Similarity and Logic Based Ontology Mapping for Security Management

Alfred Ka Yiu Wong, Nandan Paramesh, Pradeep Kumar Ray

University of New South Wales, Australia
alfred.ky.wong@gmail.com, p.ray@unsw.edu.au, paramesh@cse.unsw.edu.au

Abstract

Ontological issues have been widely researched, especially in the semantic web where ontologies are developed to strengthen the semantic layer of web information. As a result, the proliferation of ontologies necessitates a mapping approach. The existing mapping approaches are generally developed for generic ontologies such as linguistic ontologies. We envisage ontologies as formal knowledge bases basing on which mobile intelligent agents will communicate and reason with in order to perform collaborative and distributive problem resolution in a dynamic environment. Furthermore, the volatility of the environment often requires approximation in reasoning. We present a similarity-based approach to ontology mapping for ontologies formally modeled in logic. The approach is based around a similarity function and uses SLD resolution as the basis to measure semantic likeness between concepts. In validation of our concept, the approach is experimented in the network security domain.

1. Introduction

The definition of ontology mapping can be defined as the matching of every concept from the source ontology to corresponding concept from the target ontology. Mapping can be either exact- or similarity- based. Exact mapping accepts total equivalence or none, while similarity-based mapping relaxes the constraint by multiply extending mathematical equality to values in the range of [0,1]. The latter approach, deemed as more practical in a dynamic environment, allows concepts to be mapped with similarity degree assigned. The similarity degree, in the range of [0,1], is measured by a similarity function S. S is conceptually the same as similarity indexes in similarity-based fuzzy approximate reasoning. The choice of S is dependent on the domain and the employed similarity model(s). The standard similarity models include geometric, transformation, featural (Tversky 1997) and network models. Geometric model measures the distance between points, while transformation model is suitable for visual configuration. Featural model measures number of common and differing features and network model assesses hierarchical distances between concepts.

The existing ontology mapping approaches are mainly based on informal ontologies with the exception (Kalfoglou, & Schorlemmer 2002). (Rodriguez, & Egenhofer 2003), (A. Doan et al. 2002) and (Wiesman, Roos, & Vogt 2001) present mapping approaches on ontologies that contain generic concepts semantically modeled in relation to WordNet. The mapping approaches base on ideas ranging from text categorization, set and graph comparison to machine learning techniques. The notion of similarity is often made explicit by the definition of a similarity function. (Kalfoglou, & Schorlemmer 2002) attempts to express ontologies with generic concepts in prolog logic. Information flow and channel theories are used to logically deduce mapping between concepts. However, the proposed mapping has to be exact and approximation is not possible.

Approximate reasoning is a domain relevant to our logical and similarity aspects of ontology mapping. (Yeung, & Tsang 1997) presents works on approximate reasoning with the use of similarity functions on fuzzy sets. Beside fuzzy logic based similarity, (Loia, Senatore, & Sessa 2001) presents a similarity-based SLD resolution on classical logic. Approximate solutions with associated approximation degree can still be computed when exact inference process fails. The main idea is to relax the equality constraint by using similarity relations between constants and predicates. The similarity relations are associated with values in the range of [0,1] that will be used to compute the approximation degree.

We propose a mapping approach for formally represented ontologies. Ontologies constructed with formal semantic definitions are essential for intelligent reasoning. The capabilities of agents in problem resolution guarantee their future importance. In our experimentation domain, security management, there are extensive studies on agent technology such as (R. Zhang et al. 2001). The ontological issues exhibited in the security domain, concern the heterogeneity and quantity of security information. Intelligent reasoning on the security information is required for correlation and to detect cyber attacks. Ontology plays an important role not only in the alleviation of the interoperability problem, but also the enabling of precise communications between mobile agents.

This paper is organized as follow: Section 2 presents the semantic model basing on which our ontology mapping will be performed. Section 3 details the approximate mapping methodology. Section 4 experiments the ideas from Section 2 and 3 in the domain of security intrusion detection. Lastly, Section 5 summarizes the works along with future research directions.

