Extracting Attack Scenarios Using Intrusion Semantics

Sherif Saad and Issa Traore

University of Victoria, BC, Canada
shsaad@ece.uvic.ca , itraore@engr.uvic.ca

Abstract. Building the attack scenario is the first step to understand an attack and extract useful attack intelligence. Existing attack scenario reconstruction approaches, however, suffer from several limitations that weaken the elicitation of the attack scenarios and decrease the quality of the generated attack scenarios. In this paper, we discuss the limitations of the existing attack scenario reconstruction approaches and propose a novel hybrid approach using semantic analysis and intrusion ontology. Our approach can reconstruct known and unknown attack scenarios and correlate alerts generated in multi-sensor IDS environment. Our experimental results show the potential of our approach and its advantages over previous approaches.

Keywords: Attack Scenario, Alerts Correlation, Intrusion Analysis, Semantic analysis

1 Introduction

In the last several years the number of computer network attacks has rapidly increased while at the same time the attacks have become more and more complex and sophisticated. Intrusion analysts and network administrators need to understand these attacks to take appropriate responses and design adequate defensive and prevention strategies. In particular, they need to reconstruct the attack scenario (also known as attack plan) to extract attack intelligence. The attack scenario elicits the steps and actions taken by the intruder to breach the system. Understanding the attack scenario allows the intrusion analyst to identify the compromised resources, spot the system vulnerabilities, and determine the intruder objectives and the attack severity.

The current generation of intrusion detection systems (IDSs) generate low level intrusion alerts that describe individual attack events. In addition, existing IDSs tend to generate massive amount of alerts with high rate of redundant alerts and false positives. Typical IDS sensors report attacks independently and are not designed to recognize attack plans or discover multistage attack scenarios. Moreover, not all the attacks executed against the target network will be detected by the IDS. False negatives, which correspond to the attacks missed by the IDS, will either make the reconstruction of the attack scenario impossible or lead to

an incomplete attack scenario. Because of the above mentioned reasons, manual reconstruction of attack scenarios is a challenging task. Hence, there is a pressing need for new techniques allowing automatic reconstruction of attack scenarios.

We propose, in this paper, a new attack scenario reconstruction technique, which improves the attack scenario reconstruction process by combining two complementary approaches: semantic-based alerts clustering and causality-based attack analysis. More specifically, an initial set of candidate attack scenarios are first identified by measuring the similarity between IDS alerts through semantic analysis. The candidate attack scenarios are then refined by analyzing the causal relationships between them using an intrusion ontology.

We evaluated experimentally our approach using two popular datasets yielding excellent performances. In the literature the ***completeness*** (also known as the true detection rate) and ***soundness*** of the alerts correlation are the most adopted metrics to evaluate attack scenario reconstruction approaches. The two metrics were proposed by Ning et al [8]. Completeness is computed as the ratio between the number of correctly correlated alerts by the number of related alerts (i.e. that belong to the same attack scenario). Soundness is defined as the ratio between the number of correctly correlated alerts by the number of correlated alerts. The completeness metric captures how well we can correlate related alerts together while the soundness metric assesses how correctly the alerts are correlated.

The experimental evaluation of our approach yielded for both datasets, soundness and completeness ranging between 96% and 100% for the sample attack scenarios considered.

The remaining of the paper is organized as follows. Section 2 summarizes and discusses previous works on attack scenario reconstruction. Section 3 introduces our semantic model and the underlying concepts and metrics. Section 4 presents in detail our attack scenario reconstruction technique. Section 5 shows the result of our experiment. Finally, in section 6 we conclude this paper and point out some future research directions.

2 Related Works

Several approaches have been proposed in the literature for attack scenario reconstruction. The proposed approaches fall into one of two main categories based on the type of data analysis techniques involved as explained below.

The first category of attack scenario reconstruction approaches use data clustering and data mining techniques, either to cluster alerts based on their attributes similarity or to mine alerts sequences in specific time interval. Under this category fall the approaches proposed by Li *et al.*, Ding *et al*, and Al-Mamory and Zhang, respectively.

Li *et al.* investigated multi-step attack scenario reconstruction using association rule mining algorithms [5]. The authors assumed that multi-step attacks often happen in a certain time interval and based on this assumption an attack sequence time window is defined and used for association rule mining. The

DARPA 2000 dataset was used to evaluate the proposed approach yielding attack scenario detection rate of 92.2%.

