Capstone Project Idea: Agentic Multi-Stage Threat Hunting and Incident Response using LangGraph

1. Project Goal

To develop a **Hierarchical Agentic System** using **LangGraph** that automates the process of network threat detection, investigation, and enrichment based on live or recorded packet data. The system will use AI packet analysis models as its "tools" and LangGraph to manage the multi-step decision-making process.

2. The Agentic Architecture (The LangGraph Design)

The core of this project is a multi-agent workflow orchestrated by LangGraph's state machine. The state will be a dictionary that tracks the suspicious flow's metadata, model scores, and investigation notes.

The Agents (Nodes):

1. Traffic Classifier Agent (L1 Analyst)
    - **Role:** First-pass detection and anomaly triage.
    - **Tool:** A pre-trained Machine Learning model (e.g., Random Forest or Simple Neural Network) that classifies flows (extracted from PCAP using a tool like **Scapy** or **CICFlowMeter**) into: Benign, Suspicious, or Known Attack Type (e.g., DDoS, port scan).
    - **Output (State Update):** Updates the shared state with the classification and a confidence score.

2. **Protocol Deep-Dive Agent (L2 Investigator)**
    - **Role:** Analyzes the internal characteristics of a flow flagged as Suspicious.
    - **Tool:** A Deep Learning model (e.g., an LSTM or CNN) designed for zero-day **encrypted traffic analysis** (ETR) that analyzes packet sizes and timing sequences within the suspicious flow.
    - **Conditional Trigger (LangGraph Edge):** Only runs if the Traffic Classifier Agent output is Suspicious.
    - **Output (State Update):** Adds deep protocol features (e.g., estimated application, steganography likelihood) to the state.

3. **Threat Intelligence Agent (Enricher)**
    - **Role:** Contextualizes a suspected threat.
    - **Tool:** Custom Python functions to query external threat intelligence APIs (e.g., VirusTotal, AbuseIPDB) using the source/destination IPs and domains found in the packet metadata.
    - **Conditional Trigger:** Runs if the flow is classified as Known Attack Type or if the Protocol Deep-Dive Agent finds high anomaly scores.
    - **Output (State Update):** Updates the state with external reputation scores, known malicious associations, and geolocation data.

4. **Reporting/Action Agent (Finalizer)**
    - **Role:** Synthesizes the full investigation and recommends an action.
    - **Tool:** A Large Language Model (LLM) instructed with a detailed system prompt and the complete investigation state.
    - **Task:** The LLM's prompt instructs it to generate a summary report and a recommended action (Ignore, Block IP, Notify Admin, Isolate Host).

The LangGraph Flow (Edges and Conditional Logic):

The LangGraph architecture dictates the flow of analysis, making the system dynamic and efficient:

1. START → Traffic Classifier Agent
2. Traffic Classifier Agent → Conditional Edge (based on classification):
    - If **Benign** → **END** (Ignore/Log)
    - If **Suspicious** → **Protocol Deep-Dive Agent**
    - If **Known Attack Type** → **Threat Intelligence Agent**

3. **Protocol Deep-Dive Agent** $\rightarrow$ **Conditional Edge** (based on anomaly score):
    - If **Anomaly Score** $<$ **Threshold** $\rightarrow$ **END** (Safe Anomaly)
    - If **Anomaly Score** $\geq$ **Threshold** $\rightarrow$ **Threat Intelligence Agent**
4. **Threat Intelligence Agent** $\rightarrow$ **Reporting/ Action Agent**
5. **Reporting/Action Agent** $\rightarrow$ **END** (Final Report Generated)

3. Key Capstone Deliverables and Technologies

Component	Technology/Skill
Data & Pre-processing	Scapy, Pandas, Public IDS Datasets (e.g., CICIDS)
AI Models (Tools)	TensorFlow / PyTorch
Agent Orchestration	LangGraph, LangChain
External Tools	Python requests, Threat Int APIs (Free/Public T.I. Feeds)
Final Output	LLM (e.g., OpenAI, Gemini, Llama 3 via API)

4. Novelty and Impact

- **Adaptive Security:** This system moves beyond static rule-based or single-model detection. By using multiple specialized agents and conditional logic, it can adapt the analysis depth to the level of suspicion, reducing false positives and accelerating response.
- **Explainable Reasoning:** LangGraph allows you to visualize and log the exact path the analysis took (which agent ran, why it ran, and the final LLM synthesis), providing transparency into the AI's "reasoning" for the security team.
- **Automation Focus:** The project demonstrates a fully autonomous security operation model, bridging the gap between raw data (packets) and actionable intelligence (an incident report).