Journal of Information Assurance and Security.
ISSN 1554-1010 Volume 7 (2012) pp. 79-88

Heterogeneous Multi-sensor IDS Alerts Aggregation using Semantic Analysis

Sherif Saad,[1], **Issa Traore**[2]

[1]ECE Department,University of Victoria,
Victoria, BC, Canada
shsaad@ece.uvic.ca

[2]ECE Department,University of Victoria,
Victoria, BC, Canada
itraore@ece.uvic.ca

Abstract: **One of the major limitations of current Intrusion Detection System (IDS) technology is alerts flooding which is a time consuming and resource intensive problem for intrusion analysts and organizations. Alerts flooding has been handled using alerts aggregation techniques. In general, the majority of IDS alerts aggregation techniques use alerts similarity to aggregate and summarize alerts. Because intrusion characteristics are expressed using symbolic attributes, measuring the similarity between IDS alerts is difficult. Previous techniques in the area of alerts aggregation mostly use perfect match or ad-hoc techniques to measure the similarity between alerts attributes. In this paper, we propose a new IDS alerts aggregation and reduction technique based on semantic similarity between intrusions. We define a new metric to measure semantic similarity between different intrusion instances. In addition we propose a new *information loss* metric to measure the quality of the alert aggregation process. Previous techniques only used alerts reduction rate to evaluate the alerts aggregation process. Alerts reduction rate is a volume metric and is not sufficient to evaluate the quality of the alert aggregation process. Experimental evaluation using existing IDS benchmark datasets shows that our proposed technique can more effectively aggregate IDS alerts and control alerts flooding compared to previous techniques in the area, while still maintaining relatively lower level of information loss.**

Keywords: Alerts Aggregation, Intrusion Detection, Semantic Analysis, Information Loss

I. Introduction

One of the main issues that have hampered the operation of intrusion detection systems (IDS) in networked environment is alert flooding. While alert aggregation has appeared as one of the common responses to this issue, there are several unresolved challenges which have limited the effectiveness of the approaches proposed to this date.

Alert aggregation consists of grouping related IDS alerts using either a rule base or some similarity metrics. While rule-based approaches are limited by the coverage of available rules, similarity-based techniques have the potential to cover more diverse types of alerts. However, to our knowledge while the existing similarity-based approaches have yielded encouraging performances in aggregating specific types of IDS alerts, they are incapable inherently to handle many other types of IDS alerts.

The main reason for such limitation lies in the types of similarity measures used by existing approaches which are either based on perfect matching or some form of binary matching of alerts attributes. While these similarity metrics make sense when matching numerical alerts attributes, they are ineffective when dealing with symbolic attributes. Likewise, many existing alert aggregation approaches cannot process alerts beyond those generated by different IDS sensors from the same vendor or using the same alert formatting standard. The aggregation of alerts produced in heterogeneous (i.e. different vendors, different alerts formats) distributed IDS multisensor environment have proven to be challenging for the existing approaches.

In this context, the use of the Intrusion Detection Message Exchange Format (IDMEF), a common formatting language, has been seen as a solution to address the heterogeneity challenge in distributed IDS multisensor environment. However, the IDMEF provides only a syntax for formatting (in a unified way) IDS alerts produced by different sensors. The lack of semantics limits the ability of IDMEF to capture the link between similar alerts formatted using syntactically different message structures. For instance, there is a known attack against SGI Telnet servers that come with a default account where the user name and password are $4Dgifts$. This attack will be detected and reported by the Bro IDS with the message $Sensitive\ Username\ In\ Password$, while Snort IDS will describe the same intrusion with the message $TELNET\ 4Dgifts\ SGI\ account\ attempt$. Even if the two IDSs use IDMEF to report this intrusion they will use their own language/vocabularies to describe the intrusion. It is not clear or obvious how these two messages are related even though they are referring to the same intrusion instance. To establish such relation we need to study the semantics or the contents of some of the fields of the messages.

We propose, in this paper, a new alert aggregation approach

that uses semantic analysis to capture the similarity between symbolic alert attributes. By analyzing and comparing the meanings (i.e. semantics) of symbolic alert attributes, the proposed approach is able to aggregate heterogeneous IDS alerts with high performance. The main performance metric used for the evaluation of alert aggregation approaches is the *alert reduction rate (ARR)*, which is computed as the difference between the original number of alerts and the alerts remaining at the end of the aggregation process over the original number of alerts. Our proposed approach achieves ARR of about 99% using three different datasets namely the DARPA 2000 dataset (commonly used by most existing approaches), a subset of the treasure hunt dataset, and a private dataset corresponding to real attacks against our lab honeynet.

Despite its popularity, we believe, however, that the ARR is not enough to evaluate the effectiveness of the aggregation process. In fact the ARR captures well the quantitative aspect of the alert aggregation process but misses altogether the qualitative perspective. To bridge this gap, we assess the quality of our aggregation process by measuring objectively the *information loss* occurring during this process.

The contribution of this paper is three-folds. Firstly, we propose a new alerts aggregation technique based on semantic analysis and ontology. Secondly, we define a new metric to capture the semantic similarity between concepts in a given ontology. Thirdly, we introduce a new quality metric to capture the information loss resulting from the alerts aggregation process.

The rest of the paper is structured as follows. Section II provides a summary of existing literature on alert aggregation approaches. Section III provides an overview of our general approach, and then introduces new metrics to capture semantic similarity for symbolic alert attributes and the information loss resulting from their aggregation. We also introduce in the same section our alert aggregation algorithm. Section **??** presents the experimental evaluation of our proposed approach and discusses the obtained results. Section **??** makes some concluding remarks.

II. Related Work

While a significant amount of literature has been produced on single sensor IDS alerts aggregation, only a few papers have been published on multi-sensor IDS alerts aggregation. In this section, we summarize and discuss related works under each of these two categories of alerts aggregation approaches.

A. Single Sensor Alerts Aggregation

As indicated above, several single sensor alerts aggregation approaches have been proposed in the literature [**?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**].

