Method Ontology for Intelligent Network Forensics Analysis

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

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

***Abstract*—Network forensics is an after the fact process to investigate malicious activities conducted over computer networks by gathering useful intelligence. Recently, several machine learning techniques have been proposed to automate and develop intelligent network forensics systems. An intelligent network forensics system that reconstructs intrusion scenarios and makes attack attributions requires knowledge about intrusions signatures, evidences, impacts, and objectives. In addition, problem solving knowledge that describes how the system can use domain knowledge to analyze malicious activities is essential for the design of intelligent network forensics systems. In this paper we adapt recent researches in semantic-web, information architecture, and ontology engineering to design a method ontology for network forensics analysis. The proposed ontology represents both network forensics domain knowledge and problem solving knowledge. 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.**

I. Introduction

Network forensics analysis is the process of extracting intrusion patterns and investigating malicious activities conducted over the network. Although there are many tools that can be used to collect and preprocess raw network forensics data, the important task of verifying the existence of malicious activities, describing their details, and identifying their sources is currently largely an ad-hoc and manual process. Considering the massive amount of data to analyze and the different data sources to cover, we can easily understand that network forensics analysis is a complex and time consuming task.

In this context, the automation of the network forensics analysis becomes 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.

Existing network forensics analysis systems usually focus on the statistical features characterizing the malicious activities. However, in real-life practice, forensics analysts focus primarily on the meaning or semantics of the malicious activities and pay little attention to statistical characteristics. For that reason intelligent network forensics systems should extract and analyze as well the semantics of the malicious activities. The use of semantics analysis in intelligent network forensics investigation requires the existence of an ontology that provides a formal representation of the set of concepts characterizing forensics entities and processes and the relations between these concepts.

Ontologies play a major role in building knowledge representation systems and Artificial Intelligence (AI) systems in general. 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 a method ontology for network forensics analysis, which to our knowledge is the first of the kind. The proposed ontology represents both network forensics domain knowledge and problem solving knowledge. The current version of our network forensics ontology contains knowledge about more than 11,000 malicious activities and 30 network forensics problem solving methods.

Ontologies are commonly categorized into lightweight and heavyweight ontologies. This categorization is based on the formality and the granularity of the knowledge represented by the ontology. Intelligent systems such as expert and reasoning systems require heavyweight ontologies. Our proposed network forensics ontology is a heavyweight ontology.

The rest of the paper is structured as follows. Section II summarizes and discusses related work. Section III describes the main phases of the proposed ontology development lifecycle. Section IV presents the basic building blocks of the proposed network forensics ontology. Section V presents and

978-1-4244-7550-6/10/$26.00 ©2010 IEEE

discusses the reasoning capabilities of the proposed ontology. Section VI illustrates the application of the proposed ontology through the case study of a real-life network intrusion scenario. Section VII makes some concluding remarks and discusses future research.

II. Related Work

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 [1] 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 [2] 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 [3]. 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.

While limited work has been done on the use of ontology in network forensics, in recent years several ontologies have been proposed for intrusion detection systems. We review in the rest of this section sample of these works, since intrusion detection is the most related area to network forensics.

Undercoffer and colleagues [4] propose the first ontology for intrusion detection system. The authors introduce a target centric ontology for intrusion detection by analyzing intrusion alerts. The proposed ontology is built by evaluating 4000 vulnerabilities and the required attack strategies to exploit these vulnerabilities. Principal component analysis and fuzzy clustering are used for feature extraction and data abstraction, while Mahalanobis distance is used for dissimilarity measurement. According to the authors, the proposed ontology can also be used to detect distributed attacks.

Hung and Liu [5], [6] develop an anomaly intrusion detection system using a network attack ontology based on the ontology developed by Undercoffer et al. [4]. The proposed anomaly IDS consists of three stages, namely, specification, mapping, and generation. The proposed approach is compared experimentally against traditional anomaly intrusion detection approaches based on algorithms such as K-means, Nearest Cluster Algorithm (NEA), and C4.5. It is reported that the ontology-based approach achieves better performance results compared to the traditional ones.

Abdoli and Kahani develop ontology for distributed intrusion detection system [7] using Protege editor. They use ontology to provide a mean of extracting semantic relations between attacks and intrusions alerts generated by different IDSs. The proposed intrusion detection system uses the ontology approach to reduce the false alarm rate in the network.