Ding *et al.* proposed an attack scenario reconstruction model by extending the apriori association rule mining algorithm to handle the order of intrusion alerts occurrence [3]. The authors introduced, more specifically, a time sequence apriori algorithm for mining intrusion alerts with respect to their order of appearance. The DARPA 1999 dataset was used to evaluate the proposed algorithm. The evaluation results show that the true scenario detection rate is 76% while the soundness of the approach is 53%.

Al-Mamory and Zhang proposed a lightweight attack scenario reconstruction technique by correlating IDS alerts based on their statistical similarity [2]. In the proposed approach, similar raw IDS alerts are grouped into meta-alert (MA) messages. An attack scenario is generated by correlating MA messages using a relation matrix (RM) that defines the similarities between every two MA messages. Using the DARPA 2000 dataset, it was shown that the completeness and the soundness of the proposed approach are 86.5% and 100%, respectively.

Attack scenario reconstruction systems that use clustering and data-mining approaches can handle large amount of IDS alerts and in general can reconstruct novel and unknown attack scenarios. They suffer, however, from several limitations. One of these limitations is the inability of the techniques to reconstruct complex or sophisticated multi-step attack scenarios. This is because clustering and data-mining approaches cannot detect causality between individual attacks. Another important issue is their proneness to construct incorrect attack scenarios. For instance, the alert clustering process may lead to overlapping alerts clusters. Alerts from the same scenario may end up in different alerts clusters, while alerts from different scenarios may be placed in the same cluster. It is not possible, however, for one alert instance to belong to two different attack scenarios at the same time. Such situation can occur because either an alert actually belongs to one scenario and is falsely clustered into the other scenario, or there is only one real attack scenario, and the reconstruction technique falsely assumes that there are two scenarios.

The second category of approaches use, in most cases, rule bases for attack scenario reconstruction, and represent attack scenarios and attack knowledge using formal methods. Examples of works that fall under this category include proposals by Ning *et al.*, Ding, and Liu *et al.*, respectively.

Ning and colleagues proposed an attack reconstruction approach that correlates intrusion alerts based on the prerequisites and the consequences of the intrusion [8]. The intrusion prerequisites are the necessary conditions for the intrusion to occur and the intrusion consequences are the outcomes of successful intrusions. The DARPA 2000 DOS 1.0 attack scenario dataset was used to evaluate the proposed technique, yielding an equal value for the completeness and the soundness of 93.96% .

Liu and colleagues proposed a multi-step attack scenario reconstruction technique using predefined attack models [7]. The proposed technique defines attack models that an attacker may follow to break in the system. Each defined attack

model follows a general attack pattern involving four phases: probe, scan, intrusion, and goal. The attack scenario reconstruction is executed over three main stages, namely, preprocessing stage, attack graph construction stage, and scenario generation stage. The proposed technique was evaluated using the DARPA 2000 LLDOS1.0 dataset achieving 87.12% completeness and 86.27% soundness.

The above knowledge-based approaches can reconstruct both known and unknown attack scenarios as long as the individual attack steps are stored in the knowledge-base. In addition some of these approaches can capture the causality between individual attacks. However, most of the proposed systems use hard coded knowledge and rely on explicit knowledge. As a result, these techniques fail to detect hidden and implicit relations between attacks, which makes it difficult for them to recognize novel attack instances in a timely fashion. Moreover, knowledge-based techniques cannot handle concurrent attacks that do not have any explicit causal relationship.

Based on the above literature review, it is clear that new approaches are needed that can handle large amount of IDS alerts and allow reconstructing automatically novel and unknown attack scenarios with high accuracy. In this regard, we propose a new attack scenario reconstruction approach that is a hybrid of clustering and knowledge-based techniques.

To improve the accuracy of clustering-based reconstruction, a robust alert clustering criteria must be defined. Clustering IDS alerts is difficult because many alerts attributes are symbolic data, and also heterogeneous IDS sensors tend to use different formats and vocabularies to describe the alerts. To address the above challenges, we propose to cluster the alerts based on their semantics and not their syntactic representations. After clustering, we refine using semantic inference the obtained clusters by identifying causally related alerts subsets and linking such subsets to specific attack scenarios.

3 Intrusion Semantic Analysis

We use an ontology to describe the intrusion domain and encode our knowledge base. The use of an ontology involves two main advantages. Firstly, it provides a common vocabulary to describe IDS alert messages generated by different IDS sensors. This allows achieving interoperability between heterogeneous IDS sensors. Secondly, it provides a semantic representation for the domain of computer and network intrusions. Using the semantic representation of IDS alerts and intrusion instances allows analyzing the alerts and the intrusion based on their semantic characteristics, and inferring the underlying relationships.

