CMPE 491

HIGH-LEVEL DESIGN REPORT

Ahmad Ismail 99000332006

Ateş Öztürk 21145665572

Esra Gürgen 40654688056

1. Introduction

1.1 Purpose of the system

The purpose of the IoT Intrusion Detection System (IDS) is to address serious cybersecurity problems in IoT environments. It uses advanced machine learning techniques and real-time monitoring. The system's primary objectives include:

1. **Improving Security in IoT Settings:** IoT devices are used in several areas such as healthcare, smart homes, industrial automation and transport. This feature provides ease of use and effectiveness. However, it also brings unique security weaknesses. Traditional cybersecurity methods cannot fully cover such risks. Therefore, the IDS project aims to provide powerful, flexible and innovative protection methods specifically for IoT networks.
2. **Utilizing AI-Driven Threat Detection**: The IDS will be able to identify threats that are known and unknown. While doing it, it will use two types of learning. These are supervised and unsupervised machine learning algorithms. With supervised learning, the IDS can analyze existing threat patterns. Also techniques like anomaly detection in unsupervised learning help detect unusual behaviors that could indicate new or emergent threats.
3. **Real-Time Monitoring and Response**: The IDS constantly monitors network traffic in real time. It allows to quickly detect and respond to security issues. This ability is very important in IoT environments since early and quickly detecting and intervention threats can stop damage, as well as protect confidential data.
4. **Minimizing False Positives**: The system aims to prevent harmless activities from being unnecessarily flagged as threats. To do this, it optimizes machine learning algorithms and uses advanced analysis tools. Thus, it will reduce the number of false alarms. In this way, it allows the security team to focus on real threats.
5. **Scalability and Adaptability:** The IDS will be created to adapt efficiently with different sizes and setups of IoT networks. It will support various communication protocols used in IoT devices such as MQTT, CoAP, HTTP, Zigbee and Bluetooth. This guarantees full and successful threat detection across diverse IoT systems.
6. **User-Friendly Interface**: The system will have an intuitive graphical user interface (GUI). Administrators and network operators can use it easily. This interface will clearly visualize security statuses, alerts and system configurations. And it facilitates monitoring and management of IDS (Intrusion Detection System). Thus, there will be no need to have cybersecurity expertise.
7. **Automated Incident Response**: When threats are detected, the system automatically will take predefined actions. Such as alert notifications to administrators, blocking suspicious

IP addresses and isolation of compromised devices. These automatic responses will reduce the damage of security incidents and prevent potential vulnerabilities within the IoT network.

8. **Compliance with Regulatory Standards**: The IDS will be created according to global cybersecurity standards and regulations such as GDPR (General Data Protection Regulation) and NIST (National Institute of Standards and Technology). Thus, the system will operate ethically and legally. It will protect user privacy and maintain trust in the handling of sensitive information.

1.2 Design Goals

The IoT Intrusion Detection System (IDS) project has design goals. These goals are determined to ensure cybersecurity management in IoT environments, increase operational efficiency and reinforce user trust. The details are:

1. **Accurate Intrusion Detection:**
    - Development of AI-Driven System: It is planned to use advanced machine learning algorithms to detect and classify security threats in IoT networks. It will use supervised learning methods to identify known threat patterns and unsupervised learning-based anomaly detection for previously unidentified threats.
    - Minimal False Positives: It is aimed to prevent real threats from being confused with unnecessary warnings. For this, it will be optimized machine learning models and algorithms. In this way, false alarms will be minimized.
2. **Real-Time Monitoring:**
    - Continuous Monitoring: It is aimed at The IDS will watch the network traffic all the time. This helps detect and stop threats as soon as they happen.
    - Anomaly Detection: The system will use real-time analysis and anomaly detection techniques to be able to quickly notice deviations from normal network behavior and suspicious activities.
3. **Automated Incident Response:**
    - Immediate Actions: It is planned to ensure that the system will automatically act when a thread is found. It can send alerts, block malicious IP addresses or isolate affected devices.
    - Enhanced Security Measures: Automatic countermeasures will be deployed. In this way, this will reduce the impact of incidents and prevent further exploitation of vulnerabilities in IoT devices.