2. Semantic Model

Semantic model provides a guideline on how to capture meanings. It serves as a schema that governs the scope of semantic to be modeled. There are semantic models developed in other domains for different purposes. For example, in the image processing domain, meanings of images are captured by paying attention to their appearance. The purpose is to formalize visual characteristics in order to facilitate image query retrieval. Semantic models are also used in domains: proof-carrying code, validation of authentication protocol … etc. We are interested in a model that is capable of modeling dynamic concepts such as events, processes.

In contrast to most existing static semantic models, we consider the semantics of events, processes as dynamic. That is, their internal structures and variable states vary over time. The existing ontological approaches where concepts are semantically modeled with respect to class references, structures and so on, fail to adequately capture the notion of time. In the network management domain, the set of concepts include not only static object (e.g. *network object – routing table*), but also computational entities (e.g. process – *security monitoring operation*). The practice of using *slots* (Protégé) to represent individual static semantic relationships such as *part-of, function* and so on, is not sufficient to capture the more complex semantic relationships such as the *state transition* of *network objects* that are manipulated by a process. We stress that ***Semantics*** is distributed ***Over Space*** (***SoS***) and ***Over Time*** (***SoT***). The ***SoS*** by itself already requires the use of complex semantic relationships to model sophisticated object interactions. The addition of ***SoT*** further complicated the notion of semantics at abstract concept level. Thus, a more flexible mechanism is required to express and capture semantics. We suggest the use of first order logic (FOL) as the underlying representation language of ontologies. The powerful expressiveness of FOL and its success as a semantics specification language for concepts such as programming languages, motivate our choice. Throughout the paper we denote the semantics of a concept C that is modeled in FOL as L(C). In order to facilitate precise mapping and similarity measurement, the different aspects *i* of semantics (such as class references, structures) are modeled in separate FOL statements, denoted $L_i(C)$.

3. Mapping Methodology

In this section, we propose a similarity- and logic-based ontology mapping approach. The approach has two components. Section 3.1 proposes a logic-based similarity function **S** to measure similarity between two concepts. Section 3.2 discusses a strategy for locating concepts to be compared by **S**. That is, given a subject concept that has to be mapped, the strategy guides the search through the ontology such that the most similar concept can be identified. The two components together form our ontology mapping methodology.

3.1. Concept Mapping

Let C_1 and C_2 be two concepts from respective ontologies O_1 and O_2. We translate the problem of similarity assessment between C_1 and C_2 to similarity measurement between $L(C_1)$ and $L(C_2)$. By definition, $L(C_1)$ is logically equivalent to $L(C_2)$ if they have the same logical content. L(C1) is equivalent to L(C2), syntactically if L(C1) ↔ L(C2) is a theorm and; semantically if SemLogic(C1) ↔ SemLogic(C2) is a tautology. In order to prove semantic equivalence, we use SLD resolution to show that $L(C_1)$ and $L(C_2)$ have the same truth value in every model.

3.1.1. Semantic Equivalence

C_1 and C_2 are semantically equivalent if $L(C_1)$ and $L(C_2)$ are logically equivalent. That is, if we can infer through the process of SLD resolution that $L(C_1) \models L(C_2)$ and $L(C_2) \models L(C_1)$, then every model of C_1 is a model of C_2 and vice versa. The property of logical entailment $\models$ allows us to deduce some form of semantic equivalence, at least as good as how well the semantics is abstracted by $L(C_1)$ and $L(C_2)$.

3.1.2. Semantic Similarity

When either $L(C_1) \models L(C_2)$ or $L(C_2) \models L(C_1)$ is proven, a certain degree of similarity exists between C_1 and C_2. In fact, equivalence is a special case where similarity is at maximum. We aim to assess similarity between $L(C_1)$ and $L(C_2)$ when any $\models$ direction fails. Our purpose is to measure semantic similarity such that the underlying logical reasoning mechanism is not altered. In order to do so, we first of all analyze and interpret logic semantically such that semantics can be estimated by the complexity of logical statements. We define a logical statement as a set of assertions and an assertion as a disjunctive constituent:

Logical Statement as Semantic Assertions

$Sem_{Logic}(C)$ is a set of semantic assertions joined by and only by logical operators: $\wedge$ and $\leftrightarrow$.

Semantic Assertion

Semantic assertion can be viewed as a logical assertion or combination of logical assertions. Let $P = \{p_1, p_2 \ldots p_n\}$ be the set of logical predicates defined over some domain D. A semantic assertion is a subset of P joined by and only by logical operators: $\vee$ and $\rightarrow$.

