfaces.
NOTE: If an IP address cannot be located on the chosen interface, the daemon will bind to all addresses.
IP Protocol
IPv4
This option is only used when a non-default address is chosen as the source above. This option only expresses a preference; If an IP address of the selected type is not found on the chosen interface, the other type will be tried.
Remote log servers
127.0.0.1:5140
IP[:port]
IP[:port]
Remote Syslog Contents
Everything
System Events
Firewall Events
DNS Events (Resolver/unbound, Forwarder/dnsmasq, filterdns)
DHCP Events (DHCP Daemon, DHCP Relay, DHCP Client)
PPP Events (PPPoE WAN Client, L2TP WAN Client, PPTP WAN Client)
General Authentication Events
Captive Portal Events
VPN Events (IPsec, OpenVPN, L2TP, PPPoE Server)
Gateway Monitor Events
Routing Daemon Events (RADVD, UPnP, RIP, OSPF, BGP)
Network Time Protocol Events (NTP Daemon, NTP Client)
Wireless Events (hostapd)
Syslog sends UDP datagrams to port 514 on the specified remote syslog server, unless another port is specified. Be sure to set syslogd on the remote server to accept syslog messages from pfSense.
Save
pfSense is developed and maintained by Netgate. © ESF 2004 - 2025 View license.
Apr 10 22:22
pfSense.home.arpa - Pack
https://192.168.1.254/pkg_edit.php?xml=syslogng.xml&id=0
pfsense COMMUNITY EDITION
System
Interfaces
Firewall
Services
VPN
Status
Diagnostics
Help
Package / Services: Syslog-ng / General
General
Advanced
Log Viewer
General Options
Enable
Select this option to enable syslog-ng
Interface Selection
DMZ
OT
WAN
loopback
Select interfaces you want to listen on
Default Protocol
UDP
Select the default protocol you want to listen on
CA
Select Certificate Authority for TLS protocol.
You can use it in your object definition as ca-dir('/var/etc/syslog-ng/ca.d') option of tls( ).
Certificate
GUI default (67a216256cdae)
Select server certificate for TLS protocol.
You can use it in your object definition as key-file('/var/etc/syslog-ng/syslog-ng.key') and cert-file('/var/etc/syslog-ng/syslog-ng.cert') options of tls( ).
Default Port
5140
Enter default port number you want to listen on
Default Log Directory
/var/syslog-ng
Enter default log directory (no trailing slash)
Default Log File
default.log
Enter default log file
Archive Frequency
Daily
Select the frequency to archive (rotate) log files
Compress Archives
Select this option to compress archived log files


Apr 10 22:22
pfSense.home.arpa - Pack
https://192.168.1.254/pkg_edit.php?xml=syslogng.xml&id=0
WAN
loopback
Select interfaces you want to listen on
Default Protocol
UDP
Select the default protocol you want to listen on
CA
Select Certificate Authority for TLS protocol.
You can use it in your object definition as ca-dir('/var/etc/syslog-ng/ca.d') option of tls( ).
Certificate
GUI default (67a216256cdae)
Select server certificate for TLS protocol.
You can use it in your object definition as key-file('/var/etc/syslog-ng/syslog-ng.key') and cert-file('/var/etc/syslog-ng/syslog-ng.cert') options of tls( ).
Default Port
5140
Enter default port number you want to listen on
Default Log Directory
/var/syslog-ng
Enter default log directory (no trailing slash)
Default Log File
default.log
Enter default log file
Archive Frequency
Daily
Select the frequency to archive (rotate) log files
Compress Archives
Select this option to compress archived log files
Compress Type
Gzip
Select the type of compression for archived log files
Max Archives
30
Enter the number of max archived log files
Include SCL
Include syslog-ng standard configuration library (SCL)
Save
pfSense is developed and maintained by Netgate. © ESF 2004 - 2025 View license.
Apr 10 22:22
pfSense.home.arpa - Pack
https://192.168.1.254/pkg.php?xml=syslog-ng_advanced.xml
pfsense COMMUNITY EDITION
System
Interfaces
Firewall
Services
VPN
Status
Diagnostics
Help
Package / Services: Syslog-ng Advanced / Advanced
General
Advanced
Log Viewer
Object Type
Object Name
Description
destination
wazuh
connect to wazuh
log
wazuh_route
route default.log to "wazuh" destination
source
_DEFAULT
Add
Save
pfSense is developed and maintained by Netgate. © ESF 2004 - 2025 View license.


Apr 10 22:22
pfSense.home.arpa - Ser
https://192.168.1.254/pkg_edit.php?xml=syslog-ng_advanced.xml&act=edit&id=0
pfsense COMMUNITY EDITION
System
Interfaces
Firewall
Services
VPN
Status
Diagnostics
Help
Services: Syslog-ng Advanced / Edit / Advanced
General
Advanced
Log Viewer
General Options
Object Name
wazuh
Enter the object name
Object Type
Destination
Select the object type
Object Parameters
{ network("192.168.1.10" transport(udp) localip(192.168.1.254));
};
Enter the object parameters
Description
connect to wazuh
Enter the description for this item
Save
pfSense is developed and maintained by Netgate. © ESF 2004 - 2025 View license.
Apr 10 22:22
pfSense.home.arpa - Ser
https://192.168.1.254/pkg_edit.php?xml=syslog-ng_advanced.xml&act=edit&id=1
pfsense COMMUNITY EDITION
System
Interfaces
Firewall
Services
VPN
Status
Diagnostics
Help
Services: Syslog-ng Advanced / Edit / Advanced
General
Advanced
Log Viewer
General Options
Object Name
wazuh_route
Enter the object name
Object Type
Log
Select the object type
Object Parameters
{ source(_DEFAULT); destination(wazuh); };
Enter the object parameters
Description
route default.log to "wazuh" destination
Enter the description for this item
Save
pfSense is developed and maintained by Netgate. © ESF 2004 - 2025 View license.

