Constructing Attack Scenarios through Correlation of Intrusion Alerts

Keywords: IDS, Alert Correlation, Attack Scenarios

Paper này giới thiệu về framework dung cho alert correlation, trình bày cách hiện thực giai đoạn off-line và đánh giá phương pháp này dựa trên tập dữ liệu 2000 DARPA intrusion detection scenario.

**Tóm tắt về framework:** framework này dựa trên sự quan sát là trong 1 chuỗi các tấn công, thì các thành phần của 1 cuộc tấn công không tách biệt nhau mà có lien quan và tạo thành các bước của cuộc tấn công đó, và từ đó đưa ra ý tưởng là correlate alerts sinh ra bởi IDSs bằng cách sử dụng prerequisites và consequences. Giống với mô hình của JIGSAW nhưng có một số khác biệt + cải tiến. Prerequisites và Consequences được xây dựng từ các thành phần cơ bản là predicates và predicates có thể được kết hợp để biểu diễn các điều kiện phức tạp bằng phép toán logic conjunction và disjunction.

**Hyper-alert type T**: là một bộ ba (fact, prerequisite, consequence) biểu diễn tri thức về một kiểu tấn công.

* Fact: tập hợp tên các thuộc tính + miền giá trị, chỉ ra các loại thông tin được report trong alert
* Prerequisite + Consequence: biểu thức logic của predicates trong đó các biến nằm trong fact. Prerequisite là các điều kiện cần phải đúng để cho tấn công thành công còn Consequence là những hành động có thể xảy ra nếu cuộc tấn công thành công.

**Hyper-alert h**: là 1 thể hiện của 1 hyper-alert type. Hyper-alert h được ngầm định là prerequisite là true và tất cả predicates trong consequence sẽ là True. Hyper-alert thể hiện các prerequisite và consequence của nó bằng cách thay thế các biến tự do bằng các giá trị cụ thể

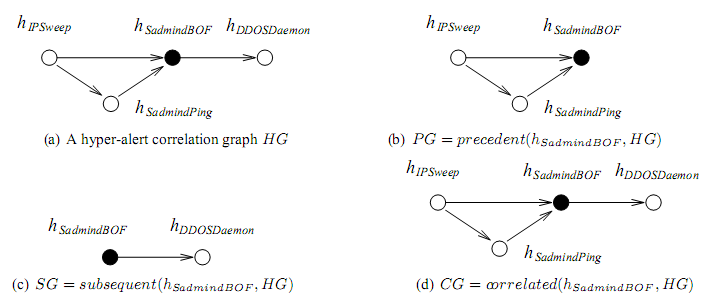
**Hyper-alert h1 prepares for hyper-alert h2** nếu tồn tại p ϵ P(h2) : prerequisite set of h2 và tập C là tập con của C(h2) trong đó tất cả các c thuộc C thì *c.end\_time < p.begin\_time* và conjunction của tất cả predicates trong C suy ra p.

**Ràng buộc về thời gian cho hyper-alerts**: vì framework cho phép nhiều alerts có thể aggregated thành 1 hyper-alert nên cần sử dụng rang buộc này để aggregation tạo thành 1 hyper-alert có nghĩa hơn đồng thời để giảm unreasonable aggregation. Có 2 mức rang buộc về thời gian:

* *Duration constraint D*: một hyper-alert h thoả mãn duration constraint D nếu Max{t.end\_time} – Min{t.begin\_time} < D
* *Interval constraint I*: một hyper-alert h thoả mãn interval constraint I nếu
  + h chỉ có 1 tuple
  + tất cả các t trong h, phải tồn tại một t’ khác t trong h thoả mãn: t.begin\_time < T < t.end\_time, t’.begin\_time < T’ < t’.end\_time và |T – T’| < I

**Hyper-alert correlation graph** HG = (N,E) là một đồ thị có hướng, trong đó các đỉnh là hyper-alerts, với mỗi cặp đỉnh n1, n2, có một cạnh nối từ n1 tới n2 nếu và chỉ nếu hyper-alert biểu diễn bởi n1 prepares for n2.

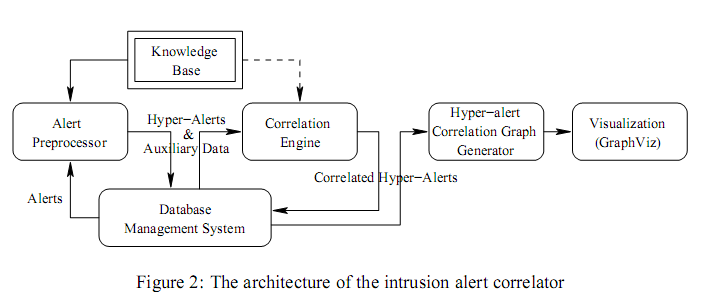
Ngoài ra còn có một số định nghĩa lien quan tới correlation graph như precedent (n, HG), subsequent (n, HG) và correlated (n, HG). Chú ý, trong thực tế không phải tất cả hyper-alerts đều liên quan tới hyper-alert n cần quan tâm nên HG với correlated (n, HG) có thể khác nhau.



Nhận xét:

* Cung cấp biểu diễn ở mức high-level của correlated alerts
* Có thể giảm thiểu ảnh hưởng bởi các cảnh báo sai.
* Không phụ thuộc vào predefined attack scenarios. (giống với JIGSAW)
* Có thể correlate alerts thậm chí khi IDSs miss.
* Nếu như critical attack lien kết 2 stage mà bị miss thì sẽ tạo thành 2 correlation graphs tách biệt nhau. Đồng thời phương pháp này cũng không hiệu quả cho trường hợp correlate alerts mà không có quan hệ prepare-for.

