2009 Eigth IEEE/ACIS International Conference on Computer and Information Science

Application of Security Ontology to Context-Aware Alert Analysis

Hui Xu, *Student Member*, *IEEE*, Debao Xiao and Zheng Wu
Institute of Computer Network and Communication
Huazhong Normal University
Wuhan, P.R. China
e-mail: xuhui_1004@hotmail.com

***Abstract*—With rapid development of computer networks, users need a new solution for network security management, aiming at integration. This paper focuses on context-aware alert analysis, which is one of its key functionalities. A practical and efficient approach to guarantee unified representation of context information, background knowledge and attack knowledge for security alerts is still lacking these days. This paper applies security ontology by means of OWL+SWRL+OWL-S based on CIM Schema to describe context information and security knowledge in a unified manner. We argue that, our proposed approach improves existing alert analysis techniques by providing formal representations with the use of security ontology, which may possibly be an important stage for implementation of unified network security management.**

Keywords- network security management; context-aware alert analysis; security ontology

I. INTRODUCTION

Nowadays, in order to overcome disadvantages of Intrusion Detection System (IDS) and resolve the conflict with current network security requirements as well, more and more researchers and engineers advocate the following two techniques.

(1) Collaboration

The adoption of collaboration is desired to play an important role in realizing unified management. However, the collaboration among different network security mechanisms, different network security products (and possibly different formats) for the same mechanism as well as different data (including various non-security data), has made the management problem harder to solve.

(2) Correlation

Low-level collaboration needs the support of upper-level data analysis. This technique correlatively analyzes data from each context source to get a result that a single data fails to provide. IDS-centric correlation can be divided into the following two parts.

- Alert evaluation
- Alert correlation

According to these two techniques, alert analysis for unified network security management can be divided into three stages, which are alert collection, alert evaluation and alert correlation. However, as for alert analysis, one of the main problems is about effective collection and unified representation for security information, non-security information and correlation knowledge. Our prior work [1] and [2] have discussed the issues related to the support of XML-based integrated network management for alert analysis. The aim of this paper is then to apply security ontology to represent security information, non-security information and correlation knowledge in a unified way, for the sake of context-aware alert analysis.

The remainder of this paper is organized as follows. In Section II, the basic architecture for context-aware alert analysis built upon XML-based integrated network management is briefly presented. Section III proposes the use of security ontology for context-aware alert analysis, and Section IV demonstrates construction of this security ontology using a four-step policy. Section V concludes the paper and prospects future work.

II. OVERVIEW OF PROPOSED ARCHITECTURE FOR CONTEXT-AWARE ALERT ANALYSIS

Network management techniques, which are remarkable for their capability in collecting information of hosts and networks, such as host process information, host system information, network topology, network flow, may be an appropriate approach for context-aware alert analysis to collect context information. Based on the support of collaboration provided by XML-based network management techniques, a basic architecture for alert analysis in the interest of unified network security management is proposed as Fig. 1 [1].

As is presented in Fig. 1, the embodiment of context consciousness is both alert information collection from IDSs and context information collection using the XML-based integrated network management platform from hosts and networks as well. At the same time, network information, host information and vulnerability information are put into corresponding database as background knowledge, which need to be unified in representation.

III. APPLICATION OF SECURITY ONTOLOGY TO CONTEXT-AWARE ALERT ANALYSIS

Context-aware alert analysis mainly focuses on IDSs for post-detection alert analysis and security management actions. Thus from this point of view, standards are urgently required for unified representation.

978-0-7695-3641-5/09 $25.00 © 2009 IEEE
DOI 10.1109/ICIS.2009.199

A. Ontology for Unified Representation

As for unified representation, Intrusion Detection Message Exchange Format (IDMEF) [3] may be a choice. IDMEF is emerging as an industry standard, and it can be used for interoperation between different IDSs. Thus IDMEF agents can then be used to perform the conversion of heterogeneous alert output into the standard IDMEF format. However, IDMEF aims at standardizing various IDS formats, which may be not quite suitable for unified representation of security information and knowledge.

Being a standard container, ontology seems prospective for unification of security information and knowledge representation. The most general and complete definition for ontology is "an explicit and formal specification of a shared conceptualization" [4]. In brief, ontology aims in defining a set of concepts, properties, and their axioms that provide rules, which govern them. It seems that, security ontology becomes a promising direction for researches on context-aware alert analysis.

