**COMSATS University Islamabad,   
Abbottabad Campus**

**SOFTWARE DESIGN DESCRIPTION   
(SDD DOCUMENT)**

**for**

**IoT-Based Honeypot Network for Cybersecurity Threat Detection and Attack Analysis**  
Version 1.0

***By***

**Salim Khan CIIT/FA21-BCS-193/ATD**

**Zakirullah Salar CIIT/FA21-BCS-186/ATD**

**Mohammad Zahid Rahmat CIIT/FA21-BCS-222/ATD**

***Supervisor*Sayed Shahab Zarin**

**Table of Contents**

Table of Contents

[Introduction 5](#_Toc186406385)

[System overview 5](#_Toc186406386)

[Architectural design 6](#_Toc186406387)

[Design models 7](#_Toc186406388)

[Data Flow Diagrams: 7](#_Toc186406389)

[Activity Diagram: 16](#_Toc186406390)

[Data Design 19](#_Toc186406391)

[Databases and Data Storage 19](#_Toc186406392)

[Requirements Traceability Matrix 21](#_Toc186406393)

[Human interface design 22](#_Toc186406394)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

**Application Evaluation History**

|  |  |
| --- | --- |
| **Comments (by committee)**  **\*include the ones given at scope time both in doc and presentation** | **Action Taken** |
|  |  |
|  |  |

**Supervised by**

**<Sayed Shahab Zarin>**

Signature\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Introduction

This SDD For the development of the IoT-based honeypot system, the Agile methodology will be adopted. This iterative and adaptive approach ensures continuous development, testing, and feedback integration throughout the project lifecycle. Agile is ideal for this system due to the following reasons:

**Focus on Evolving Threats:**

Agile emphasizes adaptability, allowing the system to respond quickly to new cybersecurity threats and evolving attack vectors.

**Rapid Prototyping and Iteration:**

Functional prototypes of the honeypot can be developed early, enabling thorough testing and refinement based on real-world attack scenarios and feedback.

**Scalability and Flexibility:**

As the system grows, Agile supports incremental enhancements to the architecture, ensuring scalability and alignment with the latest security trends.

**Collaboration with Experts:**

Frequent collaboration with cybersecurity specialists and IoT experts ensures the honeypot’s design remains relevant and effective.

**Continuous Improvement:**

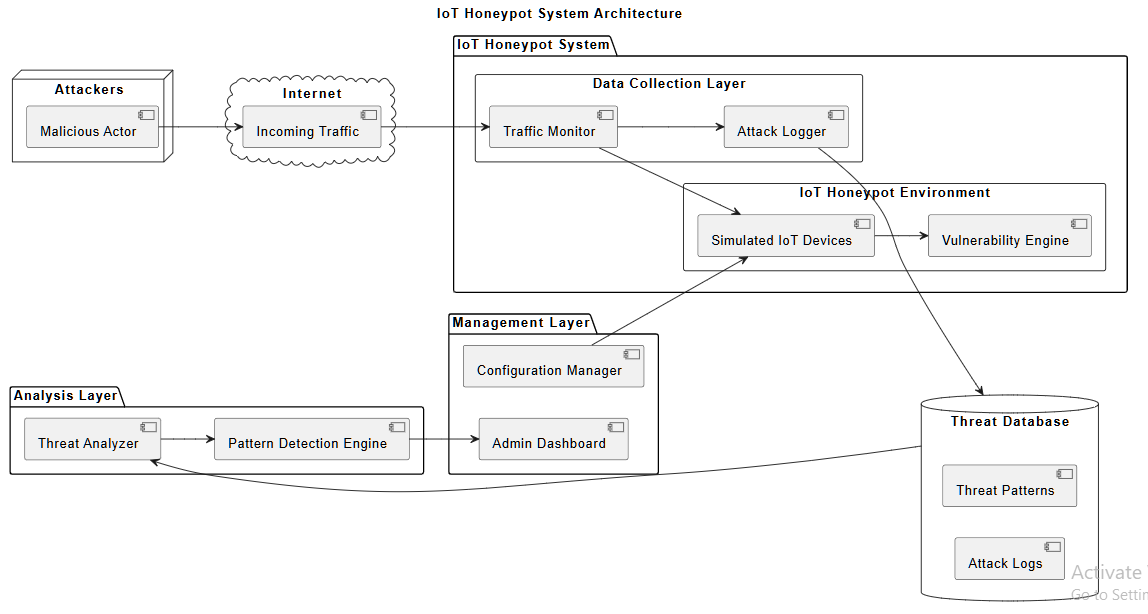
Regular sprints allow the team to incorporate insights from captured data, refine detection capabilities, and enhance the system's functionality iteratively.