Isaza and colleagues use ontology to develop an intelligent intrusion detection system [8], [9]. The authors use artificial neural network and multi-agent system (MAS) to build an intelligent distributed intrusion detection system. The ontology proposed in this work captures the knowledge related to intrusion signatures, reaction, and prevention rules. According to the authors, the detection accuracy of the proposed ontology-based IDS is superior to that obtained for traditional signature-based IDS.

III. Ontology Development Lifecycle

There are many methods to build an ontology that represents a domain of knowledge and supports reasoning over such knowledge. 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 which is based on the work of Lopez and Perez [10]. Most of the ontologies developed today are based on this approach. METHONTOLOGY divides the ontology developing process into eleven main tasks. In addition, the process itself is based on evolving prototypes. While we use METHONTOLOGY as our main ontology development approach we also select some key feature from other approaches mainly the Uschold and King [11] approach and the work of Gruninger and Fox [12]. We discuss in this section the key steps involved in our network forensics ontology development.

A. Specification

Based on METHONTOLOGY approach we begin by creating a specification for our network forensics ontology. 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 represent the network forensics domain and the network forensics investigation process.

B. 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.

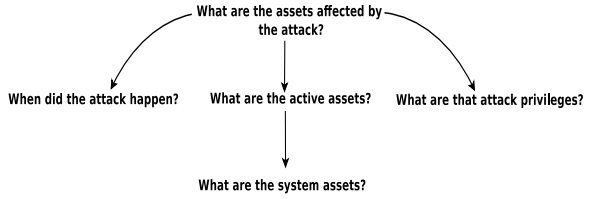

Fig. 1. Example of Competency Questions Tree Structure

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, 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 a 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 [11]. 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 [13], [14], [4], [15].

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 (an 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.

C. 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.

IV. 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 I. The taxonomic relations are binary relations (2-ary relations) used to categorize classes in the network forensics domain in a taxonomic structure.

From Table I we can see that we use in our network forensics ontology four taxonomic relations. All of these relations are transitive, reflexive, and anti-symmetric. 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 a 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 II.

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 II
General Ontological Relations

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.

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.

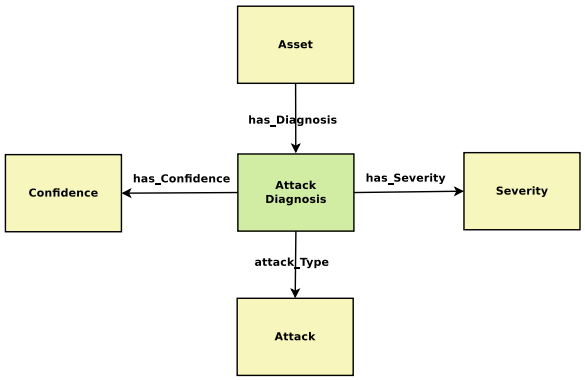

Fig. 2. Attack Diagnosis Relation

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

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

TABLE I
TAXONOMIC RELATIONS AND THEIR PROPERTIES

depicted in Figure 3, the attack attribution links the following individuals: ***attacker, location, malicious, and attack***.

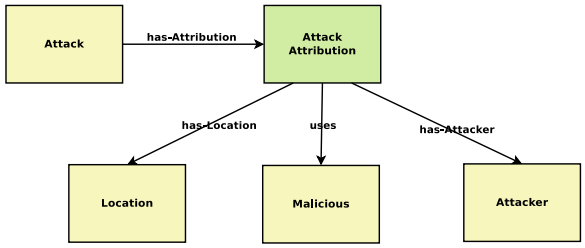

Fig. 3. Attack Attribution Relation

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.

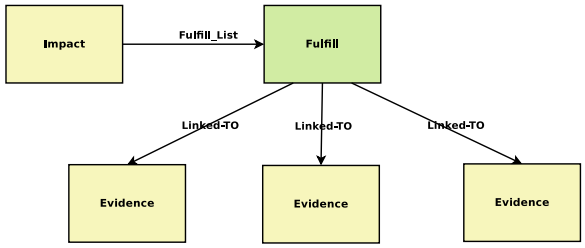

Fig. 4. Fulfill Relation

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.