Zhigong proposed a real-time alert aggregation and correlation system [**?**] that uses five attributes, namely, source IP, source port, destination IP, destination port and intrusion signature. Three metrics are defined to capture attributes similarity. These metrics, however, are very trivial. For instance, one of the metrics, which captures the similarity between intrusion signatures, simply returns 1 if two signatures are equal and zero otherwise. With the proposed approach, alerts based on different intrusion patterns would probably not be aggregated. Heterogeneous Multi-sensor IDS Alerts Aggregation using Semantic Analysis

Xu and colleagues proposed a graph based approach to aggregate alerts based on the intrinsic order between them referred to as *happened before* relation [**?**]. The approach was evaluated with the DARPA 2000 dataset yielding an alerts reduction rate of 64.2%. The main issue with this approach is the high run time required to construct an alert graph and the assumption of low false positive rate generated by the IDS which is not always the case in practice.

Hofmann and Sick proposed an online intrusion alert aggregation system [**?**] in which alerts attributes are divided into two types: categorical attributes and continuous attributes. Examples of categorical attributes are intrusion class, IP address and port number. Examples of continuous attributes are alert time and packet size. Several metrics are defined to capture the similarity between categorical attributes. It is assumed that categorical attributes have a multinomial distribution while continuous attributes have a normal distribution. A maximum-likelihood estimation (MLE) method is used to design a parametrized probabilistic model that clusters or aggregates alerts. Experimental evaluation of the proposed approach with the DARPA dataset and two private datasets yielded alerts reduction rates above 98%.

Wen et al. proposed a lightweight intrusion alert fusion system [**?**]. The proposed system, called cache-based alert fusion scheme, was inspired from the working mechanism of the cup cache by applying the concept of Least Recently Used (LRU). The authors believe that the cache-based mechanism can improve the run-time of the aggregation algorithm. Experimental evaluation of the proposed technique with different IDS datasets (DARPA, Treasure hunt and Defcon) yielded an average alert reduction rate of about 91.0%.

Two other alerts aggregation approaches have been proposed in [**?**]. The first approach, known as attack thread reconstruction, aggregates a series of raw IDS alerts into a hybrid alert if there is a perfect match between raw alerts attributes, which as mentioned above is limited. Experimental evaluation of this approach using the DARPA 2000 dataset yielded an alerts reduction rate of 6.61%. The second approach, known as attack focus recognition, can aggregate IDS alerts based on different intrusion patterns, such as, one-to-many or many-to-one attack scenarios. However, the approach cannot aggregate alerts that are the results of the same intrusion attempt but have different intrusion signatures. Experimental evaluation of this second approach yielded an alerts reduction rate of 49.58% when using the DARPA 2000 dataset.

Zhuang et al. proposed an alerts aggregation approach using a set of similarity metrics to capture the similarity between alerts attributes [**?**, **?**]. Experimental evaluation of the approach yielded an alerts reduction rate of 98.7% when using the DARPA 2000 dataset. The proposed approach, however, cannot be used to aggregate alerts generated by different IDS sensors.

Jie et al. [**?**] proposed an alerts aggregation model that uses binary matching to aggregate the attack type attribute and simple similarity metrics to aggregate other attributes like IP address and port number. Evaluation with the DARPA

dataset shows that the proposed approach can reach an alerts reduction rate equal 90%.

B. Multi-Sensor Alerts Aggregation

To our knowledge, the first multi-sensor IDS alert aggregation approach was proposed by Valdes et al. [**?**]. The proposed approach uses a similarity function to aggregate alerts that match closely but not necessarily perfectly. Meta alert and alert templates are defined and used to describe IDS alerts. Given a pair of alerts, the similarity function returns for each alert attribute a value between 0 and 1 that reflects the similarity between corresponding attributes. To deal with different intrusion patterns a set of rules referred to as *Situation-Specific Similarity Expectation* are defined. It is not clear, however, how the authors measure the distance between different intrusion classes. Likewise the proposed approach seems to lack a general mechanism to measure the similarity between different intrusion classes. Evaluation of the approach using a private dataset collected from the lab of the authors, yielded alerts reduction rates between 50%-67%. However, an important limitation of the evaluation process, was that while the proposed approach was intended for multi-sensor alerts aggregation only a single IDS sensor was used to generate the alerts involved in the evaluation dataset.

Xu et al. proposed an alerts aggregation and fusion approach that can aggregate alerts generated by multiple IDS sensors [**?**]. The approach uses a multi-keywords scheme to cluster IDS alerts and routes clustered alerts to a sensor fusion center (SFC). Each SFC aggregates received alerts based on their source, destination, and attack class. This approach, however, cannot process alerts generated from different intrusion patterns. Although a dataset obtained from the D-Shield project was used to illustrate the approach, no quantitative performance measure was provided.

Fan et al. proposed a distributed IDS alert aggregation approach [**?**]. In the approach, raw IDS alerts collected from different IDS sensors are first converted to IDMEF format. Then, the converted alerts are processed by an alerts aggregation algorithm that categorizes them into four intrusion classes named *discovery, scan, DOS, and privilege escalation*. For each class of intrusions a similarity function is used to measure the similarity between alerts attributes. Alerts that belong to the same category will be aggregated or fused into meta-alert. Experimental evaluation of the approach using the DARPA 99 dataset yields an alert reduction rate of about 43.42%.

Debar et al. proposed an alerts aggregation and correlation approach for alerts generated by sensors from different vendors [**?**]. Alerts received from different sensors are expected to be in a standard format such as the Intrusion Detection Message Exchange Format (IDMEF). Four alerts attributes are used for the aggregation, namely, the source, target, alert class, and alert severity. The received alerts are aggregated based on a set of aggregation rules called aggregation-situations. Each aggregation rule generates a different meta-alert for the same set of raw IDS alerts, which leads to different aggregation views for the same set of raw IDS alerts. One of the main limitations of the proposed approach is the requirement of perfect match which means that alerts based on different intrusion patterns may not be aggregated. The authors illustrated the proposed approach through a case-study, and as a result their did not provide any information about the alert reduction rate.

C. Discussion

As mentioned above, only a small number of multi-sensor alerts aggregation approaches have been proposed in the literature. These approaches mostly use a common format to represent alert messages from different sensors such as the IDMEF standard. However, this only solves the alert message format problem, but cannot ensure that the keywords used by the different sensors to describe the same alert attributes have the same meanings. This of course will limit the performance of the aggregation approach. Likewise, the few existing multi-sensor aggregation approaches either achieved relatively low alert aggregation rates or simply did not report any quantitative performance results. This raises the need of formal alerts representations that consider both the structures and the semantics of the alert messages.