Several network intrusion ontologies have been proposed in the literature [1, 13, 4]. We use in our work a new intrusion ontology, introduced in our previous work [11] that contains the required knowledge to extract intrusion intelligence.

The intrusion ontology contains many classes representing different concepts from the intrusion analysis domain. The upper level classes of our intrusion ontology are illustrated in Figure 1. Classes in the ontology are connected by arcs representing the relations between them.

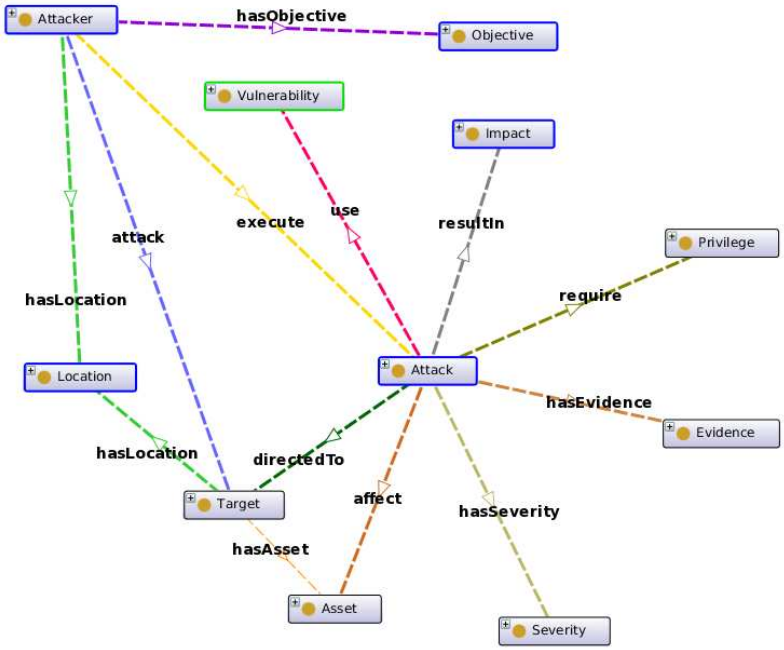

Fig. 1: Intrusion Ontology Screenshot

The relations between concepts can be quantified by measuring their *semantic relevance*. In knowledge engineering and information retrieval, the notion of *relevance* expresses how two objects are related with respect to the matter at hand. Semantic relevance occurs between classes and individuals in the same ontology through either explicit relations or implicit relations. Several approaches have been proposed to calculate the semantic relevance between concepts, objects or resources in specific domain of knowledge [10, 9]. We propose in this work, a new metric to capture the semantic relevance between intrusion alerts based on the relations occurring between them through our ontology. More specifically, we compute the semantic relevance between two alerts x and y as the summation of the weights of all the relations occurring between them divided by the summation of the weights of all the relations that can occur between any two alerts.

Given two alerts $x \in A, y \in A$, let R_{xy} denote the set of all relations between x and y. Let R denote the set of all relations between alerts pairs from A, i.e., $R = \cup_{x \in A, y \in A} R_{xy}$. Given a relation $r \in R$, let $w(r)$ denote the weight associated with r. We define the semantic relevance between alerts x and y as follows:

$$sem_{rel}(x, y) = \frac{\sum_{r \in R_{xy}} w(r)}{cardinality(R)} \quad (1)$$

In order to compute the semantic relevance between two alerts, we need to identify their relations, both implicit and explicit. While explicit relations are drawn from predefined ontological relationships, implicit relations are discovered through semantic inference.

A subset of the ontological relations used to calculate the semantic relevance between alerts are shown in Figure 2. Using these relations, a set of inference rules were designed. The rules are represented in the Semantic Web Rule Language (SWRL) and stored as XML files in the knowledge-base. Table 1 shows some of the predicate sentences (used to define the rules) and their meanings.

Predicate Sentence	**Description**
Alert(?x)	check if variable x is an Alert instance
Attack(?a)	check if variable a is an Attack instance
report(?x,?a)	check if variable a which is an attack instance is reported by x which is an alert instance
Impact(?m) ∧ resultIn(?a,?m)	check if variable a which is an attack instance has an impact m which is an instance of attack impact class

Table 1: Predicates Sample

The following is an example of an inference rule that finds if two alerts have the same attacker:

$$Alert(?x) \wedge Alert(?y) \wedge Attacker(?a) \wedge hasSource(?x, ?a) \wedge hasSource(?y, ?a) \rightarrow hasSameAttacker(?x, ?y)$$