According to the featural similarity models, similarity can be assessed by measuring the number of common and differing assertions. Our idea of similarity measurement is based around the paradigm of database query where approximate matching is performed implicitly between loosely specified query and accurate database records.

The notion of logical statement as a set of assertions is structurally the same as a database record or query compositing of attributes. The values of semantic assertions symbolize attribute values or query constraints.

The matching between query constraints and database record values is therefore similar to the matching between $L(C_1)$ and $L(C_2)$. Hence, the attempt in deriving $L(C_1) \models L(C_2)$ can be viewed as Database Record $\models$ Query. By employing the database record-query analogy, we are proposing an asymmetric similarity measurement scheme (how close is Database Record to Query). In fact, the asymmetric property is an integral essential feature of our ontology mapping strategy presented in Section 3.2. Throughout the forth coming discussion, lets assume the following scenario for similarity computation: $L(C_1) \models L(C_2)$, that is how close is $L(C_1)$ to $L(C_2)$.

The major difference between the database query paradigm and our settings is that attributes are explicitly labeled and semantic assertions are not. The identification problem is fortunately overcome by employing SLD resolution as part of the similarity assessment mechanism. SLD resolution is a logical process that attempts to prove $\models$. Its derivation can be formalized as follow:

Given a logic program P in first order language and a goal G. The derivation consists a sequence G_0, G_1, ... G_m of negative clauses from G, associated with a sequence Q_0, Q_1 ... Q_m of variants of clauses from P, and a sequence of substitution θ_0, θ_1 ... θ_m. G_i and Q_i resolve into G_{i+1}, and G_0, G_1 ... G_m yields the corresponding computed substitution θ_0, θ_1 ... θ_m.

As mentioned earlier, a semantic assertion is composed of predicates. The SLD resolution of $L(C_1) \models L(C_2)$ involves the identification of Q_i such that G_i can be resolved into G_{i+1}. Q_i is technically a variant of semantic assertion of C_1. Each resolution from G_i to G_{i+1} contributes to the progression towards the inference of a semantic assertion from C_2 in C_1. The identification of assertions to be compared is therefore implicit to SLD resolution. We propose a similarity function, at conceptual level, that measures how many semantic assertions from C_2 can be inferred in C_1.We define *Complement*$_{res}$(p,p') as the process of predicates complementation during resolution when p = ¬p'. That is, the process of using Q_i to resolve with G_i into G_{i+1}. When all compositional predicates of assertion a_n is complemented, we denotes it as *Complement*$_{res}$(a_n).

Definition 1: Similarity Function (Abstract)

$$S_{abstract}(L(C_1), L(C_2)) =$$
$$|\{a \mid a \in L(C_2), Complement_{res}(a)\}| \ / \ |L(C_2)|$$

where a denotes assertion, || denotes size; and $|Sem_{Logic}(C_2)|$ denotes the number of assertions in $Sem_{Logic}(C_2)$.

$S_{abstract}$ measures the number of semantic assertions from C_2 that can be inferred from C_1. The similarity value approaches 1 as more assertions of C_2 can be inferred from C_1. Note that $S_{abstract}$ assumes that if any semantic assertion can be inferred in C_1, it is completely inferred. That is, each assertion inference contributes 1 / $|L(C_2)|$ to $S_{abstract}$. We generalize the similarity function to not only consider how many assertions but also how much of them can be inferred. Consider the scenario of comparing two value sets: {a,b} and {a,b,c}. A similarity measure between the two value sets is to separately compute similarity for each value pair from the sets. The similarity value between the sets is the maximum possible sum of the similarity values for each pair, divided by the number of pairs formed. Assume the following matrix that indicates similarity value for the pairs:

	a	b	Null
a	1	0.3	0
b	0.3	1	0
c	0.5	0	0

The maximum summation would be the pairs (a,a) (b,b) and (c null), hence the similarity value of 2 / 3 between the sets. We are interested in introducing such a similarity matrix to $S_{abstract}$ such that partially inferred semantic assertions can be assessed and correctly weighted in the similarity function. We introduce a function ∂ that serves as a similarity matrix. ∂ measures how well a semantic assertion is inferred.

