Automatic Monitoring, Threat Detection, Alarm Generation

Bashar Al-Saify, Max Wetterström, Erik Nordström, Patric Rönn

***Abstract*—The rapid expansion of Internet of Things devices and the adoption of cloud-edge computing architectures have revolutionized data processing and security management. However, these advancements have also exposed systems to new security threats, requiring solutions for monitoring, detecting threats and alarm generation. This paper presents a literary analysis of these approaches, including Intrusion Detection Systems and multi-layer security in a cloud-edge environment. This was done by categorizing the techniques used, their methodologies and the strengths and weaknesses. A comprehensive summary is also presented using the metrics of impact, robustness and complexity.**

***Index Terms*—Monitoring, Alarm generation, Privacy, Machine learning, IoT Security, Cloud Security, Cloud-Edge, Multi-Layer Security**

I. Introduction

In today's digital landscape, the rapid proliferation of Internet of Things (IoT) devices and cloud-edge has revolutionized the way organizations manage and secure their data. As industries increasingly rely on these technologies to enhance operational efficiency, scalability and time sensitive data, the vulnerabilities associated with interconnected systems have also risen dramatically. Cyber threats are becoming more sophisticated requiring advanced strategies for automatic monitoring, threat detection and alarm generation.

Automatic monitoring is crucial for identifying anomalies and potential security breaches in real time enabling organizations to respond swiftly to emerging threats. Traditional methods of intrusion detection often fall short due to their reliance on predefined signatures and static rules making them inadequate against new and unknown attacks. There is a pressing need for intelligent systems capable of adaptive learning and proactive threat identification.

This paper explores the integration of automatic monitoring, threat detection and alarm generation within cloud edge environments. We examine the roles of Intrusion Detection Systems (IDS), Cyber Threat Intelligence (CTI) and collaborative approaches to enhance security measures. By investigating the current landscape and addressing the challenges in effective alarm generation and false positive reduction this study aims to provide valuable insights into improving the resilience of cloud-edge networks against evolving cyber threats.

In this report we categorize the different technologies and discuss their respective architecture and methods. We take a deeper look on cloud edge systems and IoT role in a cloud edge system. We also classify the different security mechanisms at the different layers, how edge device monitoring works with different IDS to generate alarms in order to detect threats. We will analyze how to mitigate vulnerabilities across the protocol layers in regards to multi-layers security.

A. Background

The rise of IoT and cloud-edge computing has transformed industries that depend on large-scale data processing and real-time operation. IoT are interconnected devices that collect, send and process data. These devices are designed to be energy-efficient and optimizing performance while minimizing power consumption. Cloud-edge computing extends the computational capabilities of the cloud closer to the edge network, reducing latency and improving overall efficiency. This enables more efficient data processing and load sharing, but it also introduces new layers of complexity, particularly in terms of managing distributed resources and securing data transmission across diverse environments and networks. An edge device in IoT offer a small amount of processing power and memory in-order to be more memory efficient. Another aspect is multi-layer security where you divide a system into their respective protocol layers, i.e the physical layer, data link layer, network layer, transport layer and application layer. Classifying both attacks and defenses into which layer or layers they operate on can be beneficial in assessing what parts of a system they target.

II. IoT

IoT devices face multiple challenges during development of the devices, since they have various architectures regarding what protocol stack and data format the system uses. IoT devices range from simple sensors to complex systems. Depending on the use case, IoT devices can be positioned at the edge of the network or within it [1]. Some of the transport messaging protocols in use are Message Queue Telemetry Transport (MQTT), which uses publishers whose task is to produce data like a sensor. It then sends it to the MQTT broker who collects the messages and analyze who it is from and the data contents as seen in Fig. 1 [2]. Wireless Sensor Networks (WSNs) are one system that efficiently utilizes MQTT, and they are a large network of low energy sensors. They utilizes the MQTT broker as a data storing point and to processes the real-time data in-order to perform scanning of an environment [3].

There are several network protocols developed for IoT devices, one such network protocol is Long Range Wide Area Network (LoRaWAN), which connects IoT devices that are off the grid using sub-gigahertz radio band and are regionally regulated.

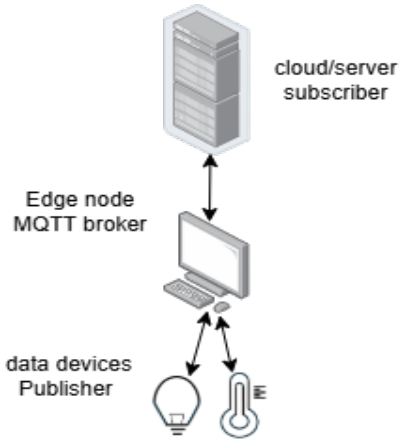

Fig. 1. MQTT data visualization.

Due to this, regulation available channels can be mixed, therefore the throughput ranges from 250bps to 22kbps [4]. Another widely adopted network protocol in use is Zigbee, it supports short range, low power communication often in used in IoT home devices. Zigbee uses coordinators similar to MQTT where the data is relayed through the network from router nodes to a root node and the root node handles the data from the edge device where the network can have the form of either a star, tree or mesh [5].

III. Cloud-Edge System

There are many of different kinds of implementations of cloud-edge systems depending on their use case. It is a system that bridges the gap between the cloud and edge systems, this makes it so when edge devices require more processing power, they can offload tasks to the cloud, which offers vast resources that can be dynamically allocated based on the specific needs of the system [6]. The system can be individual devices connected to the cloud or a network connected to an edge node and then connected to the cloud like the system in Fig. 2.

A. Edge systems

An edge system is a system that connects multiple sources and are close to the data source. It is used to ease with computing, storage and networking to smaller devices that are made to be energy efficient and memory effective. Edge devices can for example, precompute requests from a device in the network and also save important information in order to, save on time and data transmission [7]. There are multiple technologies that use an edge system, for example 6G, to reduce response time, increase data throughput on the network, and the ability to add AI to the technology [8]. Edge systems are also utilized in 5G networks through Multi-access Edge Computing (MEC), which reduces latency and traffic congestion and enabling cloud-offloading in the network [9].

B. Data security at the edge

There are multiple approaches to secure data generated at edge networks. Since there are multiple types of networks, there are various different aspects to control in order to ensure security at the edge. One critical aspect to monitor and ensure security is data integrity, as data packets are sent through intermediate devices that can be altered, for example, via a Man in the Middle attack.

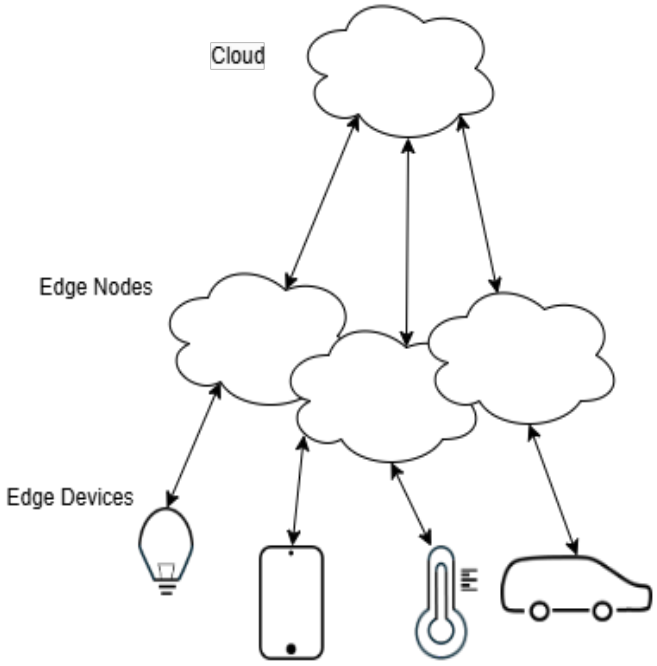

Fig. 2. High level design of a cloud-edge system.

Zigbee addresses this issue by using a cryptographic Message Integrity Code (MIC) to provide data integrity and authenticity for the MAC header [10]. Similarly, LoRaWAN uses a Cipher based Message Authentication Code (CMAC) to ensure data integrety and authenticity [11]. The primary difference between MIC and CMAC implementations is that MIC employs hash based functions whereas CMAC uses block cipher and is generally more energy efficient making it suitable for off-grid devices [12].

Both protocols utilize the AES stream cipher to encrypt the data payload in order to ensure data confidentiality. While Zigbee does permit the reuse of the same encryption keys in order to simplify the encryption, this creates an encryption vulnerability. However, with the current known techniques and computational power today, breaking the encryption is impractical.

The main difference between LoRaWan and Zigbee is their range: LoRaWan is long range while Zigbee is short range and therefore they have different security aspects to examine. LoRaWan need to be more tolerant against sniffing attacks because of the longer range. In contrast, sniffing a ZigBee connection requires a close proximity to the devices.

