A Semantic Analysis Approach to Manage IDS Alerts Flooding

Sherif Saad
Electrical and Computer Engineering
University Of Victoria
Email: shsaad@ece.uvic.ca

Issa Traore
Electrical and Computer Engineering
University Of Victoria
Email: itraore@ece.uvic.ca

***Abstract*—In this paper we propose a new approach to manage alerts flooding in IDSs. The proposed approach use semantic analysis and ontology engineering techniques to combine and fuse two or more raw IDS alerts into one summarized hybrid/meta-alert. Our approach applying a new method based on measuring the semantic similarity between IDS alerts attributes to decide the alerts that are suitable for aggregation and summarization. In contrast to previous works our approach ensure that the aggregated alert will not loss any valuable information that exist in the raw alerts set. The experimental results show that our approach is effective and efficient in fusing massive number of alerts comparing to previous works in the area.**

***Keywords*-Alerts Aggregation; Intrusion Detection; Semantic Analysis; Information Loss; Ontology;**

I. Introduction

Nowadays, intrusion detection systems (IDS) has a major role in implementing security solutions for computer networks. One of the main issues that have hampered the operation of IDS in networked environment is alerts flooding. IDS can generates hundred of thousands of alerts per hour, which create an alert flooding problem. Alerts flooding is a time consuming and resource intensive problem for intrusion analysts and organizations. Alerts flooding has been handled using alerts aggregation techniques. In the last few years several alerts aggregation techniques have been proposed to deal with alerts flooding.

IDS alerts aggregation is the process of grouping two or more raw IDS alerts that are close in time and have similar features into one hybrid-alert [1]. In other words it fuses together different views (alerts) of the same event (intrusion). IDS alerts aggregation techniques use alerts similarity to aggregate and summarize alerts. However, it is a challenge to define similarity measure between alerts. This is mainly because most of the alerts attributes are symbolic attributes not numerical attributes. Usually alert attributes are attacker IP address, target IP address, attack signature, alert time, protocol, packet size, etc. Most of the previous approaches use some trivial hamming distance or other ad-hoc techniques to measure the similarity between alters attributes. These similarity measurements are usually limited to specific attack pattern or alerts set and can not be generalized to handle any group of alerts or attack pattern.

Another problem in alerts aggregation is the quality of the generated hybrid-alerts. In other words the aggregation process should ensure that any valuable information in the raw alerts will exist in the resulting hybrid-alerts set. To our knowledge none of the proposed approaches can prevent information-loss. In fact, the main performance metric used for the evaluation of alert aggregation approaches is the *alert reduction rate* (ARR). The ARR equal 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. The percentage of the reduction means nothing if there is valuable information in the raw alerts set that is absent in the resulting hybrid-alerts set.

To address the problems that occur in the previous alerts aggregation techniques, we proposed a new alerts aggregation technique that can address these problems. In our approach we use semantic similarity and an intrusion analysis ontology to aggregate raw IDS alerts. Our aggregation approach start by clustering alerts into n number of clusters and then fuse the alerts in each cluster into one or more hybrid-alerts. We aggregate the alerts based on the semantic similarity between alerts attributes. During the alerts fusion two or more low level-alerts attributes are represented in the hybrid-alert by one high-level attribute from the ontology.

The contribution of this paper is three-folds. First, we proposed a new alerts aggregation technique based on semantic analysis and ontology. Second, we proposed a new metric to measure semantic similarity between concepts in any given ontology. Finally, we proposed a new quality-metric to measure the information-loss that result from the alerts aggregation process. The rest of this paper is organized as follows: Section II presents the related works. Section III describes the proposed approach in details. Section IV presents our experiment and we discuss the results with respect to previous work. Finally, the conclusion of the paper is presented in section V.

II. Related Work

In the literature several alerts aggregation and fusion approaches have been proposed to deal with alerts flooding. The majority of these approaches use the similarity of alerts attributes to aggregate and fuse alerts. To measure the similarity between the attributes of different alerts some approaches use perfect-match between alerts attributes. For instance, Debar and colleagues proposed an alerts aggregation approach that

aggregate two or more alerts if the attributes of these alerts are equal [2]. Zhigong proposed a real-time alert aggregation and correlation System [3] that uses five features, namely, source IP, source port, destination IP, destination port and intrusion signature. Only if the values of the five features for a given alerts set are equal, this alerts set will be aggregate into one meta-alert. There are many alert aggregation techniques that require perfect-match between alerts attributes [4], [5]. However, these approaches are not very effective with alerts flooding since it is limited to duplicated and redundant alerts.

