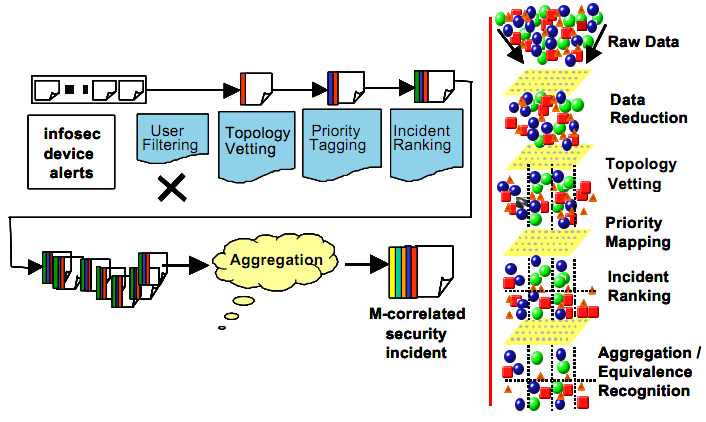
A mission-impact-based Approach to INFOSEC Alarm Correlation

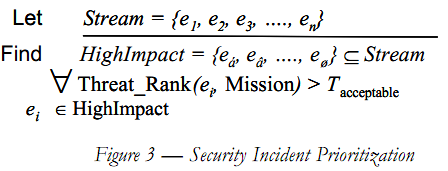
1. Introduction
   * Tác giả giới thiệu các nguồn của INFOSEC (information security) như firewall, IDS,antivirus ... (Số nguồn > our số nguồn hiện tại)
   * Đề ra yêu cầu cần phải tổng hợp lại
   * Nêu ra 1 số lí do tại sao cần phải tổng hợp (có thể dùng để chém gió trong paper của mình – plagiarism)
     + Số lượng chuyên gai trên lĩnh vực ít mà alert thì thông tin ít => cần phải có cơ chế tăng tính abstract của alert
     + Số lượng alert lớn nên time cần nhiều để analyze =>
     + Thiếu cơ chế phối hợp giữa các INFOSEC source
   * Tác giả xây dựng 1 Mission Impact Intrusion Report Correlation System (gọi tắt là M-Correlator)
   * Về mission impact based: mission impact analysis thật ra dùng để kết hợp các alert liên quan (correlate) để tạo alert ở mức tổng quát hơn và rank mấy cái alert tổng quát này
2. Các giải thuật của M-Correlator



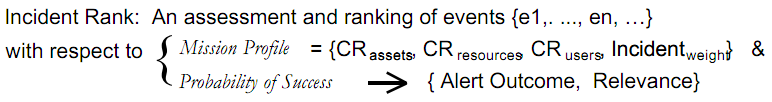
Hình vẽ quá xấu :|

* + Các bước thực hiện:
    - Dynamic alert controllable filters: bỏ các alert không quan trọng (làm thế nào bit cái nào là quan trọng???)
    - Kiểm tra alert với topology của mạng để xác định mức độ liên quan(relevance score): kiểm tra với các vul đã biết
      * Các vul này được nhập vào thông qua Incident Handling Fact Base
    - Tính toán priority (tren từng alert)
    - Tính toán Incident rank dựa trên các thông số trên và đưa ra xác suất thành công
    - Cluster các alert liên quan dựa trên giải thuật clustering dựa trên attr (cái này tác giả nói là correlate thật ra chỉ là aggregate như trong định nghĩa của mình)

1. Incident handling fact base (step 0)
   * Nhiệm vụ:
     + Kết hợp các incident với các thông tin mô tả
     + Các thông tin về incident như OS version, hardware platform
     + Các incident nào có thể merge bởi giải thuật clustering
   * Các field của fact base
     + Incident code: mã được lấy từ CSDL CISL
     + COTS code: mã lấy từ các soft khác
     + Incident class:
     + Vul OS and Hardware: các OS và hardware cần thiết cho incident
     + Bound ports and app: các app và port cần thiết cho incident:
     + Cluster list: dùng để cluster lại dựa trên cùng 1 cluster name
     + Refernce: no7is ẽ lấy mã incident code về.
2. Relevance information (step 2)
   * M-Correlator có topology của network được xây du75gn bởi expert
     + Có thể được xây dựng bởi NMap
   * Mỗi alert sẽ có 1 số field và sẽ được match với thông tin trong fact base
     + OS type and version
     + Hardware type
     + Service suite
     + Enabled network service
     + Application
     + Không nói rõ cách tính như thế nào: chỉ nói alert có thể có relevant score từ 0 – 255
3. Tính toán priority (step 3)
   * Mission là nhiệm vụ mà hệ thống đó được xây dựng để thực hiện
   * Mission được định nghĩa thông qua mission specification: gồm 2 phần
     + Thông tin về các service và data asset quan trọng nhất đối với mission như
       - Các thông tin nhạy cảm
       - Các service quan trọng
       - Các account nghi ngờ
     + Thông tin về những incident nguy hiểm nhất mà admin quan tâm
   * Một số thành phần quan trọng của mission specification sẽ được dùng để tạo interest profile như class của incident,
   * Mission impact analysis sẽ giúp admin tìm incident nào nguy hiểm nhất
   * Cách tính



* + Các alert đã có incident class và sẽ được match với interest profile
  + Tác giả có định ra 1 số incident class cơ bản:
    - Privilege Violation
    - DoS
    - Probe
    - Access Violation
    - ...
  + Paper có phần ví dụ về mission specification

1. Incident rank analysis:
   * Dùng để đánh giá incident dựa trên
     + Ảnh hưởng của incident trên profile được thể hiện thông qua priority caculation ???
     + Khả năng của incident đó đã thành công
       - Lí do là vì IDS không hỗ trợ việc đánh giá incident đã thành công hay chưa
   * Cách tính
   * 
2. Alert clustering algorithm:
   * M-Correlator sử dụng giải thuật clustering để gom các Netowrk và host alert lại.
   * Các Network alert s4 được gọp lại dựa trên các thuộc tính như IP, port....
   * Các Host alert sẽ được gọp lại dựa trên process ID, user ID...
   * M-Correlator sẽ thiết lập các policy
     + Khi có 1 incident mới, M-Cor sẽ kiểm tra xem incident đó có thuộc policy đó không
   * 1 Policy :

Profile Cross\_Sensor\_Signature\_And\_Session\_Match

Policy Liberal

Candidate\_If [ OR

[ IN\_GROUP observer\_name Network\_Sensors ]

[ AND

[ IN\_GROUP observer\_name Host\_Sensors ]

[ NOT

[ AND

[ NULL source\_pid ]

[ NULL source\_username ]

]

]

]

]

Match\_If [ AND

[ EQ incident\_signature ]

[ NULL\_OR\_EQ observer\_stream ]

[ OR

[ AND

[ IN\_GROUP observer\_name Network\_Sensors ]

[ ELEMENTS\_EQ source\_IParray ]

[ ELEMENTS\_EQ target\_IParray ]

]

[ AND

[ IN\_GROUP observer\_name Host\_Sensors ]

[ ELEMENTS\_EQ target\_IParray ]

[ NULL\_OR\_EQ source\_pid ]

[ NULL\_OR\_EQ source\_username ]

]

]

]

Delay\_Until\_Expire 600

Delay\_Until\_Flush 90

Initial\_Flush\_Delay 90

Enable true

Unique\_Match true

Merge\_Action fuse