V. ONTOLOGY REASONING

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. Reasoning over ontology is the process of finding implicit facts given explicitly stated facts in the ontology. Ontology reasoning is useful for generalization, prediction, diagnosis, and drawing conclusions from facts. In general there are three main forms of reasoning that can be implemented over ontologies. These forms of reasoning are deductive reasoning, inductive reasoning and abductive reasoning.

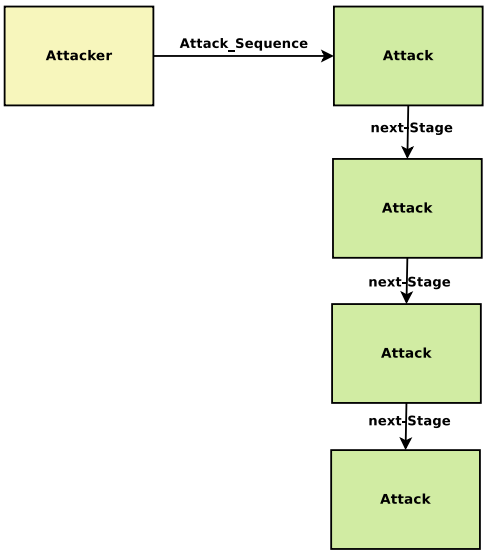

Fig. 5. Attack Sequence Relation

Deductive reasoning is the most common reasoning model over ontologies. Deductive reasoning is used to draw a conclusion by narrowing down general domain knowledge encoded in the ontology. For instance, let us consider the following premises: ***P1: all probing attacks are Reconnaissance attacks*** and ***P2: IPsweep is a probing attack***. Given the above premises, using deductive reasoning, we can conclude that: ***IPsweep is a reconnaissance attack***.

Inductive reasoning is a down-top reasoning approach that is based on observing instances, recognizing patterns and making generalizations based on those patterns. An important difference between inductive reasoning and deductive reasoning is that in inductive reasoning the truth of the premises does not guarantee the truth of the conclusion. For example, let us assume that we have n number of intrusions and all of these intrusions are ***instance-of*** Privilege Escalation attack. In addition, each of these instances ***Affects*** a FTP server and has the impact of root privileges. Given these facts and using inductive reasoning we can conclude that all privilege escalation attacks that target FTP servers will allow the intruder to gain root privileges.

Abductive reasoning aims at finding the best explanation for an observed case or fact. Specifically, abduction reasoning allows the precondition ***a*** to be inferred from the consequence ***b***. For instance, let us consider three intrusion instances ***A, B, and C***, respectively, executed in sequence by the same intruder. Let us also assume that from the ontology we know that the

intrusion instance ***A*** is an FTP probing attack and that the intrusion instance ***C*** is based on an FTP exploit that requires the intruder to have root privileges. Given these facts we can infer that the intrusion instance ***B*** is a privilege escalation attack that has the impact of root privileges.

As mentioned above, inductive reasoning and abductive reasoning over ontologies are less common compared to deductive reasoning. Likewise most of the available inference engines for ontologies focus on deductive reasoning. We found, however, that for some forensics tasks both abductive and inductive reasoning are very useful. For instance, during a forensics investigation we believe that abduction reasoning is very useful as a diagnosis approach to explain after the fact situation. In addition, inductive reasoning is useful in drawing good generalized conclusions from analyzing specific forensic cases. Our proposed ontology supports all three forms of reasoning as we illustrate in the case study presented in the next section.

VI. Case Study

In this section we illustrate the use of our ontology in network forensics by presenting a real network intrusion case occurred 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 a vulnerable FTP server installed in our honeynet to break into our network and take control of one of the machines.

A. Case Study Overview

As depicted in Figures 6 and 7, 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.

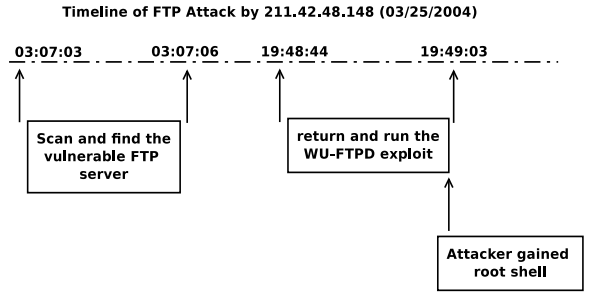

Fig. 6. Time-Analysis for a Multistage FTP Attack

The network forensics data collected for the above intrusion case include the network binary file in pcap format, the snort alerts file, and the target system configuration data. The snort alerts file contains 90 alerts messages over six unique intrusion signatures that 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.