In contrast, to trivial approaches that use perfect-match between alerts attributes, other approaches were designed to aggregate alerts that generated as a result of specific attack pattern. Fan and colleagues proposed a distributed IDS alert aggregation model [6]. The proposed techniques categorizes the alert into four intrusion classes. Alerts that belong to the same category will be aggregated into meta-alert. The alerts category affects the calculation of the similarity between alerts attributes. For example if the alerts belong to the scan-attack category then alerts can be aggregated even if their IP addresses are not equal. Experimental evaluation of the model using the DARPA 99 dataset yields an alert reduction rate of about 43.42%. In [7] an 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 can not aggregate alerts that are the results of the same intrusion attempt but have different intrusion signatures. This approach yielded an alerts reduction rate of 49.58% when applied to the DARPA 2000 dataset.

In addition to perfect-match similarity and known attack pattern there more sophisticated techniques to measure the similarity between alerts attributes. Some approaches define a specific similarity metric for each alert attribute [8], [9], [10], [11]. For example they defined a metric to measure the similarity between two IP addresses and defined a different metric to measure the similarity between attack signature. Most of these similarity metric are trivial hamming-distance functions. The performance of these approaches is in general better comparing to perfect-match similarity and known attack pattern. In fact, some of these approach reach an alerts reduction rate of 99%. However, there is no guarantee that the generated alerts will not loss any valuable information in the raw IDS alerts. In fact, high alert reduction can lead to the generation of very poor-quality hybrid/meta-alerts.

III. The Proposed Approach

In our approach we consider an IDS alert as a set of attributes, these attributes are either symbolic attribute such as intrusion signature or none-symbolic attributes such as alert-time. In our opinion the goal of alerts aggregation is to cluster raw IDS alerts that belong to the same attack instance and summarize these alerts without losing important information.

The key idea of our approach is that alerts that belong to the same attack instance are semantically similar, even if they described by different format (keywords). Therefore, if we can measure the semantic similarity between alerts we can effectively aggregate them. Our approach assumes the existence of an ontology encompassing all the concepts underlying each of the symbolic 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 [12]. Kruegel and Christopher argued that an ontology for intrusions is a prerequisite for true interoperability between different IDSs [13]. In the last few years several network intrusion ontologies and taxonomies have been proposed [14], [15], [16], [17], [18], [19], [20], [21]. 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.

Concepts within the ontology are organized into hierarchical structures known as taxonomies. Where each taxonomy contains concepts that are linked by subclass and superclass relations. This hierarchical structure play a major role in calculating the semantic similarity between two concepts and generating the hybrid-alerts. This is because the semantic similarity between concepts that belong to the same taxonomy is greater than the semantic similarity between concepts that belong to different taxonomies. In addition the first common ancestor of two concepts in the same taxonomy can be use to represent the two concepts in the generated hybrid-alert.

To demonstrate the usage of our semantic similarity based approach we present a simple example. Let us assume that we use a symbolic attribute to represent the type of intrusion in formatting alerts. An Information-Gathering is a subtype of the intrusion type. In the ontology we have a taxonomy structure (concept-tree) corresponding to intrusion (type) attribute (see [21] for more details). For the sake of simplicity we will only consider part of the Intrusion-Type taxonomy to explain our approach. Figure 1 is a subtree that describes Information-gathering attack type.

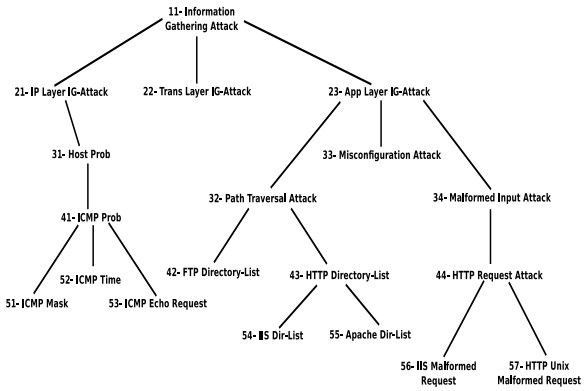

Fig. 1. Information-Gathering Attack Ontology (Partial)

Information-Gathering attack class describes the set of attacks that the attacker use to gather intelligence about his target. The attacker collects information about running services, live-hosts, or obtains credential information such

as user name and password. The root class in Figure 1 is the ***Information-Gathering Attack*** class. This class has three properties, namely, affected-system, method and impact, each subclass in the figure inherits these three properties. In general each subclass either will add a constraint on the inherited properties or will define a new properties. For example in the second level of the Information-Gathering ontology we have three classes. These classes are ***IP Layer IG-Attack***,***Trans Layer IG-Attack*** and ***App Layer IG-Attack***.The three classes added one additional constraint on the affected-system property. In the remaining of this paper we will use the partial ontology in Figure 1 to explain how we measure both the semantic similarity between intrusion attributes and the amount of information-loss in the aggregation process.