B. Related Ontology Languages

The Web Ontology Language (OWL) [5], proposed by the World Wide Web Consortium (W3C) for the definition of ontology in Semantic Web, is based on Resource Description Framework (RDF) and RDF Schema (RDFS) and provides greater machine interpretability of Web content by offering additional vocabulary along with formal semantics. Since OWL is a very complete ontology language, it can be directly used to specify context information because it has most of the constructions included in management information languages and even those facets, which are not included, can be defined by extending OWL and RDFS [6].

A step forward in integrating ontology-based security management information is to add behavior information, such as axioms and constraints, to the OWL security ontology. As a Semantic Web rule language combining OWL and RuleML, SWRL [7] is proposed to extend the OWL ontology with rule axioms. A rule axiom consists of an antecedent and a consequent, each of which consists of a (possibly empty) set of atoms. In the "human-readable" syntax of SWRL, a rule has the form *Antecedent* → *Consequent*, with an intended meaning as whenever the conditions specified in the *Antecedent* hold, then the conditions specified in the *Consequent* must also hold. For the case of alert analysis, SWRL is mainly used to define correlation rules.

As is indicated in [8], the ontology described by terms of OWL+SWRL is enough to a security management information definition, and OWL-S [9] then specifies security management actions invoked by the behaviors defined in SWRL. When a particular condition defined in the management OWL ontology occurs, the manager invokes one security management service according to the rule defined by SWRL, and the corresponding service will then be executed to those selected security products or network devices. Thus in this way, the automation of context-aware alert analysis can be partially implemented from the viewpoint of formalization.

IV. Construction of Security Ontology

As for context-aware alert analysis, construction of the security ontology is based on well-known standards in the field of information management, knowledge management and security management. And delighted by the work presented in [10], this construction adopts the following four-step policy based on our prior work [8].

(1) Modeling at the level of concept;

(2) Description in OWL with CIM extension;

(3) Defining correlation rules by means of SWRL;

(4) Defining security management services for automatic response by means of OWL-S.

We will explain each step in detail as follows, in order to illuminate the construction process of security ontology.

A. Modeling at the Level of Concept

According to the method along with its proposed conceptual model in [10], a similar logical model for this security ontology can be constructed in the interest of context-aware alert analysis. Key concepts of this logical model include *Context*, *Asset_Owner*, *Vulnerability*, *Threat* and *Countermeasure*. And taking context-aware alert analysis into consideration, *Context* becomes the most important concept.

B. Description in OWL with CIM Extension

Design of security ontology for context-aware alert analysis is based on CIM Schema, with collaboration requirements of *Context* information (used for alert evaluation) and *Countermeasure* information (used for upper-level treatment).

To help the design of security ontology for context-aware alert analysis, Protégé-OWL editor [11] has been used. The Protégé-OWL editor is an extension of Protégé that supports OWL, providing functions such as edits and visualizing classes, properties and SWRL rules.

Construction of this security ontology for context-aware alert analysis can be realized through transforming only the *CIM_ManagedElement* concept. Take *Context* for example, Fig. 2 and Fig. 3 respectively provide the logical view and the properties view of this construction with the use of Protégé-OWL editor.

Furthermore, Fig. 4 shows part of relationships among OWL classes for this security ontology by the OWLViz tool of Protégé-OWL editor. Based on Fig. 4, *Context* and *Countermeasure* are demonstrated in Table I and Table II.

TABLE I. Descriptions for Context Types

Context Type	Description
Vulnerability	As a significant context, vulnerability information is mainly used to evaluate the importance of alerts. Typical sources are Nessus, Bugtrap and CVE
Operation System (OS)	Vulnerability of OSs themselves may possibly threaten network security. For example, IIS Exploit attacks target at the Linux OS, not the Windows OS
Service	Generally speaking, port, protocol and version information of one service are closely related to network security attacks
Generic	Generic information mainly includes network flow, network user and process, which can be actively gained by XML-based integrated network management tools

TABLE II. DESCRIPTIONS FOR COUNTERMEASURE TYPES

Countermeasure Type	Description
Firewall	Interaction of IDS and firewall is one significant countermeasure, which aims in automatic response
Antivirus	As a post-IDS countermeasure, antivirus needs to be integrated with IDS and firewall for the purpose of unified network security management
NetworkCountermeasure	It is of great importance to define network security policies from a whole viewpoint and demonstrate its motivation, and based on this, concrete countermeasure actions can be defined

