**SDI 3203 Bonus Project Assignment**

**Course:** SDI 3203 Network Fundamentals  
**Instructor:** Dr. Chenggang Wang  
**Due Date:** April 29, 2025 (Tuesday), 11:59 PM

**Format:** Please submit a zip file, which should include your source code, a pdf file that summarize the execution of your code.

**Total Points:** 20 bonus points addon to your final grade

**Note: This is a individual project**

**1. Overview**

In this project, you will work independently on a research project focused on basic network traffic analysis and anomaly detection. You will use open‐source simulation tools and public datasets to explore concepts in network behavior and security. The aim is to develop hands-on experience with network simulation, data collection, and simple machine learning techniques.

**Your Role:**  
You are responsible for defining your research question, setting up your experiments, analyzing the data, and presenting your results. I will provide guidance and support as needed.

**2. Objectives**

* **Network Simulation and Traffic Capture:**  
  Learn to use **Mininet** to build a simple network topology and generate traffic.
* **Basic Data Analysis:**  
  Capture and analyze network traffic using tools like **tcpdump** and **Wireshark**.
* **Simple Machine Learning Integration:**  
  Use Python (with **pandas** and **scikit-learn**) to perform basic classification or anomaly detection based on extracted traffic features.
* **Reporting:**  
  Develop a final report and a brief presentation that describe your methodology, analysis, and findings.

**3. Project Topics and Specific Tasks**

You may choose one of the six projects below. Each project includes step-by-step tasks and recommended tools. If you have a similar idea, please discuss it with me for approval.

**A. Basic TCP Traffic Analysis and Anomaly Detection**

**Objective:**  
Simulate a simple network with normal and slightly abnormal TCP traffic. Analyze captured traffic to distinguish between typical flows and those with unusual behavior.

**Recommended Tools:**

* **Mininet** for network simulation.
* **tcpdump/Wireshark** for traffic capture.
* **Python** with **pandas** and **scikit-learn** for data handling and classification.

**Tasks:**

1. **Setup and Traffic Generation:**
   * Install Mininet following the Mininet Walkthrough.
   * Create a topology with at least three hosts.
   * Generate normal TCP traffic (e.g., file transfers using iperf or netcat).
   * Introduce abnormal behavior in one flow (e.g., by scripting excessive retransmissions or packet delays).
2. **Data Collection:**
   * Capture network traffic using tcpdump.
   * Export captured data to CSV via Wireshark for selected fields (e.g., packet size, inter-arrival times, TCP flags).
3. **Data Analysis:**
   * Use Python and pandas to load and process your data.
   * Visualize differences between normal and anomalous flows using matplotlib.
   * Train a basic classifier (e.g., decision tree) with scikit-learn to label flows.
4. **Reporting:**
   * Write a report detailing your setup, data collection, analysis, and results.
   * Include visualizations (graphs/charts) to illustrate your findings.

**B. Intrusion Detection Using Public Dataset Analysis**

**Objective:**  
Analyze a public dataset (e.g., NSL-KDD) to build a simple intrusion detection model that classifies traffic records as normal or malicious.

**Recommended Tools:**

* **NSL-KDD dataset** (download from here).
* **Python** with **pandas** and **scikit-learn**.

**Tasks:**

1. **Data Acquisition and Preparation:**
   * Download the NSL-KDD dataset.
   * Explore and clean the dataset using pandas.
   * Preprocess the data by handling missing values and encoding categorical features.
2. **Exploratory Data Analysis:**
   * Visualize key feature distributions with matplotlib or seaborn.
   * Identify patterns that differentiate normal from attack records.
3. **Model Development:**
   * Split the data into training and testing sets.
   * Train a simple classifier (e.g., logistic regression or decision tree).
   * Evaluate the model using accuracy, precision, and recall.
4. **Reporting:**
   * Document your data exploration, preprocessing, model training, and evaluation.
   * Include charts and tables that support your findings.

**C. DNS Traffic Analysis and Simple Anomaly Detection**

**Objective:**  
Analyze DNS query data (simulated or public) to understand typical query patterns and identify anomalies that might indicate security issues.

**Recommended Tools:**

* **Mininet** (or Python scripts) to simulate DNS traffic.
* **Wireshark/tcpdump** for traffic capture.
* **Python** with **pandas** and **scikit-learn**.

**Tasks:**

1. **Data Collection:**
   * Simulate DNS traffic in a small Mininet topology or use a simple Python script.
   * Capture DNS traffic with tcpdump and export relevant fields (query type, frequency, response times) to CSV.
   * Alternatively, use a small public DNS dataset if available.
2. **Feature Extraction and Analysis:**
   * Use Python to extract and analyze features from your DNS data.
   * Visualize common DNS query patterns.
   * Implement a simple anomaly detection method (e.g., clustering or threshold-based) to flag irregular behavior.
3. **Reporting:**
   * Prepare a report detailing your data acquisition, analysis, and any anomalies detected.
   * Include visualizations that help explain your findings.

**D. Port Scan Detection Using Simulation**

**Objective:**  
Simulate a network environment to generate port scanning activity and analyze the traffic to detect scanning behavior.