A. Semantic Similarity

As we mentioned before 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. 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 $x_{i1}, ..., x_{ip}$ attribute vector 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 $a_i = [x_{i1}, ..., x_{ip}$ and $a_j = [x_{j1}, ..., x_{jp}$, we define our semantic similarity metric between symbolic attribute values x_{ik} and x_{jk} $(1 \leq k \leq s)$ as follows:

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

Where $path(x_{ik}, x_{jk})$ is the length of the shortest path from concept x_{ik} to the common ancestor of x_{ik} and x_{jk} in the concept tree, and $depth(x_{ik})$ is the depth (height) of concept x_{ik} in the concept tree. The proposed similarity measure is a number between $[0, 1]$ where 1 correspond to exact match and 0 correspond to no match between the concepts. 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.

Now, let us consider the Information-Gathering Attack ontology in Figure 1. Using the equation 1, we can compute the semantic similarity between any two classes in the ontology. For example the semantic similarity between the IIS Dir-List class and the Apache Dir-List equal $sim(IISDir - List, ApacheDir - List) = 0.8$. Where the class HTTP Directory-List the least common ancestor of IIS Dir-List and Apache Dir-List and the depth of IIS Dir-List and Apache Dir-List equal 5. Using the ontology and the semantic similarity metric we calculate one semantic similarity matrix for each intrusion alert symbolic attribute. Given a concept tree consisting of n concepts, the corresponding semantic similarity matrix is an $n \times n$ matrix, in which each cell s_{ij} corresponds to the semantic similarity measure between concepts numbers i and j.

Given equation 1 we can calculate the semantic similarity between two alerts a_i and a_j where both have the same number of attributes number of attributes using equation 2.

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

B. Information-Loss

Aggregating raw IDS alerts into hybrid alerts will lead unavoidably to loss of information. In fact any data aggregation or summarization process will lead to loss of information. In our work, two concepts that belong to the same domain are aggregated by replacing them by their least common ancestor (LCA) in the ontology. 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 between concept c_1 and it's 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.

Several approaches have been proposed to calculate the information content of a concept in an ontology [22], [23]. In our work we calculate the information content of a concept c based on the importance of the semantic properties that identify the concept c. To measure the importance of each semantic property we use the approach of semantic-relevance introduced by Sartori and Lombardi [24]. According to Sartori the semantic relevance of a property for a given concept can be understood as a combination of two different components; the dominance of the property for that concept and the distinctiveness of the semantic property in a targeted semantic domain. Using the definition of semantic relevance (SR) we can compute the IC of a concept as follows:

$$IC(c) = \sum_{i=1}^{|P|} SR(p_i) \quad (3)$$

Where P is the set of semantic properties of the concept c and $SR(p_i)$ is the semantic relevance of the i^{th} property in P.

Now, given a set of concepts C, we define the information loss (IL) that result from replacing the concepts of C by their least common ancestor a by the following metric:

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

Where $LCA(C)$ corresponds to the least common ancestor of the concepts in C. The information loss is a value between **0** and **1**. The information loss for a hybrid-alert H is summation of the information loss of each attribute in H over the total number of attributes in H.

C. Alert Aggregation Algorithm

Our alerts aggregation algorithm groups alerts into several alerts clusters based on the semantic similarity between the alerts. The algorithm generates for each alerts cluster a hybrid-alert that represents the alerts of that cluster. The algorithm insert an alert into an alert cluster if the semantic similarity between that alert and the hybrid-alert of that cluster greater than a predefined semantic similarity threshold. The algorithm regenerates the hybrid-alert of the cluster when it inserts an alert into that cluster by fusing the attributes of the hybrid-alert with the attributes of the alert that was inserted in the cluster. The fusion of two attributes of two different alerts by replacing them with their least common ancestor in their concept tree in the ontology. For example, let us assume 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 $HTTPDirectory - List$ their least common ancestor according to the concept tree in Figure 1.