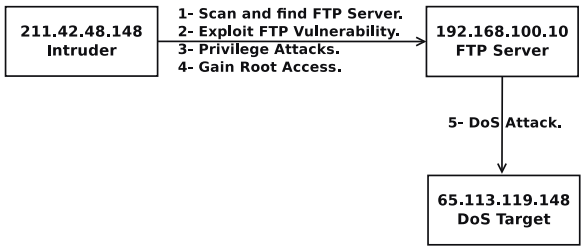

Fig. 7. Multistage FTP Attack Sequence

B. Attack Intelligence Extraction

We illustrate in this subsection how an intelligent network forensics analysis system can use our proposed ontology and the reasoning mechanisms discussed earlier to reconstruct automatically the above attack scenario.

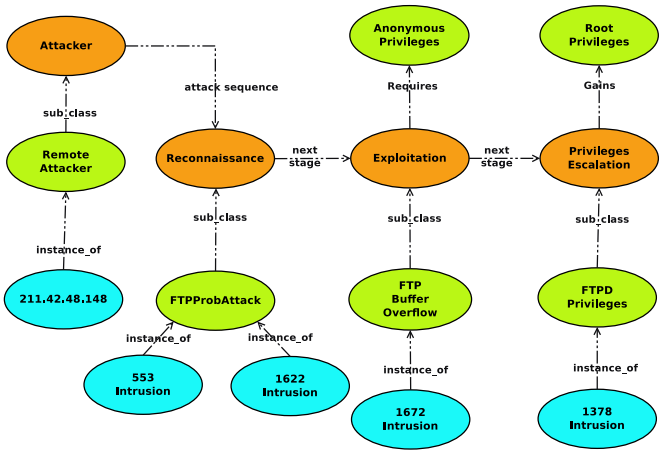

Fig. 8. Ontological Representation for the Multistage FTP attack

Figure 8 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 ex-

ample, 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.

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 sub-classes ***FTPProbAttack***, ***FTPBufferOverflow*** and the ***FTPD-Priviliages*** form 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.

C. Attack Attribution and Evidence Extraction

In the previous section, we illustrate the use of deductive reasoning. In this section, we will illustrate inductive reasoning and deductive reasoning over attack attribution and evidence extraction cases, respectively.

We use the term ”attack attribution” to refer to the process of attributing attack events by identifying their sources, methods, severity and underlying evidences. In our network forensics ontology, there are two N-ary relations that are useful for attributing network intrusions. These two N-ary relations are the ***attack attribution*** and the ***fulfill*** relations.

Figure 9 depicts the representation of the attack attribution of the FTP privilege escalation attack within the ontology. According to our ontology the intrusion instance 1378 is an instance of ***FTPD Privilege Attack*** and attributed by the ***FTPD_ Privi Attribution*** class. The attribution provides information about the tools used in the attack, the origin of the attack and the individual behind the attack. In addition to the ***attack attribution relation***, we have the ***Affects*** and the ***Requires*** relations that show the asset affected by the attack and the vulnerability exploited during the attack, respectively. Given the knowledge encoded in the ontology in Figures 8 and 9, using inductive reasoning we can conclude the following.

- Other versions of WU-FTPD server software contain software bug and may allow an attacker to execute arbitrary code.
- Networks running WU-FTPD server software have a severe vulnerability that can allow a remote intruder to gain root access.

Although the above facts seem trivial for a human expert, reaching such conclusions is all but straightforward for a machine.