C. Defining Correlation Rules by Means of SWRL

In order to supplement the expression of the OWL ontology based on CIM Schema, SWRL is added for the definition of rules (used for alert correlation), which are applied to build attack scenarios. Take one correlation rule in DDoS attack for example. This SWRL description in RDF format is as follows.

```
<swrl:Variable rdf:ID="x">
<swrl:Variable rdf:ID="y">
<swrl:Variable rdf:ID="z">
<ruleml:imp>
 <ruleml:body rdf:parseType="Collection">
  <swrl:classAtom>
   <swrl:classPredicate rdf:resource="#DestIPAddress"/>
   <swrl:argument1 rdf:resource="#x"/>
  </swrl:classAtom>
  <swrl:classAtom>
   <swrl:classPredicate rdf:resource="#FTPService"/>
   <swrl:argument1 rdf:resource="#y"/>
  </swrl:classAtom>
  <swrl:individualPropertyAtom>
   <swrl:propertyPredicate rdf:resource="#ExistFTPService"/>
   <swrl:argument1 rdf:resource="#x"/>
   <swrl:argument2 rdf:resource="#y"/>
  </swrl:IndividualPropertyAtom>
  <swrl:classAtom>
   <swrl:classPredicate rdf:resource="#OS"/>
   <swrl:argument1 rdf:resource="#z"/>
  </swrl:classAtom >
 </ruleml:body>
 <ruleml:head rdf:parseType="Collection">
  <swrl:individualPropertyAtom>
   <swrl:propertyPredicate rdf:resource="#GainOSInfo"/>
   <swrl:argument1 rdf:resource="#x"/>
   <swrl:argument2 rdf:resource="#z"/>
  </swrl:IndividualPropertyAtom>
 </ruleml:head>
</ruleml:imp>
```

By the SWRLTab tool of Protégé-OWL editor, these SWRL correlation rules can be directly generated in XML format. Fig. 5 presents the process of building SWRL expressions for correlation rules. As is depicted in Fig. 5, with the use of the SWRLTab tool, the following SWRL correlation rules can be easily edited.

DestAddress(?x)∧FTPService(?y)∧ExistFTPService(?x,?y)∧OS(?z)
→ GainOSInfo(?x,?z)

DestAddress(?x) ∧ OSSolaris(?y) ∧ GainOSInfo(?x,?y) ∧ VunerableSadmind(?z)
→ HasVunerableSadmind(?x,?z)

DestAddress(?x) ∧ VunerableSadmind (?y) ∧ HasVunerableSadmind(?x,?y)∧Access(?z)
→ GainAccess(?x,?z)

D. Defining Security Management Services for Automatic Response by Means of OWL-S

OWL-S is expected to enable automatic Web service discovery, invocation, composition and interoperation. The ontology of services in OWL-S is composed of several classes, the main one of which is the *Service* class. In fact, it is the *Resource* class that "provides" a *Service*. In order to construct the ontology of services, it is needed to provide the following three essential types of knowledge about a service.

· The *ServiceProfile* class tells about what the service requires of the users or other agents, and provides for them. Thus, the *Service* "presents" a *ServiceProfile*.

· The *ServiceModel* class tells about how the service works and details the semantic content of each request. Thus, the *Service* is "describedby" a *ServiceModel*.

· The *ServiceGrounding* class tells how the service is used and defines a mapping with a Web Services Description Language (WSDL) document of the Web service. Thus, the *Service* "supports" a *ServiceGrounding*.

The semantic description of Web services in OWL-S can be separated into three parts: two abstract parts containing the *ServiceProfile* (the **what** part) and the *ServiceModel* (the **abstract how** part), and one concrete part, which is the *ServiceGrounding* (the **concrete how** part) [12].

OWL-S models services as processes, with definition of the *Process* class as follows.

```
<owl:Class rdf:ID="Process">
 <rdfs:comment>
    The most general class of processes
 </rdfs:comment>
 <owl:unionOf rdf:parseType="Collection">
  <owl:Class rdf:about="#AtomicProcess"/>
  <owl:Class rdf:about="#SimpleProcess"/>
  <owl:Class rdf:about="#CompositeProcess"/>
 </owl:unionOf>
</owl:Class>
```

Thus in this case, security management services for automatic response can be defined as *AtomicProcess* or *CompositeProcess*, while *SimpleProcess* provides the view for these two.

V. CONCLUSIONS AND FUTURE WORK

This paper proposes the use of security ontology for unified representation of information and knowledge in order to realize context-aware alert analysis, and discusses the construction issues about this security ontology by means of OWL+SWRL+OWL-S in detail.