The algorithm proceeds in n number of rounds where n equal the number of attributes in one alert (note that all the alerts have the same number of attributes). The input of each round is a set of alerts and a semantic similarity thresholds vector. The thresholds vector is a vector of size n that contains a semantic similarity threshold value for each type of symbolic alert attribute. So there is a different threshold variable for the source attribute, the target attribute, the attack-type attribute and so on. At each round the algorithm cluster the alerts using the thresholds vector and generates one hybrid-alert for each cluster. At the end of each round the algorithm update the semantic similarity thresholds vector and use the updated thresholds vector and the hybrid-alerts that were generated during that round as the inputs to the next round.

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 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. 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 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.The main steps of our alerts aggregation algorithm are illustrated in Algorithm 1.

IV. Experiment

In this section we evaluate the proposed alert aggregation algorithm. We use two will known intrusion datasets, namely, the the DoS1.0 version of the DARPA 2000 dataset [25] and the Treasure Hunt dataset [26] and a private dataset collected in our lab. In our evaluation we use three attribute to represent each IDS alert. These attribute are the attack-source, attack-target and the intrusion-type. We used snort IDS version 2.8.4 to analyze the datasets and generate the raw IDS alerts. We compute the alert reduction rate which is the most widely used metric to evaluate alert aggregation approaches. In addition we use the information loss metric to measure the amount of information loss generated from our approach.

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 6 different threshold vectors listed in Table I.

Table II shows some statistical information about each dataset after analyzing it with Snort such as the numbers of alerts, number of hosts, durations and number of different intrusion instances.

For each dataset, we run our alert aggregation algorithm 6 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 III 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 2 illustrates the APCs obtained for the different datasets.

As we can see from Table III our alerts aggregation can reach high level of alerts reduction rate with reasonable information loss rate. By analyzing the results we found that in

Algorithm 1: IDS Alerts Aggregation Algorithm

```
/* A a set of intrusion alerts of size n                    */
/* T semantic similarity threshold vector                   */
Input: A, T
Output: H
1  begin
       /* Th: threshold vector of size p                    */
       /* C: set of alerts clusters                         */
2      Th ← [1, ..., 1] ;
3      i ← 1;
4      A' ← A;
5      C ← φ ;
6      while i ≤ p do
7          for j = 0 to size(A') do
8              a ← A'[j];
9              for s = 0 to size(C) do
10                 c ← C[s];
11                 h ← hybrid-alert of cluster c;
12                 x ← true;
13                 for l = 1 to p do
14                     if (similarity(a[l], h[l]) ≤ Th[l])
                       then
15                         x ← false;
16                         break;
17                     end
18                 end
19                 if x = true then
20                     insert a into c;
21                     h ← fuse a with h;
22                     H ← h;
23                     break;
24                 end
25             end
26             if x = false then
27                 let c new cluster and insert a into c;
28                 let h hybrid-alert of c = a;
29                 H ← h;
30             end
31         end
32         A' ← H;
33         Th[i] ← T[i];
34         i ← i + 1;
35     end
36     return H;
37 end
```

general higher alerts reduction rate means higher information-loss rate.

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

Vector ID	Source	Target	Attack-Type
V0	1	1	1
V1	0.8	0.8	0.9
V2	0.8	0.8	0.8
V3	0.6	0.6	0.8
V4	04	0.4	0.45
V5	0.16	0.16	0.12

TABLE I
SEMANTIC SIMILARITY THRESHOLD VECTORS

Dataset	Treasure Hunt	DARPA 2000	Lab Traffic
Alerts	199587	2170	2048
Intrusions	18	16	63
Intruders	5	273	115
Target	4	738	836
Duration	≈ 3 min	≈ 100 min	≈ 900 min

TABLE II
INTRUSION DATASETS INFORMATION

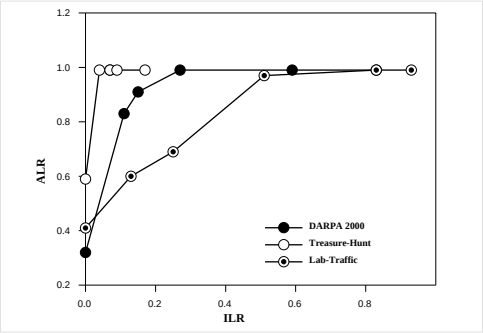

Fig. 2. APCs for single sensor IDS alerts aggregation

Vector	DARPA		Treasure Hunt		Lab Traffic	
	ALR	ILR	ARL	ILR	ALR	ILR
V0	0.32	0	0.59	0	0.41	0
V1	0.83	0.11	0.99	0.04	0.6	0.13
V2	0.91	0.15	0.99	0.07	0.69	0.25
V3	0.99	0.27	0.99	0.07	0.97	0.51
V4	0.99	0.59	0.99	0.09	0.99	0.83
V5	0.99	0.83	0.99	0.17	0.99	0.93

TABLE III
SINGLE SENSOR ALERTS AGGREGATION

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 a 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 [6] or define a set of rules to aggregate the alerts based on the type of attack pattern (see for instance, [27], [2]).