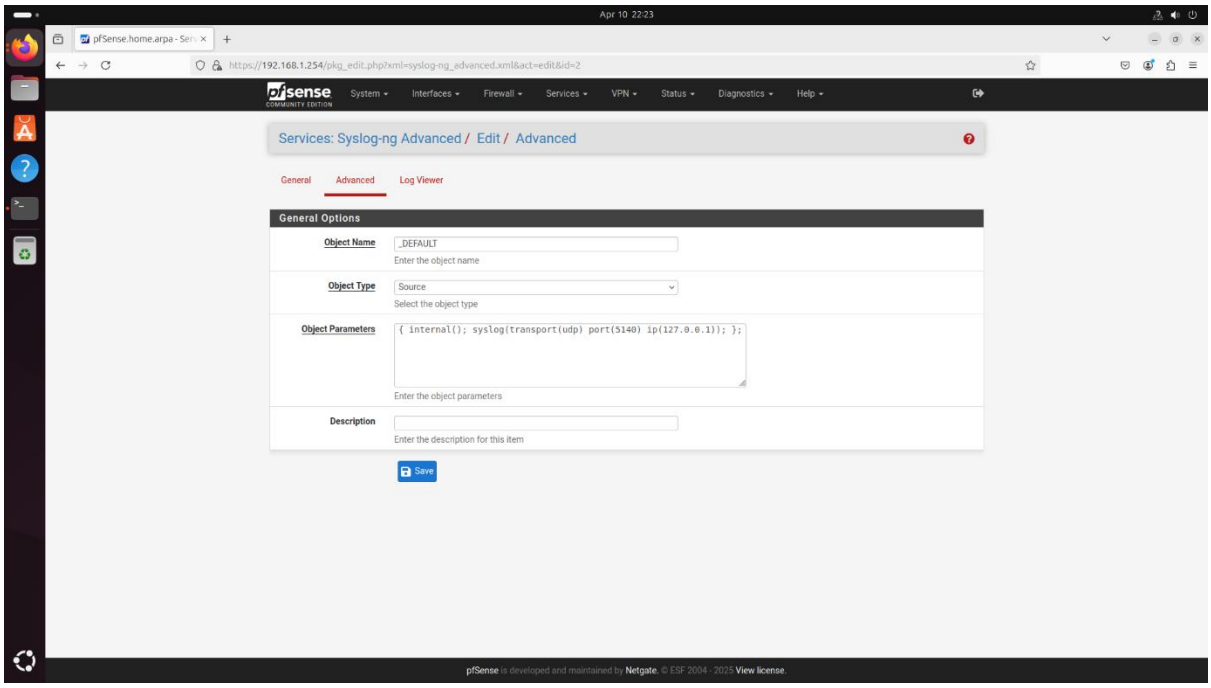


Figure B. 9: pfSense Syslog-ng Log Forwarding Setup Screenshot.

Appendix C: Wazuh SIEM Configuration Screenshots

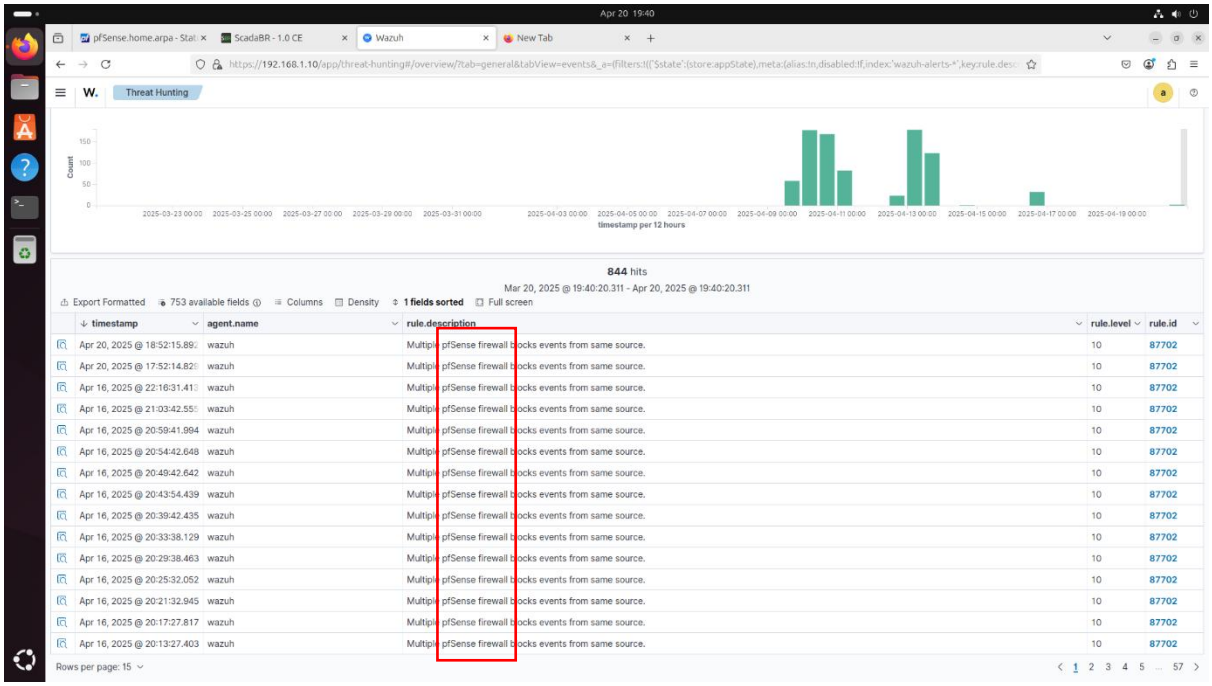


Figure C. 1: Wazuh Syslog Events from pfSense.

