**THM Hackathon Documentation**

The Group: THM

Topic Name: Telco Transport Topology and Alarm Visualization

**Some Information about Topic:**

Telco transport topology is essential for modern communication, integrating various technologies and topological structures to create a reliable, scalable, and efficient network. This infrastructure supports the vast amounts of data transmission required by businesses, consumers, and governmental organizations.

**How It Works**

* **Data Transmission**: Data is packetized and transmitted over fiber optic cables, copper lines, or wireless links.
* **Routing and Switching**: Routers and switches direct data packets based on destination addresses and routing protocols (e.g., OSPF, BGP).
* **Redundancy and Failover**: Multiple pathways and backup systems ensure continuous service in case of a failure in one part of the network.

**Topological Structures**

1. **Point-to-Point**:
   * Idea: A direct link between two nodes in a network.
2. **Star Topology**:
   * Idea: All nodes are connected to a central node (hub or switch).
3. **Ring Topology**:
   * Idea: Each node is connected to two other nodes, forming a closed loop.
4. **Mesh Topology**:
   * Idea: Nodes are interconnected, with multiple paths between any two nodes.
5. **Tree Topology**:

* Idea: A hierarchical structure with a root node connected to one or more nodes, which in turn are connected to further nodes.

**Our Idea:**

We create the dashboard which shows us the alerts , count of logs, unsuccessfull connection’s count, connection between endpoints and its graphics.

Build a Transport Topology helps us to view the log counts alerts counts, some rules, graphics of alerts and etc.

And our project divided into two parts: CLI and GUI as a front end part and back-end part

The facilities of Our project:

* To view the logs
* See the malicous activity
* See the alerts logs
* And graphics of Alerts
* Develop an effective alerts system
* system identify the new types of alarms

Our project functions by continuously monitoring and capturing all network traffic, saving this data into log files for detailed analysis. It utilizes predefined rules, specified by users in a file named **local.rules**, to detect anomalies in network connections, ensuring any suspicious activity is promptly identified. Additionally, the system monitors for connection losses and generates alerts to notify users of any disruptions. To enhance its analytical capabilities, the project integrates with Splunk, allowing for advanced analysis of the logged data. This comprehensive approach ensures robust network traffic monitoring, effective anomaly detection, and sophisticated log analysis.

**The code explanation of our code**

The Programming Tools: Html/Css , Javascript, Python

The facilities of code in python

1. To identify malicious activity
2. To identify the connection lost logs

The Front part- we should use javascript html css and some tools etc.

We can see the graphic in dashboard which helps us to see the chart of activity

The Back part- We should use Python

**Responsibilities**

**Hajagha Agayev- Backend Developer**

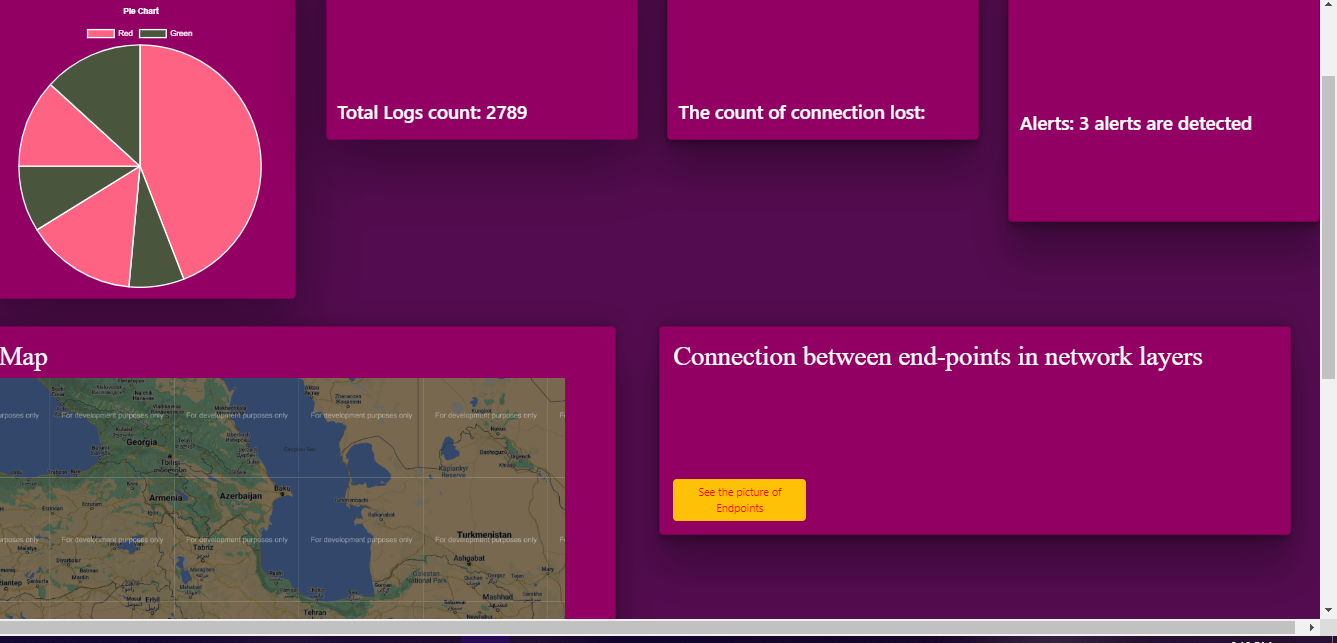
* **Backend Development**: Responsible for developing the backend logic of the application using Python.
* **Network Programming**: Utilizes the Scapy Python library to handle network-related tasks, including packet manipulation and network scanning.
* **API Development**: Works closely with the integration specialist to develop APIs that facilitate communication between the backend and other components.