A chain of rules can be used to infer an indirect relation between two alerts. For example, it can be established by inference that two different alerts that report two different attack types while having the same impact are relevant. An example of SWRL rule to infer alerts with similar attack impact is given by:

$$Attack(?a) \wedge Attack(?b) \wedge Impact(?m) \wedge resultIn(?a, ?m) \wedge resultIn(?b, ?m) \rightarrow hasSameImpact(?a, ?b)$$

$$Alert(?x) \wedge Alert(?y) \wedge Attack(?a) \wedge Attack(?b) \wedge report(?x, ?a) \wedge report(?y, ?b) \wedge hasSameImpact(?a, ?b) \rightarrow reportSameImpact(?x, ?y)$$

4 Attack Scenario Reconstruction

4.1 General Approach

Our attack scenario reconstruction process starts by collecting raw alerts generated by different (heterogeneous or homogeneous) IDS sensors, with different formats and containing possibly some false positives. The collected raw alerts are

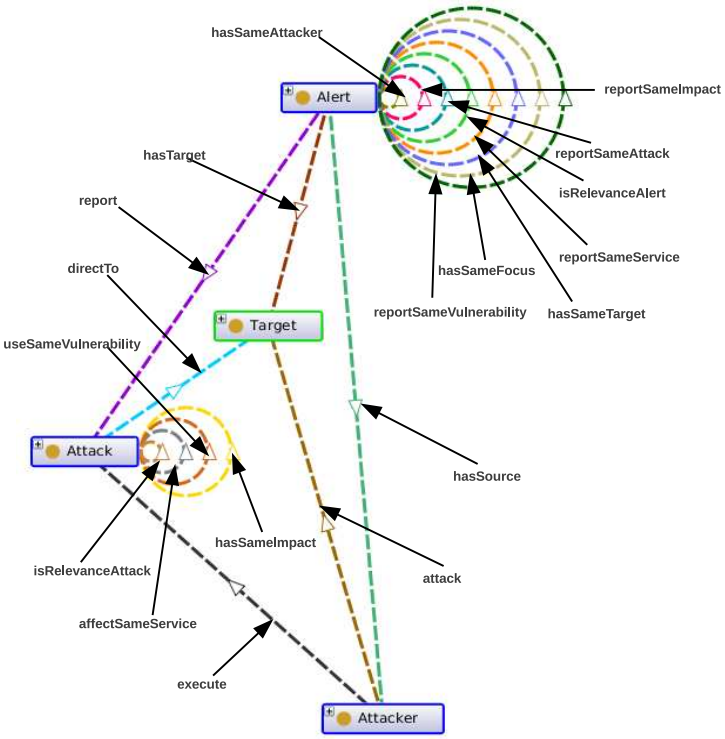

Fig. 2: Ontological Relations between Alerts, Attack, Attacker and Target

preprocessed by converting them into a common format that takes into account both the structures and semantics of the alert messages. Then, the converted alerts are validated by eliminating possible false positives. To convert the alerts into a common format, a separate profile is built for each IDS sensor. Each sensor profile contains a set of formatting rules used to convert raw alerts into a predefined format based on the vocabularies in the intrusion ontology.

The alerts resulting from the previous phases are grouped into several clusters based on their semantic relevance. The obtained clusters are analyzed using semantic inference to detect the causality relation between corresponding alerts. Then, the attack scenarios are extracted using semantic inference.

4.2 Semantic-based Alerts Clustering

The objective of semantic-based alerts clustering is to find groups of alerts that are semantically relevant with respect to particular attack scenarios. A cluster of semantically relevant alerts represents a candidate attack scenario. Given a set A of n number of alerts there are $2^n - 1$ possible alerts groupings, where

each alert grouping corresponds to a candidate attack scenario. A generated candidate attack scenario may correspond to a true or false attack scenario.

Based on the inferred relations between alerts, we calculate the semantic relevance between them and construct what we refer to as the alerts correlation graph (ACG). The ACG is an undirected weighted graph $G = (V, E)$, where V is a set of vertices representing alerts and E is a set of edges representing the relations between alerts. The edges in the ACG are labeled by the values of the semantic relevance between the alerts corresponding to adjacent vertices.

As an example, suppose we want to construct the ACG for the set of alerts given in Table 2.