4. **Scalability and Adaptability:**
    - Flexible Deployment: The IDS will be designed to integrate with IoT networks of different sizes and configurations. Thus, it will support several communication protocols such as MQTT, CoAP, HTTP, Zigbee and Bluetooth. Also it will work on various devices without creating any problem.
    - Compatibility: IOT world constantly changes and develops. Therefore, it is aimed to adapt to this change and be flexible.
5. **User-Friendly Interface:**
    - Intuitive GUI: It is aimed at developing a user-friendly graphical interface. In this way, administrators and operators can easily monitor, manage and configure the system.
    - Ease of Use: The interface will be designed in a simple and understandable way. So that users who have different technical knowledge levels can use it easily. In this way, it minimizes the need for cyber security knowledge and it will be accessible for everyone.
6. **Compliance and Ethical Considerations:**
    - Regulatory Compliance: To protect user privacy and ensure lawful data processing practices, compliance with international cybersecurity standards and regulations such as GDPR and NIST will be demonstrated. In this way, user rights will be protected and a reliable structure will be created.
    - Ethical Principles: In the process of developing the system, not only technical but also ethical principles such as transparency, justice and accountability will be considered. AI-supported decision-making and threat detection processes will be managed responsibly.

1.3 Definitions, Acronyms, and Abbreviations

- IDS: Intrusion Detection System
- IoT: Internet of Things
- AI: Artificial Intelligence
- ML: Machine Learning
- GUI: Graphical User Interface
- GDPR: General Data Protection Regulation

- NIST: National Institute of Standards and Technology
- MQTT: Message Queuing Telemetry Transport
- CoAP: Constrained Application Protocol
- HTTP: Hypertext Transfer Protocol
- IP: Internet Protocol
- API: Application Programming Interface

1.4 Overview

This high-level design report presents the architecture and key design components of an AI-enabled IDS developed specifically for IoT environments. The IDS aims to provide solutions to the increasing security problems that arise with the proliferation of IoT devices in areas such as healthcare, smart cities, industrial automation and home networks.

IoT systems are different from traditional IT infrastructures. IoT ecosystems include many types of devices, often have limited resources, and usually work without much human control. These features make insufficient classical security solutions such as firewalls and antivirus software. That is why the system which is developed on scope of the project aims to present a new generation IDS by providing real-time threat detection, automatic incident response and continuous self-improvement capability with using advanced machine learning-based algorithms.

The developed system has a modular structure consisting of many subcomponents such as network traffic monitoring, threat analysis, automatic response mechanisms and user interface. These modules communicate with each other with APIs and work integrated with a central control mechanism. The key features of the system are:

- Real-Time Monitoring of network traffic in the IoT network by supporting different communication protocols such as MQTT, CoAP, HTTP, Zigbee and Bluetooth.
- Hybrid Detection Engine that combines signature-based and anomaly-based detection to address both known and emerging threats.
- Automated Countermeasures such as alert generation, IP blocking and device quarantine. These actions minimize response time and limiting threat impact.
- Scalability to adapt both small-scale local networks and large and complex business-level IoT structures.
- Compliance with global security standards such as GDPR and NIST to ensure safe and ethical processing of user data.
- User-Friendly Interface that allows even non-expert administrators to configure, monitor and manage system operations effectively.

This report covers each critical component of the system like subsystem decomposition, persistent data management strategies, access control mechanisms and handling of boundary conditions such as power loss or connectivity issues.

By implementing this IDS, IoT network operators can achieve enhanced cybersecurity, reduce operational risks, and ensure the uninterrupted, secure functioning of connected devices.

2. Current Software Architecture

Today, most IoT systems still use basic or legacy security methods. These systems usually consist of firewall rules, static access control lists (ACLs) and standard antivirus software. However, these traditional solutions were not designed for the dynamic and complex nature of IoT networks. They are often made up of many small, limited-resource devices.

The main applications commonly seen in current IoT systems are:

- Manual Supervision: Security often relies on administrators manually reviewing logs and system behavior. This method is not scalable and open to error.
- Firewall-Based Filtering: Generally, basic filtering methods based on IP address or port number are used. These systems lack the intelligence to distinguish between malicious and harmless traffic.
- Predefined Rule Sets: Some IDS solutions used work with signature-based fixed rules. Therefore, they cannot detect zero-day attacks or advanced anomalies.
- Lack of Real-Time Analytics: In many cases, data is analyzed after the fact. Due to this situation, detection is delayed and there is not enough time to respond.
- Protocol Limitations: Most existing tools are designed to work well with web protocols like HTTP. IoT-specific protocols such as MQTT, CoAP, or Zigbee are not supported.
- Centralized Architecture: Many traditional systems assume are built with the idea that all data and processing occur in one location either in cloud or on-premises servers. This causes delays and is not appropriate IoT scenarios with low latency edge.