Several of the existing alerts aggregation approaches require perfect match of the alerts attributes in the aggregation process. While these approaches do not suffer from information loss, they have very poor performances and do not really address the alert flooding problem. In fact they are mostly limited to eliminating redundant alerts only. On the other hand approaches that use attribute similarity yield promising performances with alert reduction rates reaching 99% for some approaches. However, none of these approaches consider the quality of the generated hybrid or meta alerts. All the proposed approaches lack an appropriate method to assess the effect of information loss that occur in the aggregated alerts. While the problem of information loss has been pointed out in the literature [**?**, **?**] no metric or approach were proposed to handle this aspect.

We propose in this paper a new multi-sensor IDS alert aggregation approach that uses semantic similarity to address the above mentioned limitations of existing approaches. To our knowledge, our work is the first semantic similarity based alert aggregation approach proposed in the literature. We introduce in detail our proposed approach in the next section.

III. Semantic-based Alerts Aggregation

In this section, we give an overview of our alerts aggregation approach, and then present in detail key aspects of the approach, including a new semantic similarity metric, a new information loss metric, and our alert aggregation algorithm.

A. Approach Overview

The key idea of our approach is that alerts that relate to the same attack instance are semantically similar, even if they are described in different formats. Therefore, if we can measure the semantic similarity between alerts we can effectively aggregate them. IDS alerts are structured using a number of attributes. These attributes can be divided between symbolic and non-symbolic categories. Our approach assumes the existence of an ontology describing the semantics of the concepts corresponding to the symbolic alerts attributes. In other words our approach require an intrusion detection domain ontology.

An ontology is a formal representation of a set of concepts and the relations between these concepts in a domain of knowledge. Kruegel and Christopher argued that an ontology for intrusions is a prerequisite for true interoperability between different IDSs [**?**]. In the last few years several network intrusion ontologies and taxonomies have been proposed [**?**, **?**, **?**, **?**, **?**, **?**, **?**]. All of these ontologies can be used to provide common vocabularies and make knowledge shareable by encoding domain knowledge. Hence, they could be used (to some extent) as knowledge bases in our aggregation model. To demonstrate our approach, we use a network forensics ontology proposed in [**?**] that contains knowledge about more than 11,000 malicious activities and 30 network forensics problem solving methods.

In our opinion an alerts aggregation approach should satisfy three requirements. The first requirement is *interoperability*, which means the ability of aggregating alerts generated by different IDS sensors with different formats. The second requirement is *threat recognition*, which means the ability of aggregating different alerts that are generated as a result of the same attack instance or pattern. The third requirement is *information preservation*, which means the aggregation process should preserve as much as possible the valuable information carried by the initial raw alerts.

To deal with the interoperability requirement we build an IDS profile for each brand of IDS such as Snort-IDS, Bro-IDS, etc. The IDS profile maps the keywords used by corresponding IDS sensor to describe intrusion instances to the vocabularies (keywords) used in our ontology. We process the alerts generated by each IDS sensor and use the IDS profile to convert them into a common format that uses the same structure and semantic to describe the alerts. The alerts are then processed by our alert aggregation algorithm and aggregated into hybrid alerts as explained later.

B. Semantic Similarity Metric

Similar concepts or classes in an ontology are structured in a taxonomy structure also referred to as *concept tree*. A *concept tree* describes the abstraction relationship (i.e. generalization/specialization) between similar concepts using a hierarchical structure. The root of the tree corresponds to the most abstract form of the concept, while intermediary nodes correspond to refined concepts, and leaves nodes correspond to instances. Our approach consists of associating with each symbolic alert attribute a concept tree in which the attribute itself is the root node while the attribute values correspond to the leaves of the tree.

As an example, let us assume that we use symbolic attributes to represent the type of intrusion, the attack source and the attack target in formatting alerts. Figure **??** is a subtree that describes *Information Gathering* attack methods from the global concept tree corresponding to intrusion (type) attribute (see [**?**] for more details). Figure **??** is a subtree that describes *network address*, which is used to represent the source and the target of network intrusion in the ontology.

We use the notion of concept tree to measure the similarity between symbolic alert attribute values as explained in the following.

Let $a_1, ..., a_n$ denote a set of IDS alerts, where each alert a_i is represented using a p-dimensional attribute vector

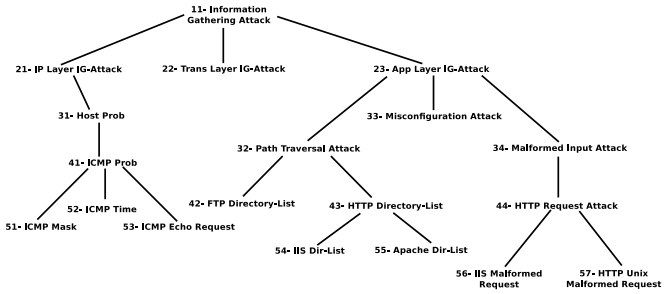

Figure. 1: Information-Gathering Attack Ontology (Partial)

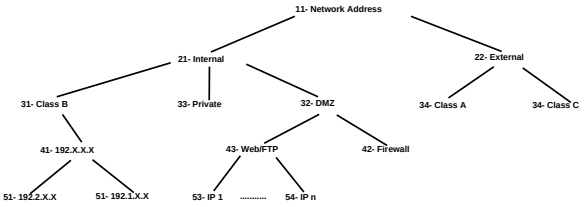

Figure. 2: Network Address Ontology (Partial)

$[a_{i1}, ..., a_{ip}]$ and only the first s attributes are symbolic attributes ($1 \leq s \leq p$). The similarity between two concepts in an ontology depends on the commonalities and the differences between the two concepts. The commonalities between two concepts are represented by their relations to their lowest common ancestor in the ontology. On the other hand the differences between them is based on their locations within the ontology structure. Based on the above considerations, given two alerts $a_i = [a_{i1}, ..., a_{ip}]$ and $a_j = [a_{j1}, ..., a_{jp}]$, we define our semantic similarity metric between symbolic attribute values a_{ik} and a_{jk} $(1 \leq k \leq s)$ as shown in equation **??**.

