Ontology-based Intelligent Network Forensics Investigation

Sherif Saad
Electrical and Computer Engineering
University Of Victoria
Victoria, B.C. V8W 3P6, Canada
shsaad@ece.uvic.ca

Issa Traore
Electrical and Computer Engineering
University Of Victoria
Victoria, B.C. V8W 3P6, Canada
itraore@ece.uvic.ca

Abstract

We propose, in this paper, a new ontology for network forensics analysis. The proposed ontology is the first cyber forensics to integrate both network forensics domain knowledge and problem solving knowledge. As such it can be used as a knowledge-base for developing sophisticated intelligent network forensics systems to support complex chain of reasoning. We use a real life network intrusion scenario to show how our ontology can be integrated and used in intelligent network forensics systems.

1 Introduction

Network forensics is a relatively new discipline that investigates cyber attacks by collecting and analyzing network traffic. Unfortunately, network forensics is currently largely an ad-hoc and manual process. These coupled with the massive amount of data involved make the automation of the network forensics analysis a necessity. Automating the network forensics process raises some key challenges as this requires converting the existing ad-hoc approaches into systematic analysis techniques and converting existing expert knowledge into intelligent analysis and decision-making mechanisms.

Ontologies play a major role in building intelligent knowledge representation and analysis systems. In fact, ontologies are the core of any knowledge representation system, because at minimum they provide the conceptualization of the vocabularies within a specific domain. Of course, without a strong conceptualization we end up with a weak knowledge-base that cannot distinguish between concepts within the domain. In this case, reasoning about the domain will be difficult and perhaps useless. Ontologies allow clarifying the structure of knowledge and concepts in the domain which improve reasoning systems.

Today, most of the proposed security or network forensics ontologies are domain ontologies that are used to provide common vocabularies and make knowledge sharable by encoding domain knowledge. These domain ontologies are less valuable for developing intelligent systems such as expert and reasoning systems. This is because of their lack of problem solving mechanisms that allow an intelligent system to use domain knowledge to solve a specific problem. Method ontology refers to a category of ontologies that contain knowledge about how to achieve a particular task.

In this work, we propose, to our knowledge, the first network forensics ontology analysis, which provides both network forensics domain knowledge and problem solving knowledge. The proposed ontology can be used as a knowledge-base for developing sophisticated intelligent network forensics systems to support complex chain of reasoning.

The rest of the paper is structured as follows. Section 2 summarizes and discusses related work. Section 3 describes the main phases of the proposed ontology development lifecycle. Section 4 presents the basic building blocks of the proposed network forensics ontology. Section 5 illustrates the application of the proposed ontology through the case study of a real-life network intrusion scenario. Section 6 makes some concluding remarks and discusses future research.

2 Related Work

In 2001, Raskin and Nirenburg proposed for the first time the use of ontology in computer and information security [1]. In their work, the focus was primarily on highlighting the advantages of using ontology to represent the domain of information security. Since then, more researches have been carried out on the use of ontology in the field of computer security. Currently ontologies are used in developing security tools such as intrusion detection systems (IDS) and, antivirus and other malware detection systems. However, to our knowledge, few network forensics ontologies have been proposed in the literature and none of these ontologies provide specific mechanisms for building intelligent network forensics system.

Brinson and colleagues [2] define an ontology that describes characteristics and concepts pertaining to the cyber forensics profession, and what the people involved in cyber-forensics environment need to do in terms of education, certification and specialization. Although the proposed ontology sheds some light on the different job functions underlying the cyber forensics profession and may assist in curriculum development, it does little to advance cyber forensics methodologies and techniques.

Park and colleagues [3] propose a cyber forensics ontology for criminal investigation in cyber space. The proposed ontology emphasizes on the definitions of the different types of cyber crimes such as cyber terrorism, general cyber crimes, hacking, fraud, etc. In addition to defining cyber crimes, the ontology focuses on defining the types of evidences that can be collected to prove criminal intention for each type of cyber crime. The authors also discuss the possibility of using ontology in mining cyber crimes data, but the proposed ontology fails to provide the required knowledge to accomplish such objective.