Because of these limitations, many IoT systems are open to threats. They can be attacked in many ways such as device hijacking, data exfiltration, protocol spoofing and unauthorized device access. They lack an integrative smart system that can deliver real-time, automated, and adaptive security customized to the needs of the IoT ecosystem.

3. Proposed Software Architecture

3.1 Overview

The proposed IoT Intrusion Detection System (IDS) will be made with modular and distributed architecture. It will meet the unique challenges of IoT environments. It combines edge

computing, cloud services and AI-based detection engines. It aims to provide a real-time, scalable and adaptable security monitoring infrastructure.

Key characteristics of architecture include:

- Modularity: The system is divided into independent subsystems. For example monitoring, detection, response, and reporting. These parts work together.
- Distributed Processing: Edge nodes such as Raspberry Pi or ESP32 devices perform preliminary data collection and filtering. This helps reduce delays and network load. At the same time, centralized cloud components handle deep data analysis and machine learning model training.
- Protocol Awareness: The architecture supports several IoT communication protocols. This provides that it becomes useful in many kinds of IoT networks.
- Security-First Design: End-to-end encryption, role-based access control and audit logging are embedded into all layers.

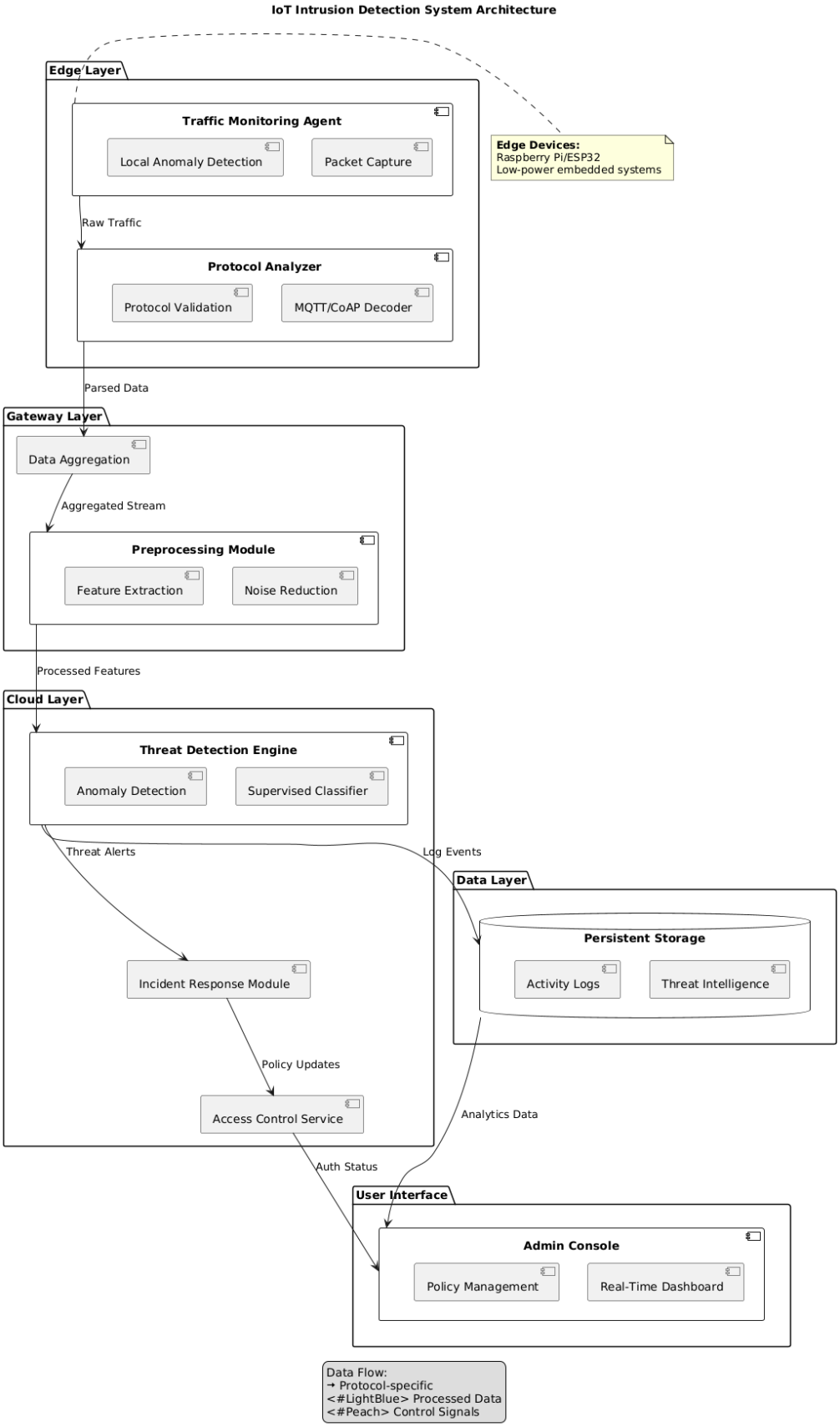
3.2 Subsystem Decomposition

The IoT Intrusion Detection System is made of six main parts. Each part has a specific role and works together with the others.

1. Traffic Monitoring Subsystem: It collects raw network traffic from IoT devices. It works at edge gateways or network sensors. It filters data and performs initial packet inspection.
2. Threat Detection Engine: This engine applies AI/ML models (both supervised and unsupervised) to identify malicious behavior or anomalies. It operates partly on edge devices for low latency detection and partly in the cloud for deep analysis.
3. Incident Response Module: When a threat is found, this module takes automatic actions. For instance, it can quarantine compromised devices, block IP addresses and alert administrators.