# System overview

The IoT Honeypot System is designed to simulate a network of vulnerable IoT devices to attract and study malicious cyberattacks. It captures and analyzes attacker behaviour, providing valuable insights into common attack methods. The system comprises modules for threat detection, data logging, and analysis, enabling continuous monitoring and improvement. Scalable and adaptable, the honeypot supports evolving IoT security needs and informs the development of more effective defence mechanisms.

# Architectural design

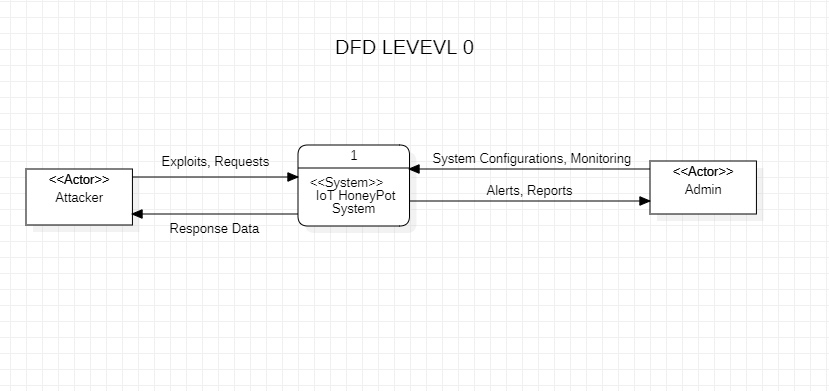
**Architecture Diagram**



## Design models

## Data Flow Diagrams:

**DFD LEVEL 0:**



**Description:**

The Level 0 DFD provides an overview of the IoT Honeypot System as a single process with its external entities. It highlights the interactions between attackers and administrators with the system, showing the main input and output data flows. Attackers interact with the system by attempting to exploit vulnerabilities, while administrators manage the system by configuring simulations, monitoring activity, and reviewing reports. The IoT Honeypot System processes these interactions by simulating IoT devices, capturing attack attempts, and providing real-time alerts and detailed reports to the administrators.