ID	Source	Target	Attack
a_1	201.134.12.11	172.16.112.10	Scan
a_2	201.134.12.11	172.16.116.44	Scan
a_3	135.13.216.191	172.16.113.84	Scan
a_4	201.134.12.11	172.16.112.10	BufferOverFlow
a_5	135.13.216.191	172.16.116.44	Scan
a_6	201.134.12.11	172.16.112.10	RootAccess
a_7	135.13.216.191	172.16.116.44	TelnetAccess

Table 2: Alerts Examples

For the sake of simplicity we will assume that only three types of relations can occur between any two alerts, namely, ***hasSameSource***, ***hasSameTarget*** and ***reportSameAttack***, and also that each relation has a weight value equal 1. This means that the maximum number of relations between any two alerts is 3. Based on the above considerations, the constructed ACG for the alerts set in Table 2 is shown in Figure 3.

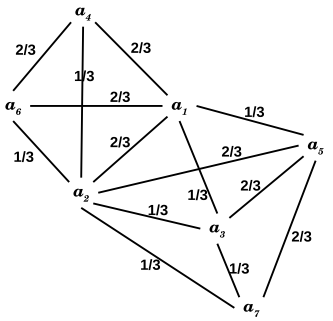

Fig. 3: Example of Alerts Correlation Graph

The edges of the ACG in Figure 3 are labelled by the semantic relevance values between corresponding alerts. For instance, alerts a_1 and a_6 being linked

by two relations (i.e. ***hasSameSource*** and ***hasSameTarget***), the semantic relevance between them is 2/3.

Algotithm 1 illustrates the steps to build the Alerts Correlation Graph. The algorithm takes a set A of hybrid or commonly formatted alerts as an input and generate the alerts correlation graph as an $n \times n$ matrix G where n is the total number of alerts in A. The entry $G[i, j]$ is zero if the semantic relevance between alerts a_i and a_j in A is less than a predefined semantic relevance threshold θ. If the semantic relevance value w is greater than or equal θ the algorithm set the value of $G[i, j]$ equal to w, which indicates that there is an edge e between a_i and a_j in G with weight w. The runtime complexity of Algorithm 1 is $O(n^2)$.

Algorithm 1: Constructing Alerts Correlation Graph

```
   /* A a set of IDS alerts                                          */
   /* G a matrix represent the ACG                                   */
   /* w a semantic relevance between a pair of alerts in A           */
   /* θ semantic relevance threshold                                 */
   /* n number of alerts in A                                        */
   Input: A, θ
   Output: G
 1 begin
 2     for i ← 1 to n − 1 do
 3         for j ← i + 1 to n do
 4             w ← sem_rel(a_i, a_j);
 5             if w ≥ θ then
 6                 G[i, j] ← w ;
 7             end
 8         end
 9     end
10     return G;
11 end
```

In graph theory a clique in an undirected graph is a subset of its vertices such that every two vertices in the subset are connected by an edge. In our case a clique in the ACG represents a subset of semantically relevant alerts. Therefore, we consider every maximum clique in the ACG as a candidate attack scenario. We use the well-known Bron-Kerbosch algorithm to find all maximum cliques in the ACG. In the ACG shown in Figure 3, there are three maximum cliques as illustrated by Figure 4.

Now let c_1, c_2, and c_3 denote the three maximum cliques in the ACG of Figure 4, where $c_1 = \{a_1, a_2, a_4, a_6\}$, $c_2 = \{a_1, a_2, a_3, a_5\}$ and $c_3 = \{a_2, a_3, a_5, a_7\}$. By looking closely at the above three candidate attack scenarios, we notice that they have some common vertices (alerts). For example, a_2 belong to all three of them. Considering that an alert can belong to only one attack scenario, we

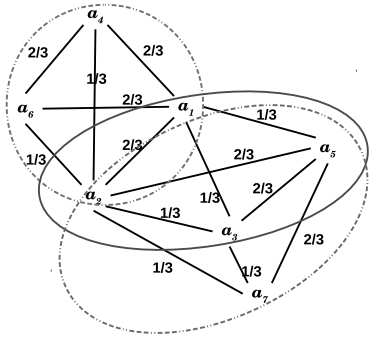

Fig. 4: Maximum Cliques in an Alerts Correlation Graph

need to refine our set of candidate attack scenarios by removing common alerts between them.

To remove a common alert from different candidate attack scenarios, we calculate the total semantic relevance of the common alert with respect to each candidate attack scenario, and assign it to the candidate attack scenario yielding the maximum total semantic relevance. This process will be repeated until each alert is assigned to only one candidate attack scenario.