Hoss and Carver suggest the use of ontology in network forensics analysis, although no specific ontology is proposed in their paper [4]. Only an abstract structure for the required ontologies and their characteristics are provided. Specifically, a framework identifying five specialized ontologies is proposed. These include Crime Ontology, Forensics Device Ontology, Legal Ontology, Digital Device Ontology and Ontology for Forensics Information Integration.

3 Ontology Development Lifecycle

To build our network forensics ontology we use a hyper approach that combines different key features from several ontology development approaches, such as the METHONTOLOGY approach by Lopez and Perez [5], the Uschold and King [6] approach and the work of Gruninger and Fox [7]. We discuss in this section the key steps involved in our network forensics ontology development.

3.1 Specification

According to our specification, our ontology is a heavyweight ontology in terms of formality and granularity and its domain of interest is network forensics. We choose description logic as our ontology knowledge representation paradigm allowing us to formalize the knowledge in our ontology. Our network forensics ontology is a method ontology that represents knowledge about the network forensics domain including concepts and their relations and attributes and facts about these concepts. In addition, as method ontology, it contains knowledge about how to use the domain knowledge for problem solving or complex chain of reasoning. In other words our ontology contains knowledge that represents the network forensics domain and the network forensics investigation process.

3.2 Conceptualization

After setting our ontology specification, we move to the conceptualization stage. In this stage we identify the basic concepts or classes that are domain specific or network forensics specific. We identify three main types of knowledge to be represented in our network forensics ontology. These three types of knowledge are:

- Problem solving goals.
- Problem solving knowledge for network forensics process
- Factual knowledge about network forensics domain

The problem solving goals are a set of network forensics process goals. These goals are expressed in a set of informal competency questions. For instance, a network forensics competency question could be something like: Given a set of intrusion alerts, what are the attacks types that appear in this set? In our initial prototype we defined 71 competency questions a sample of which are listed here:

- What vulnerabilities exist in the target system?
- What are the critical attack assets?
- Given a set of privileges, what is the attacker capable of?
- Given a set of attack impacts, what are the attacks that result in these impacts?
- Given a set of assets, which assets are vulnerable assets?

As shown in Figure 1, the competency questions are structured in hierarchical tree structure or taxonomic structure, such that the answer of any parent competency question requires the answers from all its children competency questions.

The competency questions are used during the design of the ontology to help us in identifying the problem solving goals for our ontology. The competency questions and their answers are useful to acquire knowledge about the scope of the problem submitted to the ontology, as well as the necessary constraints,

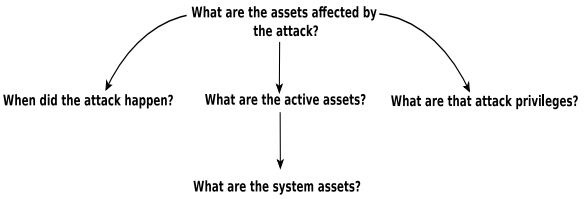

Figure 1: Example of Competency Questions Tree Structure

assumptions, input and output data. In addition, we use these competency questions to evaluate our ontology, by ensuring that the implemented ontology is able to answer all the competency questions identified during the ontology design stage.

Another important knowledge that must be represented by our ontology is the problem solving knowledge for network forensics process. This knowledge is represented by the set of constraints, roles, and property functions that process the knowledge to achieve the goals of the forensics analysis. As an example let's go back to our first competency question: "Given a set of intrusion alerts, what are the attacks types that appear in this set? ". According to this question we have a diagnostic goal with respect to the problem which is in this case the attacks types (classes). The ontology must contain knowledge describing the diagnostic goal. This knowledge is represented in the ontology as a set of constraints and functions. The problem solving knowledge explains how the domain knowledge can be used to reach the problem solving goals.

The third type of knowledge that we represent in the network forensics ontology is factual knowledge or domain knowledge. Such knowledge include classes, sub-classes, properties, and relations between classes. We need to represent all the abstract and core concepts that exist in the network forensics domain. Classes that share the core concepts, such as the class ***computer-virus*** and the class ***computer-worm*** , are connected by taxonomic relations. Classes from different taxonomies are also connected, for instance, the class ***web-server*** and the class ***denial-of-services*** both are connected by non-taxonomic relation.