4. Protocol Analyzer: It decodes and interprets traffic from MQTT, CoAP, HTTP, Zigbee, Bluetooth and other IoT protocols. Then, it extracts meaningful features for detection.
5. Reporting and Analytics Module: It aggregates logs, generates visual dashboards and produces audit-ready reports for administrators. It helps administrators understand what is happening in the system.
6. User Interface Module: It provides a web-based GUI. Users can make system configuration, real-time monitoring, alert management and forensic analysis.

3.3 Hardware/Software Mapping

- **Hardware Components:**
    - Edge Devices: They are low-power embedded systems like Raspberry Pi or ESP32. They are deployed near IoT networks for traffic capture and preliminary processing.
    - Central Server/Cloud: It hosts ML model training, historical data storage and global system control.
- **Software Components:**
    - Detection Engine: This is the AI part of the system. It is Python-based and runs on edge and cloud.
    - Backend Services: These are background programs that manage data and system actions. Java or Python are used. These microservices manage data flow, incident response, and integration.
    - Frontend: This is the web GUI for users. React or Angular is used. It allows users to watch, control and manage the system easily.
    - Database: The system uses cloud-hosted databases such as MongoDB (NoSQL) or PostgreSQL (SQL). These store logs and configurations.
    - Communication: For internal communication, Secure REST APIs and MQTT are used. This makes sure the system shares data safely and quickly.


IoT Intrusion Detection System Architecture
Edge Layer
Traffic Monitoring Agent
Local Anomaly Detection
Packet Capture
Edge Devices:
Raspberry Pi/ESP32
Low-power embedded systems
Raw Traffic
Protocol Analyzer
Protocol Validation
MQTT/CoAP Decoder
Parsed Data
Gateway Layer
Data Aggregation
Aggregated Stream
Preprocessing Module
Feature Extraction
Noise Reduction
Processed Features
Cloud Layer
Threat Detection Engine
Anomaly Detection
Supervised Classifier
Threat Alerts
Log Events
Data Layer
Persistent Storage
Activity Logs
Threat Intelligence
Incident Response Module
Policy Updates
Access Control Service
Analytics Data
Auth Status
User Interface
Admin Console
Policy Management
Real-Time Dashboard
Data Flow:
→ Protocol-specific
<#LightBlue> Processed Data
<#Peach> Control Signals