The total semantic relevance of an alert with respect to a specific attack scenario is the sum of the semantic relevance between this alert and other alerts in the same attack scenario. For example, in Figure 4 the total semantic relevance of vertex a_1 in c_1 is $(2/3 + 2/3 + 2/3 = 2)$ and in c_2 is $(2/3 + 1/3 + 1/3 = 1.3)$. Therefore, a_1 will be removed from c_2 and reassign to only c_1. By applying the same method to other common vertices, we will end up with only two candidate attack scenarios s_1 and s_2, where $s_1 = \{a_1, a_2, a_4, a_6\}$ and $s_2 = \{a_3, a_5, a_7\}$.

Algorithm 2 illustrates the main steps to extract the candidate attack scenarios from an alert correlation graph. The algorithm takes as input an alert correlation graph G generated by Algorithm 1. First the set C of maximum cliques are extracted from G using the Bron-Kerbosch algorithm. The alerts (or vertices) in each clique are sorted based on the alert number. To detect alerts that belong to more than one clique we apply a simple set intersection method, where each clique in C is treated as a set. The set intersection returns a list $A^{'}$of alerts (vertices) that belong to more than one clique. Then, the algorithm iterates for n times, where n is the total number of alerts in $A^{'}$. In each iteration the algorithm calculates the alert membership to each clique in C based on the total semantic relevance. At the end of each iteration an alert a is assigned to a clique c, where the membership of a with c is maximum. Then, a is removed from the other cliques in C. Finally the algorithm removes a from $A^{'}$ and terminates when $A^{'}$ is empty. In addition to extracting candidate attack scenarios,

Algorithm 2 addresses also the problem of shared alerts between the candidate scenarios.

Algorithm 2: Extracting Candidate Attack Scenario from ACG

```
   /* A a set of IDS alerts                                                    */
   /* G a matrix represent the ACG                                             */
   /* C a set of maximum clique in ACG                                         */
   /* A' a set of alerts that belong to more than one clique                   */
   /* m membership between an alert a and clique c                             */
   /* n number of alerts or vertices in ACG                                    */
   /* s number of alerts in A'                                                 */
   /* l number of maximum cliques in ACG                                       */
   Input: G
   Output: C
 1 begin
 2     C ← BronKerbosch(G);
 3     for i ← 1 to n do
 4         β ← 0 ;
 5         for j ← i to l do
 6             if a_i ∈ c_j then
 7                 β ← β + 1;
 8                 if β ≥ 2 then
 9                     add a_i to A';
10                     Break ;
11                 end
12             end
13         end
14     end
15     while A' ≠ ∅ do
16         max ← −1;
17         for i ← 1 to s do
18             for j ← 1 to l do
19                 m ← sum of the weights of all adjacent edges of a_i in c_j;
20                 if m ≥ max then
21                     max ← m ;
22                     sAlert ← a_i ;
23                     sClique ← c_j ;
24                 end
25             end
26         end
27         remove sAlert from A';
28         foreach clique c ∈ C do
29             if c ≠ sClique and sAlert ∈ c then
30                 remove sAlert from c;
31             end
32         end
33     end
34     return C;
35 end
```

The run time complexity of Algorithm 2 is $O(3^{n/3})+O(n\times l)+O(s^2\times l)$, where n is the number of alerts, s is the number of alerts shared between candidate attack scenarios, and l is the number of candidate attack scenarios in ACG.

4.3 Attack Causality Analysis

The semantic clustering only groups alerts that belong to the same attack scenario into one cluster. Likewise, the candidate attack scenarios generated from the semantic clustering do not provide any information about the sequencing of the attack or the steps the attacker executes to reach his objective. However, the main goal of the attack scenario reconstruction is to identify the sequence of steps and actions taken by the intruder to break into the system. An effective way to elicit the attack sequencing consists of analyzing the causality between the individual attacks reported in the IDS alerts.

To detect the causality between different attack instances, each attack instance is associated with both a set of prerequisites and a set of consequences. The attack prerequisites are the set of logical conditions to be satisfied for the attack to succeed while the attack consequences are the set of logical conditions that will become true when the attack succeeds. Two attacks a and b are causally related if at least one of the consequences of one of them is among the prerequisites of the second one.