The next step after identifying the goals of our network forensics ontology is to determine the concepts and the classes that are specific to the network forensics domain. There are three main methods in ontology engineering to construct domain classes, namely Top-Down, Middle-Out, and Bottom-Up approaches [6]. We choose the middle-out method because experience with ontology design shows that middle-out is the most effective method. The middle-out method is very simple to apply. It begins by the identification of the most important concepts or classes that exist in the domain, followed by the more abstract and more specific classes. For instance, the class ***alert*** is a middle class in the ***evidence*** class taxonomy, while the class ***evidence*** is more abstract and the class ***intrusion alert*** is more specific. To cover a wide range of classes in the network forensics domain we studied several network intrusion taxonomies and ontologies [8–11].

Now we have to build the classes taxonomies to group our ontology classes into categories. At this point we use the class description and properties to group similar classes in a single taxonomy. Note that classes in a single taxonomy are connected with taxonomic relations of type **is-A**, **superClass-Of**, and **subClass-Of**. At the end of this step we obtain 12 independent taxonomies, each containing a set of network forensics related classes. Then we build our ontological-relations which are the relations that link classes from different taxonomies together. These ontological-relations are divided into general relations and specific relations. Examples of general ontological-relations are **part-Of**, **has-A**, **use-A**, etc. General ontological-relations are strictly 2-ary relations (binary relations). On the other hand specific ontological relations are N-ary relations (N-ary relation maps between a subject and two or more subject/values).

Binary relation is the most common type of relation in ontology. However, heavyweight ontologies, method and task ontologies that are used to represent complex knowledge for sophisticated processes require the use of N-ary relations. For instance, the representation of the following knowledge: "Apache web server is target of denial of service attack with higher severity" requires the use of N-ary relations and cannot be represented by two binary relations because the relations in this case are all interconnected. After obtaining the classes and the relations between them, the final step in the conceptualization is to provide, for each class and relation in the ontology, a detailed description informally using natural language.

3.3 Formalization and Implementation

We start formalizing the network forensics ontology using the appropriate ontology language. The final step in our ontology lifecycle is the actual implementation of the ontology and feeding the ontology with individuals (classes instances). There are different description logic languages that can be used to implement our network forensics ontology. As mentioned before, we use description logic as our ontology knowledge representation paradigm. We use in this work the Web Ontology Language (OWL) as our ontology encoding language. OWL is a description logic based

language to develop ontologies and represent knowledge in semantic web.

4 Proposed Ontology

The initial prototype of our ontology contains 111 classes that represent the network forensics domain. Here, we represent the top-level classes in our ontology and the relations between them. These classes represent the main concepts in the network forensics domain. The top-level classes in our network forensics ontology are listed below:

Attack	Malicious	Evidence	Impact
Attacker	Objective	Motive	Asset
Vulnerability	System	Privilege	Location

There are two main types of relations in our network forensics ontology, namely, taxonomic relations and ontological relations. The taxonomic relations in our ontology are listed in Table 1. The taxonomic relations are binary relations (2-ary relations) used to categorize classes in the network forensics domain in a taxonomic structure.

From Table 1 we can see that we use in our network forensics ontology four taxonomic relations. All of these relations are transitive, reflexive, and antisymmetric. The first relation **is-A** is used to identify the type of a property. For instance, the class ***attack*** has the property "tool" , where "tool is-A Malicious". The **superclass-Of** and **subclass-Of** relations are used to express inheritance. For instance, the class "location" is the superclass-Of "remote-location" and "local-location" and so both "remote-location" and "local-location" are subclass-Of "location". Finally, the **instance-Of** relation is used to link an individual to specific class. For instance, the "code-red" is instance-Of "computer-worm".

The second type of relations in our ontology, as mentioned before, is the ontological relations. These relations are either binary or N-ary relations (ternary or more). As indicated before, there are general ontological relations and specific ontological relations. Specific ontological relations represent more complex relations than the general ones. Our general ontological relations are listed in Table 2.

In the initial prototype of our ontology, four specific ontological relations are defined. These relations are N-ary relations represented by two major patterns. The first pattern is by creating a new class with ***N*** properties to represent the N-ary relation. The second pattern is by using lists for arguments to identify a N-ary relation that represents a sequence of arguments.