**Recommended Tools:**

* **Mininet** for network simulation.
* **Nmap** for generating port scans.
* **tcpdump/Wireshark** for capturing traffic.
* **Python** with **pandas** and **scikit-learn** for analysis.

**Tasks:**

1. **Simulation Setup and Traffic Generation:**
   * Set up a simple Mininet topology with several hosts.
   * Use Nmap from one host to perform a port scan on another host.
   * Generate normal traffic concurrently to compare behaviors.
2. **Data Collection:**
   * Capture network traffic during the port scan using tcpdump.
   * Export relevant fields (number of connection attempts, source/destination ports) to CSV.
3. **Data Analysis:**
   * Analyze the captured data with Python to extract features indicative of a port scan (e.g., a high number of connection attempts within a short period).
   * Implement a simple rule-based or ML classifier to distinguish scanning from normal activity.
4. **Reporting:**
   * Document your simulation setup, data analysis, and detection method.
   * Include visualizations to demonstrate how scanning traffic differs from normal traffic.

**E. Network Flow Visualization and Basic Metrics Analysis**

**Objective:**  
Capture and visualize network flows from a simulated network to analyze basic performance metrics such as throughput, latency, and packet loss.

**Recommended Tools:**

* **Mininet** for creating a network topology.
* **tcpdump/Wireshark** for capturing traffic.
* **Python** with **pandas**, **matplotlib**, and optionally **networkx** for visualization.

**Tasks:**

1. **Simulation Setup:**
   * Create a Mininet topology with multiple hosts and links.
   * Generate continuous traffic (e.g., using iperf or custom Python scripts).
2. **Data Collection:**
   * Capture network traffic with tcpdump.
   * Export key metrics (e.g., throughput per link, latency between hosts) into CSV files.
3. **Visualization and Analysis:**
   * Use Python to parse the CSV data and compute basic network metrics.
   * Visualize the network flows and performance metrics (graphs, flow diagrams, etc.).
   * Analyze and discuss any observed performance issues.
4. **Reporting:**
   * Write a report explaining your simulation, the metrics collected, and your visualizations.
   * Include charts and any observations regarding network performance.

**F. ICMP Ping Analysis for Network Diagnostics**

**Objective:**  
Simulate ICMP (ping) traffic under different conditions and analyze the responses to understand basic network diagnostics and detect potential issues.

**Recommended Tools:**

* **Mininet** for network simulation.
* **Ping** command for generating ICMP traffic.
* **Wireshark/tcpdump** for capturing ICMP packets.
* **Python** with **pandas** and **matplotlib** for data analysis.

**Tasks:**

1. **Simulation Setup:**
   * Create a basic Mininet topology with at least two hosts.
   * Use the ping command between hosts to generate ICMP traffic.
   * Introduce artificial network delays (using Mininet’s link parameters) to simulate different conditions.
2. **Data Collection:**
   * Capture ICMP traffic using tcpdump.
   * Extract key metrics (e.g., round-trip time, packet loss) and export them to CSV.
3. **Data Analysis:**
   * Use Python to analyze the ping responses.
   * Plot round-trip times and packet loss under normal and delayed conditions.
   * Discuss how these metrics can be used for basic network diagnostics.
4. **Reporting:**
   * Prepare a report detailing your simulation setup, data analysis, and findings.
   * Include visualizations to illustrate the differences in network performance under varying conditions.

**4. Timeline and Milestones**

1. **Weeks 1–2: Initial Proposal**
   * Select your project topic and define your research question.
   * Submit a brief proposal outlining your objectives, chosen tools, and a rough plan of your tasks.
2. **Weeks 3–4: Background Research & Planning**
   * Conduct a literature review or basic research to familiarize yourself with the tools and concepts.
   * Submit an updated project plan with specific tasks and anticipated challenges.
3. **Weeks 5–8: Implementation & Data Collection**
   * Set up your simulation environment (or download the dataset) and begin data collection.
   * Start initial analysis and simple model development.
4. **Weeks 9–10: Analysis & Reporting**
   * Finalize your analysis and create visualizations.
   * Write your final report documenting your methodology, results, and insights.
5. **Week 11: Final Presentation & Submission**
   * Present your findings to the class.
   * Submit your final report, source code, and documentation.

**5. Evaluation Criteria**

Your project will be evaluated on:

* **Proposal Clarity (20%):**  
  Clear definition of the research question, objectives, and methodology.
* **Implementation (30%):**  
  Successful setup of the simulation or dataset analysis, quality of data collection, and execution of tasks.
* **Analysis and Results (30%):**  
  Depth of data analysis, appropriate use of basic ML techniques or visualizations, and clarity of findings.
* **Reporting and Presentation (20%):**  
  Quality of the written report and clarity of your presentation.

**6. Additional Notes**

* **Resources:**
  + **Mininet Documentation:** Mininet Walkthrough
  + **Wireshark Documentation:** Wireshark User Guide
  + **Python Libraries:** Documentation for **pandas**, **scikit-learn**, and **matplotlib** is available online.
* **Support:**  
  I am available during office hours and by appointment to help with any challenges. Please reach out early if you encounter difficulties.
* **Collaboration:**  
  While discussions with peers are encouraged, all submitted work must be your own. Properly cite any external resources.

Good luck, and I look forward to seeing your innovative projects in network traffic analysis and basic anomaly detection!