An OWL-based Security Incident Ontology

Luciana Andréia Fondazzi Martimiano, Edson dos Santos Moreira

Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo (ICMC-USP)
Avenida Trabalhador Sãocarlense, 400, Centro. CEP 13560-970
São Carlos - São Paulo – Brazil
{luciana, edson}@icmc.usp.br

1. Introduction

As the Internet evolves, more people and computational systems are connected and more information needs to be protected. In this context, the concerning about information security has increased inside the organizations, mainly because information is their most valuable assets. However, nowadays, the computational security has generated a great volume of information, making its manipulation and management even harder. Due to this great volume of information generated by different sources, such as systems logs, firewall logs and vulnerabilities alerts, the security administrators face out a difficult problem, which is to efficiently generate knowledge about security information problems to make decisions and to solve incidents.

Different research institutes have making efforts to classify and catalogue security data and information. The CVE Project (Common Vulnerabilities and Exposures), developed in the Mitre Institute[1], aims to standardize the names for all publicly known vulnerabilities and security exposures, making easier to identify the same vulnerability in different security tools [Mann and Christey, 1999]. The CERT®/CC (Computer Emergency Response Team/Coordination Center)[2], created in the Carnegie Mellon University, is an information center that stores information about computational security, including vulnerabilities alerts, security incidents, virus alerts and others.

Although the great importance of the CVE Project and the CERT®/CC initiatives, they do not implement semantic into information stored and publish. Not only the semantic makes it possible to structure the information but it also helps to process the meaning of the information. Without this semantic meaning, the software agent or the system administrator is unable to make important and implicit correlations among security incidents.

To ease and make it possible to correlate different security incidents from different sources, and also to ease the knowledge and information management about security incidents, we propose a security incident ontology, defining a unique vocabulary of terms and relations related to this domain (Martimiano *et al.*, 2004). In the context of this work, a security incident is "*the act of violating an explicit or implied security policy*", which is a CERT®/CC definition. The Security Incident Ontology has been developed using Protégé 3.0 and the OWL plugin, available for the tool.

2. The Security Incident Ontology

According to CERT®/CC statistics, the number of security incidents has increased. From 1988 to 2003, 319,992 security incidents have been reported to CERT®/CC. Similarly to CERT®/CC statistics, in Brazil an exponential growing also starts in 1999. According to CAIS (Security Incident Center)[3], created by RNP (National Research Network - NRN) to prevent, deal with and alert the Brazilian security community about incident in its backbone, 68,275 security incidents have been reported from 1997 (year CAIS was created) to November 2004.

Considering this scenario in the security incident domain, the administrators need constantly make decisions about different security alerts they face out. Most of these decisions are made based on the administrators' tacit knowledge. However, as security data and information have increased, it has become more difficult to the administrators to correlate and manage security incidents to make efficiently and effectively decisions. In this sense, the ontologies can be applied.

Before starting the development of the ontology, it is important to answer some questions:

- **Which is the domain the ontology will represent?**

No ontology is wrong, it only represents the point of view of a group of researchers with respect to a restrict domain knowledge. As developing ontologies is not an easy task, it is important to define exactly which domain the ontology will represent. In this sense, this work proposes a security incident ontology, which will represent the terms and relations related to the security incident domain.

- **Why is the ontology needed? For what types of questions the ontology should provide answer(s)?**

The former question is related to the reasons described by Noy and McGuinness (2001): sharing common understanding, reusing of domain knowledge, interoperability and making domains assumptions explicit. The latter question is related to what Grüninger and Fox (1995) have defined in their work about ontologies development

[1] http://cve.mitre.org. Last access: April 8, 2005.
[2] http://www.cert.org. Last access: April 8, 2005.
[3] http://www.rnp.br/cais. CAIS is a Brazilian center similar to CERT®/CC. Last access: April 8, 2005.

methodologies – **competency questions**. As the authors argued, it is important to define what kind of information the ontology will provide. In our case, what the system administrator can learn about security incidents and how he/she can correlate incidents using the ontology. Some competency questions are: which **virus** can explore which **vulnerabilities**? Which **security incident** proceeds, or is correlated to, another **security incident**? Which **consequences** a specific **security incident** can imply to? Which **assets** a specific **security incident** can act on?

- **Who will use the ontology?**

It is important to define who the users of the ontology are: people and/or systems. In our case, systems administrators and security tools are the potential users.

2.1 Developing the Security Incident Ontology

Developing an ontology is as difficult as developing a software. But differently of software development, there is not a consensus among the researchers about the ontology development process. As few works have been published about how to proceed during the ontology development process [Uschold and King, 1995; Gruninger and Fox, 1995; Noy and McGuinness, 2001], there are absences of standardized activities and systematic methodologies.

In the sense to minimize such problems, Fernández *et al*. (1997) developed the **Methontology**. This methodology describes what activities should be carried out during the ontology development. The main activities are: planning and specification, knowledge acquisition, conceptualization, formalization, integration, implementation, evaluation, documentation and maintenance. The knowledge acquisition, evaluation and documentation are carried out during all the ontology development process. As this methodology is based on the IEEE Standard 1074-1995 (1996), which describes software development process, it is the most mature one. Because of its maturity, this methodology has been used to develop the Security Incident Ontology.

