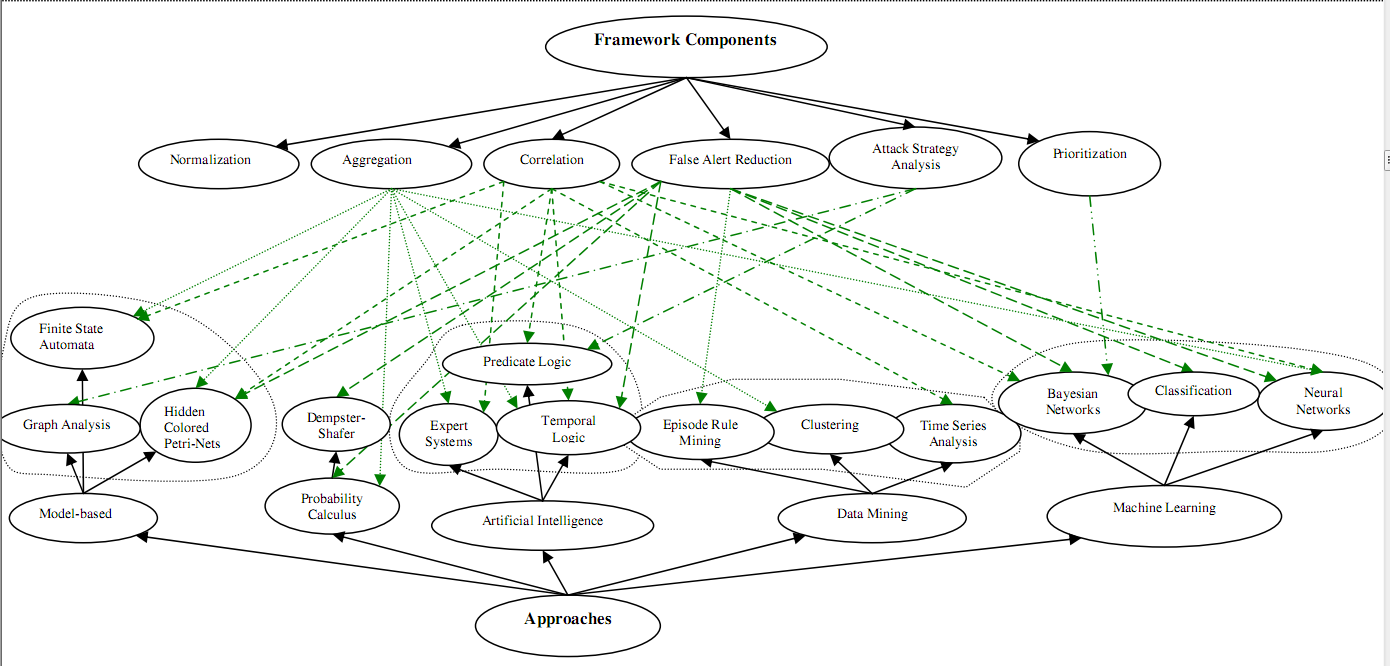
**Alert Correlation Survey: Framework and Techniques**

Keywords: survey, alert correlation, framework, security

In this paper, a framework with 6 components is proposed. These components consist of Normalization, Aggregation, Correlation, False Alert Reduction, Attack Strategy Analysis, Prioritization. The emphasize of this survey is to map those components with techniques that have been used in many frameworks, and to give a better understanding of this area.



Mapping between Framework Components and Approaches

1. Normalization:

* Translate every alerts from many different IDS into a standardized format.
* Intrusion Detection Message Exchange Format (IDMEF) data model is the most notable effort in standardizing alert formats.
* IDMEF is object-oriented and is implemented in XML.
* Main design goal: express relationships between alerts which is actually essential requirement of alert correltion -> this can be a good format for IDS alert.
* Issues:
  + Different IDSs use their own naming conventions to fill the fields of the alerts.
  + A common classification of attacks is required, but it still doesn’t exist.

1. Aggregation:

* Purpose: group all similar alerts together. “Similar” means that:
  + Alerts match in all attributes except with little time difference.
  + Alerts share the root cause.
* This paper use the extended view of aggregation. Similar alerts tend to have similar root causes or similar effects on network resources.
* In *Probabilistic Alert Correlation*: similarity is estimated based on similarity between alert attributes.
* In *Clustering instrusion detection alarms to support root cause analysis*:
  + Alerts is clustering based on their root causes. Root causes of an alert is the reason for which the alert has been raised.
  + Central to this techniques is hierarchy structures (generalization hierarchies), which decompose the attributes of alerts from the most general values to the most specific values.
  + The dissimilarity of two attributes (of two alerts, means that attribute A of alert 1 and attribute of alert 2) is defined to be the longest path between values of that attribute in the generalization hierarchy.
  + The dissimilarity of two alerts is defined to be the sum of the dissimilarity of their attributes.
* In *Managing alerts in a multi-intrusion detection environment*: the CRIM framework is designed, includes two components:
  + Alert clustering component: creates cluster of similar alerts using expert rules.
  + Alert merging component: creates the representative alert for a cluster.
* There are some other approaches like data mining, machine learning, a unified method.