Moreover there are not only external threats against the network but also internal threats that needs to be addressed. Devices within a network are often given higher trust due to authentication and inclusion, which poses a security risk [13]. Addressing this trust issue in the network is not yet standardized, but there has been some research to build and

optimize the trust in a zero trust network model [14]. The authors use a trust value in order to use a path that has the highest trust in a WSN network, this makes that the device has to build up trust in order to be included and trusted within a network. If a device has a low score it could be flagged for further analysis.

C. Monitoring

Monitoring a system can be a rather complex task, especially when the system is large and distributed. The main purpose of monitoring is to record events in the environment. These records can later be used to debug issues and find the root cause of a problem. In the security domain, logs are useful when attempting to generate alarms of ongoing cyberattacks and when one wants to determine what damages an attack was able to inflict on a system.

There are a multitude of possible combinations of which information can be logged. The NIST standard SP 800-92 [15] contains information that can aid a security expert in selecting what types of data are relevant, see the following list for a summary.

- System Events
- Audit Records
- Client requests and server responses
- Account information
- Usage information
- Significant operational actions.

The same standard also defines a way to guarantee the integrity of the logs. A message digest have to be calculated for each log file to ensure that it has not been altered after creation. When ingesting the file to a storage facility the digest should be recomputed, if they to not match the log has been altered and cannot be trusted. A message digest is a digital signature most commonly implemented through a hashing algorithm such as MD5 or SHA.

Furthermore, the value of the collected data can drastically vary depending on where a system is monitored. A cloud system is thought to have 7 layers; facility, network, hardware, operation system, middle-ware, application and user. In order to not fail to collect vital information about an attack, all of these layers should be monitored [16].

Traditionally, one was inclined to place e.g. an IDS on the main link that intersects the gateway router. When attempting to monitor devices on the cloud-edge, this is rather unfeasible due to the sheer amount of data that are normally flowing through such a link and the fact that isolated IDS's have weaknesses. There is a rather large probability that such a naive approach would cause more harm than gain in the form of network congestion and unidentified cyberattacks.

Fung et. al build on the issues with isolated IDS's [17], arguing that they are ineffective in detecting unknown threats. They thus propose an Intrusion Detection Network (IDN) as a possible solution, similar to Fig. 3. Simply put, this is a network of collaborating IDS's that share detections and knowledge between each other using a peer-to-peer protocol. More specifically they investigate consultation-based IDN's, meaning that an IDS in the network can consult and receive feedback from other IDS's in the network when it is unsure of whether an activity is malicious or not. The proposed IDN uses a fully distributed Bayesian trust model that yields a high degree of robustness and scalability. Furthermore the solution aggregates and uses feedback from previous consultations when making a decision on wether to raise and alarm or not. This decreases the rate of false positives and false negatives. The IDS's in the network uses an incentive when allocating resources to a neighboring IDS. This incentive is based on trustworthiness and how much resources other IDS's are allocating to the specific IDS. Combined with an algorithm that is able to pair collaborators given any arbitrary context a high efficiency can be obtained at a low cost.

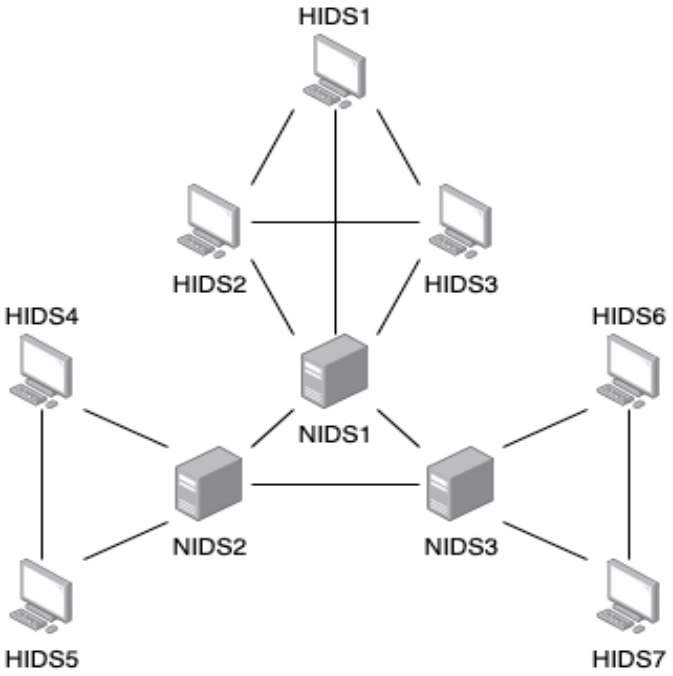

Fig. 3. High level design of a Collaborative IDN.

Even though the use of an IDN seems to be more effective than isolated IDS's when it comes to detection of potential threats, one prominent question remains. How should the probes, or in this case the IDS's that constitute the IDN be deployed? Tundo et al. presents an array of probe deployment patterns [18] that could provide guidance. The authors define a probe holder as an object that hosts one or many executable probes. If an adversary could obtain control of a holder, all of its probes would be compromised. A reserved pattern uses a reserved holder for each user in the system, see Fig. 4. It contains the damage if an adversary were to obtain access to a holder. This would naturally increase failure containment and mitigate security risks, since there are no interference between users. It does however come with some drawbacks. The number of holders and probes grow proportional to the number of users, and a higher economic cost will be incurred to utilize the resources effectively.

In the industry monitoring could be implemented using Filebeat, Fleet, Logstash and Elasticsearch [19],[20]. Fleet is

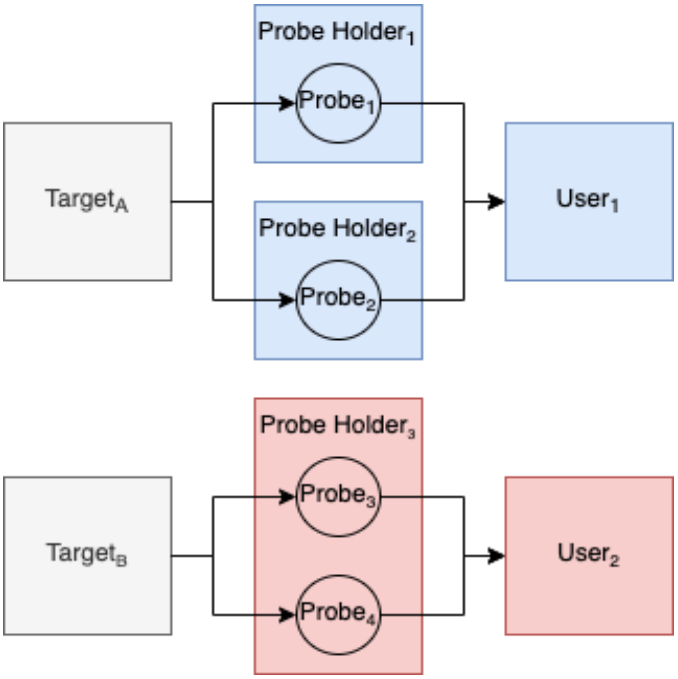

Fig. 4. One way to implement a reversed pattern.

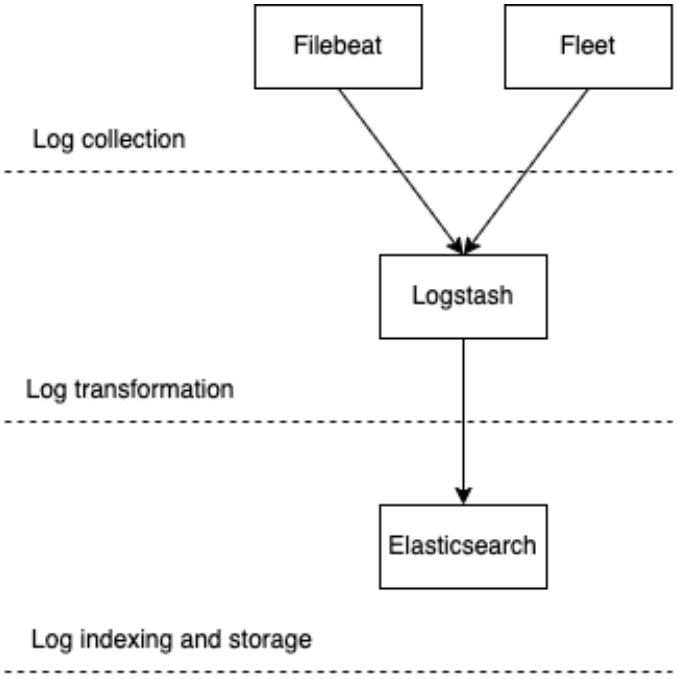

Fig. 5. Collection and processing of logs for monitoring.

a command and control server that manages Elastic Agents deployed on nodes in the network. The main purpose of these agents is to collect data and create logs. Filebeat then locates all of the logs created by the agents and assigns a harvester to each one. A harvester reads a log and sends the events to Libbeat. There, the events are centralized in order to be deduplicated and aggregated. Next, the processed events are sent to Logstash where they are filtered, enriched and transformed into structured JSON objects. Finally, the objects are ingested into Elasticsearch where they are indexed and stored. An overview of the process is visualized in Fig. 5.

