A logo of a university

Description automatically generated**COMSATS University Islamabad,   
Abbottabad Campus**

**Project Proposal   
(SCOPE DOCUMENT)**

**For**

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

***By***

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

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

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

***Supervisor*Sayed Shahab Zarin**

**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:

Our project, "IoT-Based Honeypot Network for Cybersecurity Threat Detection and Attack Analysis," is about building a special network that will help us study how hackers target IoT devices. With the growing use of IoT devices like smart home cameras, thermostats, and routers, these devices have become a popular target for cyberattacks. Unfortunately, many of these devices lack strong security, which makes them vulnerable to unauthorized access, data theft, malware, and even large-scale attacks like Distributed Denial of Service (DDoS).

This project focuses on creating a honeypot network that simulates various vulnerable IoT devices, such as smart cameras, thermostats, and routers, to attract and log malicious activity. By monitoring and analyzing these interactions, the system aims to provide valuable insights into the tactics, techniques, and procedures (TTPs) commonly used by attackers in IoT environments. Additionally, this project integrates real-time traffic monitoring, anomaly detection, and data visualization to enhance threat detection and support proactive cybersecurity strategies. Ultimately, this honeypot network will contribute significant data to IoT security research, aiding in the development of more effective defenses and preventive security measures for the IoT ecosystem.

Purpose:

This document defines the requirements for developing an IoT Honeypot System aimed at enhancing IoT security. By establishing a honeypot network that simulates vulnerable IoT devices, the system attracts, and records cyberattacks, providing valuable insights into attacker behaviors and strategies. The system supports common IoT protocols, captures a wide range of attacks, and utilizes real-time data collection, visualization, and machine learning techniques to analyze threats, identify patterns, and improve proactive security measures.

Simulate IoT Protocols**:**

|  |  |
| --- | --- |
| **Use case ID** | UC -01 |
| **Use case Name** | Simulate IoT Protocols |
| **Actors** | Attacker |
| **Description** | The system mimics the behaviour of IoT devices and protocols to attract attackers and capture malicious activity. |
| **Trigger** | An attacker initiates an interaction with the system’s simulated IoT devices. |
| **Preconditions** | The IoT Honeypot system is active and configured to simulate IoT protocols.  Attacker has access to the simulated environment |
| **Postconditions** | Attack data is captured and logged for analysis.  The system remains operational for subsequent interactions. |
| **Normal Flow** | The attacker sends malicious data or exploits vulnerabilities within the simulated IoT environment.  The IoT simulator receives the malicious data or detects the exploit attempt.  The simulator analyzes the received data or the detected activity to identify the attack type and source.  The simulator records the attack data, including timestamp, source (e.g., IP address, device ID), and type of attack.  The simulator continues to operate, potentially implementing mitigation measures based on the captured attack data. |
| **Alternative flow**  **Alternative Flow 1 (Simulation Error):** | If the simulation fails, the system displays an error message and restarts the simulation. |
| **Alternative flow**  **Alternative flow 2**  **(Image Upload Error)** | If an error occurs during image upload (e.g., unsupported file format, exceeding file size limit), the system displays an error message. The user must choose a valid image and try again. |
| **Exceptions** | If the system crashes, attack data is not captured. |
| **Business Rules** | Simulated protocols must mimic real IoT devices to attract attackers effectively.  The system must ensure sufficient security to prevent unauthorized access. |
| **Assumptions** | The attacker is motivated to interact with the simulated environment. |

|  |  |
| --- | --- |
| **Use case ID** | UC -02 |
| **Use case Name** | Monitor Real-Time Attacks |
| **Actors** | Admin |
| **Description** | Admin monitors ongoing attacks on the simulated IoT environment in real time. |
| **Trigger** | Admin accesses the monitoring dashboard. |
| **Preconditions** | Attack data is being actively captured by the system.  Admin has access credentials to the monitoring system. |
| **Postconditions** | Admin gains insights into ongoing attacks.  Real-time attack activity is logged and displayed. |
| **Normal Flow** | The administrator successfully authenticates with the system using valid credentials.  The system presents a real-time dashboard to the administrator. This dashboard displays live attack activity, including:  The source IP address of the attacker.  The specific IoT protocol being targeted (e.g., MQTT, CoAP).  The exact time of each attack attempt.  The administrator reviews the live attack data on the dashboard.  The administrator analyzes the displayed information to gain critical insights, such as:  Identifying the most frequent attack sources.  Detecting emerging attack trends.  Understanding the impact of attacks on the simulated environment. |
| **Alternative flow**  **(No Attack Data)** | If no attack data is available, the system displays “No ongoing attacks” on the dashboard. |
| **Exceptions** | Dashboard fails to load due to system overload. |
| **Business Rules** | Data updates must occur in real time (e.g., every second). |
| **Assumptions** | Admin has basic knowledge of interpreting dashboard information. |

|  |  |
| --- | --- |
| **Use case ID** | UC -03 |
| **Use case Name** | Generate Alerts |
| **Actors** | Admin |
| **Description** | The system notifies the admin of critical attack patterns or anomalies in real time. |
| **Trigger** | The system detects unusual activity or a critical threshold breach. |
| **Preconditions** | The system is actively monitoring attacks.  Admin is subscribed to notifications. |
| **Postconditions** | Admin is informed of critical events. |
| **Normal Flow** | The IoT Simulator continuously monitors for suspicious activity.  upon detection of a critical attack pattern by the Security Module, an alert is generated.  The Notification Service then dispatches both an email notification to the administrator's registered email address and an SMS message to their mobile phone number.  Ensuring timely awareness and response to the critical security threat. |
| **Alternative flow**  **(Alert Delivery Failure)** | If notification delivery fails, the system logs the issue and retries. |
| **Exceptions** | Alerts are not generated if attack thresholds are incorrectly set. |
| **Business Rules** | Alert messages must include details like time, type, and severity. |
| **Assumptions** | Admin regularly checks their email or SMS for notifications. |