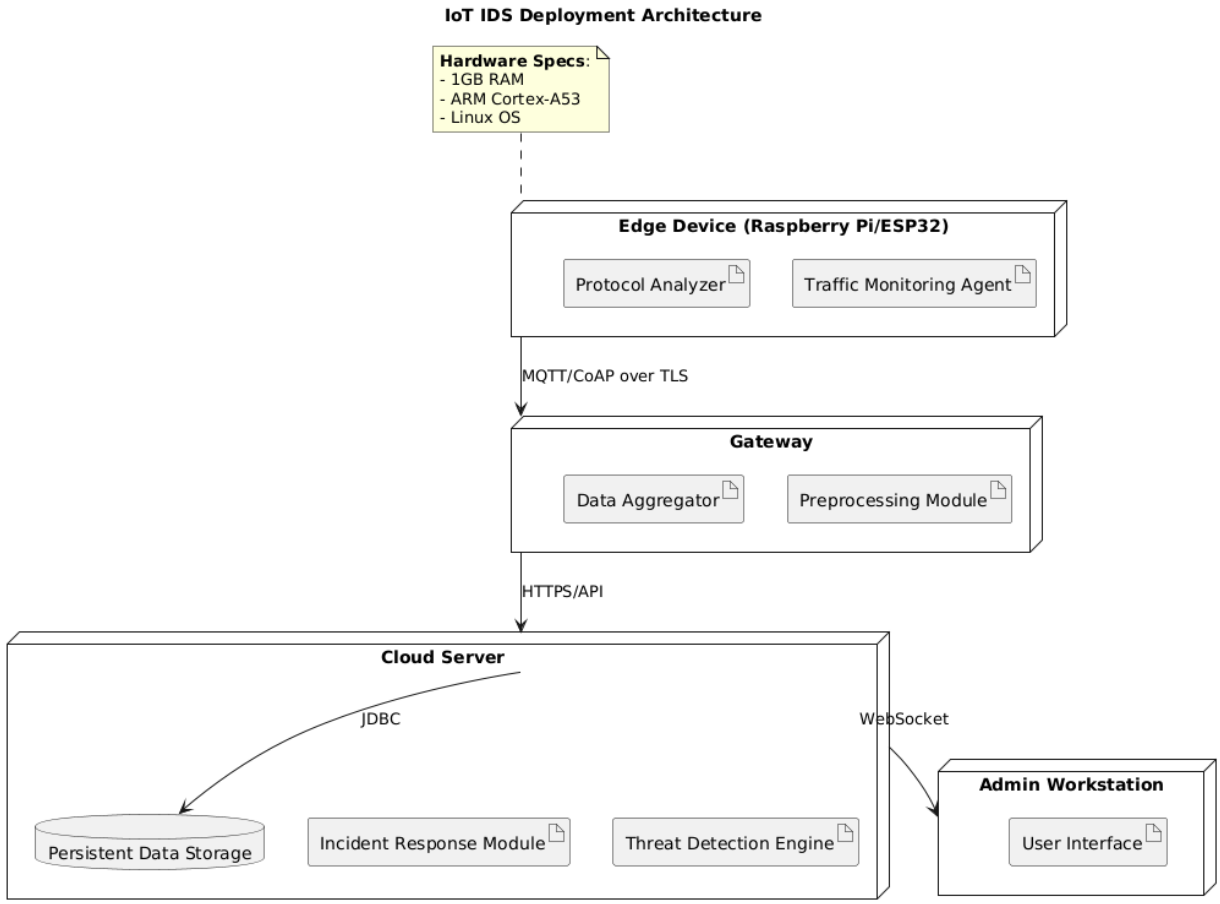


Figure 1 (A & B): UML Deployment Diagram showing how software components are distributed across edge devices, gateways and cloud servers.

3.4 Persistent Data Management

The system stores all important data in cloud databases. It includes raw traffic metadata, detection logs and user activity. Data encryption is applied both at rest (AES-256) and in transit (TLS 1.3). The system also follows data retention policies and backups. Thus, it is ensured that durability and compliance with legal standards.

3.5 Access Control and Security

- Role-Based Access Control (RBAC): The system allows users to have different authorization levels. It is based on their roles and responsibilities. Permission configurations are supported by roles such as administrator, analyst, and auditor.
- Authentication: Multi-factor authentication (MFA) is applied. It is particularly for administrator-level access. This helps to reduce unauthorized access risk.
- Encryption: All data communications are protected with end-to-end encryption protocols. In this way, data will be private and secure during communication.