D. Intrusion detection system

IDS has been an essential tool for cybersecurity due to the rapidly growing complexity and sophistication of cyberattacks. IDS plays a pivotal role in protecting critical data and systems. As the digital landscape expands including multiple environments such as IoT devices, cloud environments and broad corporate networks.

IDS are essential as cybersecurity tools designed to monitor, detect and respond to unauthorized or malicious activities within a network [21]. Traditional IDS are designed to identify and respond to unauthorized or malicious activities within a network or host environment as can seen in Fig. 6 [22]. These systems use two main detection methods, signature based detection and anomaly based detection [21]. Both methods require computational capabilities and workflows to ensure robust monitoring and alerting mechanisms.

Signature based IDS operates by matching analyzed data against a database of predefined attack patterns or signatures [21]. Sensors such as network sniffer capture real-time network packets and host activity logs and then analyze the data to extract the important components, such as packet headers, payloads or system logs [22]. The identified components are then compared with stored signatures using algorithms like Boyer-Moore or Knuth Morris Pratt for string matching and more complex patterns are identified using regular expressions [21]. When a signature has matched, the system generates an alert and logs the event at the same time it notifies the administrator of the incident [22]. Tools like Snort are example of the implementation of signature based IDS by offering a large signature database for network packet analysis [17]. This approach is efficient for known threats due to the use of optimized structures like hash tables for faster lookups [22]. But at the same time it is limited to only detect known threats, and for that reason in order to remain effective, the IDS has to be regularly updated.

An anomaly based IDS identifies deviations in behavior compared to an established baseline of normal behavior [21]. The baseline of normal behavior is based on the patterns of the data that is used to train the system [21]. The patterns can be such as network traffic, system usage, login frequencies and bandwidth consumption. When the training phase is over, the IDS uses real-time data to continuously compare it with the learned baseline and any significant deviations will be flagged as anomalies [21]. Detection relies on statistical thresholds or clustering algorithms such as K-Means or DBSCAN (Density Based Spatial Clusting of Applications with Noise) to identify and categorize anomalies. K-Means is a well-known clustering method used in data mining and machine learning to group data points into clusters [23]. The second algorithm, DBSCAN, is a popular clustering algorithm that is used to identify clusters in data by grouping points that are closely packed together and marking points in low-density regions as noise [24]. Anomaly based IDS can be efficient

in identifying unknown threats and zero-day attacks, but it often struggle with high false-positive rates, especially in more dynamic environments with shifting behaviors. At the same time, the computational demands require significant power due to the need for ongoing model updates and complex real-time processing [21].

Traditional IDS faces several challenges as the amount of data increases. As network traffic volumes grow, maintaining real-time analysis becomes difficult without distributed processing and multi-threaded architectures [22]. More advanced attacks use techniques like packet fragmentation or encrypted payloads to evade detection requiring more complex tools to detect, such as Deep Packet Inspection (DPI) [25]. When it comes to anomaly based systems, they are efficient for a certain amount of attacks, but in a more dynamic environment the baseline can lead to frequent false positives reducing the level of trust in the system [21]. Despite the challenges traditional IDS systems face, it remains fundamental in cyberscurity. In order to detect and mitigate threats, the IDS leverages sophisticated data collection, statistical analysis and machine learning [22]. The evolution and development of IDS continues to play a vital role in securing systems against an ever-expanding threat landscape [17].

The growing complexity of network attacks, especially in IoT environments where limited computational resources and specialized protocols, make traditional IDS approaches less effective [26]. Instead of relying on inspecting packet payloads, which is challenging in encrypted or complex networks, an anomaly based IDS analyzes the behavioral patterns in network traffic to identify malicious activity [26]. Generative Adversarial Networks (GAN) are a concept in machine learning, especially in the field of deep learning [27]. GAN consists of two neural networks, a generator and a discriminator that works in opposition to each other in a process known as adversarial training. The framework allows GAN perform different things such as learn the underlying distributions of data, generate realistic data, and improve the robustness of machine learning models [27]. The generator in the network is responsible for creating synthetic data that mimic the real data it learns during training. For example, in image processing, the generator creates images that appear as if they are part of the original dataset, even though they are entirely synthetic [27]. In applications like intrusion detection it works similarly, the generator produces traffic patterns that mimic malicious or adversarial perturbations in order to simulate real world scenarios [28]. On the other hand, the discriminator acts as a classifier that has the task to distinguish between the real data and the synthetic data created by the generator. The discriminator learns to evaluate and improve its accuracy over time, improving its accuracy in identifying patterns and detecting anomalies in the data. This allows the model to generate synthetic examples that help the IDS to recognize and neutralize perturbations effectively, de-noising the data. The improved IDS can then classify threats more reliably even under more complex and sophisticated attack conditions.

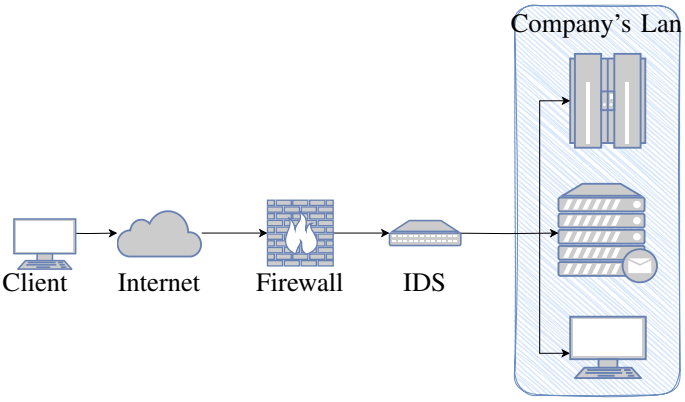

Fig. 6. Simplified example of where IDS are placed in the system network.

E. Cyber threat intelligence

The authors of [29] explain that CTI is necessary for organizations because it provides a comprehensive understanding of potential and active threats. The CTI is important for proactive threat hunting and mitigation because it does not only involve responding to detected threats, but anticipating and preventing attacks before they occur. By adopting a proactive strategy, the security teams will be able to detect early Indicators Of Compromise (IOC) and respond quickly to reduce potential damage. CTI can be enhanced by integrating open-source intelligence (OSINT) by expanding the range of data and automating its processing.

The paper [29] presents a system called ThreatRaptor, designed to enhance cyber threat hunting by utilizing OSINT. The ThreatRaptor improves the efficiency of detecting sophisticated cyber threats by leveraging OSINT through different types of mechanisms. The system uses an unsupervised Natural Language Processing (NLP) pipline to indentify and extract IOCs and their relationship to each other from OSCTI texts. The external repositories can help to find IOC (such as IP address, files hashes, domains and URL) which is pieces of evidence that can indicate a system breach or malicious activity. This is achieved by using an algorithm that breaks down the text into blocks to process sentence by sentence and identify dependencies between words.The Threatraptor uses the identified words and their relationship to each other to construct a structured threat behaviour graph of it consisting of nodes that represents each IOC and edges denoted the relationships. The graph can be used to provide a visual representation of the identified IOCs and how they are connected to each other such as communication or data transfer actions.
The ThreatRaptor uses the data from the created graph together with Threat Behavior Query Language (TBQL) to automatically synthesize queries to find matching system audit records in the monitored log data, as seen in Fig. 7. This synthesis allows translation of the graph into actionable search queries that scan system logs for potential threats without manual quert crafting [29].
The traditional method for cyber-threat hunting requires security analysts to manually construct queries based on log data or observed suspicious behavior to identify potential threats.

This process can be time-consuming and labor-intensive when dealing with large datasets or complex attacks. By using ThreatRapotor it simplifies the process by leveraging OSCTI to automatically generate queries for detecting potential threats. When ThreatRaptor ingests external threat intelligence, it uses this intelligence to automatically create queries that align with the data. These queries are designed to search for specific signs of compromise or malicious activity in system logs or network traffic.

The *ThreatRaptor* is considered a light-weight system that operates effectively without requiring continuous human oversight or intricate configuration. Those properties of a system enhance its scalability, making it suitable for deployment in environments that involve large datasets or complex infrastructures. The nature of the systems ensures efficient resource usage providing comprehensive threat hunting capabilities and monitoring with minimal manual intervention.

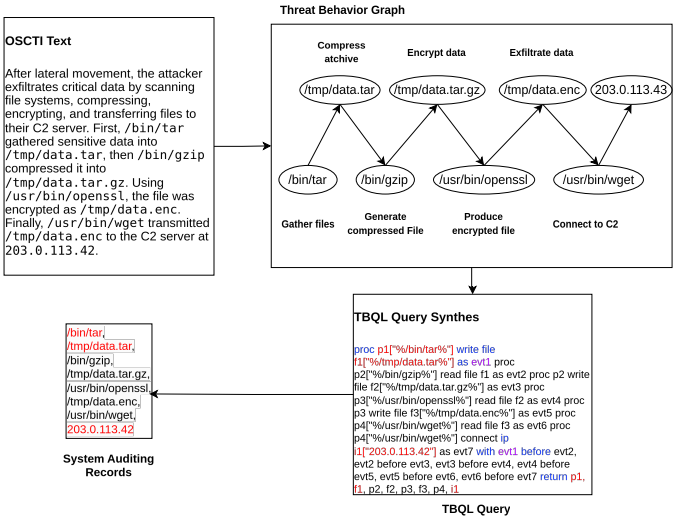

Fig. 7. An example data leakage attack case demonstrating the whole processing pipeline of THREATRAPTOR.