Future work includes implementation of all-rounded building of security ontology used for context-aware alert analysis, taking extensibility into consideration.

ACKNOWLEDGMENT

This work has been partially supported by the Independent Research Planning Project of Public Security Department of Hubei Province in P. R. China under Grant No. 2007STZZCX001 and the Scientific and Technological Planning Project of Wuhan City, P. R. China under Grant No. 200710421130.

REFERENCES

[1] H. Xu, D. Xiao, X. Xia, and Y. Chang, "A Collaborative Architecture for Post-IDS Alert Analysis", Proceeding of 3rd International Conference on Computer Science and Education, Xiamen University Press, July 2008, pp. 666-671.

[2] H. Xu, D. Xiao, X. Xia, and Z. Wu, "Collaborative Post-IDS Alert Analysis Based on Network Management Techniques", Proceeding of 1st International Colloquium on Computing, Communication, Control, and Management, IEEE Press, August 2008, pp. 220-224.

[3] H. Debar, D. Curry, and B. Feinstein, "The Intrusion Detection Message Exchange Format (IDMEF)", RFC4765, March 2007.

[4] R. Studer et al., "Knowledge Engineering: Principles and Methods", Data and Knowledge Engineering, vol. 25, no. 1, January 1998, pp. 161-197.

[5] P. F. Patel-Schneider, P. Hayes, and I. Horrocks, eds., "OWL Web Ontology Language Semantics and Abstract Syntax", W3C Recommendation, February 2004.

[6] J. E. López, V. A. Villagra, and J. Berrocal, "Applying the Web Ontology Language to Management Information Definitions", IEEE Communication Magazine, vol. 42, no. 7, July 2004, pp. 68-74.

[7] I. Horrocks et al., "SWRL: A Semantic Web Rule Language Combining OWL and RuleML", W3C Member Submission, May 2004.

[8] H. Xu, X. Xia, D. Xiao, and X. Liu, "Towards Automation for Pervasive Network Security Management Using an Integration of Ontology-based and Policy-based Approaches", Proceeding of 3rd International Conference on Innovative Computing, Information and Control, IEEE Press, June 2008, pp. 87.

[9] D. Martin, ed., "OWL-S: Semantic Markup for Web Services", W3C Member Submission, November 2004.

[10] B. Tsoumas, and D. Gritzalis, "Towards an Ontology-based Security Management", Proceeding of 20th International Conference on Advanced Information Networking and Applications, IEEE Press, April 2006, pp. 985-992.

[11] Stanford Medical Informatics, Protégé-OWL editor, http://protege.stanford.edu/overview/protege-owl.html.

[12] H. Xu, and D. Xiao, "Applying Semantic Web Services to Automate Network Management", Proceeding of 2nd IEEE Conference on Industrial Electronics and Applications, IEEE Press, May 2007, pp. 461-466.

