TRƯỜNG ĐẠI HỌC
SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
HCMC University of Technology and Education
HCMUTE

KHOA CÔNG NGHỆ THÔNG TIN

MÔN: LẬP TRÌNH MẠNG

BÁO CÁO ĐỒ ÁN CUỐI KỲ

***

PCAP PACKET ANALYZER SỬ DỤNG CREWAI

GVHD: NGUYỄN ĐĂNG QUANG

SVTH:

1. Nguyễn Ngọc Đông Phương 22162033
2. Nguyễn Quỳnh Hương Quyên 22162036

Mã lớp: 242NPRO430980_01

Thành phố Hồ Chí Minh, Tháng 5 Năm 2025

1. **Hoạt động của PCAP Packet Analyzer**

- Tự động phân tích file **pcap_normal.pcap**, phát hiện giao thức, địa chỉ IP nguồn và đích, tổng số packet bắt được
- Thống kê và liệt kê các kết nối giữa địa chỉ IP nguồn và đích kèm theo giao thức của các host duy nhất
- Đếm số lượng và liệt kê danh sách host
- Hiển thị danh sách kết nối của các packet sau đó thông kê các giao thức có bao nhiêu gói tin
- Vẽ biểu đồ mạng (networkx) thể hiện kết nối giữa các host, biểu đồ tròn cho phân bố giao thức, bản đồ nhiệt (heatmap) cho sống lượng kết nối giữa các host
- Chạy CrewAI: Tạo và thực thi một "Crew" gồm các Agent, mỗi Agent thực hiện một nhiệm vụ (task) tương ứng với các bước trên → Các thông tin sẽ được lưu lại vào biểu đồ cột và biểu đồ mạng dưới dạng dạng file ảnh.

2. **Tool, Agent**

2.1. Tool

Sử dụng các hàm Python làm công cụ xử lý logic chính, các hàm sẽ đóng vai trò tương tự như tool – còn được gọi là callback của Task:

- **extract_packet_info(pcap_file):**
  - *Đặc điểm:* đọc file PCAP, trích xuất thông tin gói tin gồm: IP nguồn, IP đích, giao thức
  - *Tham số đầu vào:* ***pcap_file: str***
  - *Package sử dụng:* ***scapy.all (rdpcap, IP, TCP, UDP, ICMP*)** – thư viện mạnh để phân tích gói tin mạng từ PCAP; cung cấp API dễ sử dụng cho việc kiểm tra và truy xuất header IP, TCP, UDP, ICMP
  - *Kết quả đầu ra:* ***List[Dict]*** - Danh sách gói tin đã trích xuất
- **analyze_hosts(packet_data):**

- *Đặc điểm*: Phân tích số host duy nhất, liệt kế các kết nối giữa IP nguồn và IP đích
- *Tham số đầu vào*: ***task_output: List[Dict]*** hoặc ***TaskOutput***
- *Package sử dụng:* ***collections.Counter, networkx*** – Dùng Counter và networkx để gom kết nối và chuẩn bị cho vẽ biểu đồ mạng sau này
- *Kết quả đầu ra*: ***Tuple[List[str], List[Tuple[str, str]]]***

- **analyze_protocols(packet_data):**
  - *Đặc điểm*: Thống kê số lượng gói tin theo từng giao thức (TCP, UDP, ICMP, Unknown) và vẽ biểu đồ cột.
  - *Tham số đầu vào*: ***task_output: List[Dict]*** hoặc ***TaskOutput***
  - *Package sử dụng:* ***collections.Counter, matplotlib.pyplot*** – Counter giúp đếm số lần xuất hiện của giao thức. matplotlib để trực quan hóa dữ liệu bằng biểu đồ cột.
  - *Kết quả đầu ra*: ***collections.Counter***
- **draw_protocol_bar(counts):**
  - *Đặc điểm*: Vẽ biểu đồ cột từ dữ liệu giao thức đã thống kê.
  - *Tham số đầu vào*: ***protocol_counts: Counter***
  - *Package sử dụng:* ***matplotlib.pyplot*** – Trực quan hóa số lượng gói tin theo giao thức giúp phát hiện bất thường, phổ biến trong phân tích mạng.
  - *Kết quả đầu ra*: File ảnh được lưu trong thư mục của chương tình ***(protocol_bar.png)***
- **draw_network_graph(hosts, edges):**
  - *Đặc điểm*: Vẽ sơ đồ mạng từ các kết nối giữa host.
  - *Tham số đầu vào*: ***hosts: List[str], edges: List[Tuple[str, str]]***
  - *Package sử dụng:* ***networkx, matplotlib.pyplot*** – giúp vẽ biểu đồ có hướng giữa các host, rất phù hợp cho mô hình mạng.
  - *Kết quả đầu ra*: ***bool*** – Trạng thái thành công
- **generate_visualization_code(...):**