F. Alarm generating systems

Creating alarm generating systems using monitoring data is not a trivial task and can be prone to errors. In order to combat this issue Albuquerque et al. proposes two patterns, *Development Tracking* and *Exception Tracking* [30] that intend to limit the number of errors and anomalies in addition to making them more visible to the engineering team. The authors classifies Deployment Tracking as a practice intended to correlate system anomalies with recent deployments of software or new hardware. The practice aims to give developers a clear answer to whether an anomaly in the system is caused by a recent deployment or not. In the context of generating alarms, this is vital since they generally have a maximum response time associated with them. If a developer does not notice that a recent deployment introduced an anomaly in the system, it could render the service useless. In order to prevent this, every change and deployment to production should be tracked in a way that clearly highlight system anomalies in relation to deployment times. It is not an uncommon practice to store exceptions in a local log file. In some scenarios, this might be sufficient, but when we are talking about cloud systems, this approach poses some real issues. The most prominent issue in this context is how a developer is supposed to figure out if a deployment is causing an anomaly if the logs are distributed. In order to prevent this scenario the Exception Tracking pattern can be used. Instead of storing exceptions that might have been caused by an unknown anomaly in a local log file, it would be beneficial to send the exceptions to a centralized tracking service. By doing this, the service can perform de-duplication and aggregate exceptions. By extension, this enables more efficient tracking of issues and their resolutions, resulting in alarms with higher trustworthiness.

Now consider a system architecture where logs from multiple edge devices are sent to a centralized tracking service. These logs might contain anomalies or exceptions that should trigger an alarm, or they could be clean, only containing normal behavior. As previously mentioned the accuracy and recall of an alarm generating system need to be very high if it are to provide any real value. Thus, we need a way to distinguish between benign and malicious activity within the collected logs.

One way to do this is by using a Convolutional Neural Network (CNN). Abdelsalam et. al [31] tried this approach by evaluating to what degree both a 2D and 3D CNN are able to correctly label malicious samples. Deep learning (DL) does however come with the same drawbacks as the rest of the Machine Learning (ML) spectrum. The model needs to be trained, and parameters need to be tuned. Even if this computational overhead is disregarded, the F1 score usually asymptotically approaches a value less than 1, regardless of additional training and optimization. Their CNN's were comprised of two convolution layers followed by a fully connected one before a prediction was made. The F1 scores were somewhat promising, reaching approximately 0.79 and 0.90 for the 2D and 3D versions, respectively.

Although the F1 scores for the 3D CNN look promising, a ML model can leak sensitive information. Differential Privacy (DP) could mitigate this leakage with an impact cost. Ababi et. al [32] implemented differentially private stochastic gradient descent (SGD) algorithms in TensorFlow, a framework that enables a programmer to distribute the training of a ML model over multiple nodes. Their results showed that an accuracy of around 70% could be achieved with this approach. The epsilon parameter was set to a value between 2 and 8 and their tests were performed using different noise levels on the CIFAR-10 dataset.

Distributed ML is not perfect though and comes with its own issues. Verbraeken et al. [33] found that the number of nodes could have to be quadrupled to increase training speedup by a factor of 1. They also concluded that distributed ML lacks in the privacy department. If the aforementioned leakage of sensitive information is excluded, it struggles with keeping training data separated across nodes, and statistical noise can limit the usefulness of the model.

One way to combat data leakage from ML models is to employ Federated Learning (FL) that improves data locality. Strong

data locality does however not guarantee privacy by itself, a system needs to prevent message inference during both the training portion and in the final model. Truex et al. [34] proposed an FL approach that utilizes both DP and Secure Multiparty Computation (SMC) to mitigate inference while achieving an acceptable accuracy. They found that their model can be used to train an arbitrary ML model with a high degree of scalability while being robust and having an F1 score in the upper 15th percentile range.

Since a system that generates alarms from logs processes a massive quantity of data, the field of Big Data research is also applicable. Distributed systems such as Apache Hadoop and Spark could thus be used. Huang et al. [35] were able to show that an implementation in Hadoop Distributed File System (HDFS) could detect the deletion of files, but it is unclear what challenges it faces.

Continuing on the example of how monitoring could be implemented in the industry, we can extend it to also include alarm generation, as seen in Fig. 8. Kibana provides a graphical interface to the JSON data stored in Elasticsearch [36]. It also allows a user to write rules that scan the data for patterns using the Event Query Languange (EQL) [37]. Thus, a user can write rules aimed to detect specific malicious sequences, interesting anomalies, and correlations. If these rules are triggered, an alarm can be generated and viewed in Kibanas security dashboard. As a side note, Kibana also offers integrated ML solutions that can perform root cause analysis, unsupervised anomaly detection, supervised classification regression and much more.

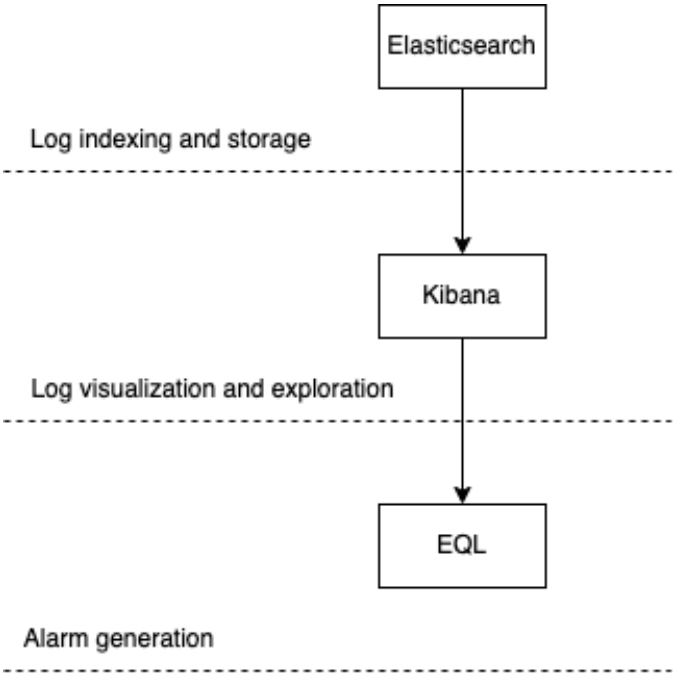

Fig. 8. Generation of alarms based on log data.

IV. Attack Detection in Multi-Layer Security

When examining security in a system, you can choose to divide it into their respective protocol layers to see which aspect of the system an attack or defense targets [38]. The layered structure of a WSN network includes the following.

1) Physical layer: The first layer represents the flow of actual binary data through some medium.
2) Data link layer: The second handles the transport of data between nodes in the network.
3) Network layer: The third layer provides transport of information with routing and forwarding.
4) Transport layer: The fourth layer breaks up data into packets and handles sending of data with protocols.
5) Application layer: The fifth and final layer is responsible for handling data to and from an end user or end system.

Breaking up security of a system into their respective layers of the protocol stack enables a more focused assessment of vulnerabilites. This helps in classification and to pinpoint its operational impact on the system, which is beneficial in accelerating vulnerability discovery and mitigation.

Different kinds of systems will have varying levels of activity across the protocol layers, so classifying a threat according to where in the system it is prevalent can help in more effectively identifying and terminating it at a quicker pace.

To effectively implement multi-layer security in scenarios involving cloud and edge networks, data aggregation can aid in reducing computational overhead [39]. By utilizing selective data transmissions, a more optimized workload could be enforced at central systems, reducing the risk for data floods.

To fuse data between protocol layers can be an additional measure in strengthening defenses. By leveraging information gathered through a cross-layered method, weaknesses in traditional NIDS systems can be mitigated, such as reducing false positives and false negatives [40].

A. Physical Layer

In the physical layer, there are mainly attacks of certain characteristics which can occur [38]. An eavesdropping attack is when you passively intercept traffic and monitor it. It is an essential prerequisite for many other attacks and are often deterred through encryption and effective protocol standards.

Another type of attack is a jamming attack, where the attacker interrupts the communication of a node entirely. This can be achieved either through electromagnetic interference or to otherwise make disruptions to being able to transmit data.

An Advanced Continuous-Time Convolution (ACTC) technique to detect jamming and replay attacks in LoRaWAN based WSN was proposed [41]. The model proposed showed an attack detection rate of 100% in the range of 100KHz to 250KHz, and for 500KHz the jamming attacks had a detection rate of 93.33% and replay attacks 95%. For the two former bandwidths, the false positive rate (FPR) was zero, but was 0.55 for 500KHz, indicating that the model becomes less effective for higher bandwidths.