2.2 The Conceptual Model

After defining the vocabulary of terms and their relations, the preliminary conceptual model was developed. Figure 1 illustrates the conceptual model of the Security Incident Ontology. Herein, the conceptual model only presents the classes and their subclasses (hierarchical relations). All the attributes of the classes are omitted. The conceptual model was generated by Protégé Tool using OWLViz.

The main class of the ontology is the **Security Incident** one. All the others classes are related to this one. The main idea of the ontology is: an **Agent** *performs* an **Attack** that can *cause* a **Security Incident**. To perform an **Attack**, an **Agent** *uses* a **Tool,** which *explores* a **Vulnerability,** to *get* **Access**. A **Security Incident** *implies to a* **Consequence**, *acts on* an **Asset** and *happens on* such **Time**. Instead of being an attribute of the class **Security Incident**, a class **Time** was created, because it is important to correlate different incidents. The correlation can, for instance, help to establish which attacks or incidents happen before such an incident. Besides these relations, a **Security Incident** can *precede* another **Security Incident** and a **Security Incident** can *proceed* a **Security Incident**. These last two relations also allow the correlation among security incidents.

The main classes of the Security Incident Ontology are described as following:

- **Access**: This class represents the type of accesses an agent can have.
- **Agent**: This class represents the entity that performs one or more attacks in order to cause a security incident.
- **Asset:** This class represents the target of a security incident
- **Attack**: This class represents the attack itself performed by the agent.
- **Consequence**: This class represents the consequences a security incident can imply.
- **Security Incident**: This is the most important class. It represents the security incident caused by an agent through an attack.
- **Time**: This class represents information about when the security incident happened.
- **Tool**: This class represents the means, used by an agent, of exploiting a computational system.
- **Vulnerability**: This class represents the types of vulnerabilities a system can have and it imports the vulnerability ontology developed by Brandão *et al*. (2004a) and Brandão (2004b).

In Protégé Tool, the hierarchical relations are called ***Asserted Hierarchy*** and the non-hierarchical relations and the attributes are called ***Properties***. The ***DataType Properties*** represent the attributes and the ***ObjectType Properties*** represent the non-hierarchical relations, which in the Security Incident Ontology represent the events between classes. For instance, the main ***ObjectType Properties*** of the class **Security Incident** are:

- ***acts_on*** with the class **Asset**;
- ***happens_on*** with the class **Time**;
- ***implies_to_a*** with the class **Consequence**;
- ***proceeds*** and ***precedes***, which are self-relations.