Relation-Name	Subject-Class	Object-Class
Executes	Attacker	Attack
Exploits	Attacker	Vulnerability
Uses	Attacker	Malicious
Located-At	Attacker	Location
Has-A	Attacker	Motive
Leaves	Attacker	Evidence
Uses	Attacker	Malicious
Target	Attacker	Asset
Gains	Attacker	Privilege
Compromises	Attacker	System
Requires	Attack	Vulnerability
Elevates	Attack	Privilege
Proved-By	Attack	Evidence
Causes	Attack	Impact
Triggered-By	Attack	Malicious
Affects	Attack	Asset
Traced-To	Attack	Malicious
Has-A	Attack	Objective
Exist-In	Vulnerability	Asset
Requires	Attack	Privileges
Extracted-From	Evidence	Asset

Table 2: General Ontological Relations

The first N-ary relation in our ontology is ***attack diagnosis*** relation. This N-ary relation describes the relation between ***asset, attack, attack confidence, and attack severity***. We create a new class in our ontology to represent this relation as depicted in Figure 2.

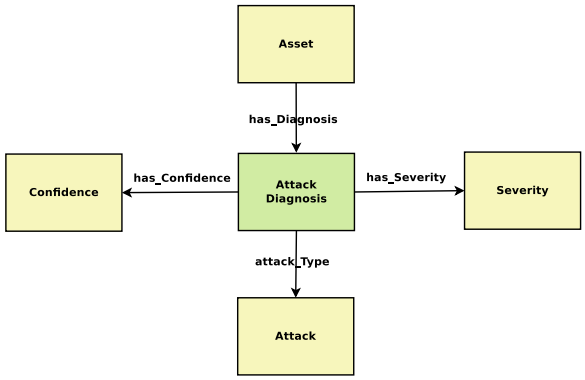

Figure 2: Attack Diagnosis Relation

The second N-ary relation is ***attack attribution*** that shows the attribution of an attack incident. We use the first N-ary relation design pattern to represent the attack attribution. As depicted in Figure 3, the attack attribution links the following individuals: ***attacker, location, malicious, and attack***.

Relation-Name	Transitive	Reflexive	anti-symmetric
is-A	√	√	√
superclass-Of	√	√	√
subclass-Of	√	√	√
instance-Of	√	√	√

Table 1: Taxonomic Relations and their properties

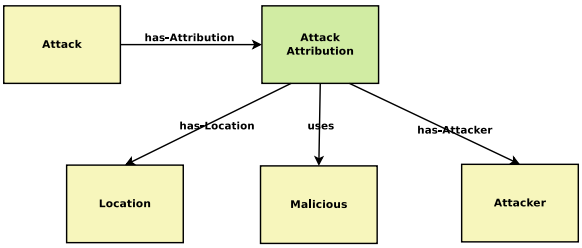

Figure 3: Attack Attribution Relation

The third N-ary relation in our ontology is the ***fulfill*** relation between an ***impact*** and a list of ***evidences***. The fulfill relation uses the second type of N-ary relation design pattern. The fulfill relation links a single impact to one or more evidences fulfilled by that impact as depicted in Figure 4.

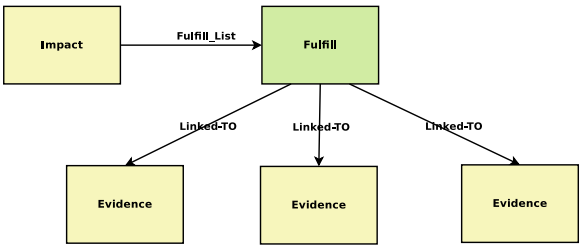

Figure 4: Fulfill Relation

The fourth and last N-ary relation in our ontology is the ***attack sequence relation***, which is used to describe the attack scenario, expressed as an ordered list of ***attacks*** executed by the ***attacker***. The attack sequence relation is described in Figure 5.