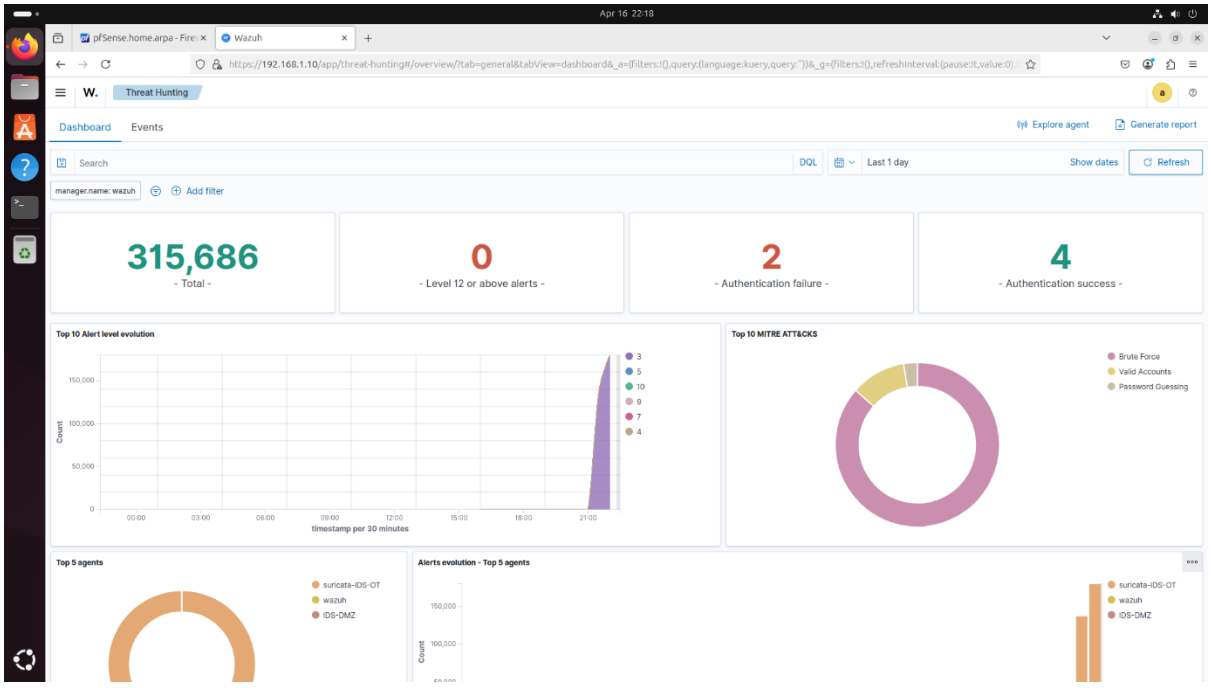


Figure C. 2: Wazuh Dashboard - General Monitoring Overview.

Appendix D: Suricata IDS Screenshots

Custom Suricata Rules for Modbus TCP

The following custom Suricata rules were developed and tested to detect suspicious Modbus TCP activities. These rules were placed in the local rules file (local.rules) on both IDS instances.

```
# Read Coils (Function Code 0x01)

alert tcp any any -> any 502 (msg:"Modbus TCP Read Coils Attempt"; content:"|00 00|"; offset:2;
depth:2; content:"|01|"; offset:7; depth:1; classtype:attempted-recon; sid:1000001; rev:1;)

# Write Single Coil (Function Code 0x05)

alert tcp any any -> any 502 (msg:"Modbus TCP Write Single Coil"; content:"|00 00|"; offset:2;
depth:2; content:"|05|"; offset:7; depth:1; classtype:attempted-dos; sid:1000002; rev:1;)

# Write Multiple Registers (Function Code 0x10)

alert tcp any any -> any 502 (msg:"Modbus TCP Write Multiple Registers"; content:"|00 00|";
offset:2; depth:2; content:"|10|"; offset:7; depth:1; classtype:attempted-dos; sid:1000003;
rev:1;)

# Preset Single Register (Function Code 0x06)

alert tcp any any -> any 502 (msg:"Modbus TCP Unauthorized Access Attempt"; content:"|00 00|";
offset:2; depth:2; content:"|06|"; offset:7; depth:1; classtype:policy-violation; sid:1000004;
rev:1;)

# Emergency Shutdown Coil Write (Target address 0x0000)

alert tcp any any -> any 502 (msg:"Modbus TCP Shutdown Coil Write (0x0000)"; content:"|00 00|";
offset:2;
```

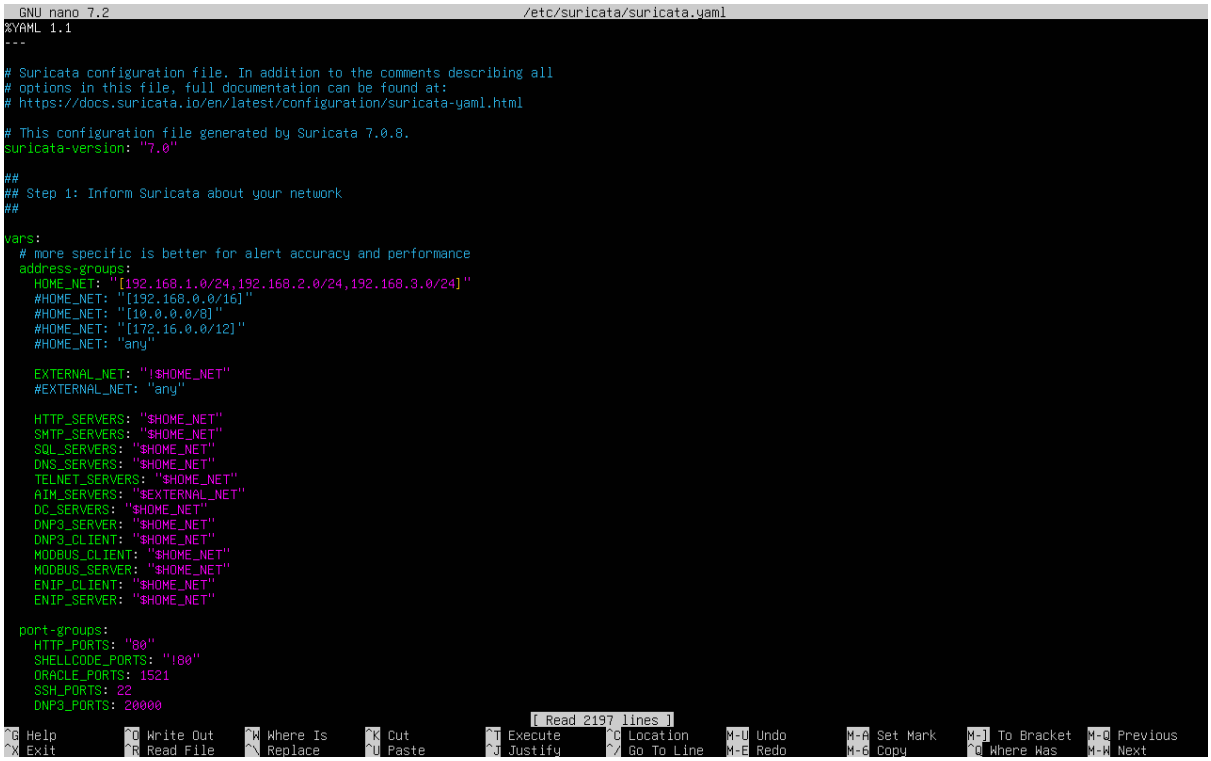


Figure D. 1: Suricata IDS Monitoring Configuration for LAN1, LAN2, and LAN3.

Figure D. 2: Suricata Detection of Suspicious Modbus Traffic.