1. Correlation:

* Purpose: find the causal relationships between alerts to reconstruct the attack scenarios. The role of this component is providing a highest level view of the actual attacks.
* Divided into 4 categories:
  + Scenario-based correlation: LAMBDA, STATL, ADeLe; alerts are correlated with each other if they can combined to construct attack scenario (scenario is defined by experts).
  + Rule-based correlation:
    - Attack sub-goals can be reached in different ways by using various exploits. So it’s hard to maintain a comprehensive attack database for the scenario-based correlation.
    - This method correlates alerts by matching prerequisites and consequences of attack steps.
    - Rule-based correlation can be considered as a special scenario-based correlation.
    - Example: JIGSAW, MIRADOR, hyper alert.
  + Statistical correlation:
    - In A *Probabilistic-Based Framework for INFOSEC Alert Correlation*: Bayesian network is used to model the causal relationship between alerts.
  + Temporal correlation:
    - Causality relationships between alerts are calculated using *Granger Causality Test*.
* The scenario-based and rule-based methods are restricted to knowledgebase.
* Statistical and temporal methods can correlate alerts which may contribute to unknown attacks (may learn new attacks). Besides, they seem to be much more time consuming comparing to knowledgebase methods in practice.
* Those are complementary to each other.

1. False Alert Reduction:

* Purpose: distinguish between false positive and true positive alerts.
* In *A data mining analysis of rtid alarms*: frequent episode rules have been used for mining normal alert patterns:

L1, L2, …, Lm -> R1, R2, …, Rn(s,c,w)

s: support c:confidence w:time window

Li’s represent the left-hand patterns which may trigger a rule.

Ri’s represend the right-hand patterns which are expected to be seen most frequently in a time window no larger than w.

Frequent episode rules encode normal sequence of alerts which are seen with adequate support and confidence and are suspicious of being false positives (support and confidence in association rule in data mining). These rules are generated in a learning process with an attack free traffic. In operation mode, sequence of alerts which match a frequent episode rule are consider as false positives.

* In *Using adaptive alert classification to reduce false positives in intrusion detection*:
  + A classification techniques has been used for false positive analysis.
  + 3 input: network topology, alert context (other similar alerts related to this alert), alert semantics (evidences on alert applicability based on the vulnerabilities of the target network).
  + A cost-sensitive variation of Ripper rule learner has been used as the classification algorithm which returns a confidence score for each of 2 output classes.
* In *Reasoning about complementary intrusion evidence*:
  + A framework that employs security state information of the system (system monitoring and vulnerability scanning tools) and alerts (IDS).
  + Causal relationshop between alerts and system attributes are modeled by alert-attribute graph. Nodes represent alerts or system attributes. Edges represent pre-postcondition relationship between alert and attributes.
  + Using Bayesian networks.
* In *Correlation of intrusion alarms with subjective logic*:
  + Aggregation and correlation of alerts using subjective logic.
  + Alerts come along with *belief, disbelief, uncertainty, relative atomicity*.
* In *A novel framework for alert correlation and understanding*:
  + Distinguish between attacker actions (transitions) and alerts (observables).
  + Attack scenarios based on pre-postcondition.
  + An algorithm for inference (deciding about most probable transitions in the model given a sequence of observables) is presented.

1. Attack Strategy Analysis:

* Purpose: reason about attacker’s actual intention. Input to this component is low-level correlated alerts.
* In *Building attack scenarios through integration of complementary alert correlation method*:
  + Using two-level correlation techniques:
    - Causality correlation: base on pre, post.
    - Similarity correlation: similarity between attributes.
  + Alerts are correlated through each of those 2 techniques.
  + Result of causality correlation: set of correlation graphs. They may have been separated by missed alerts by IDS.
  + Based on similarity correlation results, graphs which share similar alerts are analyzied -> add possibly missed alerts (virtual alerts) between 2 similart alerts if first alert can prepare for the second one.
* In *Attack plan recognition and prediction using causal networks*:
  + Raw alerts are correlated by using knowledgebase or statistical techniques.
  + Attack plans are maintained by Attack Trees.
  + For prediction, attack trees are converted to Bayesian Networks. In runtime, the Bayesian networks are used to compute the probabilities of goals and sub-goals of attacker.

1. Prioritization:

* Purpose: classify alerts based on their severity and take appropriate actions for dealing with each alert class.
* This component should take into account various domain information in addition to alert types. Eg: Security policy, network topology, vulnerability analysis of the network services and installed softwares…
* The focus of previous works in this category has been mainly on modeling the domain knowledge.

Questions:

1. In aggregation, what is the main purpose of “root cause analysis” in *Clustering instrusion detection alarms to support root cause analysis?* Can its result be used in next step in correlation? Why are similarity alerts grouped based on same reason of these alerts but not same user/attacker or same target?
2. What is the algorithm used for creating representative alert for cluster in CRIM?
3. What does time series - variable mean?
4. How can “the distinguish between attacker actions and alerts” reduce false alert?
5. Why were “Graphs share similar alert” correlated to one graph?
6. Where is the prioritization component’s position in this framework?