- Audit Logs: All access and transaction activities on the system are recorded with tamper-proof audit logs for intrusion detection and accountability purposes.

3.6 Global Software Control

There is a centralized control module. It arranges subsystem operations, manages software updates, monitors health/status of components and coordinates anomaly threshold adjustments. It enables dynamic tuning of detection models based on feedback and incident outcomes.

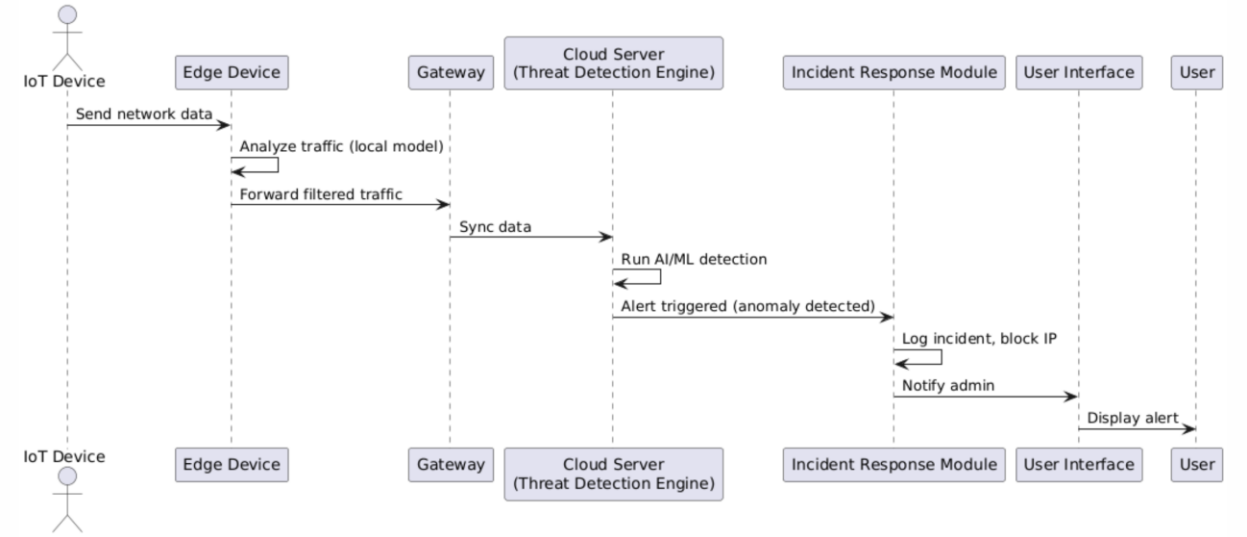


Figure 2: UML Sequence Diagram illustrating the interaction flow when a network anomaly is detected by the IDS.

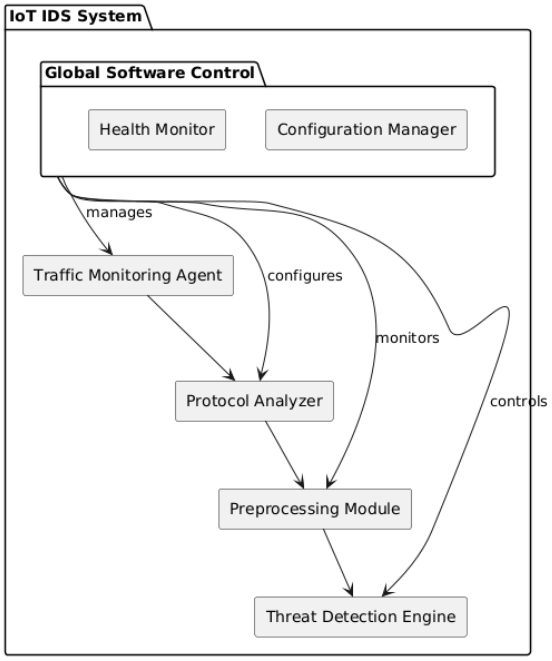


Figure 2: Threat detection workflow sequence

3.7 Boundary Conditions

The system design anticipates and gracefully handles boundary conditions. These are:

- Network Failures: When connectivity restores, edge nodes cache data locally and synchronize with the cloud.
- Power Outages: Critical system states are checkpointed to enable rapid recovery.
- False Positives/Negatives: Continuous feedback mechanisms update ML models to minimize detection errors.
- Device Overload: Load balancing between edge and cloud processing ensures scalability and responsiveness.