Definition 2: Resolution Quality of Assertion

$$\partial(a) = \forall p_i \in a \ \ (Max(\Omega(p_i, Q_i)))$$

where a is an assertion from C2 such that ¬a ∈ G_j; p_i denotes a predicate, Q_i is the variant of clauses from C_1 used along with G_i to resolve into G_{i+1}; and **Ω**, *defined below in Definition, is a function that assesses how well p_i is resolved during the resolution from G_i to G_{i+1}.*

Ω employs information theory to measure how well a predicate is resolved during SLD resolution. We argue that the more random the information is embedded in an assertion (Q_i in Defintion 2), the worse it is in resolving the predicate p_i. We analyze exhaustively how a predicate p(x) can be complemented:

A. (1) ~p(x) ∈ G_i (2) p(x) ∈ Q_i
(2) complements (1) through identity
B. (1) ~p(x) ∈ G_i (2) p(x) ∨ q(y) ∈ Q_i
(2) complements (1) with q(y) remains
Given an axiom: h(x) → p(x)
C. (1) ~p(x) ∈ G_i (2) h(x) ∈ Q_i
(2) complements (1) through axiom
D. (1) ~p(x) ∈ G_i (2) h(x) ∨ q(y) ∈ Q_i
(2) complements (1) through axiom with q(y) remains

Given the cases above, we employ *information theory* to quantify the information randomness of (2). The well known formula, $-\log_2$ prob, determines the information content of an event that has prob chance of occurring. The formula $\sum prob_i * -\log_2 prob_i$ is referred as *entropy*. Entropy, on the one hand, can be viewed as the measurement of information content of an event sequence. On the other hand, it can be deemed as measurement of the randomness and impurity of information. The latter notion is more suitable in semantic sense. The randomness of information embedded in (2) provides an indication in measuring how well (1) is resolved. The main idea is that the more random the information content is, the less accurate (2) is in resolving with (1) from G_i to G_{i+1}. When information randomness is 0, Ω should return maximum value 1. Hence we have Ω (in range of [0,1]) as follow:

Definition 3: Resolution Quality of Predicate

$$\boldsymbol{\Omega} \, (p_i, Q_i) = e^{-Entropy(Q_i)} * S_p(p_i, p')$$

where p' is a predicate from Q_i; and S_p is a function that measures the similarity between pi and p'. S_p can take the simple function $1 \,/\, 1 + d(C_{pi}, C_{p'})$ to measure the hierarchical distance between the corresponding concepts of the predicates (or other approaches).

The entropy of Q_i can be calculated according to the cases A, B, C and D. In case A, Q_i being p(x) firmly states that p(x) is true, hence the randomness of information is 0. In case B, Q_i being $p(x) \vee q(y)$ vaguely specifies that p(x) might be true. The information content is random in that p(x) can be estimated as dominating prob = 0.5 of the information content. In fact p(x), in a uniform context, can be generalized as dominating 1 / N of the information content where N is the number of predicates in Q_i. Similar to case A, the information randomness of (2) in case C is 0. However, S_p takes the similarity value between p(x) and h(x) instead of 1. Finally, in case D, the information randomness of (2) is similarly interpreted as in case B.

In complex scenarios where A, B, C and D co-exist, we select the case with maximum Ω. Having introduced the similarity matrix function ∂, the similarity function $S_{abstract}$ can be redefined as:

Definition 4: Similarity Function (Logic)

$$S_{Logic}(L(C_1), L(C_2)) =$$
$$(\forall a \in L(C_2) \sum \partial(a)) \,/\, |\{a \mid a \in L(C_2)\}|$$

S_{Logic} measures how similar $L(C_1)$ is to $L(C_2)$. We further generalize S_{Logic} to S such that similarity is measured between the actual concepts. Assume that a concept C whose semantics is modeled in logic with the different semantic aspects represented in separate FOL statements. We define S as:

Definition 5: Similarity Function (Concept)

$$S(C_1, C_2) =$$
$$\sum \omega(i) * S(L_i(C_1), L_i(C_2))$$

where ω(i) is an application dependent weight distribution function that controls the importance of different aspects I in affecting S.