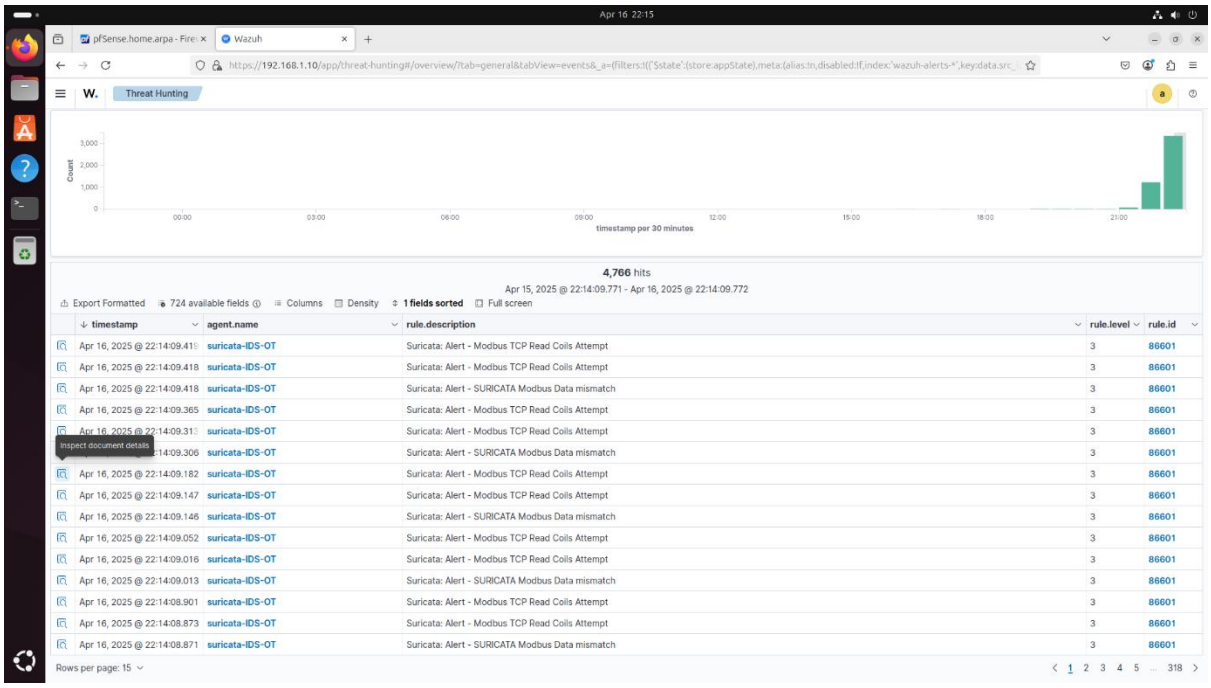


Figure D. 3: Suricata Alerts Visualized in Wazuh Dashboard.

Appendix E: Jumpserver MFA Configuration (Google Authenticator)

To enhance the security of SSH access to the Jumpserver, **multi-factor authentication (MFA)** was implemented using **Google Authenticator**. This setup added an additional layer of protection beyond traditional username and password logins, aligning with Zero Trust principles.

Installation of Google Authenticator PAM Module

On the Jumpserver (a Debian/Ubuntu-based Linux VM), the `libpam-google-authenticator` package was installed:

```
sudo apt update
sudo apt install libpam-google-authenticator
```

Each authorized user then executed the following command to initialize their MFA secret:

```
google-authenticator
```

This process generated a QR code (for scanning into the Google Authenticator app), recovery codes, and secret keys.

PAM Configuration

To enforce TOTP-based MFA during SSH logins, the PAM configuration for SSH was updated:

File: `/etc/pam.d/sshd`

```
auth required pam_google_authenticator.so
```

SSHD Configuration

To support TOTP as a second factor, SSH server settings were modified:

File: `/etc/ssh/sshd_config`

```
ChallengeResponseAuthentication yes
UsePAM yes
AuthenticationMethods password,keyboard-interactive
```

The SSH daemon was restarted to apply changes:

```
sudo systemctl restart sshd
```

Testing and Logging

Users were required to:

1. Enter their standard SSH password
2. Enter the 6-digit TOTP code from their Google Authenticator app

Successful and failed login attempts were logged under `/var/log/auth.log` and forwarded to the **Wazuh SIEM** system via the Wazuh agent. These logs enabled centralized audit and monitoring of authentication events.

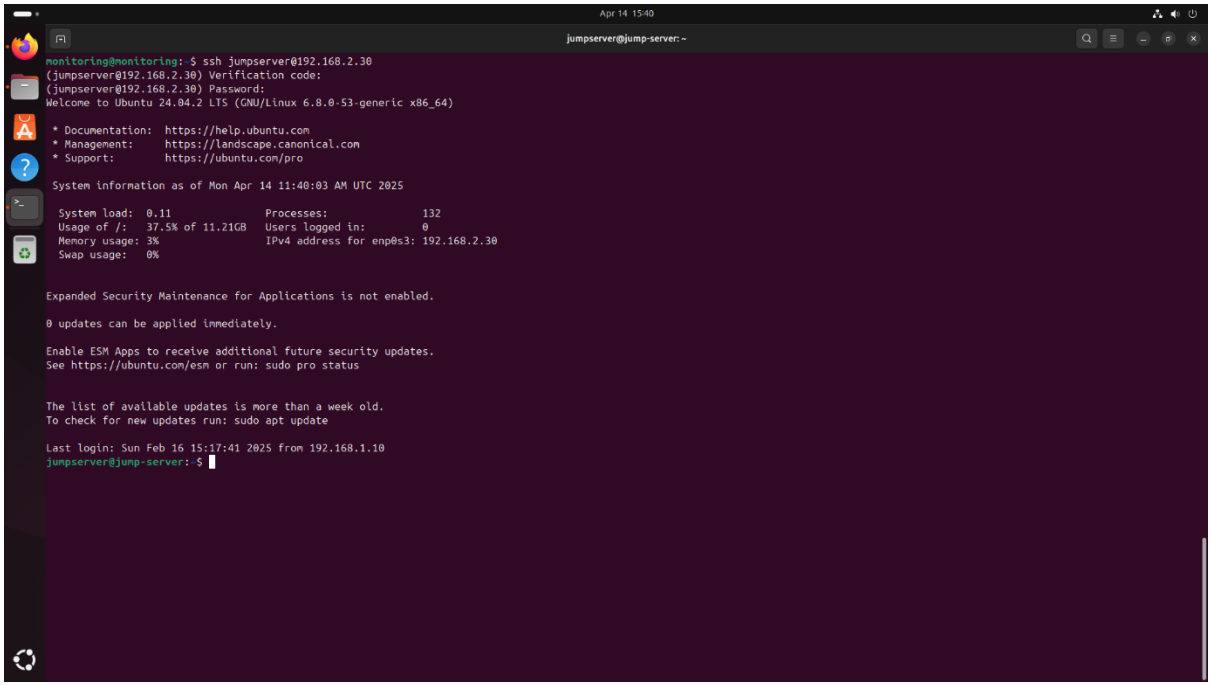


Figure E. 1: SSH Session to Jumpserver from IT Network.