Where $path(a_{ik}, LCA(a_{ik}, a_{jk}))$ is the length of the shortest path from concept a_{ik} to the least common ancestor (LCA) of a_{ik} and a_{jk} in the concept tree, and $depth(a_{ik})$ is the depth of concept a_{ik} in the concept tree. The metric has two important properties. The first property is that the semantic similarity between higher-level concepts are less than the semantic similarity between lower-level concepts. This reflects the fact that two general concepts are less similar than two specialized ones. The second property is that the semantic similarity between a parent concept and any child concept of this parent is greater than the similarity between this child concept and any other child concept of the same parent.

For example, using the Information Gathering Attack ontology in Figure **??**, the computation of the semantic similarity between any two concepts in the ontology is straightforward. For instance, using equation **??**, the semantic similarity between the IIS Dir-List class and the Apache Dir-List is computed as $sim(IISDir - List, ApacheDir - List) = 0.8$. In this example, the class HTTP Directory-List is the first common ancestor of IIS Dir-List and Apache Dir-List, and the depth of IIS Dir-List and Apache Dir-List equal 5.

We define the semantic similarity between two alerts a_i and a_j as follows.

$$sim(a_i, a_j) = \frac{\sum_{k=1}^{s} sim(a_{ik}, a_{jk})}{s} \quad (2)$$

$$sim(a_{ik}, a_{jk}) = 1 - \frac{(path(a_{ik}, LCA(a_{ik}, a_{jk})) + path(a_{jk}, LCA(a_{ik}, a_{jk})))}{(depth(a_{ik}) + depth(a_{jk}))} \quad (1)$$

The semantic similarity between any pair of alerts or between any pair of alert attributes is a value between **0** and **1**. Where **1** indicates the maximum similarity and **0** indicates that there is no similarity at all between the two attributes or alerts.

C. Information Loss Metric

In our work, two concepts that belong to the same domain are aggregated by replacing them by their least common ancestor (LCA) from the corresponding concept tree. However, this will lead unavoidably to loss of information. To capture the information loss we need to measure the amount of information represented by each concept or class in the ontology. The difference between the amount of information of concept c_1 and its subclass c_2 represents the information loss occurred by replacing c_2 by c_1. The information content (IC) of a concept c can be used to measure the amount of information represented by c. Recently several approaches inspired by information theory have been proposed to measure the IC of a given concept in an ontology based on the taxonomic structure of the concept within the ontology [**?**, **?**]. We use in our work the IC metric proposed by Sánchez and colleagues [**?**] and defined as follows:

$$IC(c) = -log\left(\frac{\frac{|leaves(c))|}{|subsumers(c)|} + 1}{maxleaves + 1}\right) \quad (3)$$

Where $subsumers(c)$ is a function that returns the set of subsumers concepts of concept c (these include concept c as well as all its parents concepts), $leaves(c)$ is a function that returns all the leaves concepts that are subclasses of concept c, and $maxleaves$ is the total number of leaves concepts of the concept tree. The subsumers of a given concept and the leaves of that concept reflect the information content of that concept. Based on the principle of cognitive saliency[1], concepts are specialized when it is necessary to differentiate them from already existing ones [**?**]. So, concepts with more sub-concepts provide less information than concepts at lower levels of the hierarchy (such as leaves concepts).

Now, given a set of concepts C, we define the information loss rate (ILR) resulting from replacing the concepts in C by their least common ancestor a as follows:

$$ILR(C) = \frac{\sum_{c \in C}(IC(c) - IC(LCA(C)))}{\sum_{c \in C} IC(c)} \quad (4)$$

Where $LCA(C)$ corresponds to the least common ancestor of the concepts in C.

Using the above formula and the notation introduced in the previous section, given two alerts $a_i = [a_{i1}, ..., a_{ip}]$ and $a_j = [a_{j1}, ..., a_{jp}]$, the information loss rate occurring from aggregating symbolic attribute values a_{ik} and a_{jk} $(1 \leq k \leq s)$ is defined as shown in equation **??**. The information loss rate is a value between **0** and **1**, where 0 corresponds to no information lost and 1 correspond to 100% loss of information.

The information loss rate resulting from the aggregation of a set of alerts into a hybrid alert h is computed as the summation of the information loss rate of each attribute of H over the total number of attributes in h.

The information loss rate for an entire aggregation process generating a set of hybrid alerts H may be obtained as the average of the information loss rate over all the hybrid alerts in H. However, we take in this work a more conservative approach by defining the information loss rate for an entire aggregation process as the maximum information loss over the hybrid alerts involved in H.

Now let us assume that we want to calculate the information loss rate resulting from aggregating the two classes $IISDir - List$ and $ApacheDir - List$ from the previous example. The first common ancestor of the two classes in the ontology is the class $HTTPDirectory - List$. Using equation **??**, we obtain the following: IC(IIS Dir-List)=0.9, IC(Appache Dir-List)=0.9 IC(HTTP Directory-List)=0.72. Using the above metric, the information loss rate from aggregating the two classes $IISDir-List$ and $ApacheDir-List$ is 0.18.

D. Alerts Aggregation Algorithm

The main steps of our semantic similarity based alerts aggregation process are illustrated by Algorithm **??**. The algorithm performs two main operations, namely, clustering and fusion. The clustering operation groups semantically similar alerts into a single cluster based on a predefined similarity threshold. The fusion operation fuses the alerts that belong to the same cluster and generates a corresponding hybrid alert.

The algorithm takes two inputs. The first input is a set of raw IDS alerts sorted by increasing order of occurrence time. The second input is a thresholds vector, where each element represents a predefined semantic similarity threshold for one of the symbolic attributes.

The output of the algorithm is a set of hybrid alerts that represent the original set of raw IDS alerts. In our work, an hybrid alert has the same format, and therefore the same types of attributes as a raw alert. The main difference between the attributes in a hybrid alert and those in the raw alert is the level of abstraction. Hybrid alerts' attributes values (i.e. concepts) will be equal or more abstract than corresponding raw alerts' attributes values. In addition, we associate with each hybrid alert its own information loss rate which depends on the level of abstraction of its attributes values.

The algorithm performs several rounds; during each round the alerts are grouped into one ore more clusters and one hybrid alert is generated for each cluster. An alert will be assigned to a cluster if the attributes similarities between the alert and the hybrid-alert that represents this cluster are greater than some predefined thresholds.