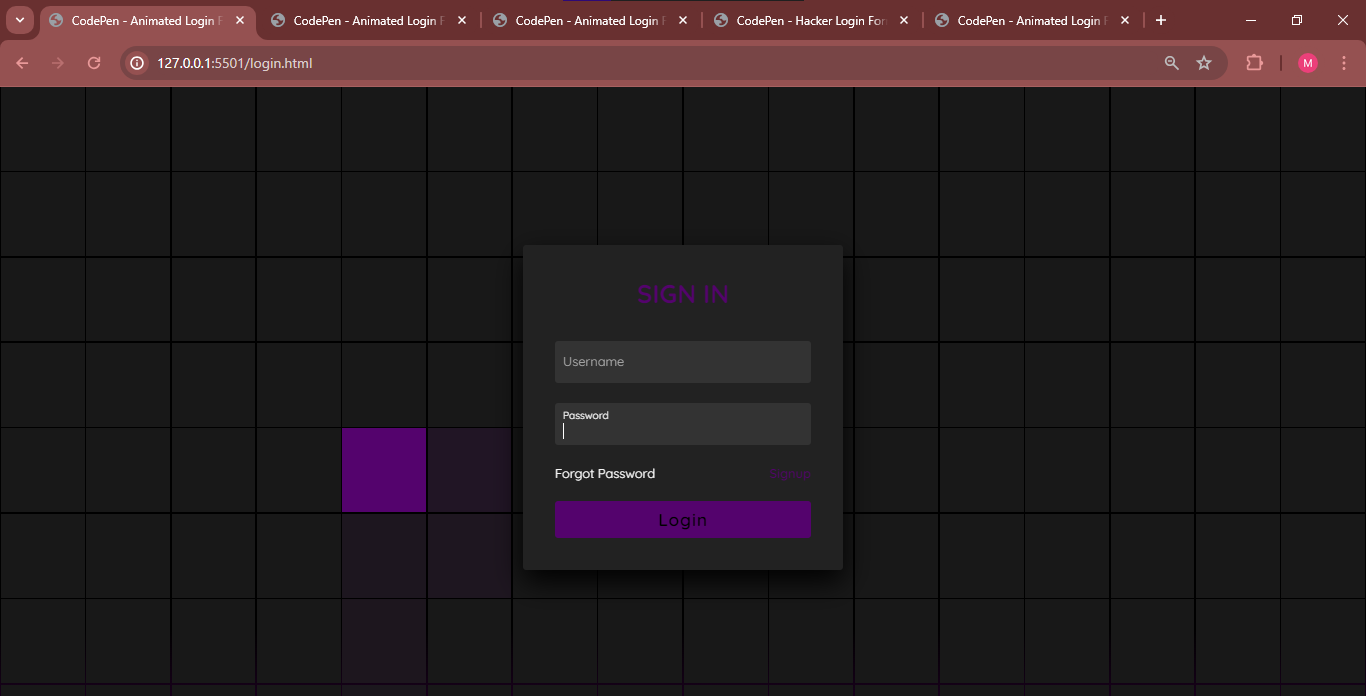
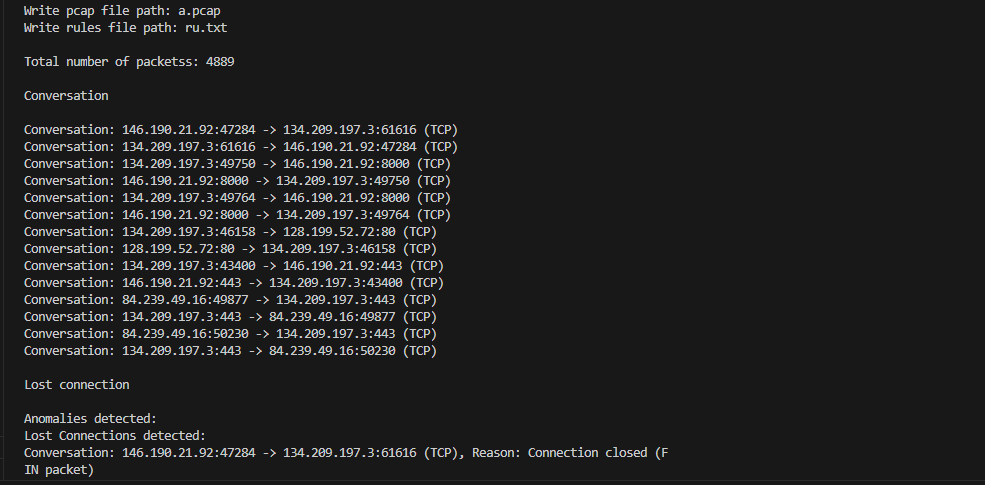
**Teymur Novruzov - Integration Specialist**

* **API Management**: Develops and manages APIs that connect the frontend and backend, ensuring smooth data flow and interaction between the two.
* **Template Engine Integration**: Selects an appropriate template engine to dynamically render content on the frontend, enhancing the user experience.
* **Database Management**: Designs and implements the database structure, ensuring efficient data storage, retrieval, and management.
* **System Integration**: Coordinates the integration of all components, including the frontend, backend, database, and template engine, ensuring they work together seamlessly.
* **Dynamic Content Management**: Establishes connections between the database and other parts of the system to enable dynamic content updates and data-driven functionalities.

**Maleyka Heybatova- Frontend Developer**

* **Frontend Design and Development**: Designs and implements the user interface of the application using HTML, CSS, and JavaScript.
* **User Experience (UX)**: Focuses on creating a responsive and intuitive user experience, ensuring that the frontend is visually appealing and easy to navigate.
* **Cross-Browser Compatibility**: Ensures that the frontend works seamlessly across different web browsers and devices.





***Thanks for creating this opportunity for us:***