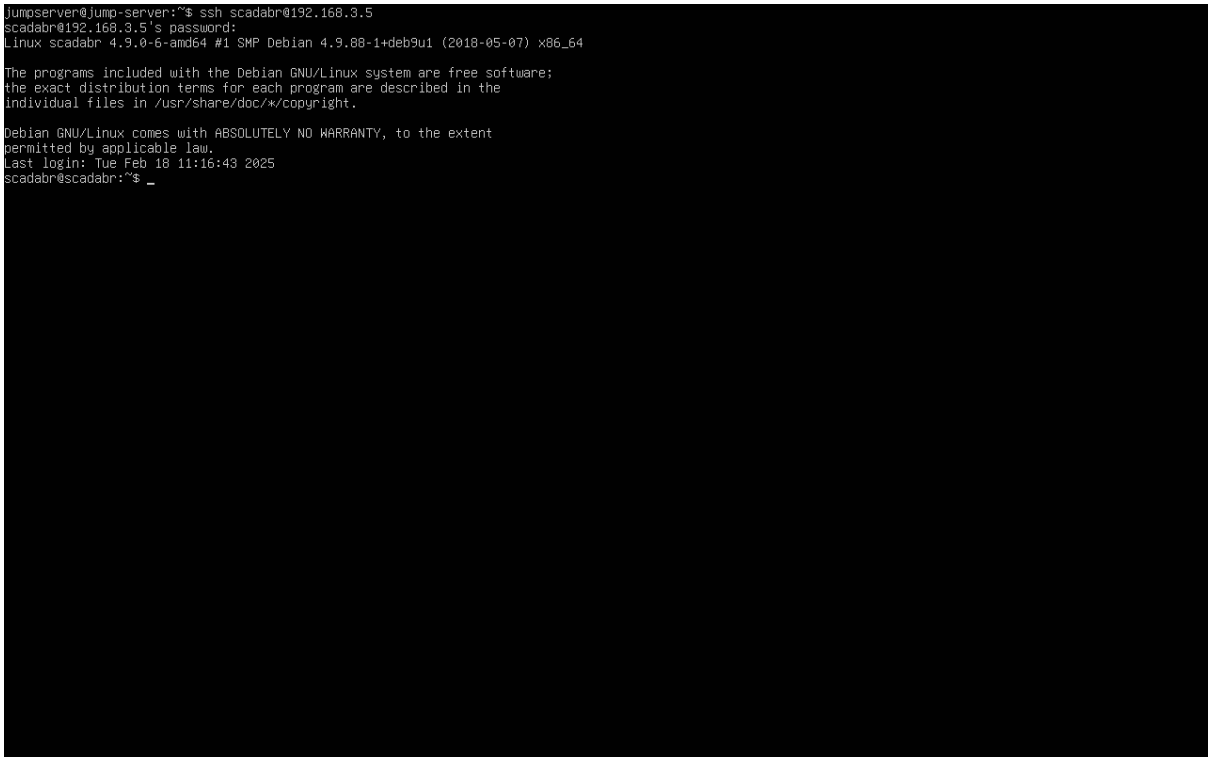


Figure E. 2: Jumpserver SSH Access to SCADA System in OT.

Appendix F: Encryption Deployment Screenshots

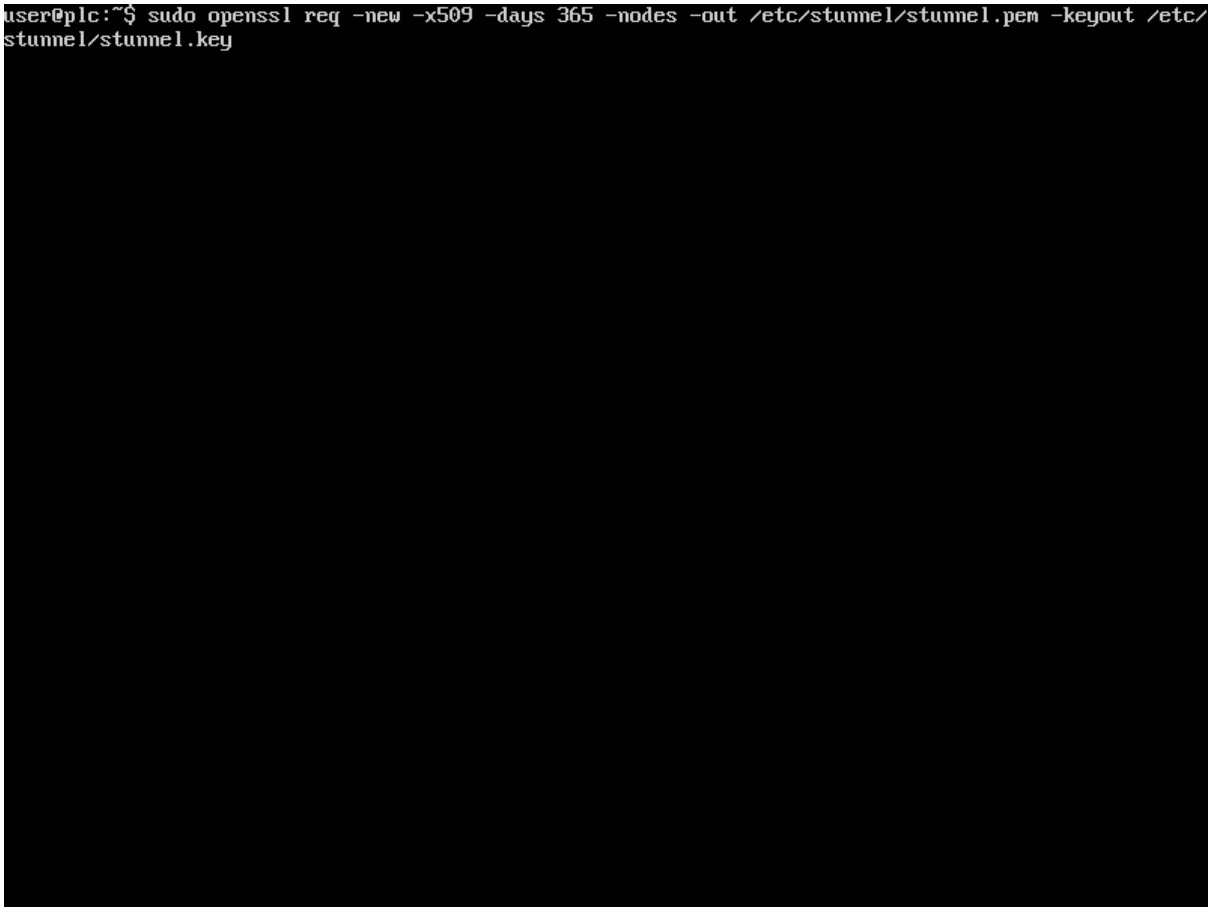


Figure F. 1: OpenSSL Certificate Generation for TLS Encryption.