Each time an alert is assigned to a cluster, we fuse that alert with the hybrid alert that represents this cluster and regen-

[1]The salience or saliency of an object or a concept corresponds to its relative standing or quality with respect to its neighbors.

$$
\begin{aligned}
ILR(a_{ik}, a_{jk}) =& \frac{(IC(a_{ik}) + IC(a_{jk}) - 2IC(LCA(a_{ik}, a_{jk})))}{2} \\
\Rightarrow ILR(a_{ik}, a_{jk}) =& log\left(\frac{|leaves(LCA(a_{ik}, a_{jk})))|}{|subsumers(LCA(a_{ik}, a_{jk}))|} + 1\right) \\
& - \frac{1}{2} log\left[\left(\frac{|leaves(a_{ik})|}{|subsumers(a_{ik})|} + 1\right)\left(\frac{|leaves(a_{jk})|}{|subsumers(a_{jk})|} + 1\right)\right]
\end{aligned} \tag{5}
$$

Algorithm 1: IDS Alerts Aggregation Algorithm

```
   /* A a set of intrusion alerts of size n                    */
   /* T semantic similarity threshold vector of size p
      */
   Input: A, T
   Output: H
1  begin
       /* Th:  threshold vector of size p                      */
       /* C: set of alerts clusters                            */
2      Th ← [1, ..., 1] ;
3      i ← 0;
4      A' ← A;
5      C ← ∅ ;
6      while i ≤ p do
7          for j = 1 to n do
8              x ← true;
9              if C ≠ ∅ then
10                 for s = 1 to size(C) do
11                     c ← C[s];
12                     h ← hybrid-alert of cluster c;
                       /* h = [h_1, ..., h_p], where h_s is the
                          s^th attribute of h                   */
13                     for l = 1 to p do
14                         if (sim(a_jl, h_l]) ≤ Th[l]) then
15                             x ← false;
16                             break;
17                         end
18                     end
19                     if x = true then
20                         c ← c ∪ {a_j};
21                         h ← fuse a_j with h;
22                         H ← H ∪ {h};
23                         break;
24                     end
25                 end
26             end
27             if (x = false) or (C = ∅) then
28                 let c new cluster such that c = {a_j};
29                 let h hybrid-alert of c such that h = a_j;
30                 H ← H ∪ {h};
31             end
32         end
33         A' ← H;
34         i ← i + 1;
35         Th[i] ← T[i];
36     end
37     return H;
38 end
```

erate the hybrid alert of the cluster. The hybrid-alert is regenerated by fusing the attributes of the hybrid-alert with the attributes of the new alert. The fusion of two attributes from two different alerts consists of replacing them with their least common ancestor in their concept tree in the ontology. For example, let us assume that we have two alerts a_1 and a_2. If we fuse the attribute attack-type where the value of that attribute in a_1 is $IISDir-List$ and in a_2 is $ApacheDir-List$ then the result will be $HTTPDirectory - List$, which is their least common ancestor according to the concept tree in Figure **??**. At the end of each round the hybrid alerts along with the remaining alerts (not aggregated yet) are sorted and passed as input to the next round of the algorithm and go through the same process outlined above.

The different rounds of the algorithm are determined by the similarity threshold vector used in the clustering. The rounds are designed so as to aggregate first the alerts that are most likely to have greater semantic similarity, and by setting the similarity threshold vector accordingly. This is performed by clustering the alerts for which a subset of attributes match perfectly (i.e. threshold = 1). The clustering is carried out iteratively by decreasing in each iteration the required number of alerts attributes that match perfectly, and lowering the thresholds for the remaining attributes to predefined levels. Hence, while in the first round the similarity thresholds are all set to one, in the last round they are set to the predefined values provided as input to the algorithm.

IV. Experiments

We present, in this section, the experimental evaluation of our framework. We describe the datasets and tools used in the evaluation, and then present the evaluation results for single sensor alerts aggregation, followed by multi-sensor alerts aggregation. We used two evaluation metrics, namely the alert reduction rate and the information loss rate.

A. Tools and Datasets

We implemented our alerts aggregation tool using Java and the Jena Ontology API to access the network forensics ontology. To evaluate the ability of our framework to aggregate multi-sensor IDS alerts, we used two different IDS sensors to analyze the datasets and generate the raw IDS alerts, namely, Snort IDS version 2.8.4 and Bro IDS version 1.5.3.

We used three different intrusion datasets, namely, the DoS1.0 version of the DARPA 2000 dataset [**?**], the Treasure Hunt dataset [**?**] and a private dataset collected in our lab. By selecting the above datasets, our goal was to use datasets with different characteristics, such that each dataset contains different attack patterns or different attack scenarios.

In the literature the DARPA 2000 dataset has been the most commonly used public dataset for evaluating alerts aggregation approaches. However, in our opinion, the DARPA 2000 dataset is not enough to evaluate alerts aggregation approaches. This is because the DARPA 2000 dataset contains only a single attack pattern and a single intruder which will usually lead to high alerts reduction rate. For that reason we used three different datasets providing a wide variety of attack patterns. For instance, the DARPA 2000 dataset contains a multistage attack scenario where the intruder scans the network, takes control of some of the hosts in the network and uses the compromised hosts to launch a DDoS attack against an off-site server. The treasure hunt dataset contains a multistage attack scenario, where several intruders penetrate an organization network comprised of several servers such as web server and database server, and perform some money

transfers from employees' accounts.
The private dataset collected through a honeynet deployed in our lab contains many standalone intrusion attempts including worm-attacks, web and FTP attacks, SQL DB attacks, privilege escalation attacks, and DoS attacks.

B. Alerts Attributes and Similarity Thresholds

In our evaluation we used (without loss of generality) three different symbolic attributes to represent IDS alerts, namely, the *attack source*, the *attack target*, and the *intrusion type*. Note that several other attributes can be added to this list such as *attack time*.
We considered for each attribute five different semantic similarity threshold values between zero and one. Using the threshold values we can generate up to 125 different threshold vectors which correspond to all possible combinations of the selected values. In our experiment, we used a subset of 10 different threshold vectors listed in Table **??**.