5 Case Study

In this section we illustrate the use of our ontology in network forensics by presenting a real network intrusion case that was conducted in 2004 against a honeynet hosted in our lab. The honeynet was accessible over the Internet and deployed some hosts involving several well-known vulnerabilities. The attack we are using here as a case study is one of several attacks captured by our honeynet. It is a multistage attack conducted over three days where the intruder targeted

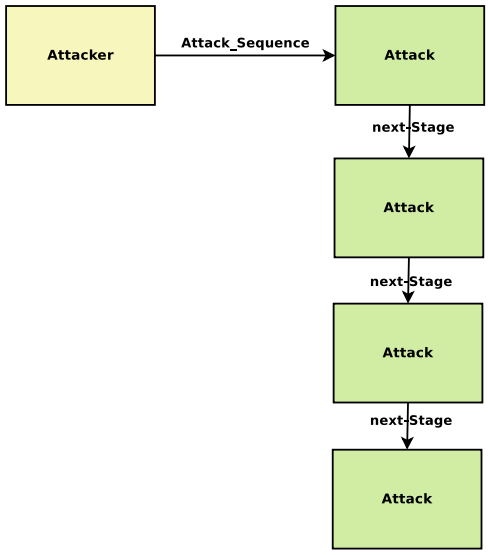

Figure 5: Attack Sequence Relation

a vulnerable FTP server installed in our honeynet to break into our network and take control of one of the machines.

As depicted in Figure 6 from host ***211.42.48.148***, the intruder started by probing our network and then found a vulnerable FTP server with IP address ***192.168.100.102***. He ran a buffer overflow exploit on the FTP server gaining root privileges on this host. Next, he used the compromised machine to execute a DOS attack against a machine with IP address ***65.113.119.148*** external to our network.

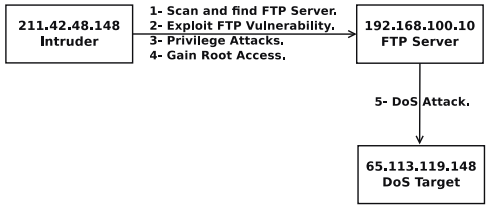

Figure 6: Multistage FTP Attack Sequence

The network forensics data collected for the above intrusion case include network binary file in pcap format, snort alerts file, and the target system configura-

tion data. The IDS alerts file contains 90 alerts messages, over six unique intrusion signatures, occurred during a period of 3 hours. Four of the six signatures are standard snort signatures with the following signature IDs: 553, 1622, 1672 and 1378.

We illustrate in the remaining of this section how an intelligent network forensics analysis system can use our proposed ontology to reconstruct automatically the above attack scenario. It must be noted that an ontology in itself is only a sophisticated knowledge representation approach. This means that we still need a reasoning system or an inference engine that can make use of the knowledge encoded in the ontology. In this case study, we use deductive reasoning, although, other forms of inference models such as inductive and abductive reasoning may be used. Likewise, there are many cases in forensics analysis where inductive and abductive reasoning are required.