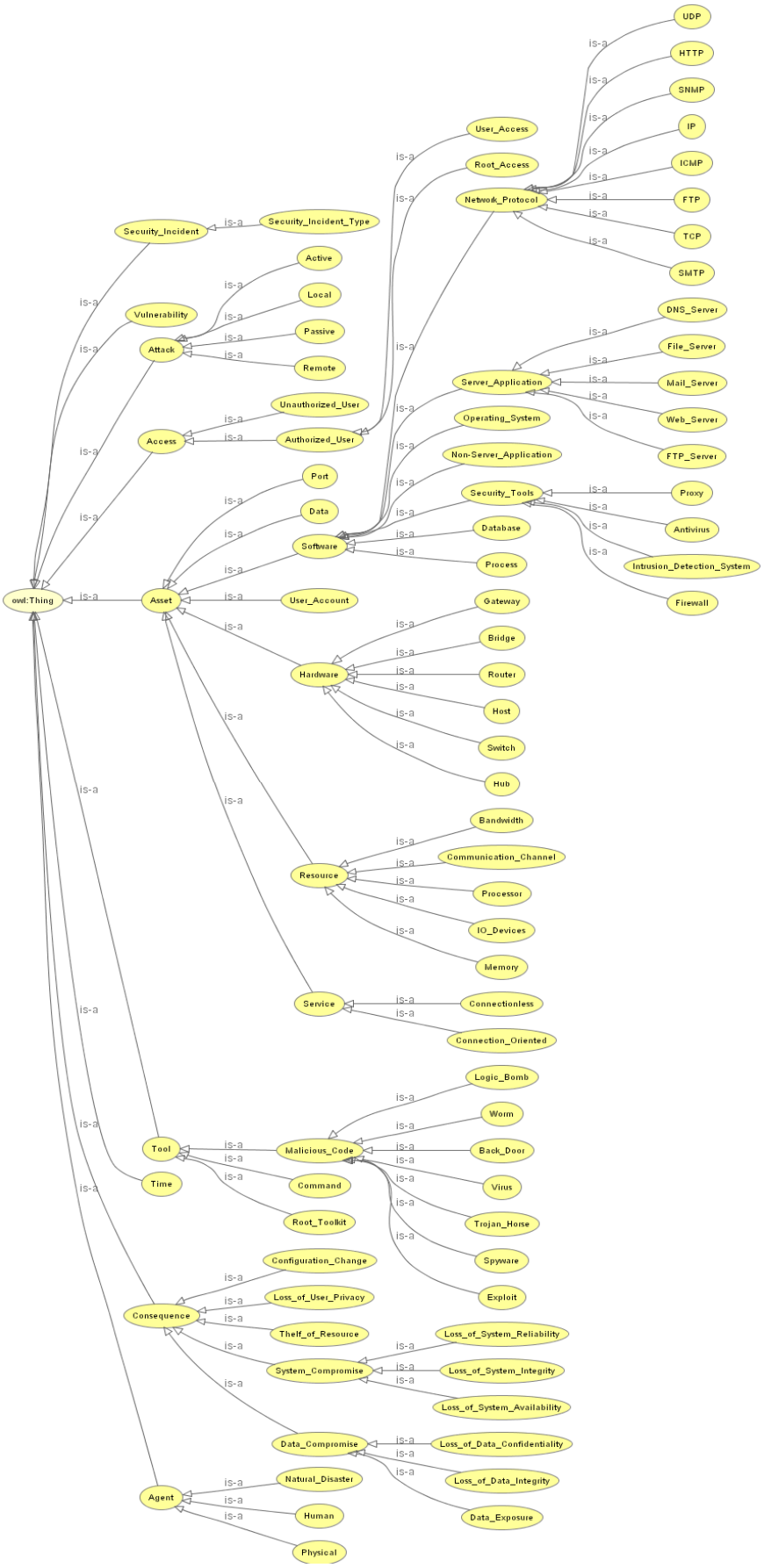

Figure 1 – The Security Incident Ontology Conceptual Model.

3. Final Remarks

Not only ontologies can fundamentally change the way in which systems are constructed, as stated in Swardout and Tate (1999), but also they will change the way people and systems can communicate to each other about domain knowledge.

Knowledge management increases the ability of the organization to learn from its environment and to incorporate knowledge. Using the Security Incident Ontology, the organization can better manage and control its security problems and issues. The system administrator also can better prevent the computational systems from previous security incidents. Besides that, we also want to make it possible to share common understanding of information among people or software agents about security incidents, to reuse knowledge about security incidents, to allow that different applications be able to share security incidents data, and to make domains assumptions and the tacit knowledge explicit.

The Validation Process is in progress. An undergraduate student is developing an OWL plugin to be used with the Snort Tool (Roesch and Green, 2005). The Snort Tool is an open source security tool that analyzes network traffic for matches against a user-defined rule set and performs several actions based upon what it sees. This rules set is being defined based on the ICMC-USP (*Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo*) security policy.

The OWL plugin will store security incident data based on the terms and relations defined by the Security Incident Ontology, being the interface between the Snort Tool and the Security Ontology Incident. Using this integration, besides being evaluated, the Security Ontology Incident can also evolve.

References

Brandão, A. J. S., Martimiano, L. A. F., Moreira E. S. (2004a); **Using Ontologies on Vulnerabilities Alerts**. 22nd Brazilian Network Computer Symposium. Proceedings of the Workshop on Computational System Security. May. (In Portuguese)

Brandão, A. J. S. (2004b); **Using Ontologies to Classify Vulnerabilities on Security Systems**. Master Thesis. ICMC-USP. São Carlos-SP-Brazil. March. (In Portuguese)

Fernández, M., Gómez-Pérez, A. Juristo, N. (1997); **METHONTOLOGY: From Ontological Art Towards Ontological Engineering**. Ontological Engineering - Working Notes. Stanford. March.

Grüninger, M., Fox, M. S. (1995); **Methodology for the Design and Evaluation of Ontologies**. Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI95), Workshop on Basic Ontological Issues in Knowledge Sharing. April.

IEEE (1996); **IEEE Standard for Developing Software Life Cycle Processes**. IEEE Computer Society, New York-USA. April.

Mann. D. E., Christey, S. M. (2003); **Towards a Common Enumeration of Vulnerabilities**. Available online on http://cve.mitre.org. Last access: April 8, 2005.

Martimiano, L. A. F., Brandão, A. J. S., Moreira, E. S. (2004); **Towards a security network incident ontology to ease the security knowledge management**. I2TS´2004 - 3rd International Information and Telecommunication Technologies Symposium. UFSCar. São Carlos-SP. December. p. 88-95.

Noy, N.F., McGuinness, D. L. (2001); **Ontology Development 101: A Guide to Create Your First Ontology**. Knowledge Systems Laboratory Technical Report KSL-01-05, Stanford University. 25p.

Roesch, M., Green, C. (2005); **SnortTM Users Manual 2.3.2 - The Snort Project**. March. 103p. Available online on http://www.snort.org/. Last access: April 8, 2005.

Swartout,W., Tate, A. (1999); **Ontologies**. IEEE Intelligent Systems. pp. 18-19.

Uschold, M., King, M. (1995); **Towards a Methodology for Building Ontologies**. Workshop on Basic Ontological Issues in Knowledge Sharing (IJCAI95).