A compromised node attack is another attack which is when a node has been compromised in a way that it is controlled by an adversary, which means it can manage and manipulate the functionalities and resources that the node has access to. A node like this is usually a point from where other attacks can

be launched from and is often a prerequisite for more complex attacks.

A more energy efficient method for detecting compromised untrusted nodes in a WSN is proposed [42]. A compromised node is often a point from where other attacks can be launched, so having mechanisms in place like Node Compromise Detection (NCD) to deter this type of attack is important. Other detection schemes for NCDs have either been behavior based or attestation based and has had their limitations. Behavior based schemes detect compromised nodes but they do not revoke them, while attestation based can be very resource intensive. In the paper they instead propose their own hybrid solution which does both attestation and revocation while keeping computational overhead low resulting in a more energy efficient detection scheme. Their simulations showed results for a reduction in false positives for compromised nodes while also reducing the computational overhead compared to other previous schemes such as ZoneTrust.

A replication attack is when a legitimate node is cloned by an adversary. Since identifying attributes like keys and id are also duplicated to the clone, it may seem legitimate and can therefore be harder to detect. If this clone is reprogrammed to act in a way that an adversary desires, then malicious activity can then be carried out through a node like this.

Strategies for detecting and mitigating node replication attacks in mobile wireless sensor networks (MWSN) have been proposed [43]. It proposes a framework using a decentralized method of nonce-based authentication. If a node fails to authenticate itself by providing an incorrect nonce, it is flagged as an impostor. It is subsequently put into quarantine by being added to a quarantine list, prohibiting further communication. Two schemes are presented, a centralized scheme where a base station collect claims, and a distributed scheme where each node independently detect and quarantine impostors. It also evaluates adaptive mechanisms for detection of replication attacks, focusing on estimating the number of impostors and adjusting its threshold dynamically. The threshold here is the number of claims against a specific node needed to flag it as an impostor. Two schemes are proposed for this. The doubling scheme which increase this threshold exponentially, but may add a delay in detection time. The other is an incremental scheme, where the threshold is increased gradually based on a constant value. Experiments on this show that a higher node density increased detection time, and that a distributed scheme offers superior resiliency than a centralized approach. It also shows that an incremental scheme offers better efficiency and lower false positives compared to doubling.

B. Data Link Layer

The main characteristic attacks when examining the data link layer are collision attacks, denial of sleep attacks and intelligent jamming attacks [38]. A collision attack is when an adversary manages to modify a packet so that upon packet checksum inspection, a mismatch will occur, triggering a packet drop. A denial of sleep attack is when the adversary targets the energy consumption of nodes in the WSN network by hindering them to go into sleep mode to save energy, which is a critical aspect to preserve in a sensor network. An intelligent jamming attack is an attack targeting the rules of communication protocols with the aim to disrupt communication and consume energy.

A scheme against denial of sleep attacks was proposed [44]. To conserve the critical aspect of energy in a sensor, a node will go into an idle state after a period of being inactive. In a denial of sleep attack, a so called anti node can repeatedly send transmissions to a node to keep it in an active state, effectively making it continuously consume power and overtime, decrease its overall lifetime. What the paper introduces is a secure scheme to authenticate nodes through the use of hash chains for authentication and symmetric encryption. The paper however does not provide any experimental results, so there is no definitive proof for the effectiveness for this proposed scheme.

C. Network Layer

The network layer in a WSN network is susceptible to a number of different kinds of attacks [38]. Common ones include, Sybil, black hole, gray hole, wormhole, sinkhole and hello flooding attack.

A black hole attack is an attack where a compromised node is used to redirect traffic to that specific malicious node [45]. The node broadcasts a message of high energy availability, making more neighboring nodes direct their messages to it. The malicious node then does not forward any messages. In a paper from Vishalo et. al, black hole attacks and their impact according to metrics in WSN networks for the LEACH protocol and detection of it using an algorithm are examined. They identify valuable metrics for evaluating the impacts of a black hole attack and subsequently measured their impact on performance and service degradation.

A gray hole attack utilizes selective forwarding and is a partial black hole attack as drops all packets except a selective few. It could for example be packets which contain certain information, or that packets are continuously dropped after a certain period of time.

A new detection method for selective forwarding attacks in WSN have been proposed by introducing three new layers [46]. MAC pool IDs layer, rule-based processing layer and an anomaly detection layer. Simulation of battlefield network shows improvement in detection performance and took measures in improving Quality of Service.

The importance of multi-layer monitoring and cross-layer data analysis has been highlighted [47]. What was explored was various network-layer attacks in WSN, with a focus on detection and consequences for attacks such as selective forwarding, black hole and sink hole. These attacks were implemented using Weighted Shortest Path (WSP) and then, an analysis was made on both victim and sink nodes in different network topologies. It concludes that threshold based IDS may be insufficient and proposes the development of more lightweight anomaly based IDSs.

D. Transport Layer

The main attacks used in the transport layer are flooding attacks and de-syncronization attacks [38]. Flooding attacks

are a kind of Denial of Service (DoS) attack, where normal communication is disrupted by transmission of a large amount of unnecessary traffic and occupying network resources, effectively inhibiting regular communication. For the transport layer, the TCP and UDP protocol are commonly used, which contain a number of security vulnerabilities, such as the previously mentioned flooding attack, and TCP prediction attack [48]. A new multi-layered focus IDS has been introduced with a focus on DoS attacks [49]. It uses anomaly based techniques to alert for excessive traffic for identifying malicious nodes. The proposed framework shows effectiveness in identifying different DoS attacks, and also other attacks in different layers.

E. Application Layer

The application layer is where services are provided and data interpretation is made and supplied to a user. Threats at this layer can can disrupt these services, expose and compromise sensitive data, and and render networks to be ineffective. Common attacks include data aggregation, replay attacks, DoS attacks and cross-layer attacks. An IDS for WSN was proposed for Wi-Fi based environments with a machine-learning based approach [50]. The threats that the IDS focuses on are flooding attacks, injection attacks and impersonation attacks. It employs a number of techniques for adressing the threats. These include using CNNs, Deep Neural Networks (DNNs) and Recurrent Neural Networks with Long Short-Term Memory (RNN-LSTM). Additionally, they used Aegan Wi-Fi Intrusion Dataset (AWID) which is a publically available dataset, labeled with normal and malicious real-world traffic. This dataset was reduced and refined from 154 original features, down to 13 features. With this, the CNN model was able to achieve an accuracy of 97% while maintaining low false positive rates, demonstrating improvements in detection capabilities. A machine-learning based attack detection method in WSN in microgrids, with focus on data integrity in smart meter data has been presented [51]. An anomaly based detection framework is used with Prediction Intervals (PIs) constructed through Lower Upper Bound Estimation (LUBE) with further enhancements through neural networks. A modified Symbiotic Organisms Search (MSOS) is then used to optimize the parameters for the neural network to address the complex non-linear data for the microgrids. Simulations based on microgrid datasets, show effectiveness in attack detection on varying levels of severity based on performance metrics. These metrics are measures through detection rates and confusion matrices. For high severity levels (above 60% data injection severity), detection rates reach 98%, proving its effectiveness.

F. Network security

As the number of edge devices in a cloud grows, it becomes more and more important to perform the majority of the computing close to the edge devices in order to avoid congestion at the network core, where resources are limited. One architecture that is able to achieve this is MEC [52]. Since the architecture includes devices and systems that are layered and interconnected it does however highlight some of the security implications associated with networking. The first step to combat these security implications could be to divide the network into different segments, so that any potential damage is contained. A networks main purpose is however to relay some type of information from one node to another. In its purest form this is just raw data that for example could be encoded in bits, radio waves or light. It thus follows that the overall network security heavily depends on how secure the information transfer is.

One way to establish a secure and private transfer of information is end-to-end communication as defined in the X.805 architecture [53]. The most important component of the architecture is the eight security dimensions that should permeate the implementation. These are access control, authentication, non-repudiation, data confidentiality, communication security, data integrity, availability and privacy.

A common acronym when talking about security is the Confidentiality, Integrity and Availability (CIA) triad. Encrypted end-to-end communication aims to achieve confidentiality and integrity, but one could argue that availability is the most important part of a network. One of the most common availability attacks is DoS and its distributed version, DDoS. These attacks aim to overwhelm a system, forcing it to drop connections due to insufficient processing power. There exist multiple methods to mitigate this, i.e. load frequency balancing, Markov processes and event-triggered transmissions [54] to mention a few. The defense strategies come with different perks and drawbacks that might depend on the system model.

The aforementioned defenses are however based on traditional tools and procedures. This results in an attacker having an asymmetric advantage, partly due to the static nature of a network. Moving Target Defense (MTD) [55] could neutralize this by altering the underlying system configurations in real-time, lowering the success rate of attacks.