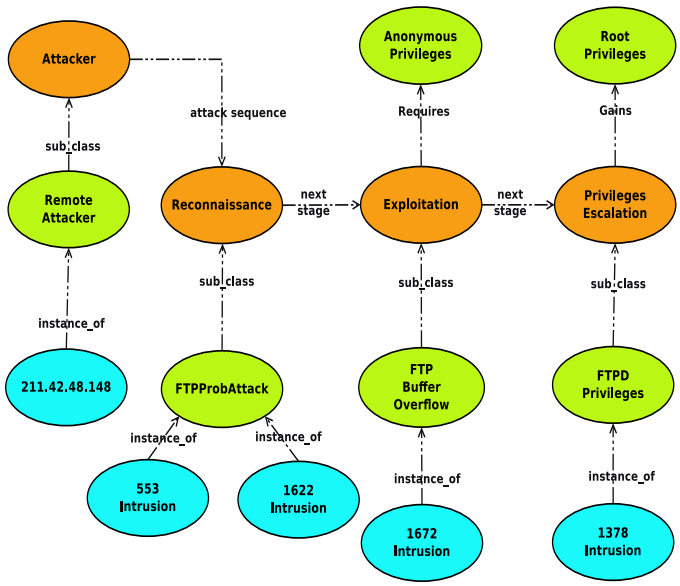

Figure 7: Ontological Representation for the Multistage FTP attack

Figure 7 depicts the ontological representation of the multistage FTP attack. Using the instance and class inference capabilities, it can be inferred that the attack signatures 553 and 1622 are instance-of FTP probing and reconnaissance attack. Such information can be obtained via the ***instance-Of*** relation between the attack instance and the ***FTPProbAttack*** class which is linked to ***Reconnaissance*** class via ***subclass-Of*** relation. The ***Reconnaissance*** class is a subclass of the ***Attack*** class. By applying the same approach for the remaining signatures, namely 1672 and 1378, it can be inferred that 1672 is a buffer overflow attack and that 1378 is a privilege escalation attack.

As we can see, using the taxonomic relations ***instance-Of*** and ***subclass-Of*** useful knowledge can be inferred about the attacks types. At this point we know that there are three main attack classes, namely, ***Reconnaissance***, ***Exploitation*** and ***Privileges Escalation***.

Knowing the attacks types, knowledge about the privileges required to execute these attacks can now be inferred. For example, the attack instance with signature 553 is an ***instance-Of FTPProbAttack*** and the ***FTPProbAttack Requires*** anonymous FTP privileges. Therefore, (it can be concluded that) this attack instance requires anonymous FTP privileges.

At this stage the ontology can also be used to infer the impact of the attack. For example, the attack instance with signature 1378 is an ***instance-Of FTPD-Privileges*** attack class. The ***FTPD-Privileges*** attack class has the impact of root privileges. Therefore, (it can be concluded that) this attack instance has the impact of root privileges, and furthermore it can be inferred that the attacker behind this attack instance had gained root privileges.

Finally, using the N-ary relation ***attack sequence*** knowledge can be inferred about reconstructing the attack scenario. The ***attack sequence*** relation links the ***Reconnaissance***, ***Exploitation*** and ***Privilege Escalation*** classes. It follows that the subclasses ***FTPProbAttack***, ***FTPBufferOverflow*** and the ***FTPD-Priviliages*** forms an attack sequence. Given this information it can be inferred that the attack instances with the signatures IDs 553, 1622, 1672 and 1378 are part of a multistage attack.

6 Conclusions

Network forensics is an expensive process that is usually time consuming and requires a team of forensics investigators. The current available network forensics systems are limited to query engines capabilities without advanced investigation techniques. Reducing the cost of network forensics by automating the forensics investigation process is a necessity.

In this paper we introduce a novel ontology-based network forensics knowledge representation approach which represents an important toward increased automation of the forensics investigation process. By combining both network forensics domain knowledge and problem solving knowledge, the proposed ontology lays down the ground for developing network forensics systems than can perform complex reasoning which is essential when investigating malicious activities.

References

[1] 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.

[2] A. Brinson, A. Robinson, and M. Rogers, "A cyber forensics ontology: Creating a new approach to studying cyber forensics," *Digital Investigation*, vol. 3, no. Supplement 1, pp. 37 – 43, 2006. The Proceedings of the 6th Annual Digital Forensic Research Workshop (DFRWS '06).

[3] H. Park, S. Cho, and H.-C. Kwon, "Cyber forensics ontology for cyber criminal investigation," in *e-Forensics*, pp. 160–165, 2009.

[4] A. Hoss and D. Carver, "Weaving ontologies to support digital forensic analysis," in *Intelligence and Security Informatics, 2009. ISI '09. IEEE International Conference on*, pp. 203–205, June 2009.

[5] M. F. Lopez, A. G. Perez, and N. Juristo, "Methontology: from ontological art towards ontological engineering," in *Proceedings of the AAAI97 Spring Symposium*, (Stanford, USA), pp. 33–40, March 1997.

[6] M. Uschold and M. Grüninger, "Ontologies: principles, methods, and applications," *Knowledge Engineering Review*, vol. 11, no. 2, pp. 93–155, 1996.

[7] M. Grüninger and M. S. Fox, "Methodology for the design and evaluation of ontologies," in *Proceedings of Workshop on Basic Ontological Issues in Knowledge Sharing held in conjunction with IJCAI-95*, 1995.

[8] 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.

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

[10] 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.

[11] 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.