4. Subsystem Services

The developed IoT Intrusion Detection System (IDS) is composed of several specialized subsystems. Each subsystem provides distinct services that collectively ensure comprehensive network security. The core services provided by each subsystem are:

4.1 Traffic Monitoring Subsystem Services

- Packet Capture and Filtering: Network traffic from IoT devices is continuously monitored and collected via edge gateways or sensor nodes. Only data that is meaningful for security purposes is processed by filtering according to predefined rules.
- Protocol Parsing: By analyzing IoT-specific communication protocols such as MQTT, CoAP, Zigbee, Bluetooth and HTTP, the necessary metadata and payload information to be used in security analysis are obtained.
- Preprocessing: Raw traffic data is normalized and formatted to reduce unnecessary noise and irrelevant information before it is fed into detection engines.

4.2 Threat Detection Engine Services

- Supervised Threat Classification: It applies trained machine learning models to identify known malicious patterns using labeled datasets.
- Anomaly Detection: By using unsupervised learning and statistical methods, deviations from the normal behavioral limits are detected and the existence of zero-day or unknown threats is understood.
- Real-Time Scoring: Each activity on the network is assigned a threat score. Therefore, it enables prioritization of alerts and response orders.

- Model Retraining and Updating: It supports dynamic model updates based on new data and feedback from incident investigations.

4.3 Incident Response Module Services

- Automated Mitigation: Predefined response steps such as blocking suspicious IP addresses, isolating compromised devices or slowing down suspicious traffic are automatically activated.
- Alert Management: The alerts created are arranged according to priority. Also, they are sent to administrators via different channels such as notifications on the dashboard, e-mail or SMS.
- Incident Logging: Detailed records of all threats detected and interventions performed are kept for use in forensic analysis and compliance reporting.
- Manual Intervention Support: Administrators are given the opportunity to disable automatic responses and when necessary, manually intervene and produce solutions.

4.4 Protocol Analyzer Services

- Protocol Identification: It detects and categorizes traffic according to supported IoT protocols.
- Feature Extraction: It extracts key characteristics such as message frequency, payload size, command types and timing patterns relevant to security analysis.
- Anomaly Flags: Flags unusual protocol usage or unexpected changes like a device suddenly switching protocols.

4.5 Reporting and Analytics Module Services

- Dashboard Visualization: Through customizable dashboards, network health, threat intensity and overall system health are visualized in real time.
- Trend Analysis: By examining historical data, recurring threats or evolving attack patterns are identified.
- Compliance Reporting: Audit-ready reports are created in accordance with regulatory standards such as GDPR and NIST.
- Data Export: Logs and reports for analysis can be exported in different formats such as PDF, CSV, JSON.

4.6 User Interface Module Services

- System Configuration: Enables secure configuration of detection parameters, alert thresholds and response policies.

- User Management: It allows for streamlined management of user accounts, roles and permissions.
- Alert Review: It offers user-friendly interfaces for reviewing and approval. Also, if necessary, escalating alerts to higher authorities.
- Forensic Tools: Supports detailed incident investigations with incident timelines and packet-level review tools.

5. Glossary

- Access Control: The process of granting or denying specific requests to obtain and use information and related information processing services and enter specific physical facilities.
- Anomaly Detection: A method for finding unknown cyber threats by spotting unusual or unexpected behavior in the system.
- API (Application Programming Interface): A set of protocols and tools that provide different software components to communicate with each other.
- Artificial Intelligence (AI): The ability of machines to think, learn and make decisions by imitating human intelligence.
- Authentication: The process of verifying that a user or device is really who they say they are. This step allows to prove who they are before they can log in to the system.
- Authorization: It is the process of determining which resources an authenticated user can access in the system. In other words, it becomes clear at this stage what is allowed and what is not.
- Bluetooth: It is a wireless communication technology that allows data transfer between devices over short distances. It is frequently used between headphones, smart watches or IoT devices.
- Cloud Computing: It is the remote provision of IT services such as servers, storage, databases, software over the Internet. It allows the use of powerful infrastructures without owning the hardware.
- CoAP (Constrained Application Protocol): A web transfer protocol designed for machine-to-machine devices.
- Data Encryption: It is the process of encoding data in a way that cannot be read, making it accessible only to authorized people. It increases security during both transfer and storage.
- False Negative: This is when a security system misses a real threat. That is, the system fails to detect the attack and the threat can move forward.