V. Hardware Security

Even though the security of hardware is outside the scope of this survey its importance should still be mentioned briefly. If the hardware that is used to i.e. collect logs are compromised an adversary could hide within a system indefinitely. It is essentially the equivalent of having a rootkit installed on a workstation, which is very hard to both find and remediate.

However, in most scenarios the hardware itself can actually be monitored. By recording the power consumption of devices in regular intervals one is able to define a normal consumption pattern. This information can, for example, be used to detect if a device has been infected by a Trojan or if it is the victim of a DOS attack [56]. The reason for this is that these types of malware are design to make use of the targets CPU, thus requiring a higher power consumption. A Trojan usually starts background processes that collect and exfiltrate data, and the objective of a DOS attack is to exhaust the victims resources. One of the most important indicators of wether a program is malicious or benign is what system calls it uses. A program is usually comprised of a large amount of these calls and they generate a large amount of data. It is thus hard to analyze them using conventional methods. Hardware, or more specifically, Field-Programmable Gate Array (FPGA) can ease this

problem since they are able to perform orders of magnitude more operation each second than a general purpose CPU [57]. GuardOL is a tool based on this concept. By encoding a machine learning model using multilayer perception in a FPGA a classifier can be trained on the features of system calls. The classifier is the able to classify unknown program samples as malicious or benign early in their execution with high accuracy.

VI. Discussion

During this chapter we will discuss some of the challenges the systems faces. We will discuss what techniques can be used, in addition to their perks and drawbacks. What ethical and privacy related concerns need to be taken into consideration. Finally we will shed some light on future trends in threat detection.

1. *Main current Challenges*
   The challenges these type of systems faces are mostly pertained to cost and scalability, it is also the main reason why this topic is hard develop. In order to scale up a system there is a complexity concern that will increase development and upkeep costs. The scaling issues IoT has results in a concern regaining how well the edge nodes are able to handle, process and offload the large amount of data generated to the cloud.
   Since monitoring is highly dependent on specific architectures there seems to be somewhat of a gap in the research pertaining to this area. This might be the result of researchers being unwilling to invest in real hardware or creating simulated environments. On the other hand is might be that monitoring is a too generalized term that has multiple definitions depending on specific scenarios. By integrating IDS's to the network environment we also introduce new challenges to the system. Because different types of IDS's have unique weakness and strengths depending on which environment they are placed in. The compatibility issues of the IDS's in the environment introduces an increased level of false positive and false negatives which undermine trust in the system and overwhelm security teams with unnecessary alerts. This type of challenges presents the need of more sophisticated algorithms and adaptive learning systems that can dynamically refine detection thresholds based on evolving network behaviors. Another important aspect that is importance of sufficient resources at the edge devices in IoT environment. IoT devices usually lack computational power and memory in order to perform advanced IDS functionality. For that reason lightweight anomaly detections use of federated learning models could be a solution for those problems. But at the same time that type of solutions have its own trade-offs, such as increased deployment complexity and potential privacy concerns due to distributed data processing. On the other hand deploying IDS's in cloud-edge environment raises multiple questions regarding the latency and real-time processing. The huge amount of data generated by interconnected devices can overwhelm traditional systems making the real-time analysis task difficult to achieve. Furthermore alarm generation seems to mostly be based on ML models that comes with privacy issues that are not negligible. If these are to be prevented the F1 score rapidly decreases, similar to the IDS case.
2. *Trade-offs for deploying an ecosystem that can monitoring automatically, detect threats and generate alarms*
   Since the cloud and IoT is constantly evolving there is no clear method for designing and implementing an ecosystem that can detect threats through monitoring and alarm generation. Edge-computing is still a relatively recent architecture intended to move both data and processing power closer to the end user. Thus a few techniques are still actively competing against each other for the title of being the default in their specific areas. In light of this we have listed the most promising techniques in Table I along with their impact, robustness and complexity.
3. *Outlook on privacy and ethical concerns*
   More and more edge users are becoming concerned with keeping their actions, location and identity confidential. By continuously monitoring the edge devices special care has to be taken in order to not accidentally expose the users private information [58]. A study from 2018 [59] highlighted that almost 90% of IoT devices gathered personal information about user in one way or another. The same paper also concluded that there are issues with the integrity of the devices and their software, furthermore the communication protocols are not perfect either. Some network protocols do however protect against information leakage, but if a device is accepted in the network then a privacy concern is raised [10]. Additionally a network can make use of trust variables, forcing less trusted devices to not be used [14].
   The main issue still remains though, an edge user loses all control of its data the moment it is outsourced to an edge node or a cloud server for data storage and analysis. Thus we would like to make sure that sensitive data is sufficiently obscured. The issue is that this is not always feasible. As an example Homomorphic encryption [60] could be used in theory, but it increases the processing requirement remarkably.
   Given the preceding reasoning it is questionable wether it is ethically sound to collect and process user data if there are no guarantees that the data will remain private and outside of an adversaries reach. We thus leave the reader to infer the answer to this question themselves.
4. *Future trends*
   As the IoT and cloud continues to grow we believe that one of the more promising technologies are Semantic Aware Security. If a system was able to incorporate an understanding of the context, and by extension the meaning, of collected log data and user behavior it would become significantly better at detecting malicious actions against itself. It might even be able to successfully defend against Advanced Persistent Threats (APTs) that influential organizations spend enormous amounts of money to develop. In the subsequent list are a few selected techniques that leverages semantics to improve

the security of a system.

- Using Specific Semantic Search [61] the detection of and localization precision of attacks can be enhanced by correlating the output of i.e. collected logs with the execution of simulated attack scenarios .
- Variations of a Graph Neural Network (GNN) [62] can be used to learn the semantics of vulnerable code by capturing structured information about the context across multiple logged statements. A GNN can also be used to secure Industrial Control Systems that are targeted by attacks relying on sophisticated inherent contextual semantics [63], for example data associativity.
- A framework that includes a semantic analysis [64] could be used to extract mutual information from IoT devices in a smart home to eliminate noise interference. Furthermore a tool, IoTSeeker [65], has been developed that takes advantage of semantics in order to identify cross-platform IoT binary vulnerabilities.

TABLE I
COMPARISON OF SECURITY TECHNIQUES BY AREA
Note: Our own thought are denoted by an asterisk (*)

Area	**Technique**	**Impact**	**Robustness**	**Complexity**
Alarm Generation	Deployment and Exception Tracking	Moderate; Reduces false positives by correlating anomalies with recent deployments in addition to centralizing exceptions [30]	High; There are no guarantees that a deployment caused an anomaly*	Low; There are minimal overhead associated with storing the deployment history and an exception log [30]
Alarm Generation	CNN, DP, Distributed ML & FL	High; Significantly improves detection of malicious behavior, DP and FL can also preserve a users privacy [31],[32],[33],[34]	Moderate; The models could leak sensitive data and generate misleading alarms [31],[32],[33],[34]	High; A lot of data has to be processed in order to train a sound model [31],[32],[33],[34]
Application Layer	ML IDS with CNN, DNN, RNN-LSTM, Anomaly detection with PIs enhanced and neural networks optimized by MSOS for microgrid data	High CNN detection rate (up to 98%) [50], Improved detection and reduced false positives	Effective for maintaining data integrity [51]	Simplifies dataset used for models from 154 to 13 features [50]
Cloud-Edge Systems	Dynamic offloading to the cloud and locally	High; improves latency and processing efficiency, scales effectively [6]	High; robust encryption and trust [9]	High; advanced infrastructure and maintenance but cost effective [6],[9]
Intrusion Detection	Anomaly-based IDS	Effective for detecting unknown and zero-day threats*	High [28]	High [26],[22],[28]
Intrusion Detection	Collaborative IDS (IDN)	Improves detection through distributed and peer-to-peer consultations*	Moderate*	High; Generate large communication overhead [17]
Intrusion Detection	Signature-based IDS	Reliable for detecting known threats*	Low; Depends on when last updated [22]	Low/Predictable [21]
IoT Security	Zigbee and LoRaWAN protocols	Zigbee: Good for short-range applications. LoRaWAN: Effective for long-range, off-grid devices. [5],[4]	moderate; trust issues in the network and encryption improvements [11],[10]	Low; lightweight protocol in implementation and operational cost[11],[10]
Link Layer	Hash chain-based authentication, Symmetric encryption against denial of sleep attacks	Preserve energy by preventing node being in active state unnecessarily [44]	Low*; Uses deprecated encryption standards [44]	Low; Uses lightweight cryptography [44]
Multi-layer Security	Cross-layer data fusion , Data aggregation and selective data transmission	Enhances anomaly detection by correlating data from different protocol layers [40], Optimize data workload to reduce data flood risks [39]	High; Depends on enough training data, but is adaptable with minimal FP [39]	Moderate; Reduce local demand but introduce added architectural complexity [39]
Network Layer	Algorithm for black hole detection in LEACH, Rule-based processing and anomaly-based detection, Multi-layer monitoring and cross-layer analysis	Evaluation of metrics and impact on performance [45], Improves detection performance and QoS, Highlights need for lightweight anomaly-based IDS's	Propose development of anomaly-based IDS's [45], Distributed schemes offer better resiliency over centralized schemes [46]	Additional layers may add complexity [47]
Network Security	MEC	Moderate; Reduces latency and traffic congestion in addition to enabling cloud-offloading [9], [52]	Not addressed in current literature	Moderate; It relies on diverse and possible vulnerable nodes with different configurations*
Network Security	MTD	Moderate; Alters the underlying system configurations in real-time to prevent attacks [55]	Moderate to High; MTDs rely on metrics like CVSS, which can lead to sub-optimal performance against zero-day attacks [55]	High; The attack representations require a lot of processing power, some are NP-Hard [55]
Physical Layer	ACTC (jamming and replay attacks), Hybrid NCD schemes, Decentralized nonce-based authentication (replication attacks)	High detection rates (up to 100% for certain bandwidths), Reduces false positives and overhead [41]	Higher effectiveness for lower bandwidths (100-250KHz) and less effective for higher (500KHz) [41], Distributed schemes have higher resiliency and lower false positives [42]	More energy efficient with less overhead [41] , Threshold schemes for detection (doubling and incremental) [43]
Transport Layer	Anomaly-based IDS with focus on DoS attacks	Effective in multi-layer DoS attack detection [49]	Effective for different types of DoS attacks [49]	Adaptable to different DoS attacks [49]
Trust Models	Dynamic scoring systems	High; Enhances reliability by reducing trust in potentially compromised devices [14]	Low to Moderate; a lot of workarounds [14]	Low; simple setup [14]