|  |  |
| --- | --- |
| **Use case ID** | UC -04 |
| **Use case Name** | Analyze Attack Data |
| **Actors** | Admin |
| **Description** | Allows the admin to study and interpret attack data to identify trends and vulnerabilities. |
| **Trigger** | Admin selects the option to analyze captured attack data. |
| **Preconditions** | The system has captured attack data. Admin has access to the analysis tools. |
| **Postconditions** | Admin obtains insights about attack patterns.  Analytical results are stored for future reference. |
| **Normal Flow** | Admin logs into the system.  Admin navigates to the data analysis section.  Admin selects datasets or attack logs to analyze.  The system processes and displays detailed attack insights, such as trends, attack sources, and vulnerabilities. |
| **Alternative flow**  **(No Data)** | If no data is available for analysis, the system displays an error message and prompts the admin to wait for data collection. |
| **Exceptions** | If the system experiences an error during analysis (e.g., processing overload), it displays an error message and logs the issue for debugging. |
| **Business Rules** | The system must visualize complex data in an intuitive format.  Data should be anonymized when necessary for security compliance. |
| **Assumptions** | Admin is familiar with interpreting analytical charts and statistics. |

|  |  |
| --- | --- |
| **Use case ID** | UC -05 |
| **Use case Name** | Analyze Attack Data |
| **Actors** | Admin |
| **Description** | Allows the admin to train machine learning models using the captured attack data to improve future threat detection. |
| **Trigger** | Admin initiates the machine learning model training process. |
| **Preconditions** | Attack data is preprocessed and ready for training.  Admin has access to the training module. |
| **Postconditions** | A trained machine learning model is saved for deployment.  System accuracy is improved for future detections. |
| **Normal Flow** | * Admin logs in and navigates to the model training section. * Admin selects the datasets to use for training. * The system preprocesses the data, removing outliers and normalizing values. * Admin configures training parameters (e.g., model type, epochs). * The system trains the model and displays progress in real-time. * Upon completion, the system saves the trained model and generates a performance report. |
| **Alternative flow**  **(Training Error)** | If the model training fails (e.g., insufficient data or incompatible parameters), the system displays an error message and logs the issue for troubleshooting. |
| **Exceptions** | System crash during training leads to a loss of progress and requires a restart. |
| **Business Rules** | Ensure high-quality, balanced datasets for accurate training. Prevent model overfitting through appropriate parameter tuning. |
| **Assumptions** | Admin has a basic understanding of machine learning concepts and processes. |

|  |  |
| --- | --- |
| **Use case ID** | UC -06 |
| **Use case Name** | Predict Future Threats |
| **Actors** | Admin |
| **Description** | Predict potential future attacks based on trends and patterns identified from historical data. |
| **Trigger** | Admin accesses the threat prediction module. |
| **Preconditions** | A trained machine learning model is available. Historical attack data exists in the system. |
| **Postconditions** | Predicted threats are displayed in a report for admin review. Predictions are logged for continuous improvement. |
| **Normal Flow** | 1. Admin logs into the system and navigates to the threat prediction module. 2. Admin specifies the prediction parameters (e.g., time frame, attack type). 3. The system uses the trained model to predict potential threats. 4. Predicted threat details, including probability and affected areas, are displayed. |
| **Alternative flow**  **(No Prediction)** | If predictions cannot be generated due to insufficient data, the system notifies the admin and prompts retraining of models. |
| **Exceptions** | System error during prediction calculation may delay results and require troubleshooting. |
| **Business Rules** | Predictions should be presented with confidence scores.  The system should update prediction models periodically to maintain relevance. |
| **Assumptions** | Admin is capable of interpreting prediction results for proactive threat management. |

|  |  |
| --- | --- |
| **Use case ID** | UC -07 |
| **Use case Name** | Visualize Attack Data |
| **Actors** | Admin |
| **Description** | Provides an interactive interface for the admin to view and interpret attack data through charts, graphs, and dashboards. |
| **Trigger** | Admin opens the visualization module. |
| **Preconditions** | Attack data exists in the database. Admin has access to the visualization dashboard. |
| **Postconditions** | Admin gains actionable insights from visualized data. Visualization preferences are saved for future use. |
| **Normal Flow** | 1. Admin logs into the system.  2. Admin navigates to the data visualization dashboard.  3. The system displays attack data in various formats (e.g., bar charts, pie charts, heatmaps).  4. Admin interacts with the visualizations (e.g., filters data by time, type, or source). |
| **Alternative flow**  **(No Data)** | If no data is available, the system displays a message and provides options to refresh or select a different dataset. |
| **Exceptions** | Visualization errors due to missing libraries or rendering issues are logged for debugging. |
| **Business Rules** | Visualization should be intuitive and easy to interpret. Data privacy must be ensured during rendering. |
| **Assumptions** | Admin has basic knowledge of interpreting charts and graphs. |