The knowledge corresponding to the attack prerequisites and consequences is represented in the intrusion ontology by introducing attack prerequisites and attack consequences relations between the *Attack* class and the *Impact* class (see Figure 1). The attack prerequisites and consequences are defined as subclasses of the *Impact* class. For any two attack instances a and b, if there is an impact p where p is a consequence of a and a prerequisite of b, then there is a causality relationship between a and b. In other words the intruder will execute first a and then b. For instance, the success of a scanning attack that detects the presence of a vulnerable FTP server is a prerequisite for a buffer overflow attack against this FTP server. It is not possible for an intruder to execute the buffer overflow attack before the scanning attack. Now, let A denote the set of consequences of attack a and let B denote the set of prerequisites of attack b. We define the strength of the causality relation between a and b as a value between 0 and 1 given by equation 2, where 0 indicates no causality and 1 indicates maximum causality:

$$causality(a, b) = \frac{|A \cap B|}{|A \cup B|} \tag{2}$$

The process of detecting attack causality and reconstructing the attack scenario graph can be described as a graph transformation operation. The attack causality detection algorithm converts the complete graph representing the candidate attack scenario into a directed acyclic graph representing the reconstructed attack scenario. The transformation consists of simply replacing the edges in the alerts correlation graph corresponding to the semantic relevance relations between alerts with new edges that represent the causality relations between the attacks reported by the alerts.

Algorithm 3 describes the key steps of the attack causality analysis. The algorithm takes a clique (i.e. a candidate attack scenario) as an input and generates an attack scenario graph as an output. The input clique is represented

by a vector V of alerts sorted in ascending order based on their timestamps. The output of the algorithm is an attack scenario graph represented by a set of matrices denoted M. The algorithm starts by creating an empty matrix m_1 and inserts the first alert in V into m_1. Then the algorithm iterates $n-1$ times, where n is the size of V. In each iteration, the algorithm checks the causality between one alert a_i from V and every alert b in every matrix m_j in M using equation 2. If the causality measure equal zero for every alert in every matrix m_j in M, the algorithm creates a new matrix m_{j+1} and adds a_i to this matrix. If the causality measure is greater than zero then the algorithm will add a_i to the matrix that returns the maximum causality with a_i.

Algorithm 3: Attacks Causality Analysis

```
   /* V a sorted vector of alerts that belong to one clique                    */
   /* M a set of matrices that represent the attack scenario graph             */
   /* n number of alerts in V                                                  */
   /* l number of matrices in M                                                */
   Input: V
   Output: M
 1 begin
 2     create m_1 as an empty matrix in M;
 3     add V[1] to m_1;
 4     l ← 1;
 5     for i ← 2 to n do
 6         max ← 0;
 7         for j ← 1 to l do
 8             foreach alert b ∈ m_j do
 9                 δ ← causality(a_i,b);
10                 if δ > max then
11                     max ← δ ;
12                     sMatrix ← m_j ;
13                     sAlert ← b;
14                 end
15             end
16         end
17         if max ≠ 0 then
18             add a_i to m_j at sAlert;
19         else
20             l ← l + 1;
21             create m_l as an empty matrix in M;
22             add a_i to m_l;
23         end
24     end
25     return M;
26 end
```

The ideal output of the algorithm is the case where M contains a single matrix, which means that the attack scenario graph is a connected graph. The case where M contains more than one matrix indicates that the attack scenario graph is not a connected graph, which corresponds either to a false negative, a novel attack, or some missing causality information.

5 Experimental Evaluation

To evaluate our approach, we use two different datasets widely used in the literature, namely, the DARPA 2000 dataset from MIT Lincoln Laboratory [6] and the Treasure Hunt dataset [12]. Specifically, we used the LLDDOS1.0 subset of the DARPA dataset and the DMZ partition from the Treasure Hunt dataset. We analyzed the tcpdump files of the datasets using SNORT IDS version 2.9.2.0 running on Ubuntu box. Table 3 shows a summary of the contents of the datasets after analyzing them with SNORT IDS. These include the number of alerts (including redundant alerts) generated by SNORT for each dataset, the number of unique intrusions or attacks reported by SNORT, the number of source and destination IP addresses and the duration of generated network traffic.

Dataset	LLDDOS1.0	Hunt-DMZ
Alerts	2170	671848
Intrusions	16	49
Sources	273	28
Targets	738	37
Duration	$\approx$ 100 minutes	$\approx$ 893 minutes

Table 3: Datasets Statistics

We used the *soundness* and the *completeness* metrics, described earlier in the Introduction, to calculate the performance of our proposed approach.

By applying our approach to the DMZ partition of the treasure hunt dataset, 6 attack scenarios were detected, five of which were attack true attack scenarios and one was a false attack scenario. The true attack scenarios detected by our approach are the following: ***Protocol Exploit***, ***Reconnaissance***,***Privilege Escalation***, and two ***Web Exploit*** attack scenarios. All of these attacks target two machines inside the DMZ, while their sources are from 2 different subnets. The attackers kept executing these attack scenarios in a brute-force manner over a period of 15 hours. The false attack scenario is ***MySQL Root Attack***. The source of that attack is one machine inside the DMZ network and the target is a host in one of the Treasure Hunt internal networks.