**DFD Level 1:**

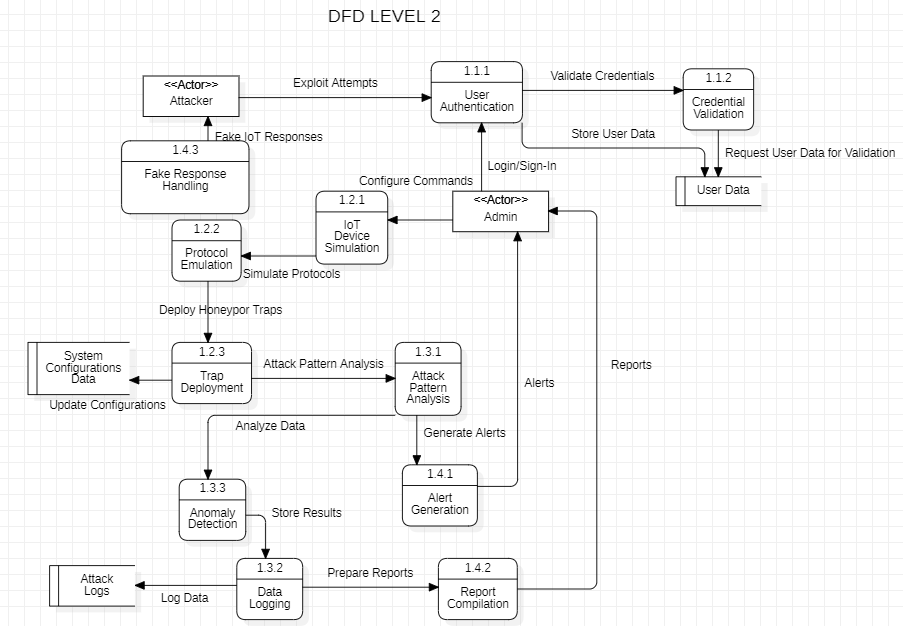
A diagram of a system

Description automatically generated

**Description:**

The Level 1 DFD for an IoT honeypot system breaks down the main process into its key subprocesses, providing more detail on how the system functions. Key subprocesses include user authentication, IoT device simulation, protocol emulation, trap deployment, attack pattern analysis, and alert/report generation. Each subprocess highlights its inputs, outputs, and the data stores involved, such as user data, attack logs, and system configurations. This breakdown offers a clear understanding of how the system manages interactions with attackers and administrators.

**DFD Level 2:**



**Description:**

The Level 2 DFD for an IoT honeypot system details the subprocesses from Level 1, including device simulation, protocol emulation, trap deployment, attack analysis, and alert/report generation. It illustrates how data flows between processes like anomaly detection and data logging, interacting with data stores such as attack logs and configurations. This level provides a comprehensive view of the system’s operations, focusing on detecting, analyzing, and responding to attacks.

SSD Diagrams:

Simulate IoT Protocols:

A screenshot of a computer

Description automatically generated

**Description:**

An attacker sends malicious data to the IoT simulator, which detects, analyzes, and records the attack. A simulation report is generated and sent to the administrator for further analysis and system improvement.

Monitor Real-Time Attacks:

A diagram of a project

Description automatically generated

**Description:**

This SSD focuses on the administrator's ability to monitor real-time attacks. After logging in with valid credentials and successful authentication, the administrator accesses the dashboard, which displays live attack details, including IP addresses, timestamps, and trends. The administrator reviews the live data and analyzes insights to identify frequent attack sources and patterns, aiding in proactive mitigation and strategy formulation.

Generate Alerts:

A diagram of a security system

Description automatically generated

**Description:**

This SSD diagram illustrates the process of generating and notifying the admin about critical attack patterns detected by an IoT system. The IoT Simulator detects a critical attack pattern and forwards the information to the Security Module, which generates an alert. This alert is sent to the Notification Server, which triggers notifications to the admin via email and SMS to ensure immediate awareness of the detected threat

Analyze Attack Data:

A diagram of a system

Description automatically generated

**Description:**

This sequence diagram illustrates how an administrator logs into the system to analyze attack data. The process begins with the administrator providing login credentials, followed by system authentication and granting access. The administrator navigates to the data analysis section, where the system displays the relevant interface. Attack logs are selected for processing, and the system analyzes the selected data, providing insights such as attack sources, trends, and vulnerabilities for further review by the administrator.

Train machine Learning Model:

A screenshot of a computer

Description automatically generated

**Description:**

The admin logs in, selects datasets, configures parameters, and starts the training process. The system processes data, trains the model, notifies the admin upon completion, displays a performance report, and saves the trained model.

Visualize Attack Data:

A screenshot of a computer program

Description automatically generated

**Description:**

The admin logs in, navigates to the visualization section, and applies filters to refine attack data visualizations. The system loads data, displays visualizations, updates them based on filters, and provides interactive, updated views for analysis.

## Activity Diagram:

A flowchart of a sign-in

Description automatically generatedA diagram of a data processing process

Description automatically generated

**Description:**

The activity diagram for the IoT Honeypot System illustrates the system's workflow, starting from user registration and authentication to monitoring and analyzing cyberattacks. Admins can register or log in with valid credentials to access the system dashboard. The system processes incoming traffic, identifying and simulating malicious activity through the IoT simulator. Data from attacks is analyzed to detect patterns, update the threat database, and generate reports. Admins review these reports to gain insights and develop mitigation strategies, ensuring continuous monitoring and improvement of IoT security.

# Data Design

**Transformation of Information Domain into Data Structures**

The IoT honeypot system’s information domain involves capturing and analyzing cyberattack behaviors, managing administrative activities, and maintaining system operations. These are transformed into structured data entities stored in a relational database.