Table 1: Semantic Similarity Threshold Vectors

Vector ID	Source	Target	Intrusion Type
V0	1	1	1
V1	0.8	0.7	1
V2	0.8	0.8	0.9
V3	0.8	0.8	0.8
V4	0.8	0.7	0.8
V5	0.6	0.7	0.8
V6	0.6	0.6	0.8
V7	0.6	0.6	0.75
V8	0.4	0.4	0.45
V9	0.16	0.16	0.12

C. Single Sensor Alerts Aggregation

First we evaluate our approach using a single IDS sensor to monitor the network traffic and generate the alerts. In this part of the experiment we use Snort to analyze the three datasets. Table **??** shows some statistics about each dataset after analyzing it with Snort such as the number of alerts, number of hosts, durations and numbers of different intrusion instances.

Table 2: Intrusion Datasets Statistics

Dataset	Treasure Hunt	DARPA 2000	Lab Traffic
Alerts	199587	2170	2048
Intrusions	18	16	63
Sources	5	273	115
Targets	4	738	836
Duration	$\approx$ 3 min	$\approx$ 100 min	$\approx$ 900 min

For each dataset, we run our alert aggregation algorithm 10 times, using each time a different semantic similarity threshold vector. Each time, we calculate the alerts reduction rate and the maximum information loss rate. Table **??** shows the results of our experiment with the single sensor alert aggregation. We plot for each dataset what we refer to as the *Aggregation Performance Curve (APC)*, which shows the relation between the alert reduction rate and the information loss rate when the threshold values vary. Figure **??** illustrates the APCs obtained for the different datasets.
By analyzing the results we find that in general higher alerts reduction rate means higher information loss rate. We also find that changing the semantic similarity threshold vector will result in one of the following three outcomes. The first

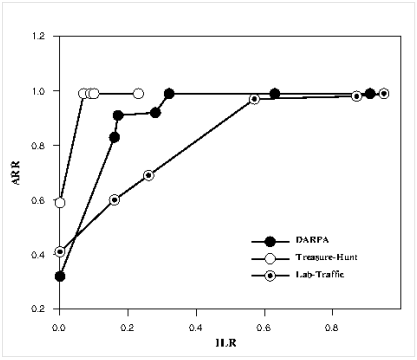

Figure. 3: APCs for single sensor IDS alerts aggregation

Table 3: Single Sensor Evaluation Results

Vector	DARPA		Treasure Hunt		Lab Traffic	
	ARR	ILR_{max}	ARR	ILR_{max}	ARR	ILR_{max}
V0	0.32	0	0.59	0	0.41	0
V1	0.32	0	0.59	0	0.41	0
V2	0.83	0.16	0.99	0.07	60	0.16
V3	0.91	0.17	0.99	0.09	0.69	0.26
V4	0.92	0.28	0.99	0.09	0.69	0.26
V5	0.99	0.32	0.99	0.09	0.69	0.26
V6	0.99	0.32	0.99	0.09	0.97	0.57
V7	0.99	0.32	0.99	0.09	0.97	0.57
V8	0.99	0.63	0.99	0.1	0.99	0.87
V9	0.99	0.91	0.99	0.23	0.99	0.95

outcome is a notable change in the ARR and the ILR; this occurs, for instance, when the threshold vector changes from $V2$ to $V3$. As we can see there is a notable change in the ARR and ILR for all datasets. The second outcome is no change at all in the ARR and ILR; for instance, this is the case when we change the thresholds from $V1$ to $V2$ or from $V6$ to $V7$. The reason for that is because the semantic similarity values between the alerts are less than the semantic similarity threshold values. The third outcome is a notable change in the ILR while the ARR barely changes. For instance, as we can see from Table **??**, changing from $V7$ to $V8$ or $V9$ does not cause any notable change in the ARR, however, there is a major change in the ILR for all datasets. Also the attack pattern has a significant impact on the alerts reduction rate, the information loss rate and the selection of the semantic similarity threshold. For example, we found that different attack patterns require different adjustments of the semantic similarity threshold for each alert attribute. For instance, 38.4% of the DARPA 2000 raw IDS alerts are related to the Mstream DDoS attack where all the alerts have spoofed, random source IP addresses. Snort uses 2 intrusion signatures to represent that DDoS attack pattern. Since the source IP address in the alerts are spoofed and random we had to set the semantic similarity of the source attribute to lower value to be able to aggregate the alerts that belong to that attack pattern. In fact several existing works in the literature set the similarity thresholds between alert attributes based on the intrusion pattern [**?**] or define a set of rules to aggregate the alerts based on the type of attack pattern (see for instance, [**?**, **?**]).
High value of alerts reduction rate should not be considered the main factor to judge the performance of any alerts aggregation approach. An intrusion analyst should also consider the amount of information loss in the generated hybrid alerts. In our experiment we found that the acceptable level of infor-

mation loss rate can be defined based on the attack pattern. For instance, when aggregating the alerts of the DDoS attack in the DARPA 2000 dataset we obtained an information loss rate of 32%. This rate is mainly the result of aggregating alerts with different source IP addresses. However, these IP addresses are randomly spoofed by the Mstream worm. In this case, the 32% of information loss is acceptable because the randomly spoofed IP addresses do not really bring any useful knowledge to the intrusion analyst. In general, we found that when applying the same semantic similarity threshold to the same intrusion pattern (e.g. DDoS attack, scan attack, etc) in different datasets we obtain the same performance rates (i.e. same ARR and ILR values).
Figure **??** depicts the hybrid alert generated for the DDoS attack in the DARPA 2000 in the aggregation process. The source of the hybrid alert is an aggregate attribute value that represents a set of IP addresses that belong to the local network. The target of the attack is an off-site IP address. The intrusion type is Mstream-DDoS, which is also an aggregation of the original two snort signatures in the raw IDS alerts.

$$HA = \left\{ \begin{array}{ccc} Source & Target & Intrusion \\ 127.X.X.X & 131.84.1.31 & mstream \\ & & DDoS \end{array} \right\}$$

Figure. 4: Hybrid alert obtained from the DARPA 2000 dataset; the hybrid alert represents a mstream DDoS Attack.

