TIAA source code studying

Đây là bài báo cáo về việc tìm hiểu các hiện thực của TIAA thông qua tìm hiểu source code của chương trình TIAA (v1.0).

Người báo cáo: TranThi (ntt.hero)

Nội dung của báo cáo bao gồm

[Tổng quan về chức năng của TIAA 1](#_Toc274395738)

[Hiện thực cơ sở dữ liệu của TIAA 2](#_Toc274395739)

[Phân chia module và chức năng của các module trong TIAA 2](#_Toc274395740)

[Hiện thực của các module trong TIAA 2](#_Toc274395741)

# Tổng quan về chức năng của TIAA

Chức năng chính của TIAA là tổng hợp các alert nhận được từ các IDS sensor có trong hệ thống được lưu trữ vào cơ sở dữ liệu hoặc trong các file XML bằng định dạng IDMEF. Sau đó, chương trình sẽ xử lý các dữ liệu này bằng cách sử dụng knowledgebase được định nghĩa sẵn trong 1 file XML để đưa về các kiến thức có thể hiểu được bởi chương trình. Sau khi các dữ liệu đã được xử lý và các thông tin cần thiết được trích xuất và lưu trữ vào cơ sở dữ liệu, chương trình sẽ hiện thực các thao tác xử lý để chuyển đổi các thông tin đó thành các attack graph để hiển thị cho người dùng.

Ngoài ra, chương trình còn hỗ trợ các tiện ích phân tích để giúp người dùng có thể dễ dàng phân tích các thông tin có được và đánh giá chính xác hơn tình hình hiện tại của hệ thống. Các tiện ích này bao gồm:

* Focused Analysis
* Clustering Analysis
* Aggregation/ disaggregation analysis
* Link Analysis
* Association Analysis
* Attack Strategy Extraction
* Missed Attack Hypotheses

Sau khi người dùng cấu hình các thông tin cần thiết cho 1 project mới như knowledgebase, các nguồn dữ liệu,… chương trình sẽ tự động tạo ra dữ liệu cần thiết rồi lưu vào cơ sở dữ liệu để dùng trong lúc chạy của chương trình.

Sau khi chuẩn bị xong các dữ liệu cần thiết, người dùng có thể xem được các graph là kết quả correlate các alert. Các alert có thể được correlated bằng cách sử dụng database hoặc sử dụng các dữ liệu đặc biệt như heap, queue, Btree, AVLTree, … để tạo nên.

Trong quá trình phân tích, chương trình sẽ cho phép người dùng chọn các tiện ích phân tích và hiển thị các kết quả tương ứng với lựa chọn của người dùng.

# Hiện thực cơ sở dữ liệu của TIAA

Các bảng trong cơ sở dữ liệu của TIAA được chia thành các nhóm chính sau:

* Các bảng để lưu trữ các thông tin về knowledgebase: Các bảng này sẽ được tạo ra trong quá trình chạy của chương trình. Các dữ liệu knowledgebase được định nghĩa trong XML file sẽ được parse và lưu trữ vào các bảng này. Dữ liệu từ các bảng này sẽ được dùng trong quá trình xử lý các dữ liệu thu được từ IDS sensors. Gồm có các bảng:
  + HATFact

|  |  |  |
| --- | --- | --- |
| HyperAlertType | FactName | FactType |

* + HATPrereq

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| HyperAlertType | P\_ID | Predicate | ArgPos | ArgName |

* + HATConseq

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| HyperAlertType | P\_ID | Predicate | ArgPos | ArgName |

* + Predicate

|  |  |  |
| --- | --- | --- |
| Predicate | ArgNum | ArgType |

* + Implication

|  |  |  |  |
| --- | --- | --- | --- |
| Predicate | Implied | P\_Arg | I\_Arg |