As stated before abductive reasoning is very useful for inferring preconditions of specific consequences. The intrusion instances 553 and 1662 are ***proved-by*** the ***ServiceProb Alert***

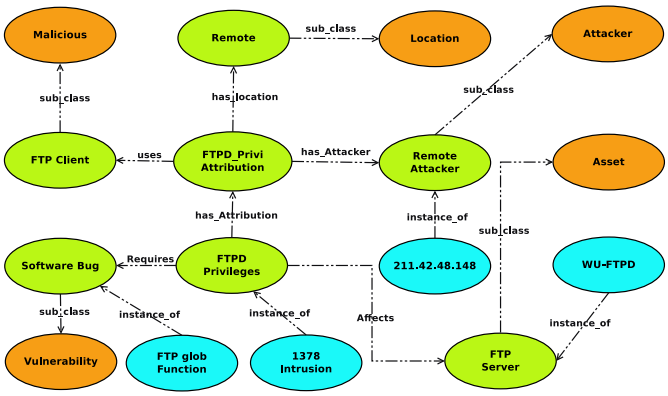

Fig. 9. Ontological Representation for the FTP Prob Attack Attribution

and the ***Service Exposed*** evidences as depicted in Figure 10. This can be inferred by deduction. But the interesting point here is the fulfill relation that shows that the evidences of the ***FTPProbAttack*** are fulfilled by the impact ***Ports Exposed***. The impact ***Ports Exposed*** is the ***causes*** of a ***Port Scan Attack***. Using this knowledge and abductive reasoning the framework can infer several explicit and interesting pieces of knowledge such as the following:

- The network was attacked by port scanning attack.
- The network is vulnerable to port scanning attack.
- The FTP server is exposed by port scanning attack.
- The NIDS generates false negative for port scanning attack.

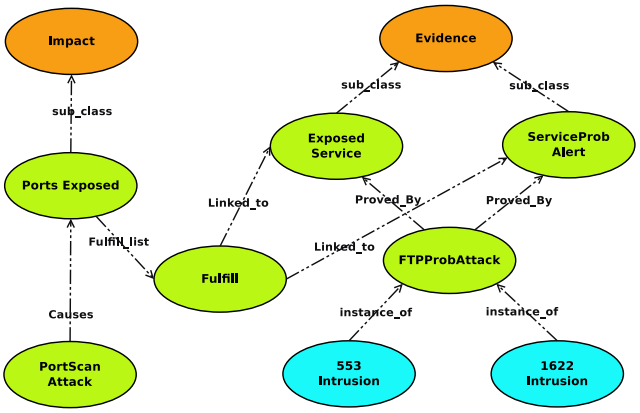

Fig. 10. Ontological Representation of the Impact Evidence Fullfill relation for the FTP Prob Attack

VII. Conclusions

In this paper we proposed a novel ontology-based network forensics knowledge representation approach. This ontology provides a formal description of the concepts characterizing the network forensics domain and describes the relationships between these concepts. In addition, the ontology is used to provide a formal description for network forensics analysis methods. Combining both network forensics domain knowledge and problem solving knowledge in a method ontology

will enable the development of network forensics systems than can perform complex reasoning which is essential when investigating malicious activities.

The benefits of building and maintaining network forensics ontology to represent network forensics Knowledge are unquestionable. However, it takes a lot of time and effort to construct and maintain it. In addition, the huge number of concepts involved in the network forensics domain and their complex relations complicate the task of constructing and maintaining the ontology. Therefore, in our future work we will investigate the design of automated ontology construction methods. We will also investigate the design of application ontology which is a more specialized form of method ontology. In this case the ontology will represent knowledge for specific network forensics problem, such as botnets forensics, DDoS forensics or web-services attacks forensics. We believe that such approach can reduce the complexity of constructing and maintaining network forensics ontologies.

REFERENCES

[1] 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).

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

[3] 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.

[4] 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.

[5] S.-S. Hung and D. S.-M. Liu, "An ontology-based network intrusion detection system: A user-oriented approach," in *ISI*, pp. 722–723, 2006.

[6] S.-S. Hung and D. Shing-Min Liu, "A user-oriented ontology-based approach for network intrusion detection," *Comput. Stand. Interfaces*, vol. 30, no. 1-2, pp. 78–88, 2008.

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

[8] M. L. Gustavo Isaza, Andrs Castillo and L. Castillo, "Towards ontology-based intelligent model for intrusion detection and prevention," in *Computational Intelligence in Security for Information Systems*.

[9] N. D. D. Gustavo Isaza, Andrs 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)*.

[10] 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.

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

[12] 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.

[13] 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.

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

[15] 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.