Figure F. 2: stunnel Server-Side Configuration on PLC.

Figure F. 3: stunnel Client-Side Configuration on SCADA System.

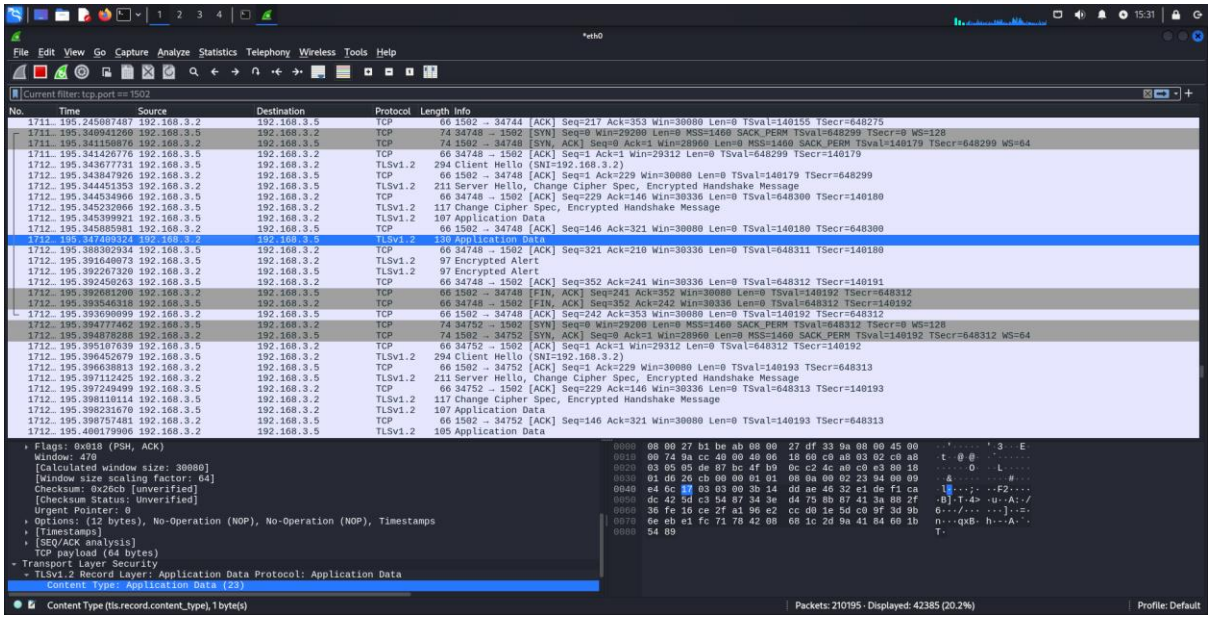


Figure F. 4: Wireshark Capture Showing TLS Handshake and Encrypted Modbus Traffic.

Appendix G: Testing and Attack Simulation Screenshots

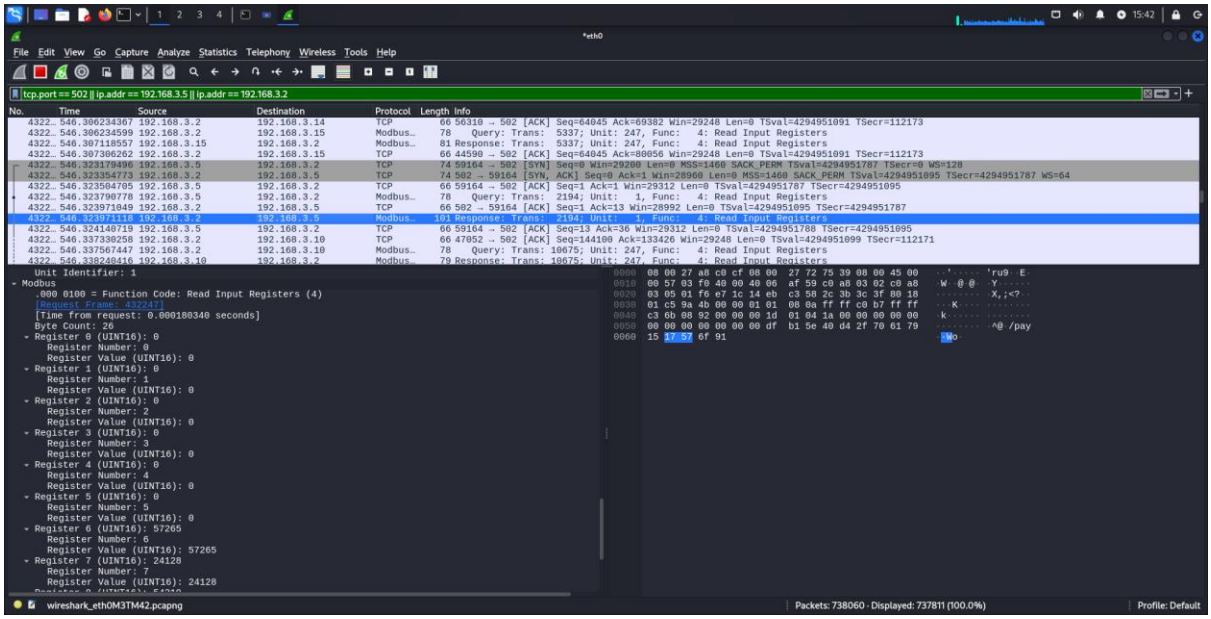


Figure G. 1: Wireshark Capture Showing Unencrypted Modbus TCP Traffic.