As indicated earlier, the DARPA dataset is the most widely used dataset for the evaluation of alert aggregation approaches. Table **??** shows a comparison between our approach and previous approaches from the literature that used the DARPA 2000 dataset.
It is important to point out that the approaches proposed in the literature used either different IDS sensors or IDS rulesets to generate the alerts from the DARPA dataset. This means Table **??** does not give a fully accurate comparison. Another important point is that none of the existing multi-sensor aggregation approaches have used the DARPA 2000 dataset in their evaluation. Likewise, all the approaches listed in Table **??** are single sensor ones. Also none of the existing approaches provided explicitly the ILR measure. The only existing approaches for which the ILR can be inferred are the ones that use perfect match to aggregate the alerts; in this case the information loss rate is always zero. The attack thread reconstruction approach proposed in [20] fits under this category and yields $(ARR = 6.61\%, ILR = 0\%)$. Our approach achieves $ILR = 0\%$ when the semantic similarity threshold vector is set to ones as shown by the case of vector V_0 in Table **??**, which is also a case of perfect match. It must be noted that approaches based on perfect match can only aggregate duplicated alerts, in which case the ARR will depend on the number of duplicated alerts available in the dataset.

D. Multi-Sensor Alerts Aggregation

In order to evaluate our approach for multi-sensor IDS alerts aggregation, in addition to the Snort IDS, we used Bro IDS to analyze the network traffic and generate the IDS alerts. In the multi sensor experiment we only used the DARPA 2000 dataset.

Table 4: Comparison of alerts aggregation approaches using the DARPA 2000 dataset in their evaluation.

Approach	Reference	ARR	ILR
Xu et al.	[24]	64.20%	not measured
Hofmann and Sick	[7]	99.00%	not measured
Wen et al	[21]	91.00%	not measured
attack thread recon	[20]	6.61%	0%
attack focus recognition	[20]	49.58%	not measured
Zhuang et al	[26]	98.70%	not measured
Jie et al	[11]	90.00%	not measured
Our approach	Current	**99.30%**	32%

As mentioned earlier, the DARPA 2000 dataset contains one attack pattern which is a multi-steps DDoS attack. The intruder probed the network and exploited a Solaris OS services vulnerability to gain root access. He then installed a malware on 3 machines and then used the malware via telnet to attack a remote site. The malware executed a *mstream DoS* attack.
Using our aggregation tool, the raw alerts generated by Snort and Bro were preprocessed and reformatted to provide unified alerts messages based on our ontology. Each attak step was reported in snort and bro by one or more attack signatures. Table **??** shows the number of raw alerts and intrusions generated by Snort and Bro based on the DARPA dataset and the number of alerts and intrusion after preprocessing the alerts messages and reformatting them based on the ontology vocabularies.
Bro IDS detected five different attack types and generated 880 corresponding raw IDS alerts when analyzing the DARPA dataset. Bro IDS, however, failed to detect the final step of the attack, which is the mstream DoS; this was the only step that was not detected by Bro. From Table **??** we can notice that after preprocessing Snort and Bro alert messages the number of intrusions increases to 18. This is because only 3 intrusions were commonly detected by both Snort and Bro.

Table 5: DARPA dataset preprocessing statistics

Alerts Format	Alerts Count	Intrusions
Snort	2170	16
Bro	924	5
Ontology-Based	3094	18

Table **??** shows the results of our experiment for multi-sensor alert aggregation, when varying the thresholds; Figure **??** illustrates the corresponding APC. The results of aggregating the DARPA dataset alerts generated by Snort and Bro are very close to the results of aggregating the alerts generated by Snort only. The main reason for that is because Bro failed to detect the mstream DDoS attack.

Table 6: Multi-sensor Alerts Aggregation Evaluation Results

Vector	DARPA	
	ARR	ILR
V0	0.37	0
V1	0.37	0
V2	0.87	0.16
V3	0.94	0.17
V4	0.99	0.32
V5	0.99	0.32
V6	0.99	0.32
V7	0.99	0.32
V8	0.99	0.63
V9	0.99	0.91

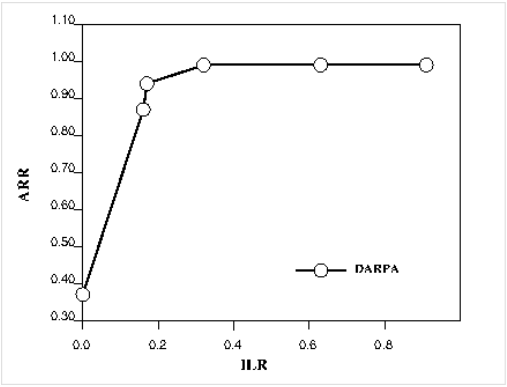

Figure. 5: APC for multi-sensor IDS alerts aggregation

Using $V4$ as threshold vector allows us to achieve $(ARR = 99\%, ILR = 32\%)$. As mentioned above, $ILR = 32\%$ can be considered acceptable because the loss is related to spoofed IP addresses, which do not bring any useful information.
It is important to emphasize that none of the existing multi-sensor aggregation approaches from the literature have actually been evaluated experimentally in true multi-sensor settings. While the proposed approaches were presented as being able to aggregate multi-sensor alerts, experimental results have been provided only for single-sensor alerts. No quantitative performance results were provided for multi-sensor alerts, which make it difficult to compare objectively our approach against these approaches.

V. Conclusion

In this paper, we proposed a new alert aggregation technique based on the semantic features of IDS alerts. The proposed technique can aggregate alerts collected from decentralized heterogeneous IDS sensors. The use of semantic features allows us to aggregate alerts that have a similar semantic description. This makes our alerts aggregation technique highly flexible compared to previous ones. In particular our technique is not specific to any attack scenario and does not require perfect match of alert features. Moreover, the use of semantic features to model IDS alerts allows us to represent alerts in a machine understandable format. This machine understandable format gives the ability to design an automated alerts aggregation technique that requires minimum human interaction. The experimental results show that our technique can be used to minimize significantly IDS alerts flooding while maintaining limited information loss. Measuring the information loss is very important when aggregating and summarizing security log files such as IDS alerts log, firewalls logs, etc. Alerts aggregation approaches that do not consider information loss will mostly result in losing important security-relevant information.

References

[1] F. Abdoli and M. Kahani. Using attacks ontology in distributed intrusion detection system. In *SCSS (1)*, pages 153–158, 2007.