The hybrid alert that represent the DDoS attack in the DARPA 2000 is illustrate in Figure 3. Where the source of

the hybrid-alert is an aggregate attribute-value that represent a set of IP addresses that belong to local network, the target of the attack is an off-site IP address. The intrusion is an mstream-DDoS which is also an aggregation of the original two snort signatures in the raw IDS alerts. Finally the time attribute that shows the duration of time where all the raw IDS alert that are represented by the hybrid-alert occurred.

$$HA = \left\{ \begin{array}{cccc} Source & Target & Intrusion & Time \\ 127.X.X.X & 131.84.1.31 & mstream & 9.27.51 \rightarrow 9.27.57 \\ & & DDoS & \end{array} \right\}$$

Fig. 3. Hybrid-Alert that presents mstream DDoS Attack

V. Conclusion

In this paper, we proposed a new alert aggregation technique based on the semantic features of IDS alerts. 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 attribute. In addition we proposed a new method to measure the quality of the alert aggregation process based on the amount of information loss. Aggregating and summarizing security log files such as IDS alerts log, firewalls log, etc without considering the amount of information loss can make the aggregation process useless. Moreover, the use of semantic features to model IDS alerts allows us 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 control significantly IDS alerts flooding and perform than existing approaches.

References

[1] T. Zang, X. Yun, and Y. Zhang, "A survey of alert fusion techniques for security incident," in *Web-Age Information Management, 2008. WAIM '08. The Ninth International Conference on*, pp. 475 –481, july 2008.

[2] 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*, (London, UK), pp. 85–103, Springer-Verlag, 2001.

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

[4] 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*, pp. 695 –700, 2010.

[5] 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*, (Washington, DC, USA), pp. 314–318, IEEE Computer Society, 2009.

[6] 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*, pp. 975 –980, 2009.

[7] F. Valeur, G. Vigna, C. Kruegel, and R. Kemmerer, "Comprehensive approach to intrusion detection alert correlation," *Dependable and Secure Computing, IEEE Transactions on*, vol. 1, pp. 146 – 169, jul. 2004.

[8] 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. vol 2, pp. 124–127, 2008.

[9] 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*, (Washington, DC, USA), pp. 469–471, IEEE Computer Society, 2007.

[10] M. Xu, T. Wu, and J. Tang, "An ids alert fusion approach based on happened before relation," pp. 1 –4, oct. 2008.

[11] A. Hofmann and B. Sick, "Online intrusion alert aggregation with generative data stream modeling," *Dependable and Secure Computing, IEEE Transactions on*, vol. vol 8, no. num 2, pp. 282 –294, 2011.

[12] V. Raskin, C. F. Hempelmann, K. E. Triezenberg, and S. Nirenburg, "Ontology in information security: a useful theoretical foundation and methodological tool," in *NSPW '01: Proceedings of the 2001 workshop on New security paradigms*, (New York, NY, USA), pp. 53–59, ACM, 2001.

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

[14] 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.*, vol. 26, pp. 211–254, September 1994.

[15] S. Hansman and R. Hunt, "A taxonomy of network and computer attacks," *Computers & Security*, vol. 24, no. 1, pp. 31 – 43, 2005.

[16] 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.

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

[18] 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)*.

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

[20] A. S. Peter, P. S, and L. V. Ekert, "An ontology for network security attacks," in *Proceedings of the 2nd Asian Applied Computing Conference (AACC 04), LNCS 3285*, pp. 317–323, Springer-Verlag, 2004.

[21] S. Saad and I. Traore, "Method ontology for intelligent network forensics analysis," in *Eight International Conference on Privacy, Security and Trust (PST 2010)*, (Ottawa, Canada), pp. 7–14, 8 2010.

[22] 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.

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

[24] G. Sartori and L. Lombardi, "Semantic relevance and semantic disorders," *J. Cognitive Neuroscience*, vol. 16, pp. 439–452, April 2004.

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

[26] UCSB, "The 2002 ucsb treasure hunt dataset." http://ictf.cs.ucsb.edu/data/treasurehunt2002/.

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