- False Positive: This is when a harmless movement is perceived as a threat. This can lead to unnecessary alarms and interventions.
- Firewall: It is a software or hardware-based security system that controls network traffic and checks whether incoming and outgoing data complies with the rules.
- GDPR (General Data Protection Regulation): It is a legal framework enacted by the European Union and aims to protect personal data. This regulation regulates the collection, processing and storage of personal information of individuals and requires that this data be protected securely.
- HTTP (Hypertext Transfer Protocol): A protocol for communication between clients and Web servers.
- IoT (Internet of Things): The network of devices that contain the hardware, software, firmware, and actuators which allow the devices to connect, interact, and freely exchange data and information.
- IP (Internet Protocol): A protocol which provides unreliable, connectionless packet delivery for the Internet.
- Intrusion Detection System (IDS): A security service that monitors and analyzes network or system events for the purpose of finding, and providing real-time or near real-time warning of, attempts to access system resources in an unauthorized manner.
- Machine Learning (ML): The development and use of computer systems that adapt and learn from data with the goal of improving accuracy.
- Malware: Types of software written to harm or take over a computer, network or device.
- MQTT: A lightweight publish/subscribe messaging protocol.
- NIST (National Institute of Standards and Technology): A non-regulatory agency that promotes innovation by advancing measurement science, standards and technology.
- Phishing: An attempt to deceive users and obtain sensitive information such as passwords and credit cards through fake e-mails or websites.
- Quarantine: The process of isolating a suspicious or dangerous device to prevent it from spreading to the network.
- RBAC (Role-Based Access Control): It is an access control mechanism that allows users to be authorized through roles, not directly. Each role has certain access rights.
- Signature-Based Detection: It is a method that tries to detect attacks by comparing with previously defined threat patterns.

- SSL/TLS: Communication protocols that encrypt web traffic and secure data transmission between users and servers.
- Threat Intelligence: Information and analysis collected so that institutions can better understand the threats they face and take precautions.
- True Positive: It is the IDS's correct detection of a real threat. In other words, the system has worked successfully.
- Zero-Day Attack: These are attacks that exploit a security vulnerability that has not yet been noticed. Systems usually do not have a patch for this vulnerability.
- Zigbee: A communication protocol that works with low power consumption, provides short-range wireless communication and is used to create networks between IoT devices.

6. References

1. **NIST Cybersecurity Framework**
   National Institute of Standards and Technology.
   https://www.nist.gov/cyberframework
   (Provides guidelines for improving critical infrastructure cybersecurity, including risk management and incident response.)
2. **General Data Protection Regulation (GDPR)**
   European Parliament and Council of the European Union.
   https://gdpr.eu/
   (Regulation on data protection and privacy in the European Union and the European Economic Area.)
3. **IEEE Cybersecurity Standards and Ethics**
   Institute of Electrical and Electronics Engineers.
   https://www.ieee.org/about/corporate/governance/ethics.html
   (Standards and ethical guidelines for cybersecurity and technology development.)
4. **International Business Machines**
   https://www.ibm.com/us-en
   (A global information technology company based in the USA)
5. **Bruegge, B., & Dutoit, A. H. (2004). Object-Oriented Software Engineering Using UML, Patterns, and Java (2nd Edition). Prentice Hall.**
   (Foundational textbook on software engineering principles relevant to system design.)
6. **Cheng, H., et al. (2018). Intrusion Detection for IoT Security: A Review. IEEE Internet of Things Journal.**
   (Comprehensive survey on IDS techniques for IoT systems.)

7. **Feng, D., et al. (2017). Machine Learning for Intrusion Detection in IoT Networks: A Survey. IEEE Communications Surveys & Tutorials.**
(Overview of machine learning applications in IoT security.)

8. **Zhou, W., et al. (2019). A Survey on Intrusion Detection Techniques in the Internet of Things. IEEE Access.**
(Analysis of various intrusion detection methodologies in IoT.)