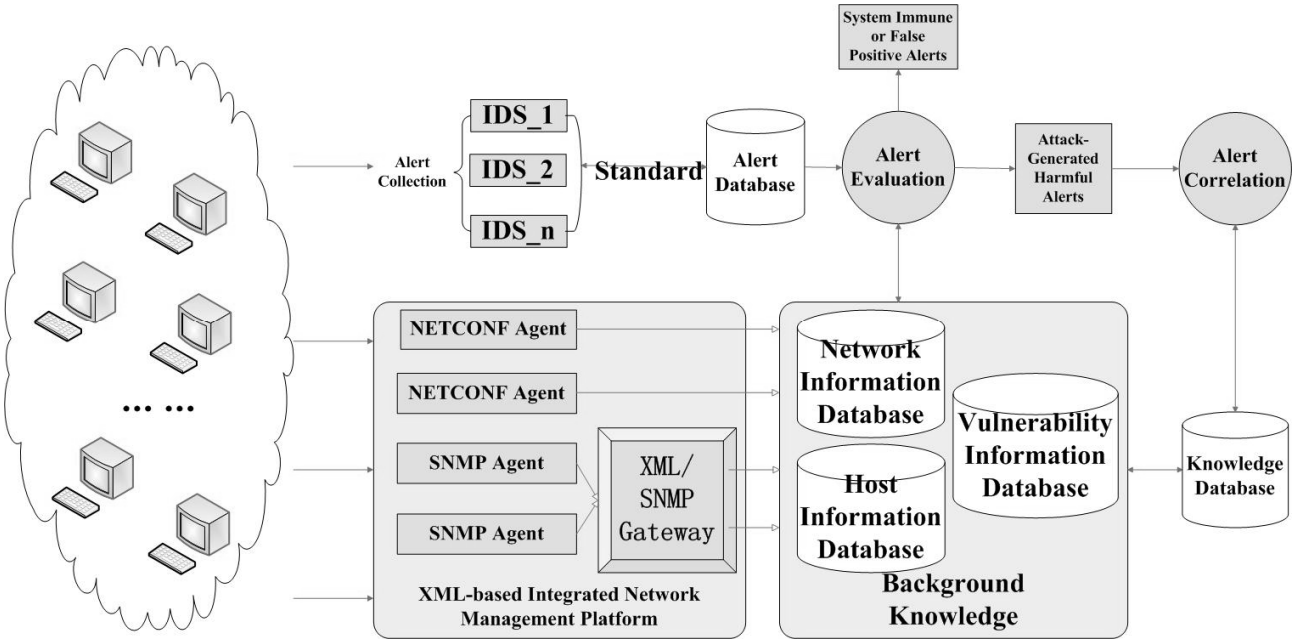

Figure 1. Proposed architecture for context-aware alert analysis based on XML-based network management techniques.

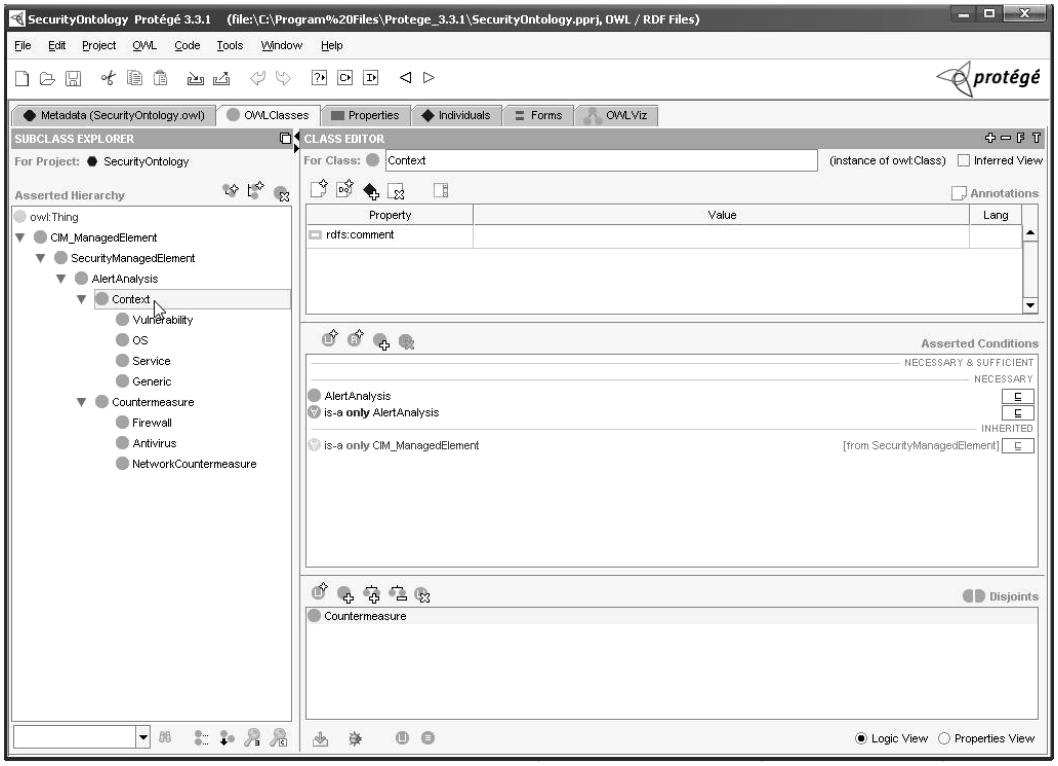

Figure 2. Logical view of constrcuting the security ontology with the use of Protégé-OWL editor.

