--- Tuần 1 ---

Bức tranh tổng quan:

Mục tiêu của đề tài này là xây dựng được mô hình triển khai tài nguyên đánh lừa dựa trên ba thuật toán học tăng cường khác nhau: DDQN, SARSA và PPO với đầu vào là trạng thái quan sát được + topology mạng dưới dạng ma trận như thầy gợi ý, so sánh và đưa ra nhận xét giữa ba mô hình này về độ chính xác, tốc độ cũng như tài nguyên sử dụng. Đưa ra phương án cải thiện tốc độ chạy của từng thuật toán.

Cách xử lí dữ liệu

Mô hình mạng được ghi nhận bằng tay để cho ra file spreadsheet (csv) thống kê thành TPG tương ứng. CSV này sẽ được code đánh giá/inference chuyển thành ma trận và kết hợp với trạng thái quan sát được (ma trận bậc 1) để tạo ra đầu vào cho model. Model nhận ma trận, kết hợp thuật toán epsilon greedy và mạng neuron để đưa ra dự đoán.

Mô hình mạng -> csv chứa TPG -> dict TPG Input model = observation array + TPG matrix

Mô hình dự kiến:

Người dùng thực hiện chuyển đổi mô hình mạng hiện tại thành file bảng thống kê CSV cùng các lỗ hổng CVE có thể có. CVE trong CSV nhận được sẽ được gán EPSS để xác định khả năng có thể bị sử dụng bới attacker để tấn công vào hệ thống. Server lắng nghe thông tin từ NMS sẽ được khởi tạo với đầu vào là thông tin sơ lược của hệ thống mạng (node quan trọng - nicr, node bình thường n, node đánh lừa nếu có - nifr), thực hiện lắng nghe log gửi về từ hệ thống NMS và phân loại theo loại CVE, trả về ma trận trạng thái quan sát được (state) cùng phần thưởng (reward) dựa trên trạng thái quan sát được. Model RL sẽ được khởi tạo, nhận đầu vào là state + reward và topology mạng theo gợi ý của thầy để tiến hành dự đoán (inference) hành động phù hợp. Kết quả trả về sau quá trình inference sẽ là ma trận triển khai (là ma trận nhị phân với một số 1 duy nhất trên mỗi hàng, tượng trưng cho vị trí thiết lập honeypot, nếu có hai hàng trùng lặp, chỉ đặt một honeypot duy nhất tại vị trí đó). Ma trận triển khai này sẽ được gửi đến server triển khai vừa khởi tạo, server triển khai được cấp sẵn một số câu lệnh/script/yaml docker cho phép triển khai honeypot image lên một node bình thường tùy theo loại node đó. Nhận được ma trận, server triển khai sẽ so khớp vị trí và gửi yaml đến docker engine để tạo honeypot và kêt nối mạng vào node bình thường. Thông tin trong mạng sẽ được NMS tiếp tục ghi nhận và gửi về cho server lắng nghe NMS.

Thuật toán dự kiến sẽ thử nghiệm:

SARSA, DDQN, PPO

Quy trình

Quy trình thiết lập tài nguyên đánh lừa gồm năm bước

- Prepare Data: nhận mô hình mạng dưới dạng csv, quét lấy EPSS và chuẩn bị TPG thành csv cho mô hình học máy

- Start Simulation Network: Khởi tạo mạng mô phỏng, ghi nhận log của console sau khởi tạo

- Start NMS listening: Bắt đầu theo dõi mạng, output ra kết quả log nhận được từ NMS đã chỉnh sửa thành observation + reward

- Start model: Khởi chạy các model: DDQN, SARSA, PPO. bắt đầu đưa ra dự đoán từ DataPrep TPG csv và NMS-Retrieval csv. Output kết quả dự đoán.

- Start Deployment Action: Bắt đầu chạy server deploy (chạy async với model), nhận JSON packet model gửi tới gồm dự đoán, lấy dự đoán đó đem so với node. Tùy theo loại node sẽ lấy một honeypot chuẩn bị sẵn đi deploy.

Công việc đã thực hiện tuần 1:

- Chỉnh sửa, hoàn tất công cụ tạo môi trường huấn luyện TPG tự động

- Hoàn thành code cho agent DDQN, đánh giá được tài nguyên tiêu thụ và độ chính xác.

- Làm quen với code cho các agent PPO và SARSA.

Khó khăn

Thuật toán DDQN hiện tại tiêu tốn quá nhiều tài nguyên khi số lượng node lớn hơn 10 cũng như độ chính xác chưa cao, cần đưa ra các giải pháp học tăng cường/cải thiện DDQN để tăng độ chính xác dự đoán và giảm tài nguyên cần thiết. Ngoài ra, nếu số tài nguyên honeypot được cấp quá ít, model cũng không thể triển khai và đưa ra dự đoán chính xác/crash.