We found that out of the total number of alerts (i.e. 671848), there are 628956 alerts related to the five attack scenarios. The remaining 42892 alerts are either false positives or single attack attempts that are irrelevant to any of the five attack scenarios. Our approach correlates 629426 alerts, 470 of which are alerts that are incorrectly considered part of the related alerts. Table 4 summarizes the performance results obtained for the different attack scenarios for the treasure hunt dataset.

Scenario	Correlated alerts	True alerts	Related alerts	Completeness	Soundness
Web Exploit 1	503337	503337	503337	100.00%	100.00%
Web Exploit 2	101071	100758	100758	100.00%	99.69%
Protocol Exploit	1730	1701	1705	99.77%	98.32%
Reconnaissance	3097	2973	3053	97.38%	96.00%
Privilege Escalation	20191	19981	20103	99.39%	98.96%

Table 4: Evaluation Results with the Treasure hunt Dataset

To compare our approach to previous approaches we used the LLDDOS1.0 attack scenario from the DARPA dataset, since most of the previous approaches used that dataset for evaluation. Table 5 shows the completeness and the soundness of our approach in comparison to previous works.

Approach	Completeness	Soundness
Ning et al	93.96%	93.96%
Liu et al	87.12%	86.27%
Al-Mamory and Zhang	86.5%	100%
Li et al	92.2%	not provided
Our Approach	100%	99.70%

Table 5: Comparison of Attack Scenario Reconstruction Approaches Using the LLDDOS1.0 Dataset

As shown by Tables 5 and 4, our approach outperforms many of the previous approaches. The completeness of our approach is promising and shows that our approach can correlate alerts that belong to the same attack scenario with high detection rate. At the same time the soundness of our approach is in general better than most of the previous approaches.

6 Conclusion

We have introduced in this paper a new attack scenario reconstruction technique using semantic and causality analysis. Our approach using semantic relevance to correlate related alerts based on their semantics. Experimental evaluation of our approach yields better results compared to previous works in the area of attack scenario reconstruction. Future work will aim at improving the run time of our approach and investigate the possibility of validating IDS alerts to effectively remove false positives and irrelevant alerts. In addition, predicting missing attack steps that result from IDS false negatives is another direction for future work. Missing attack steps can prevent or hinder the reconstruction of true attack scenario, therefore predicting missing attack steps is an essential requirement to improve the attack scenario reconstruction.

References

1. F. Abdoli and M. Kahani. Using attacks ontology in distributed intrusion detection system. In SCSS (1), pages 153–158, 2007.
2. S. O. Al-Mamory and H. L. Zhang. Scenario discovery using abstracted correlation graph. In Computational Intelligence and Security, 2007 International Conference on, pages 702–706, Dec. 2007.
3. Y.-X. Ding, H.-S. Wang, and Q.-W. Liu. Intrusion scenarios detection based on data mining. In Machine Learning and Cybernetics, 2008 International Conference on, volume 3, pages 1293–1297, July 2008.
4. 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).
5. W. Li, L. Zhi-tang, L. Dong, and L. Jie. Attack scenario construction with a new sequential mining technique. In Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Eighth ACIS International Conference on, volume 1, pages 872–877, 30 2007-Aug. 1 2007.
6. Lincoln-Laboratory-MIT. Darpa intrusion detection evaluation. http://www.ll.mit.edu/mission/communications/ist/CST/index.html.
7. Z. Liu, C. Wang, and S. Chen. Correlating multi-step attack and constructing attack scenarios based on attack pattern modeling. In Information Security and Assurance, 2008. ISA 2008. International Conference on, pages 214–219, April 2008.
8. P. Ning, Y. Cui, and D. S. Reeves. Constructing attack scenarios through correlation of intrusion alerts. In CCS '02: Proceedings of the 9th ACM conference on Computer and communications security, pages 245–254, New York, NY, USA, 2002. ACM.
9. S. K. Rhee, J. Lee, and M.-W. Park. Semantic relevance measure between resources based on a graph structure. In Computer Science and Information Technology, 2008. IMCSIT 2008. International Multiconference on, pages 229 –236, oct. 2008.
10. T. Ruotsalo and E. Hyvönen. A method for determining ontology-based semantic relevance. pages 680–688. 2007.
11. 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, 8 2010.
12. UCSB. The 2002 ucsb treasure hunt dataset. http://ictf.cs.ucsb.edu/data/treasurehunt2002/.
13. 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.