3.2. Ontology Mapping

Now that we have a similarity function S, the process of ontology mapping could be defined as a search procedure for every concept $C_1 \in O_1$, a concept $C_2 \in O_2$ such that $S(C_2,C_1)$ has maximum value. We present such search procedure in this section. W

e observe that If $L(C_1) \models L(C_2)$, but not $L(C_2) \models L(C_2)$, C_2 is a more general concept than C_1. That is, every model of C_1 is also a model of C_2, but not vice versa. Hence the set of world models of C_1 must be a subset of C_2. We employ this observation and generalize it in terms our similarity concepts. C_2 is conceptually more general than C_1 if $S(C_1,C_2) > S(C_2,C_1)$. That is, if C_1 is more similar to C_2 than C_2 is to C_1, then C_2 is intuitively more general. In fact, the observation of C_1 having a subset of world models of C_2 is the special case where $S(C_1,C_2) = 1$ and $S(C_2,C_1) < 1$. We are actually suggesting the following:

More General Rule: $\mathbf{S}(C_1,C_2) > \mathbf{S}(C_2,C_1)$
C_2 is more general if $S(C_1,C_2)=1$
C_2 is approximately more general otherwise

More Specific Rule: $\mathbf{S}(C_1,C_2) < \mathbf{S}(C_2,C_1)$
C_2 is more specific if $S(C_2,C_1)=1$
C_2 is approximately more specific otherwise

The taxonomy of an ontology is structured according to the object-oriented paradigm where generalization relationships edge the parent and child concepts together. By employing our proposed rules, we are guided on the search within the ontology. The strategy is presented as follow:

Assume C_2 belong to the ontology O_2 is the subject concept to be mapped. O_1 is the ontology where the closest concept C_1 is be searched. Let c be some concept in O_1.

Step1:	Start with c = root of O_1, and an empty cache H that is used to store concepts.
Step2:	If c is more general than C_2, push the search downwards to children of c.
Step3:	Select amongst the children of c, a node c' that produces highest S between c' and C_2. Set c = c'. If c' is more general than C_2, repeat Step 2 on c'. Else if c' is more specific than C_2, go to Step 4. Else, go to Step6.
Step4:	Cache c in H. Select amongst the siblings of c, a node c' that produces highest S between c' and c' is more general than C_2. If any such c' exists, repeat Step 2 on c' with c = c'. Else, go to Step 5.
Step5:	Select amongst the concepts in cache H, a node c', that has highest S between c' and C_2. Return c' as the closest match for C_2.
Step6:	If $S(c,C_2) = 1$ i.e. c is an exact match of C_2, return c as the closest match for C_2. Else compute S values between C_2 and every child c' of c. If $\exists c'$ $S(c',C_2) > S(c,C_2)$, perform Step 2 on c. Else cache c in H and go to Step 5.

4. Application: Intrusion Detection

In this section, we validate our mapping methodology by applying it to the network security domain. We identify intrusion detection as an area that requires approximate matching. Section 4.1 outlines the modeling of security events with respect to ***SoS*** and ***SoT***. Section 4.2 details the mapping between security events.

4.1. Semantic Modeling: Intrusion Detection

In the domain of misuse intrusion detection system, the elements that are of mapping interest are security events. We consider two aspects of semantics: behavioral and structural. In the behavioral perspective, security event as a process is described in terms of state transition. That is, the state changes of resources when the security event occurs. In the structural perspective, a security event can be described in terms of its composite components.

Structural Semantic can be viewed as a topology that depicts the participants involved and the network actions that connect the participants together. For example, a relay DoS attack event can be described as having two hosts with Windows OS such that the attacker sends an invalid DNS query to the second host by spoofing the IP address of the first host. The participants are the two hosts with property Windows OS, and the attacker, while the composite actions, send invalid DNS query and spoof IP address, bridge the participants together to form the structural semantics.

Behavioral Semantic can be described by stating the state transition of the affected resources or variables. That is, the behavioral semantic specifies the impact of the security event. The impact can be expressed as the post conditional states of resources or the set of negative behavioral actions that are inflicted in the network. For example, a Trajon horse may inflict the negative actions such as deletion and exposure of data.