**Major Data Entities**

**1. Admin**

* **Attributes**: admin\_id, email, password\_hash, name, created\_at, updated\_at
* **Function**: Stores administrator account and authentication details.

**2. AttackLog**

* **Attributes**: log\_id, source\_ip, attack\_type, target\_device, timestamp, severity\_level
* **Function**: Logs details of malicious traffic and detected attacks.

**3. IoTDevice**

* **Attributes**: device\_id, device\_name, device\_type, status, vulnerability\_level, created\_at, updated\_at
* **Function**: Stores information about simulated IoT devices in the honeypot network.

**4. ThreatPattern**

* **Attributes**: pattern\_id, attack\_type, frequency, affected\_devices, mitigation\_suggestions, updated\_at
* **Function**: Maintains data about identified attack patterns and mitigation strategies.

## Databases and Data Storage

Database: IoT Honeypot system uses a DBMS (Elastic search).

Below is a JSON code that includes example documents for each of the collections: **Admin**, **IoTDevice**, **AttackLog**, **ThreatPattern**, and **Report**. This JSON code is designed for saving data and can be used to initialize the IoT Honeypot System database.

{

"Admin": [

{

"admin\_id": "1",

"email": "admin@example.com",

"password\_hash": "hashed\_password",

"name": "System Admin",

"created\_at": "2024-12-29T10:00:00Z",

"updated\_at": "2024-12-29T10:00:00Z"

}

],

"IoTDevice": [

{

"device\_id": "101",

"device\_name": "Smart Camera",

"device\_type": "Camera",

"status": "active",

"vulnerability\_level": "high",

"created\_at": "2024-12-28T14:00:00Z",

"updated\_at": "2024-12-29T10:00:00Z"

}

],

"AttackLog": [

{

"log\_id": "1001",

"source\_ip": "192.168.1.100",

"attack\_type": "Brute Force",

"target\_device": "101",

"timestamp": "2024-12-29T10:15:00Z",

"severity\_level": "critical"

}

],

"ThreatPattern": [

{

"pattern\_id": "5001",

"attack\_type": "DDoS",

"frequency": "high",

"affected\_devices": ["101", "102"],

"mitigation\_suggestions": "Rate limiting, blocking suspicious IPs",

"updated\_at": "2024-12-29T09:45:00Z"

}

],

"Report": [

{

"report\_id": "9001",

"generated\_at": "2024-12-29T11:00:00Z",

"summary": "Analysis of recent attacks indicates increased brute force attempts.",

"recommendations": "Strengthen device credentials, enable IP blacklisting"

}

]

}

## Requirements Traceability Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| **Req. Number** | **DFD** | DFD Level 2 | Function Name(s) |
| R1 | Admin Registration | Admin Registration Process | Register admin |
| R2 | Admin Login | Admin Authentication Process | Authenticate admin |
| R3 | Monitor Traffic | Traffic Monitoring Process | Monitor incoming traffic |
| R4 | Simulate IoT Environment | IoT Simulation Process | Simulate vulnerable IoT environment |
| R5 | Detect Malicious Activity | Attack Detection Process | Detect malicious activity |
| R6 | Log Malicious Traffic | Logging Process | Log attack details |
| R7 | Analyze Attack Patterns | Data Analysis Process | Analyze attack patterns |
| R8 | Generate Reports | Reporting Process | Generate attack reports |
| R9 | Predict Threats | Threat Prediction Process | Predict future threats |
| R10 | Admin Logout | Logout Process | Logout admin |

# Human interface design

The interfaces for the IoT Honeypot System encompass admin management, IoT device simulation, traffic analysis, threat detection, and reporting. Admins can register, authenticate, and manage their accounts. The system simulates IoT devices to attract and analyze malicious activity, logging and categorizing attack data. Traffic analysis identifies patterns and potential threats, while predictive models forecast future risks. Reporting interfaces allow admins to view detailed insights and recommendations for mitigation. These interfaces collectively provide a robust framework for monitoring, analyzing, and responding to cyber threats targeting IoT environments.