* Chuẩnhóacác alert từcác data sources
* Cáchkếthợpcác alert thành hyper-alert saukhiđãchuẩnhóa(aggregation)
* Cáchxácđịnh pre-post củacác hyper-alert
* ~~Cáchkếthợpcác hyper-alert tạothành attack scenario~~
* Giải thuật kết hợp các hyper-alert thành attack scenario bằng tiền đk và hậu đk.
* Cáchtrừutượngcácbướctrong scenario thànhcác class
* Cáchkếthợp ontology vàohệthống
* Ngônngữđặctả attack tree
* ~~Các match giữa~~ giải thuật so trùng causal network và attack tree
* Cácphươngpháp visualize
* Cácthông tin cầncungcấpđể visualize
* ~~Xácđịnhlỗhổngbảomậtcủahệthống~~
* System scan
* phát hiện lỗi của hệ thống
* - nghiên cứu cách lấy alert từ IDS và từ host server -> tổng hợp lên hype alert
* - nghiên cứu cách xây dựng attack tree từ database và từ kết quả của scanning trên server
* - nghiên cứu cách xây dựng module để phát hiện tấn công (matching) và suy luận bước tiếp theo của attacker(ontology)
* - nghiên cứu cách hiển thị thông tin từ 2 nguồn : từ raw data ( của IDS và server ) , từ hoạt động tấn công .
* Alert Correlation:
  + Nghiên cứu những thành phần cơ bản của một module Alert Correlation nhất định
  + Tạm thời chia theo các phần sau:
    - Phần tiếp nhận và chuẩn hoá các alert từ IDS:
      * <1>Cấu trúc output của Snort
      * <2>Cấu trúc output của Windows Event logs
      * <3> Phần tiếp theo có hai hướng đi:
        + Định nghĩa một cấu trúc riêng để đồng bộ các alert từ <1>,<2> ,sau khi chuyển đổi sang format mới , các module trên sẽ thao tác trực tiếp trên định dạng dữ liệu mới này(nếu chọn lựa theo cách này thì ví dụ như việc kết hợp các alert trong một khoảng thời gian nhất định chỉ cần xây dựng một module)
        + Xây dựng hai module phân tích từng định dạng alert từ <1>,<2> chuyển trực tiếp sang các predicate lưu vào cơ sở dữ liệu để các module trên xử lý(với hướng này thì ta phải xây dựng 2 module tạo predicate từ 2 nguồn khác nhau…)
        + <Ai cho ý kiến chỗ này thử hén , chỉ là suy nghĩ của riêng tui>
    - Phần Alert aggrelation :<nghiên cứu về cách thức nhóm các alert trong khoảng thời gian nhất định , giảm trùng lắp …> ,hệ thống các predicates cần thiết
    - Phần kết hợp những Alert đã qua xử lí ở phần dưới tạo các hyper-alert, hệ thống các hyper-alert cần thiết
    - Phần chính giải thuật sâu chuỗi các hyper-alerts , xử lý trường hợp IDS báo cáo thiếu về các bước tấn công hiện thời => tạo các chuỗi hyper-alerts <attack scenarios>
    - Xác định định dạng output cho toàn bộ module <sử dụng CAML hay XML….>
* Plan recognition :
  + Nếu ở đây ta tổng quát them một bước nữa như C.Thắng nói thì cần nghiên cứu cụ thể:
    - Hệ thống các bước chuẩn bao hàm các loại tấn công khác nhau để đưa ra các khái niệm trong ontology hợp lý
    - Ngoài ra tìm hiểu cách thức để tổng quát causal network cung cấp bởi alert correlation module lên .
  + Cơ chế matching :
    - Giải thuật matching .
    - Giải thuật dự đoán bước tiếp theo
    - Các phần cần them vào để đáp ứng yêu cầu giải thuật <ví dụ như áp dụng xác suất để dự đoán bước tiếp theo thì phải xem xét ngôn ngữ định nghĩa lỗ hổng bảo mật có hỗ trợ định nghĩa các giá trị xác suất tương ứng >
* Xây dựng database lỗi:
  + - Chọn lựa ngôn ngữ đặc tả phù hợp với phần mềm chúng ta
    - Giải thuật phân giải tạo nên attack tree từ các đặc tả
    - <Sử dụng lại giải thuật tổng quát hoá bên trên nếu cần tổng quát them một tầng nữa>
    - <Về phía database có cần thiết nghiên cứu đến các kỹ thuật hỗ trợ việc lưu trữ lượng dữ liệu lớn ,động,truy xuất thường xuyên không???>
* Visualization
  + Quyết định mức độ chi tiết ,cầu kỳ của phần visualize ^^
  + Các kỹ thuật , yêu cầu cơ bản của visualization
* System Scan Module:
  + Kỹ thuật scan:
    - Scan cái zì , scan như thế nào
  + Database:
    - Định dạng lưu trữ những thong tin scan được
* Bữa giờ chúng ta chưa đề cập đến việc cung cấp các khả năng phản ứng của phần mềm trước một số lỗi tấn công , nếu có xây dựng thì cần nghiên cứu cơ chế , hướng giao tiếp với admin , với tool trực tiếp thực hiện phần ngăn chặn…..

# Issues

* Chọn ngôn ngữ biểu diễn alert
* Kỹ thuật abstract alert thành các hyper-alert
* Liên kết các hyper-alert thành causal network
* ~~Chọn kĩ thuật để expert đặc tả các attack trees~~
* Cách xây dựng attack tree vào database
* Chọn giải thuật để matching giữa causal network và attack trees.
* Chọn các giải thuật để suy luận ra hành động tấn công kế tiếp
* Visualize những thông tin có được (thông tin chưa xử lý và những cảnh báo, các thông tin đã được xử lý)
* Tìm hiểu các lỗi bảo mật, chọn framework phù hợp để scan lỗi

Issues

* Alert correlation:
  + Alert format (recommend IDMEF).
  + Alert aggregation into hyper-alert.
  + Connect hyper-alerts to graph.
  + Abstract nodes of a graph to general class 🡪 to make causal network.
* Security flaw (attack tree):
  + How a professional define a flaw as an attack tree.
  + How to match causal networks (from above) and attack trees (in database).
* System scan
* Visualization