We illustrate below the semantic modeling of a security event FINGER search query from Snort with ID - 1:332:

Structural Semantic

$(\bullet\exists x Host(x) \wedge \blacksquare\exists y Host(y) \wedge \exists z(Tcp(z,x,y) \wedge DstPort(z,79) \wedge Payload(z,\text{"search"}))) \rightarrow \Diamond(Attacker(x) \wedge \exists q(Tcp(q,x,y) \vee Udp(q,x,y)))$

Behavioral Semantic

$\blacksquare(\exists y,d(Host(y) \wedge Data(d,y) \wedge Private(d))) \wedge Exposed(d) \rightarrow \Diamond RemoteAccess(y)$

Note that, the semantics are distributed over time. For illustrative purposes, a variant of FOL, temporal FOL, is used to represent the intuitive meanings of the security event. Since our proposed mapping methodology bases on SLD resolution for FOL, separate FOL logical statements are in fact used (to model the semantics for different time interval: past, present and future) during the mapping process.

The notion of time is important in security monitoring. One might think intuitively the future implications of the security event. The event (ID - 1:332) is in fact a reconnaissance attempt that gathers information from the victim host through application exploit. The immediate impact of exposing information is not harmful, but it implies that future attacks of remote access might be launched. Hence the semantics are also distributed in the future. We illustrate in Figure 1 a snapshot of the structure of our developed security ontology with each node modeled in logic.

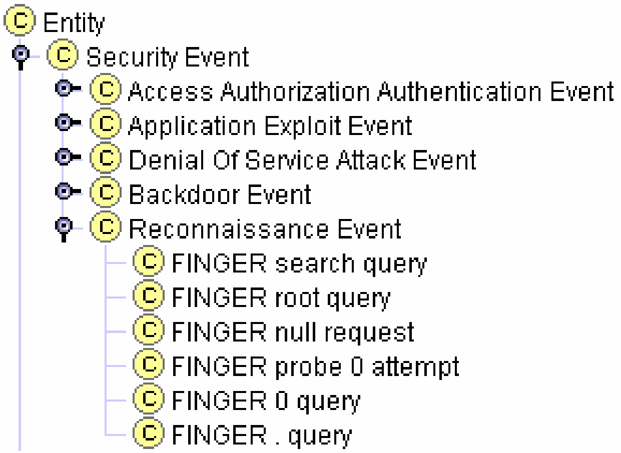

Figure 1 - Snapshot of Security Ontological Structure

4.2. Ontology Mapping: Intrusion Detection

The needs for matching between security events are envisioned through two observations. Firstly, there exist numerous intrusion detection systems. The signatures and security events can be defined proprietarily or originated from different knowledge bases such as snort, whitehats, shoki and dragon sensor. Interoperability issue arises when communication has to be performed between the intrusion detection systems in order to perform cross-network security management. Secondly, number of virus and attack strategies is growing everyday. The growth can be the results of the emergence of new attacks or variants of existing ones. The intrusion detection effort in keeping up with the attack growth is limited. By employing an approximate mapping strategy, we are able to at least provide indication on what might seem to be an attack (new or variant) base on what we already know (approximately map to attacks modeled in the security ontology). That is, the mapping can be regarded as a solution in reducing the deficiency of an intrusion detection system during the period in which security knowledge base is not up to dated.

The semantics distributed in the intervals past, present and future can be regarded as different aspects in which similarity comparisons can be performed. We have S as:

$$\boldsymbol{S(E_1,E_2) = \omega(t)*S(L_{past}E_1,L_{past}E_2) + \omega(t)*S(L_{present}E_1,L_{present}E_2) + \omega(t)*S(L_{future}E_1,L_{future}E_2)}$$

where E_1 and E_2 are the two security events in comparison; and $t \in \{past, present, future\}$.

An intuitive choice of ω(t) can be the following distribution illustrated in Figure 2. The weight distribution models the idea that the importance of similarity fades along with time.

