EXPERIMENT NO: 07

| EXPERIMENT NO:07 | |
| --- | --- |
| **Objective** | To design and implement an anomaly detection system that monitors IoT devices and identifies suspicious activities based on behavior patterns or known attack signatures. |
| **Aim:** | Develop a system that continuously monitors IoT device logs or traffic, detects anomalies, and generates alerts for potential security threats such as unauthorized access or abnormal usage patterns. |
| **Apparatus:** | 1. **IoT Device** (or simulator): Any device with a network interface (e.g., Raspberry Pi, IoTGoat, or an IoT service). 2. **Computer System** with a Linux OS (Ubuntu/Kali) or VM. 3. **Anomaly Detection Tools**:     * Python 3.x    * Scikit-learn (for machine learning-based anomaly detection)    * TensorFlow/Keras (optional, for advanced anomaly detection models) 4. **Data Collection**:     * Log files from IoT device or network traffic logs (Wireshark/Tcpdump). 5. **Monitoring Tool** (optional): Grafana or Kibana for visualizing logs and alerts. |
| Theory | **Anomaly detection in IoT systems is crucial to identifying unauthorized access, device manipulation, or unusual device behavior. IoT devices often have limited resources and are exposed to various attacks such as Denial of Service (DoS), unauthorized control, or information leakage.**  **Anomaly detection can be achieved using several techniques:**   * **Statistical Methods: Methods like Z-score or Interquartile Range (IQR) assess deviations from normal behavior by defining a baseline or threshold.** * **Machine Learning Models: Algorithms such as Isolation Forest, One-Class SVM, and clustering techniques (k-means) learn a "normal" behavior profile and flag anything that deviates significantly.** * **Deep Learning Models: Autoencoders are a type of neural network that can learn normal patterns in the data and flag anomalies based on reconstruction errors.**   **The goal is to detect abnormal activities such as:**   * **Unauthorized login attempts** * **Abnormal traffic patterns** * **Unusual power consumption or other device-specific metrics.** |
| **Procedure** | **Set up the environment**:   * Deploy the IoT device or simulate IoT traffic. * Set up a monitoring system to collect logs (e.g., syslog or network traffic logs).   **Data Collection**:   * Gather normal device operation logs and network traffic for a period of time. Logs should capture metrics like timestamps, device actions, and states. * Use monitoring tools like Wireshark or Tcpdump to capture network traffic data.   **Data Preprocessing**:   * Clean and preprocess the data. This may include removing noise, normalizing values, parsing timestamps, and converting data into a usable format. * Perform feature engineering by selecting relevant features such as CPU usage, memory usage, device-specific metrics (e.g., power consumption).   **Model Selection and Implementation**:   * Choose an anomaly detection algorithm. Options include:    + **Statistical Methods**: Z-score or IQR-based methods.   + **Machine Learning**: Isolation Forest, One-Class SVM, or k-means clustering.   + **Deep Learning**: Autoencoders for more complex anomaly detection. * Train the model using normal data collected in the previous step.   **Model Evaluation**:   * Evaluate the trained model on known anomaly data (e.g., simulated attacks like DoS or unauthorized access). * Tune model hyperparameters to minimize false positives and false negatives.   **Real-Time Monitoring**:   * Deploy the anomaly detection model to monitor the device or network in real-time. * Set up alerting mechanisms (e.g., email, Slack, or system logs) to notify administrators when suspicious activities are detected.   **Test and Analyze**:   * Simulate attacks or abnormal behaviors, such as DDoS, unauthorized logins, or abnormal device usage. * Ensure that the system generates alerts for these events while minimizing false alarms. |
| **observation** | * **The anomaly detection system should successfully identify deviations from normal activity.** * **Alerts should be triggered for simulated attacks or suspicious behaviors.** * **False positives and false negatives should be minimized but may occur depending on model tuning.**   . |
| **Result&** | **Anomaly Detection**:   * The anomaly detection system identified unusual patterns in IoT device behavior, such as:    + Unauthorized login attempts.   + Traffic spikes or patterns indicating DDoS.   + Abnormal device states (e.g., power usage).   **False Positives/Negatives**:   * Some false positives may have been observed, especially with complex models (e.g., deep learning-based ones), requiring fine-tuning. * The model performed well with minimal false negatives, but occasional outliers (non-malicious behavior) were flagged.   **Alerts Generated**:   * Alerts were successfully generated for simulated attacks such as unauthorized access or suspicious network traffic patterns. * Alert notifications were sent via email or system logs depending on the configuration.   **System Performance**:   * The system was able to monitor and detect anomalies in real-time with minimal computational overhead, especially for simpler models like Isolation Forest. |