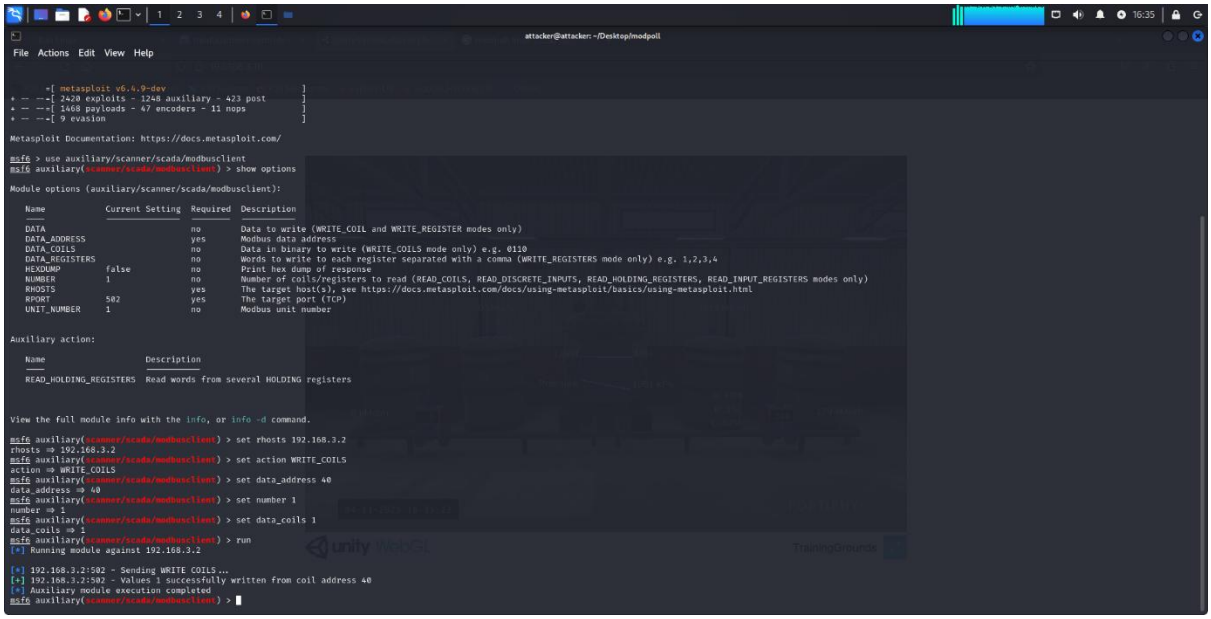


Figure G. 2: Metasploit Modbus Client Module Setup for Attack.

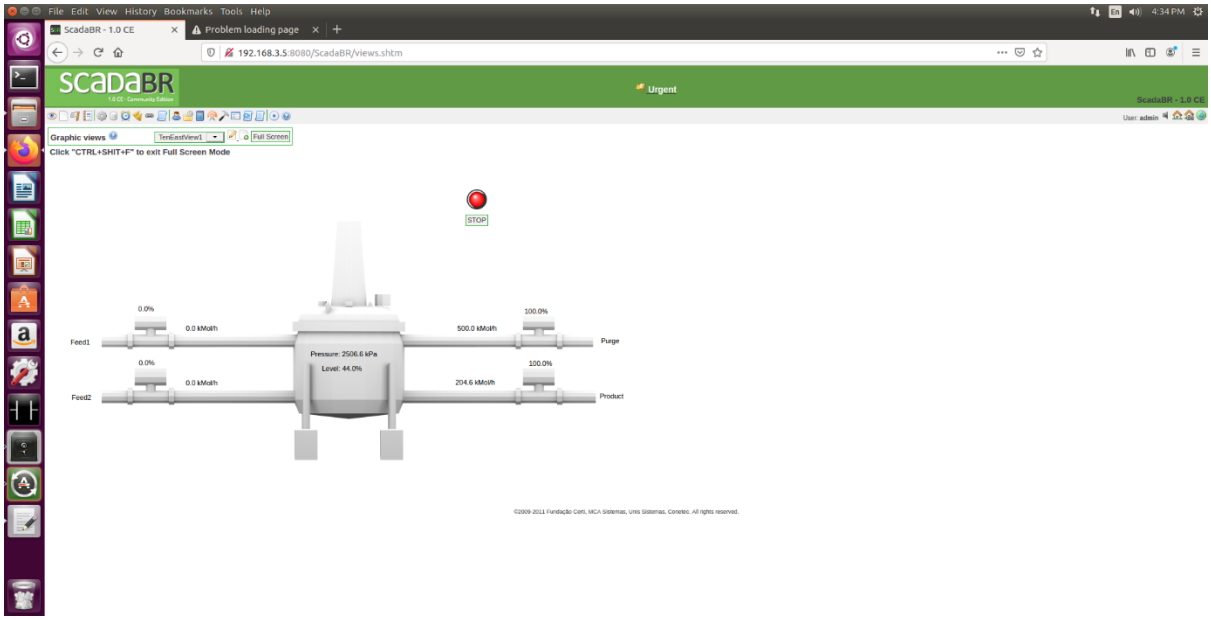


Figure G. 3: SCADA HMI Showing Emergency Shutdown.

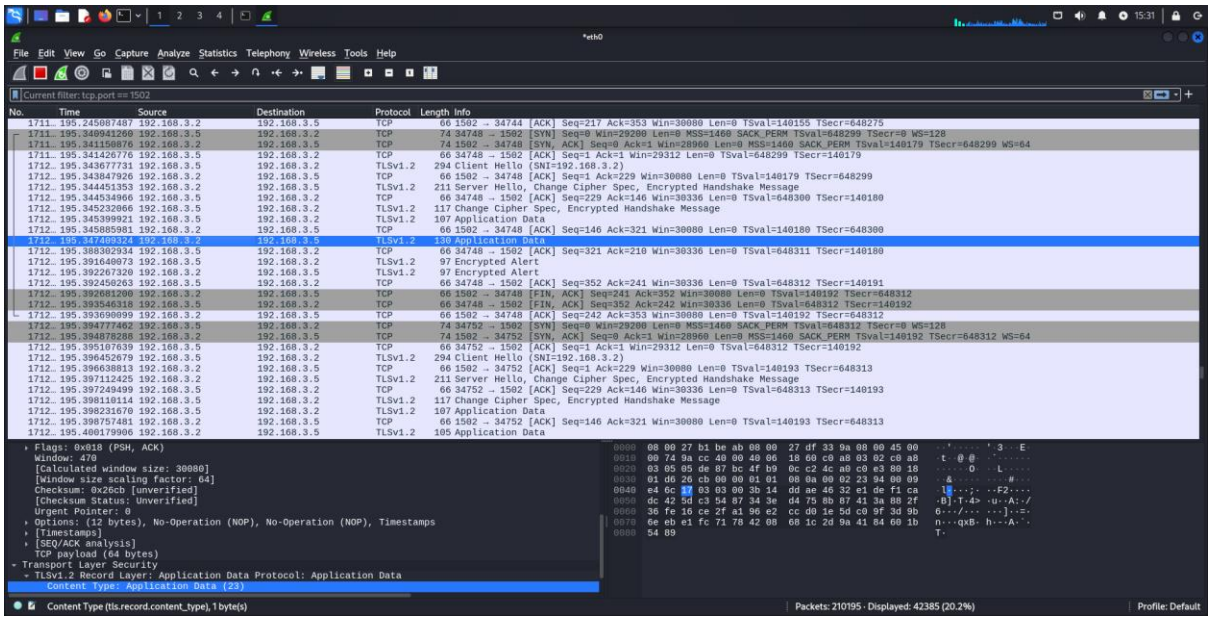


Figure G. 4: Wireshark Capture Showing TLS-Secured Modbus Traffic.