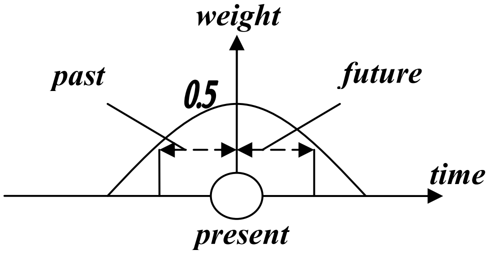

Figure 2 – Weight Distribution

Referring to the sample ontology presented in Figure 1. We consider the modeling of the node reconnaissance attack as follow:

Structural Semantic

$(\bullet\exists x Host(x) \wedge \blacksquare\exists y Host(y) \wedge \exists z(Tcp(z,x,y) \vee Udp(z,x,y)) \rightarrow \Diamond(Attacker(x) \wedge \exists q(Tcp(q,x,y) \vee Udp(q,x,y)))$

Behavioral Semantic

$\blacksquare(\exists y,d(Host(y) \wedge Data(d,y) \wedge Private(d)) \vee \exists x Host(x)) \wedge (Exposed(d) \vee Exposed(x)) \rightarrow \Diamond(Attack(y) \vee Attack(x))$

where there exists an axiom: $RemoteAccess(x) \rightarrow Attack(x)$ with $S(RemoteAccess,Attack) = 1$ and $S(Attack,RemoteAccess) = 0.9$.

We consider the mapping between the event ID-1:332 (E_2) and the reconnaissance attack node (E_1) as an illustrative example. In computation on the structural semantic, we have $S(E_2,E_1) = 1$ and $S(E_1,E_2) \approx 0.123$. We obtain $S(E_2,E_1) = 1$ and $S(E_1, E_2) \approx 0.361$ in behavioural computation. The result suggests that E_1 is strictly a more general concept of E_2.

5. Conclusion

This paper has presented an ontology mapping approach that supports approximate matching on formally represented ontologies. Matching and reasoning in approximation is often required in a dynamic environment. The similarity and logic features of the proposed approach satisfy such requirement. However, the approach is not without its limitations. Due to the use of exponential measurement in computing the resolution quality of a predicate, the density of similarity values is unevenly distributed. The problem limits the ability of the numeric similarity values in reflecting the intuitive notion of similarity.

Intrusion detection has been identified as a domain that exhibits ontological issues, namely, the interoperability problem and the lack of a semantic-based detection. The proposed solution is being experimented in the domain in the hope that the problems can be alleviated.

Extensions to current work may include the study on the use of temporal logic and its inference engine in place of first order logic and SLD resolution, and its application to different domains with ontological issues: other areas in network management, medical and the finance domain.

References

Chandrasekaran B., Josephson J. R., and Benjamins V. R. 1999. *What Are Ontologies, and Why Do We Need Them?.* IEEE Intelligent Systems, vol. 14, no. 1, pp. 20-26.

Kalfoglou Y., and Schorlemmer M. 2002. *Information-Flow based Ontology Mapping.* Confederated International Conferences DOA. CooplS and IDBASE, pp. 1132-1151.

Rodriguez M. A., and Egenhofer M. J. 2003. *Determining Semantic Similarity among Entity Classes from Different Ontologies.* Knowledge and Data Engineering, IEEE Transaction, Issue 2, vol. 15, pp. 442-456.

Doan A., Madhayan J., Domingos P., and Halevy A. 2002. *Learning to Map between Ontologies on the Semantic Web.* Proceedings of the 11th international conference on the World Wide Web, pp. 662-673.

Wiesman F., Roos N., and Vogt P. 2001. *Automatic ontology mapping for agent communication.* MERIT-Infonomics Research Memorandum series, June 2001-023.

Kemmerer R. A., and Vigna G. 2002. *Intrusion Detection: A Brief History and Overview.* Computer, Issue 4, vol. 35, pp. 27-30.

Zhang R., Qian D., Bao C., Wu W., and Guo X. 2001. *Multi-agent Based Intrusion Detection Architecture.* Computer Networks and Mobile Computing, 16-19 Oct, pp. 494-501.

Tversky A. 1997. *Features of Similarity.* Psychological Review 84(4), pp. 327-352.

Yeung D. D., and Tsang E. C. C. 1997. *Comparative Study on Similarity-Based Fuzzy Reasoning Methods.* Temporal Representation and Reasoning, pp. 136-139.

Loia V., Senatore S., and Sessa M. I. 2001. *Similarity-based SLD Resolution and its implementation in an Extended Prolog System.* Fuzzy Systems, vol. 3, pp. 650-653.