* + HATProtocol

|  |  |
| --- | --- |
| HyperAlertType | Protocol |

* Các bảng để lưu trữ các alert thu được từ IDS sensors. Alerts cũng có thể được lưu trong các XML file. Dữ liệu từ các nguồn này sẽ được xử lý để đưa vào 1 bảng chung là AlertTable. Các table này được tạo từ đầu, và chương trình chỉ query để lấy dữ liệu
  + iphdr

|  |  |  |  |
| --- | --- | --- | --- |
| Sid | cid | Ip\_src | Ip\_dest |

* + event

|  |  |  |  |
| --- | --- | --- | --- |
| Sid | cid | Timestamp | Signature |

* + signature

|  |  |
| --- | --- |
| Sig\_id | Sig\_name |

* + tcphdr

|  |  |  |  |
| --- | --- | --- | --- |
| Sid | cid | Tcp\_sport | Tcp\_dport |

* + udphdr

|  |  |  |  |
| --- | --- | --- | --- |
| Sid | cid | Upd\_sport | Udp\_dport |

* AlertTable: lưu các dữ liệu về alerts. Có 2 bảng riêng cho alert table, 1 bàng lưu alerts từ snort, 1 bảng để lưu alerts từ real secure.

Snort alerts table: alert\_info (SensorID, AlertID, begin\_time, end\_time, AlertName, SrcIPAddress, DestIPAddress, SrcPort, DestPort).

Ngoài ra còn bảng events, chứa alerts thu được từ real secure tool, nhưng trong chương trình không dùng tới nó.

* Các table để lưu kết quả của correlation. Các table này được khởi tạo trong lúc thực thi chương trình. Các table này được dùng đến trong trường hợp người dùng chọn cách correlate là sử dụng database. Gồm có các table:
  + CorrelatedAlerts (PreparingHyperAlertID, PreparedHyperAlertID)
  + HyperAlert (HyperAlertID, HyperAlertType, begin\_time, end\_time)
  + PreqSet (HyperAlertID, EncodedPredicate, begin\_time, end\_time)
  + ConseqSet (HyperAlertID, EncodedPredicate, begin\_time, end\_time)
  + HyperAlertFact (HyperAlertID, factName1, factName2, …) (fact name được lấy từ HATFact table trình bày ở trên).
* HyperAlertAbstraction (HyperAlertType, AbstractedType, HyperAlertLevel, AbstractLevel) table này lưu thông tin kết quả về graph.
* HyperAlertMapping (OriginalName, HyperAlertName): table này lưu thông tin về tên của alert tương ứng với hyper alert mà nó được aggregated.

# Phân chia module và chức năng của các module trong TIAA

Chương trình có 6 module chia theo tính năng như sau:

* GUI: hiện thực phần user interface của chương trình. Cho phép người dùng tương tác với chương trình và xem kết quả.
* Correlation: Thực hiện correlate các alerts bằng cách sử dụng cơ sở dữ liệu. Các alerts lấy từ trong cơ sở dữ liệu sẽ được chuyển đổi, tạo các bảng kết quả rồi lưu các dữ liệu này vào các bảng kết quả. Để lấy được các thông tin kết quả cần thiết thì sẽ thực hiện các câu truy vấn vào cơ sở dữ liệu để lấy thông tin
* In\_mem\_correlation: Thực hiện correlate bằng cách sử dụng các giải thuật, cấu trúc dữ liệu đặc biệt, thực hiện trực tiếp trong memory của bộ nhớ.
* Tiền xử lý dữ liệu: dữ liệu thu được từ các nguồn dữ liệu (DB hoặc XML file) sẽ được chương trình xử lý để đưa về 1 loại chung dùng cho các phương pháp correlate khác nhau, các dữ liệu này được lưu vào trong DB.
* Các tiện ích: thực hiện các tiện ích được nêu ở trên để đưa ra kết quả tương ứng dùng cho quá trình hiển thị được thực hiện ở module GUI
* Reasoning About Missed Attack: hiện thực khả năng suy luận của chương trình về các missed attack bằng cách được trình bày trong papers.

# Hiện thực của các module trong TIAA