ACKNOWLEDGMENTS

Firstly we recognize that our survey is not exhaustive as there most definitely exists a large amount of equally applicable research that was not included in this paper. Naturally there are relevant techniques that we have not touched upon, and some of the presented one might have been deemed ineffective.
We are however still of the opinion that we have captured the most important information and that this work could serve as a stating point for designing and implementing an ecosystem intended to detect threats close to the cloud edge.
Finally we would like to express our gratitude to our supervisor Nikolaos Pappas for his uninterrupted guidance.

REFERENCES

[1] B. N. Shaker, B. Q. Al-Musawi, and M. F. Hassan, "A comparative study of ids-based deep learning models for iot network," in *Proceedings of the 2023 International Conference on Advances in Artificial Intelligence and Applications*, ser. AAIA '23. New York, NY, USA: Association for Computing Machinery, 2024, p. 15–21. [Online]. Available: https://doi-org.e.bibl.liu.se/10.1145/3603273.3635058

[2] M. B. Yassein, M. Q. Shatnawi, S. Aljwarneh, and R. Al-Hatmi, "Internet of things: Survey and open issues of mqtt protocol," in *2017 International Conference on Engineering & MIS (ICEMIS)*, 2017, pp. 1–6.

[3] J. A. Stankovic, "Wireless sensor networks," *Computer*, vol. 41, no. 10, pp. 92–95, 2008.

[4] LoRa Alliance, "What is lorawan?" LoRa Alliance, White paper, 2017, retrieved November 18, 2024. [Online]. Available: https://resources.lora-alliance.org/document/what-is-lorawan

[5] C. M. Ramya, M. Shanmugaraj, and R. Prabakaran, "Study on zigbee technology," in *2011 3rd International Conference on Electronics Computer Technology*, vol. 6, 2011, pp. 297–301.

[6] J. Pan and J. McElhannon, "Future edge cloud and edge computing for internet of things applications," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 439–449, 2018.

[7] W. Meng, Y. Wang, W. Li, Z. Liu, J. Li, and C. W. Probst, "Enhancing intelligent alarm reduction for distributed intrusion detection systems via edge computing," in *Information Security and Privacy*, W. Susilo and G. Yang, Eds. Cham: Springer International Publishing, 2018, pp. 759–767.

[8] B. Mao, J. Liu, Y. Wu, and N. Kato, "Security and privacy on 6g network edge: A survey," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1095–1127, 2023.

[9] G. Nencioni, R. G. Garroppo, and R. F. Olimid, "5g multi-access edge computing: A survey on security, dependability, and performance," *IEEE Access*, vol. 11, pp. 63 496–63 533, 2023.

[10] S. Khanji, F. Iqbal, and P. Hung, "Zigbee security vulnerabilities: Exploration and evaluating," in *2019 10th International Conference on Information and Communication Systems (ICICS)*, 2019, pp. 52–57.

[11] X. Yang, E. Karampatzakis, C. Doerr, and F. Kuipers, "Security vulnerabilities in lorawan," in *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*, 2018, pp. 129–140.

[12] O. Seller, "Lorawan security," *Journal of ICT Standardization*, vol. 9, no. 1, pp. 47–60, 2021.

[13] Y. Ren, W. Liu, T. Wang, X. Li, N. N. Xiong, and A. Liu, "A collaboration platform for effective task and data reporter selection in crowdsourcing network," *IEEE Access*, vol. 7, pp. 19 238–19 257, 2019.

[14] T. Wang, L. Qiu, A. K. Sangaiah, A. Liu, M. Z. A. Bhuiyan, and Y. Ma, "Edge-computing-based trustworthy data collection model in the internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4218–4227, 2020.

[15] K. K. . M. Souppaya, "Guide to computer security log management," https://doi.org/10.6028/NIST.SP.800-92, 2006, [Accessed 01-12-2024].

[16] G. Aceto, A. Botta, W. de Donato, and A. Pescapè, "Cloud monitoring: A survey," *Computer Networks*, vol. 57, no. 9, pp. 2093–2115, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128613001084

[17] C. J. Fung and R. Boutaba, "Design and management of collaborative intrusion detection networks," in *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, 2013, pp. 955–961.

[18] A. Tundo, M. Mobilio, O. Riganelli, and L. Mariani, "Monitoring probe deployment patterns for cloud-native applications: Definition and empirical assessment," *IEEE Transactions on Services Computing*, vol. 17, no. 4, pp. 1636–1654, July 2024.

[19] N. Zhao, H. Wang, Z. Li, X. Peng, G. Wang, Z. Pan, Y. Wu, Z. Feng, X. Wen, W. Zhang, K. Sui, and D. Pei, "An empirical investigation of practical log anomaly detection for online service systems," in *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2021. New York, NY, USA: Association for Computing Machinery, 2021, p. 1404–1415. [Online]. Available: https://doi-org.e.bibl.liu.se/10.1145/3468264.3473933

[20] Elastic, "Elastic docs - fleet and elastic agent guide [8.16]," https://www.elastic.co/guide/en/fleet/current/fleet-overview.html, 2024, [Accessed 24-11-2024].

[21] J. Lansky, S. Ali, M. Mohammadi, M. K. Majeed, S. H. T. Karim, S. Rashidi, M. Hosseinzadeh, and A. M. Rahmani, "Deep learning-based intrusion detection systems: A systematic review," *IEEE Access*, vol. 9, p. 101574–101599, 2021. [Online]. Available: http://dx.doi.org/10.1109/ACCESS.2021.3097247

[22] W. Li, W. Meng, and L. F. Kwok, "Surveying trust-based collaborative intrusion detection: State-of-the-art, challenges and future directions," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, p. 280–305, 2022. [Online]. Available: http://dx.doi.org/10.1109/COMST.2021.3139052

[23] M. Mardi and M. R. Keyvanpour, "Gbkm: A new genetic based k-means clustering algorithm," in *2021 7th International Conference on Web Research (ICWR)*, 2021, pp. 222–226.

[24] M. Zhang and H. Liao, "Privacy-preserving dbscan clustering algorithm based on negative database," in *2020 5th IEEE International Conference on Big Data Analytics (ICBDA)*, 2020, pp. 209–213.

[25] A. Takai, N. Yamai, and R. Nakagawa, "Fast blocking of malicious traffic by excluding benign flow monitoring in ids/sdn cooperative firewall systems," in *Proceedings of the 17th Asian Internet Engineering Conference*, ser. AINTEC'22, vol. 118. ACM, Dec. 2022, p. 62–69. [Online]. Available: http://dx.doi.org/10.1145/3570748.3570757

[26] A. Al-Bakaa and B. Al-Musawi, "A new intrusion detection system based on using non-linear statistical analysis and features selection techniques," *Computers & Security*, vol. 122, p. 102906, Nov. 2022. [Online]. Available: http://dx.doi.org/10.1016/j.cose.2022.102906

[27] A. Popuri and J. Miller, "Generative adversarial networks in image generation and recognition," in *2023 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 2023, pp. 1294–1297.