[2] K. Burbeck and S. Nadjm-tehrani. Adwice - anomaly detection with real-time incremental clustering. In *Proceedings of the 7th International Conference on Information Security and Cryptology, Seoul, Korea*. Springer Verlag, 2004.

[3] H. Debar and A. Wespi. Aggregation and correlation of intrusion-detection alerts. In *RAID '00: Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection*, pages 85–103, London, UK, 2001. Springer-Verlag.

[4] G. Fan, Y. JiHua, and Y. Min. Design and implementation of a distributed ids alert aggregation model. In *Computer Science Education, 2009. ICCSE '09. 4th International Conference on*, pages 975 –980, 2009.

[5] N. D. D. Gustavo Isaza, Andrés Castillo. An intrusion detection and prevention model based on intelligent multi-agent systems, signatures and reaction rules ontologies. In *7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2009)*.

[6] S. Hansman and R. Hunt. A taxonomy of network and computer attacks. *Computers & Security*, 24(1):31 – 43, 2005.

[7] A. Hofmann and B. Sick. Online intrusion alert aggregation with generative data stream modeling. *Dependable and Secure Computing, IEEE Transactions on*, vol 8(num 2):282 –294, 2011.

[8] Kruegel and Christopher. *Intrusion Detection and Correlation: Challenges and Solutions*. Springer-Verlag TELOS, Santa Clara, CA, USA, 2004.

[9] C. E. Landwehr, A. R. Bull, J. P. Mcdermott, and W. S. Choi. A taxonomy of computer program security flaws, with examples. *ACM Comput. Surv.*, 26(3):211–254, September 1994.

[10] Lincoln-Laboratory-MIT. Darpa intrusion detection evaluation. `http://www.ll.mit.edu/mission/communications/ist/CST/index.html`.

[11] J. Ma, Z. T. Li, and H. W. Zhang. A fusion model for network threat identification and risk assessment. In *AICI '09: Proceedings of the 2009 International Conference on Artificial Intelligence and Computational Intelligence*, pages 314–318, Washington, DC, USA, 2009. IEEE Computer Society.

[12] A. S. Peter, P. S, and L. V. Ekert. An ontology for network security attacks. In *Proceedings of the 2nd Asian Applied Computing Conference (AACC04), LNCS 3285*, pages 317–323. Springer-Verlag, 2004.

[13] S. Saad and I. Traore. Method ontology for intelligent network forensics analysis. In *Eight International Conference on Privacy, Security and Trust (PST 2010)*, pages 7–14, Ottawa, Canada, August 2010.

[14] D. Sánchez, M. Batet, and D. Isern. Ontology-based information content computation. *Know.-Based Syst.*, 24:297–303, March 2011.

[15] N. Seco, T. Veale, and J. Hayes. An Intrinsic Information Content Metric for Semantic Similarity in WordNet. In *ECAI'2004, the 16th European Conference on Artificial Intelligence*, 2004.

[16] UCSB. The 2002 ucsb treasure hunt dataset. `http://ictf.cs.ucsb.edu/data/treasurehunt2002/`.

[17] J. L. Undercoffer, A. Joshi, T. Finin, and J. Pinkston. A Target-Centric Ontology for Intrusion Detection. In *The 18th International Joint Conference on Artificial Intelligence*, July 2003.

[18] A. Valdes and K. Skinner. Probabilistic alert correlation. In *RAID '00: Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection*, pages 54–68, London, UK, 2001. Springer-Verlag.

[19] F. Valeur, G. Vigna, C. Kruegel, and R. Kemmerer. Comprehensive approach to intrusion detection alert correlation. *Dependable and Secure Computing, IEEE Transactions on*, 1(3):146 – 169, jul. 2004.

[20] S. Wen, Y. Xiang, and W. Zhou. A lightweight intrusion alert fusion system. In *High Performance Computing and Communications (HPCC), 2010 12th IEEE International Conference on*, pages 695 –700, 2010.

[21] S. Xiao, Y. Zhang, X. Liu, and J. Gao. Alert fusion based on cluster and correlation analysis. *Hybrid Information Technology, International Conference on*, pages 163–168, 2008.

[22] M. Xu and W. Han. Distributed intrusion alert fusion based on multi keyword. In *ISDPE '07: Proceedings of the The First International Symposium on Data, Privacy, and E-Commerce*, pages 469–471, Washington, DC, USA, 2007. IEEE Computer Society.

[23] M. Xu, T. Wu, and J. Tang. An ids alert fusion approach based on happened before relation. In *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on*, pages 1 –4, oct. 2008.

[24] T. Zhihong, Q. Baoshan, Y. Jianwei, and Z. Hongli. Alertclu: A realtime alert aggregation and correlation system. In *Cyberworlds, 2008 International Conference on*, pages 778 –781, 2008.

[25] X. Zhuang, D. Xiao, X. Liu, and Y. Zhang. Applying data fusion in collaborative alerts correlation. *Computer Science and Computational Technology, International Symposium on*, vol 2:124–127, 2008.

Author Biographies

Sherif Saad received his B.Sc in Computer Science from Helwan University, Egypt (2003), M.Sc in Computer Science from Arab Academy for Science, Technology and Maritime Transport, Egypt (2007). In 2008 he received the University of Victoria Fellowship and started his PhD in the Department of Electrical and Computer Engineering of the University of Victoria. His primary research interests are in advancing machine-learning methods and their application to computer and network security. Since 2009, he is working as an information security engineer for Plurilock Security Solutions (`www.plurilock.com/`).

Issa Traore Dr. Traore obtained a PhD in Software Engineering in 1998 from Institute Nationale Polytechnique (INPT)-LAAS/CNRS, Toulouse, France. He has been with the faculty of the Department of Electrical and Computer Engineering of the University of Victoria since 1999. He is currently an Associate Professor and the Coordinator of the Information Security and object Technology (ISOT) Lab (`http://www.isot.ece.uvic.ca`) at the University of Victoria. His research interests include biometrics technologies, computer intrusion detection, network forensics, software security, and software quality engineering. He has published over 100 technical papers in computer security and software engineering and supervised 23 Master and PhD graduate students in the last 10 years. He is currently serving as Associate Editor for the International Journal of Communication Networks and Distributed Systems (IJCNDS).