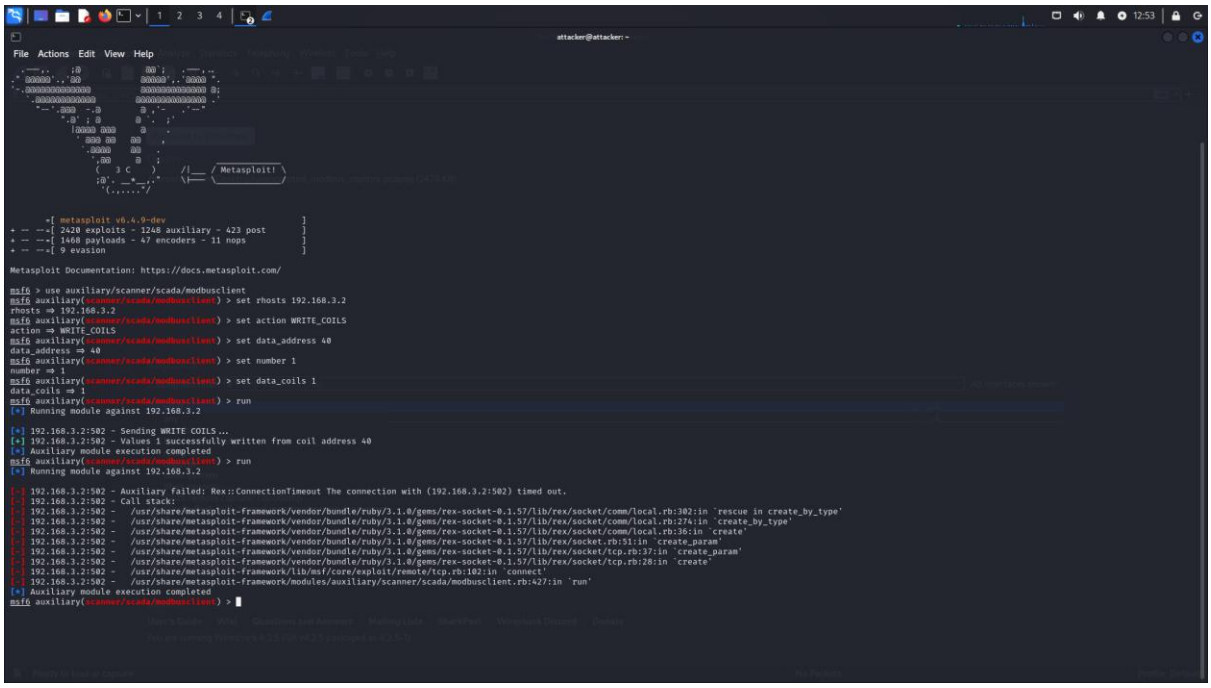


Figure G. 5: Failed Attack Attempt on TLS-Protected Modbus Communication.

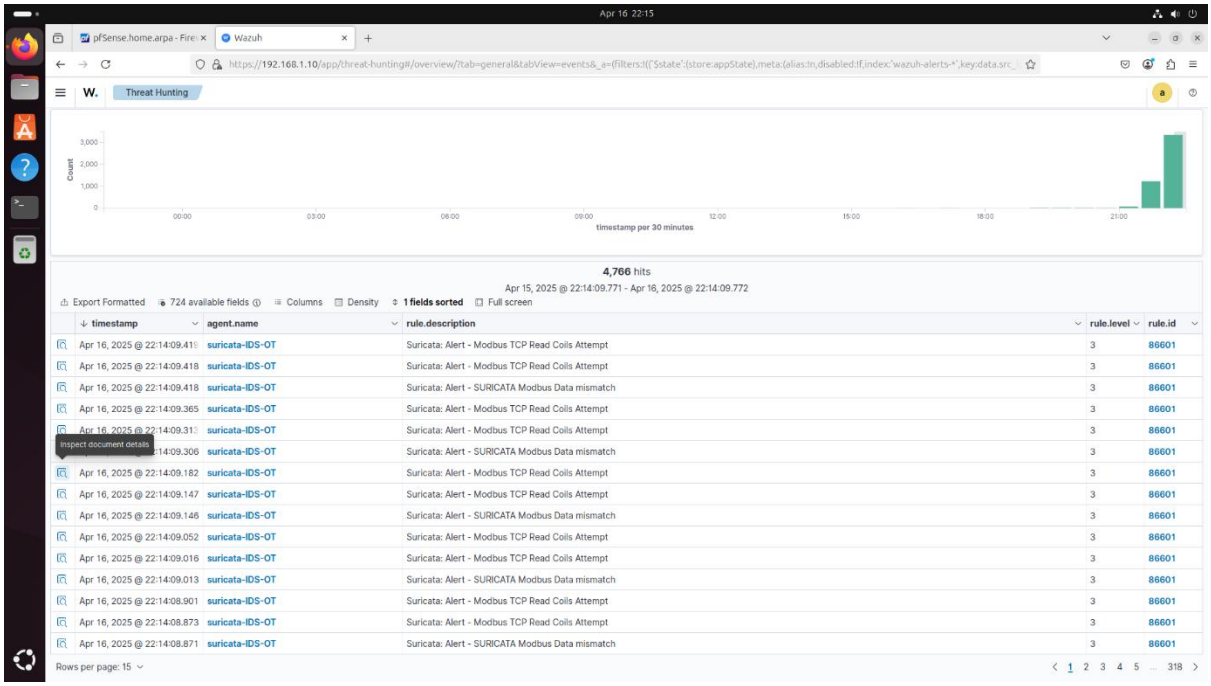


Figure G. 6: Wazuh Alert Log Showing Unauthorized Modbus Activity.

Figure G. 7: Suricata Alert Showing Suspicious Modbus Communication.