- *Đặc điểm*: Thực thi mã Python để vẽ các biểu đồ mạng nâng cao: network graph, pie chart, heatmap
- *Tham số đầu vào*: ***task_output, packet_data, hosts, edges***
- *Package sử dụng*: ***matplotlib.pyplot, networkx, collections.Counter, numpy*** – Sử dụng nhiều kỹ thuật trực quan hóa để trình bày dữ liệu mạng nâng cao: pie chart phân bố giao thức, heatmap thể hiện mức độ kết nối giữa host.
- *Kết quả đầu ra:* ***bool*** – Trạng thái thực thi

- **call_deepseek(prompt):**
  - *Đặc điểm*: Gửi prompt tới DeepSeek API và nhận phản hồi văn bản.
  - *Tham số đầu vào*: ***prompt: str***
  - *Package sử dụng*: ***litellm.completion*** – Giao tiếp với mô hình LLM (DeepSeek) để mô phỏng trả lời thông minh hoặc tạo mã, tự động hóa phân tích..
  - *Kết quả đầu ra:* ***str*** – Nội dung trả lời từ DeepSeek

2.2. Agent

- **Packet_extractor**
  - *Vai trò*: Trích xuất thông tin IP và giao thức từ file PCAP.
  - *LLM:* DeepSeek (deepseek/deepseek-chat).
- **Host_analyzer**
  - *Vai trò*: Đếm host duy nhất và liệt kê kết nối giữa IP nguồn-đích.
  - *LLM:* DeepSeek
- **Protocol_analyzer**
  - *Vai trò*: Thống kê và trực quan hóa phân bố giao thức.
  - *LLM:* DeepSeek
- **Visualization_agent**
  - *Vai trò*: Tạo các biểu đồ nâng cao từ dữ liệu phân tích.
  - *LLM:* DeepSeek

2.3. Tương tác giữa các Agents

Trong đoạn mã, các Agent trong hệ thống sử dụng CrewAI tương tác với nhau theo cách **tuần tự (sequential)**, nhưng có sự chia nhỏ nhiệm vụ (modular): Các Task được định nghĩa và tổ chức theo thứ tự **phụ thuộc lẫn nhau**:

1. **extract_task** (Agent: packet_extractor)
   - Trích xuất thông tin gói tin từ file PCAP. Đây là task gốc, cung cấp dữ liệu đầu vào cho các task sau.
2. **host_task** (Agent: host_analyzer)
   - Phân tích danh sách các gói tin đã được trích xuất từ extract_task. **Phụ thuộc** vào extract_task.
3. **protocol_task** (Agent: protocol_analyzer)

- Phân tích các giao thức từ dữ liệu extract_task. **Phụ thuộc** vào extract_task.

4. **visualization_task** (Agent: visualization_agent)
   - Tổng hợp kết quả từ extract_task, host_task và protocol_task.

3. Kết luận

3.1. Đánh giá kết quả

- *Độ chính và chi tiết*: Đọc được file PCAP và trích xuất đầy đủ thông tin của một gói tin cơ bản (IP nguồn, IP đích, giao thức). Thống kê số lượng host, packet cũng như đường đi của các gói tin giúp phân tích được lưu lượng mạng một cách khá chính xác
- *Trực quan hóa dữ liệu:* sử dụng biểu đồ mô tả các giao thức xuất hiện trong PCAP kết hợp với sơ đồ mạng (network graph) thể hiện các host và kết nối giúp nắm bắt và phân tích lưu lượng dễ dàng, trực quan
- *Tích hợp CrewAI:* Sử dụng mô hình AI giúp tự động hóa việc phân tích, trả lời thông minh, hỗ trợ người dùng trong việc khai thác dữ liệu linh hoạt.

3.2. Hướng phát triển

- *Phân tích nâng cao:*

- Tích hợp kỹ thuật phân tích payload để phát hiện malware, intrusion detection.

- *Xử lý hiệu năng:*
  - Tối ưu đọc và xử lý file PCAP lớn (chia nhỏ file, xử lý đa luồng).
  - Phát triển hệ thống realtime giám sát lưu lượng mạng.