**HIện thực:**



Chương trình hiện thực được viết bằng Java với kết nối JDBC và sử dụng GraphViz để trực quan thông tin. Knowledgebase bao gồm các thông tin cần thiết về *hyper-alert types* và *các quan hệ dẫn xuất giữa predicates*, lưu trữ bằng XML file. Khi chương trình khởi động thì nó sẽ đọc XML file, chuyển đổi sang thông tin rồi lưu trữ trong KB. Tất cả alerts cảnh báo từ IDSs sẽ được lưu trữ trong DB.

Khi alert correlator đọc hyper-alert types, nó sẽ tạo ra những tập hợp prerequisite và consequence của mỗi hyper-alert type. Sau đó, nó mở rộng tập hợp consequence bằng cách thêm vào predicates trong KB được dẫn xuất từ tập hợp consequence đó.

Alerts được report từ IDSs sẽ được alert preprocessor xử lý để tạo ra hyper-alerts. Thể hiện của predicates được biểu diễn dưới dạng string. Tập hợp của prerequisite và consequence được lưu trữ trong 2 tables là PrereqSet và ExpandedConseqSet, với các thông tin: HyperAlertID, EncodedPredicate, begin\_time, end\_time.

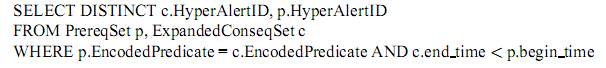
***PrereqSet***

|  |  |  |  |
| --- | --- | --- | --- |
| HyperAlertID | EncodedPredicate | Begin\_time | End\_time |

***ExpandedConseqSet***

|  |  |  |  |
| --- | --- | --- | --- |
| HyperAlertID | EncodedPredicate | Begin\_time | End\_time |

Sử dụng truy vấn sau có thể lấy được kết quả của alert correlation.



Trong paper có chứng minh là hiện thực trên sẽ lấy được tất cả và chỉ những cặp hyper-alert có sự chuẩn bị cho nhau.

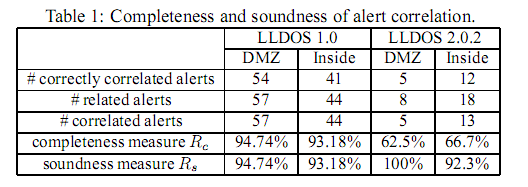
**Đánh giá kết quả:**

Sử dụng 2 tập dữ liệu của 2000 DARPA intrusion detection scenario specific datasets là LLDOS 1.0 và LLDOS 2.0.

Ở đây đưa ra 2 công thức để đánh giá completeness và soundness trong việc nhận dạng tấn công:



Và đây là kết quả của phương pháp này dựa trên 2 công thức trên:

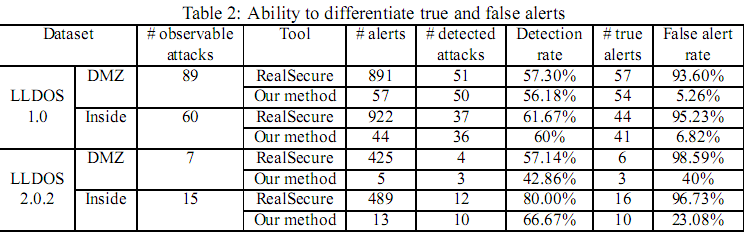


Ngoài ra, ở đây còn đánh giá hệ thống này dựa vào khả năng phân biệt giữa false alerts và true alerts.

Để xác định mức độ nhận diện dung công thức: 

Còn xác định mức độ false alert dung công thức: 

Kết quả đánh giá trên:



Paper còn có 2 Appendixes:

1. Trình bày về hyper-alert types trong ứng dụng:

* Không sử dụng predicates quá tổng quát
* Để tăng performance cho chương trình, trong prerequisite/ consequence của một hyper-alert type, không đưa tất cả predicates vào mà chỉ đưa vào predicates mà cuộc tấn công dựa chính vào nó hoặc dẫn đến điều đó. Từ đó dẫn đến 1 việc cần nghiên cứu là cần chọn predicate nào và bỏ cái nào?
* Có một số dẫn xuất giữa predicates không phải là sự thật. (ví dụ: OSSolaris(DestIP) 🡪 OSUNIX(DestIP)). Các dẫn xuất này không phải lúc nào cũng đúng, nhưng nếu không có thì sẽ dễ dẫn đến mất cơ hội để correlate alerts khi mà những thông tin tấn công không được tìm thấy bởi IDSs.

1. Hyper-alert Correlation Graphs: giới thiệu các đồ thị của chương trình trên 3 tập dữ liệu thử nghiệm khác.

Câu hỏi :

1. Trong câu Query để thực hiện việc tìm các cặp h1, h2 , tại sao điều kiện lại là p.EncodedPredicate = c.EncodedPredicate, theo paper thì đây là compare String, tức là 2 cái predicate này giống nhau như đúc thì mới match được. Nhưng cái thực chất chỉ cần cái Prerequisite có nằm trong tập ExpandedConsequence là dc rồi.

Ans : mỗi hyper-alert có thể có nhiều hàng trong bảng, tức là có thể có nhiều HyperAlertID. Khi kiểm tra điều kiện như thế thì chỉ có 1 predicate của prerequisite hoặc expanded consequence được so sánh thôi. Nên kết quả vẫn là như ông nói.