**PROJECT TITLES:**

1. Deauthentication detector
2. DDOS prediction system
3. Web vulnerability scanner

**TITLE : Real-time Deauthentication Attack Detector for Wireless Networks**

**PROBLEM STATEMENT:**

Wireless networks are vulnerable to deauthentication attacks, which can lead to unauthorized access, data breaches, and network disruptions. Current solutions rely on manual monitoring and logging, making it challenging to detect and respond to these attacks in real-time. There is a need for an automated system that can detect and alert on deauthentication attacks, enabling swift action to secure the network.

**SOLUTION:**

**Objective:**

Design and develop a real-time deauthentication attack detector for wireless networks, leveraging machine learning and packet analysis to identify and alert on potential threats.

**Key Features:**

1. Real-time packet capture and analysis

2. Machine learning-based deauthentication attack detection

3. Automated alerting and logging

4. Customizable threshold settings for detection sensitivity

5. Integration with existing network security infrastructure

**Functional Requirements:**

1. Capture wireless network packets using Wireshark or Tcpdump

2. Analyze packets for deauthentication frames (subtype 12)

3. Implement machine learning algorithms (e.g., SVM, Random Forest) to classify legitimate and malicious deauths

4. Develop a web-based interface for configuration, monitoring, and alerting

5. Integrate with existing security information and event management (SIEM) systems

**Non-Functional Requirements:**

1. High accuracy in detecting deauthentication attacks

2. Low false positive rate

3. Real-time detection and alerting

4. Scalability for large wireless networks

5. User-friendly interface for easy configuration and monitoring

**TITLE: DDOS Attack Prediction System**

**Problem Statement:**

Distributed Denial-of-Service (DDOS) attacks are a growing threat to network security, causing significant financial losses and reputational damage. Traditional security measures often struggle to detect and mitigate these attacks in real-time. Develop a predictive system that can accurately identify potential DDOS attacks before they occur, enabling proactive measures to prevent or minimize their impact.

**Solution Outline:**

**Objective:**

Design and implement a predictive system using machine learning algorithms to identify potential DDOS attacks in real-time.

**Methodology:**

1. Data Collection: Gather network traffic data from various sources (e.g., NetFlow, packet captures).

2. Data Preprocessing: Clean, transform, and feature-engineer the data for training.

3. Model Selection: Choose suitable machine learning algorithms (e.g., Random Forest, SVM, ANN).

4. Model Training: Train the models using the preprocessed data.

5. Model Evaluation: Assess the performance of each model using metrics (e.g., accuracy, precision, recall, F1-score).

6. Deployment: Integrate the best-performing model into a real-time monitoring system.

**Key Components:**

1. Data Ingestion: Collect and process network traffic data in real-time.

2. Feature Extraction: Extract relevant features from the data (e.g., packet headers, traffic patterns).

3. Model Deployment: Deploy the trained model to predict potential DDOS attacks.

4. Alert System: Generate alerts for predicted attacks, enabling proactive measures.

**Tools and Technologies:**

1. Programming Languages: Python, R.

2. Machine Learning Libraries: scikit-learn, TensorFlow, PyTorch.

3. Data Processing: Apache Kafka, Apache Spark.

4. Data Storage: Apache Cassandra, MongoDB.

**Expected Outcomes:**

1. Accurate Predictions: Identify potential DDOS attacks with high accuracy.

2. Real-time Detection: Detect attacks in real-time, enabling prompt action.

3. Proactive Measures: Prevent or minimize the impact of DDOS attacks.

**TITLE: Web Vulnerability scanner**

**Problem Statement:**

Websites and web applications are vulnerable to various security threats, including SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). Manually identifying these vulnerabilities is time-consuming and prone to errors. Develop an automated web vulnerability scanner tool to detect and report potential security threats, enabling proactive measures to secure web applications**.**

**Solution Outline:**

**Objective:**

Design and implement a tool that scans web applications for potential vulnerabilities, providing detailed reports to facilitate remediation.

**Methodology:**

1. Crawling: Crawl the web application to identify URLs, forms, and inputs.

2. Vulnerability Scanning: Scan identified components for vulnerabilities (e.g., SQL injection, XSS, CSRF).

3. Reporting: Generate detailed reports highlighting vulnerabilities and remediation suggestions.

**Key Components:**

1. Crawler Module: Crawl web applications to identify potential entry points.

2. Vulnerability Scanner Module: Scan identified components for vulnerabilities.

3. Reporting Module: Generate reports and provide remediation suggestions.

**Tools and Technologies:**

1. Programming Languages: Python, JavaScript.

2. Web Scraping Libraries: BeautifulSoup, Scrapy.

3. Vulnerability Scanning Libraries: OWASP ZAP, Burp Suite.

4. Database: Store scan results and reports in a database (e.g., MySQL, MongoDB).

**Expected Outcomes:**

1. Automated Scanning: Scan web applications for vulnerabilities with minimal human intervention.

2. Comprehensive Reporting: Provide detailed reports highlighting vulnerabilities and remediation suggestions.

3. Proactive Security: Enable proactive measures to secure web applications and prevent attacks.