| **Discussion** | **1. Challenges in Anomaly Detection**  * **Defining Normal Behavior**: IoT devices often exhibit varying usage patterns depending on the context. A fixed baseline for "normal" might not always be accurate, especially if devices have highly dynamic usage patterns. * **Real-Time Processing**: Processing data in real-time without introducing significant delays can be challenging, especially for deep learning-based anomaly detection models, which require substantial computational resources.  **2. Machine Learning vs. Statistical Methods**  * **Machine Learning** models like Isolation Forest are generally more robust for detecting complex anomalies but may require more data to train effectively. * **Statistical Methods** like Z-score are faster and simpler but might not capture sophisticated attacks or behavior anomalies in highly variable systems.  **3. Types of Attacks Detected**  * **Unauthorized Access**: The system successfully detected unauthorized login attempts based on abnormal login patterns or failed authentication attempts. * **Denial of Service (DoS)**: Traffic spikes and unusual packet rates were identified as potential signs of a DoS attack. * **Device Manipulation**: Unusual changes in device metrics (e.g., sudden power consumption spikes) were flagged as potential indicators of manipulation or malicious behavior.  **4. Impact of False Positives**  * False positives could be problematic in IoT systems where device behavior fluctuates. For example, routine maintenance could trigger an alert. Fine-tuning the model and incorporating context-aware monitoring can help reduce false positives. * Excessive false positives may lead to alert fatigue, causing administrators to ignore legitimate alerts.   . |
| **Conclusion:** | This experiment highlights the importance of anomaly detection in securing IoT devices by identifying suspicious activities. Implementing an effective detection system can help protect against common IoT attacks such as unauthorized access and DoS. However, challenges such as false positives and real-time processing need careful consideration. A combination of statistical and machine learning-based methods offers a good balance between detection capability and system efficiency. |
| **Discussion Question** | What are the main challenges when implementing anomaly detection on IoT devices?  How does an anomaly detection system distinguish between normal and malicious activities on IoT devices?  Which anomaly detection algorithms are most suitable for IoT devices with limited resources?  How can false positives be minimized without compromising the detection of real attacks?  What ethical issues should be considered when implementing a monitoring system on IoT devices? |

| **Marks-Sheet for Assessment of Practical AY25-26** | | | | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Scaling Factor for practical Assessment.** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **Total** | **Grade** |
| **Attenda-nce** | **Perform-ance** | **PRO-1** | **PRO-- 2** | **PRO-**  **3** | **Technical -Knowledge** | **Technical -Documentation** | **Level of Interaction** | **Behaviour-Attitude towards Learning** | **Compliance** |
| **Weight (MAX10)** | **10** | **10** | **10** | **10** | **10** | **10** | **10** | **10** | **10** | **10** | **100** |  |
| **Points**  **(MAX10)** |  |  |  |  |  |  |  |  |  |  |  |  |
| **Marks (MAX10)** |  |  |  |  |  |  |  |  |  |  |  |  |
| **Score** | **EP(Earned Points)=---- (MAX-10)** | | | | | **Total Marks =--------------------( (MAX100)** | | | | | | |
| **Scaling Factor for practical Assessment.** | | | | | | | | **Department of Electronics &Computer Science (AY2025-26)** | | | | |
| 1..Attendance | | | | **7.TechnicalDocumentation**  **(content as per format )** | | | | **Date of Performance**  **DOP) = / /2025** | | | | |
| **2.. Performance** | | | | **8. Level of Interaction**  **Developing Expression power** | | | | **Date of Correction**  **(DOC) = / / 2025** | | | | |
| **3 PRO-1-Practtical Objective 1 4.PRO-1-Practical Objective 2**  **5. PRO-1-Practical Objective3** | | | | **9. Behaviour-Attitude towards Learning**  **(Regularity/Team Work)** | | | | **Name of Student:** | | | | |
| **6. Technical –Knowledge**  **(preparedness &Execution)** | | | | **10. Compliance**  **(Attainment of objectives(Learning Outcome))** | | | | **Roll No-** | | | | |
| **Learning Outcome:**  **1.Write Program for each expt.**  **2.Use Proper Software tool to edit program /compile program**  **3.Execute and get the RESULT of expt.** | | | | **4.Attache-/plot /Graph of Result obtained**  **5.Maintain Proper Discipline in Lab hrs**  **6.Write D-QA on Write-up sheet for Assessment** | | | | **Signature of Faculty with Date** | | | | |