[28] J. a. Costa, F. Apolinário, and C. Ribeiro, "Argan-ids: Adversarial resistant intrusion detection systems using generative adversarial networks," in *Proceedings of the 19th International Conference on Availability, Reliability and Security*, ser. ARES '24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: https://doi-org.e.bibl.liu.se/10.1145/3664476.3669928

[29] P. Gao, F. Shao, X. Liu, X. Xiao, Z. Qin, F. Xu, P. Mittal, S. R. Kulkarni, and D. Song, "Enabling efficient cyber threat hunting with cyber threat intelligence," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, Apr. 2021. [Online]. Available: http://dx.doi.org/10.1109/ICDE51399.2021.00024

[30] C. Albuquerque and F. F. Correia, "Deployment tracking and exception tracking: monitoring design patterns for cloud-native applications," in *Proceedings of the 28th European Conference on Pattern Languages of Programs*, ser. EuroPLoP '23. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: https://doi.org/10.1145/3628034.3628038

[31] M. Abdelsalam, R. Krishnan, Y. Huang, and R. Sandhu, "Malware detection in cloud infrastructures using convolutional neural networks," in *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, July 2018, pp. 162–169.

[32] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 308–318. [Online]. Available: https://doi-org.e.bibl.liu.se/10.1145/2976749.2978318

[33] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, and J. S. Rellermeyer, "A survey on distributed machine learning," *ACM Comput. Surv.*, vol. 53, no. 2, Mar. 2020. [Online]. Available: https://doi-org.e.bibl.liu.se/10.1145/3377454

[34] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, and Y. Zhou, "A hybrid approach to privacy-preserving federated

learning," in *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, ser. AISec'19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1–11. [Online]. Available: https://doi-org.e.bibl.liu.se/10.1145/3338501.3357370

[35] L. Huang, S. Hall, F. Shao, A. Nihar, V. Chaudhary, Y. Wu, R. French, and X. Xiao, "System-auditing, data analysis and characteristics of cyber attacks for big data systems," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, ser. CIKM '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 4872–4876. [Online]. Available: https://doi-org.e.bibl.liu.se/10.1145/3511808.3557185

[36] Elastic, "Elastic docs - kibana guide [8.16]," https://www.elastic.co/guide/en/kibana/current/introduction.html, 2024, [Accessed 24-11-2024].

[37] ——, "Elastic docs - elastic security solution [8.16] - detections and alerts," https://www.elastic.co/guide/en/security/current/about-rules.html, 2024, [Accessed 24-11-2024].

[38] H. Xie, Z. Yan, Z. Yao, and M. Atiquzzaman, "Data collection for security measurement in wireless sensor networks: A survey," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2205–2224, 2019.

[39] Y. Wang, W. Meng, W. Li, Z. Liu, Y. Liu, and H. Xue, "Adaptive machine learning-based alarm reduction via edge computing for distributed intrusion detection systems," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 19, p. e5101, 2019, e5101 cpe.5101. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.5101

[40] G. Li, Z. Yan, Y. Fu, H. Chen, and J. Díaz-Verdejo, "Data fusion for network intrusion detection: A review," *Sec. and Commun. Netw.*, vol. 2018, Jan. 2018. [Online]. Available: https://doi.org/10.1155/2018/8210614

[41] M. Monjur and Q. Yu, "Advanced continuous-time convolution framework for security assurance in wireless sensor networks," in *Proceedings of the Great Lakes Symposium on VLSI 2024*, ser. GLSVLSI '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 319–322. [Online]. Available: https://doi.org/10.1145/3649476.3658785

[42] M. Thaile and O. Ramanaiah, "Node compromise detection based on nodetrust in wireless sensor networks," in *2016 International Conference on Computer Communication and Informatics (ICCCI)*, 2016, pp. 1–5.

[43] T. Dimitriou, E. A. Alrashed, M. H. Karaata, and A. Hamdan, "Imposter detection for replication attacks in mobile sensor networks," *Computer Networks*, vol. 108, pp. 210–222, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128616302717

[44] C.-T. Hsueh, C.-Y. Wen, and Y.-C. Ouyang, "A secure scheme against power exhausting attacks in hierarchical wireless sensor networks," *IEEE Sensors Journal*, vol. 15, no. 6, pp. 3590–3602, 2015.

[45] V. Bansal and K. K. Saluja, "Anomaly based detection of black hole attack on leach protocol in wsn," in *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2016, pp. 1924–1928.

[46] N. M. Alajmi and K. Elleithy, "A new approach for detecting and monitoring of selective forwarding attack in wireless sensor networks," in *2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, 2016, pp. 1–6.

[47] C. Ioannou and V. Vassiliou, "The impact of network layer attacks in wireless sensor networks," in *2016 International Workshop on Secure Internet of Things (SIoT)*, 2016, pp. 20–28.

[48] P. Sinha, V. K. Jha, A. K. Rai, and B. Bhushan, "Security vulnerabilities, attacks and countermeasures in wireless sensor networks at various layers of osi reference model: A survey," in *2017 International Conference on Signal Processing and Communication (ICSPC)*, 2017, pp. 288–293.

[49] B. Santhosh Kumar and S. Sinha, "An intrusion detection and prevention system against dos attacks for internet-integrated wsn," in *2022 7th International Conference on Communication and Electronics Systems (ICCES)*, 2022, pp. 793–797.

[50] H. Sadia, S. Farhan, Y. U. Haq, R. Sana, T. Mahmood, S. A. O. Bahaj, and A. R. Khan, "Intrusion detection system for wireless sensor networks: A machine learning based approach," *IEEE Access*, vol. 12, pp. 52 565–52 582, 2024.

[51] A. Kavousi-Fard, W. Su, and T. Jin, "A machine-learning-based cyber attack detection model for wireless sensor networks in microgrids," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 1, pp. 650–658, 2021.

[52] B. Ali, M. A. Gregory, and S. Li, "Multi-access edge computing architecture, data security and privacy: A review," *IEEE Access*, vol. 9, p. 18706–18721, 2021. [Online]. Available: http://dx.doi.org/10.1109/ACCESS.2021.3053233

[53] ITU-T, "Security architecture for systems providing end-to-end communications," https://www.itu.int/rec/T-REC-X.805-200310-I/en, 2003, [Accessed 24-11-2024].

[54] W. Duo, M. Zhou, and A. Abusorrah, "A survey of cyber attacks on cyber physical systems: Recent advances and challenges," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 5, pp. 784–800, May 2022.

[55] S. Sengupta, A. Chowdhary, A. Sabur, A. Alshamrani, D. Huang, and S. Kambhampati, "A survey of moving target defenses for network security," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1909–1941, thirdquarter 2020.

[56] H. Mohammed, F. Khalid, P. Sawyer, G. Cataloni, and S. R. Hasan, "Intrust-iot: Intelligent ecosystem based on power profiling of trusted device(s) in iot for hardware trojan detection," in *Proceedings of the 10th International Workshop on Hardware and Architectural Support for Security and Privacy*, ser. HASP '21. New York, NY, USA: Association for Computing Machinery, 2022. [Online]. Available: https://doi-org.e.bibl.liu.se/10.1145/3505253.3505262

[57] S. Das, Y. Liu, W. Zhang, and M. Chandramohan, "Semantics-based online malware detection: Towards efficient real-time protection against malware," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 2, pp. 289–302, 2016.

[58] A. Alwarafy, K. A. Al-Thelaya, M. Abdallah, J. Schneider, and M. Hamdi, "A survey on security and privacy issues in edge-computing-assisted internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4004–4022, March 2021.

[59] I. Makhdoom, M. Abolhasan, J. Lipman, R. P. Liu, and W. Ni, "Anatomy of threats to the internet of things," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1636–1675, Secondquarter 2019.

[60] D. Liu, Z. Yan, W. Ding, and M. Atiquzzaman, "A survey on secure data analytics in edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4946–4967, June 2019.

[61] D. Zhang, Q. Zhao, Y. Wang, S. Li, Z. Tian, and Y. Sun, "Attack-aware capability assessment method based on specific semantic search," in *2024 6th International Conference on Electronic Engineering and Informatics (EEI)*, June 2024, pp. 1531–1535.

[62] S. Cao, X. Sun, L. Bo, R. Wu, B. Li, X. Wu, C. Tao, T. Zhang, and W. Liu, "Learning to detect memory-related vulnerabilities," *ACM Trans. Softw. Eng. Methodol.*, vol. 33, no. 2, Dec. 2023. [Online]. Available: https://doi-org.e.bibl.liu.se/10.1145/3624744

[63] S. L(y)u, K. Wang, Y. Wei, H. Liu, Q. Fan, and B. Wang, "Gnn-based advanced feature integration for ics anomaly detection," *ACM Trans. Intell. Syst. Technol.*, vol. 14, no. 6, Nov. 2023. [Online]. Available: https://doi-org.e.bibl.liu.se/10.1145/3620676

[64] K. Li, Z. Li, Z. Gu, J. Guo, Z. Wang, and L. Sun, "Compromised iot devices detection in smart home via semantic information," in *ICC 2022 - IEEE International Conference on Communications*, May 2022, pp. 4986–4992.

[65] J. Gao, X. Yang, Y. Jiang, H. Song, K.-K. R. Choo, and J. Sun, "Semantic learning based cross-platform binary vulnerability search for iot devices," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 971–979, Feb 2021.