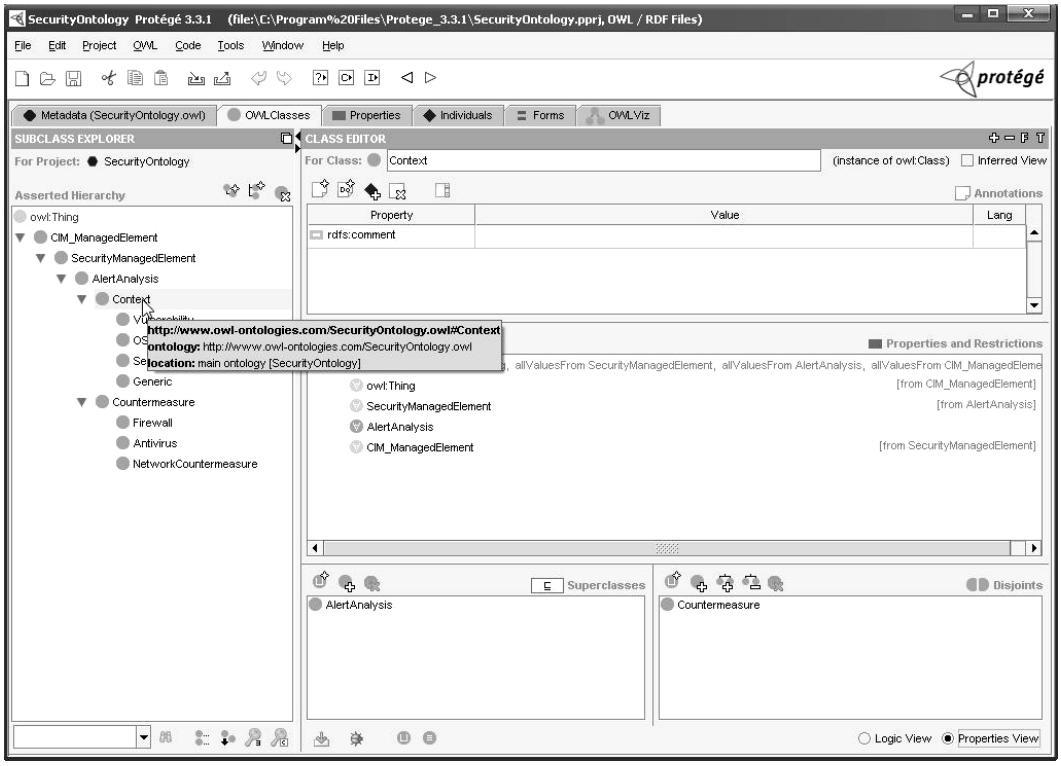

Figure 3. Properties view of constrcuting the security ontology with the use of Protégé-OWL editor.

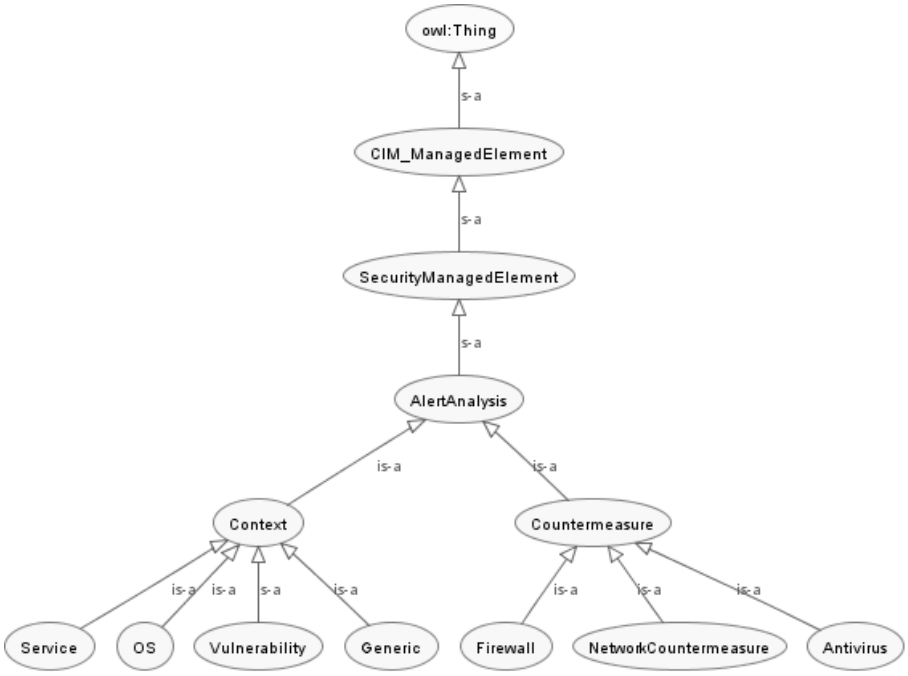

Figure 4. Part of relationships among security ontology OWL classes by the OWLViz tool of Protégé-OWL editor.

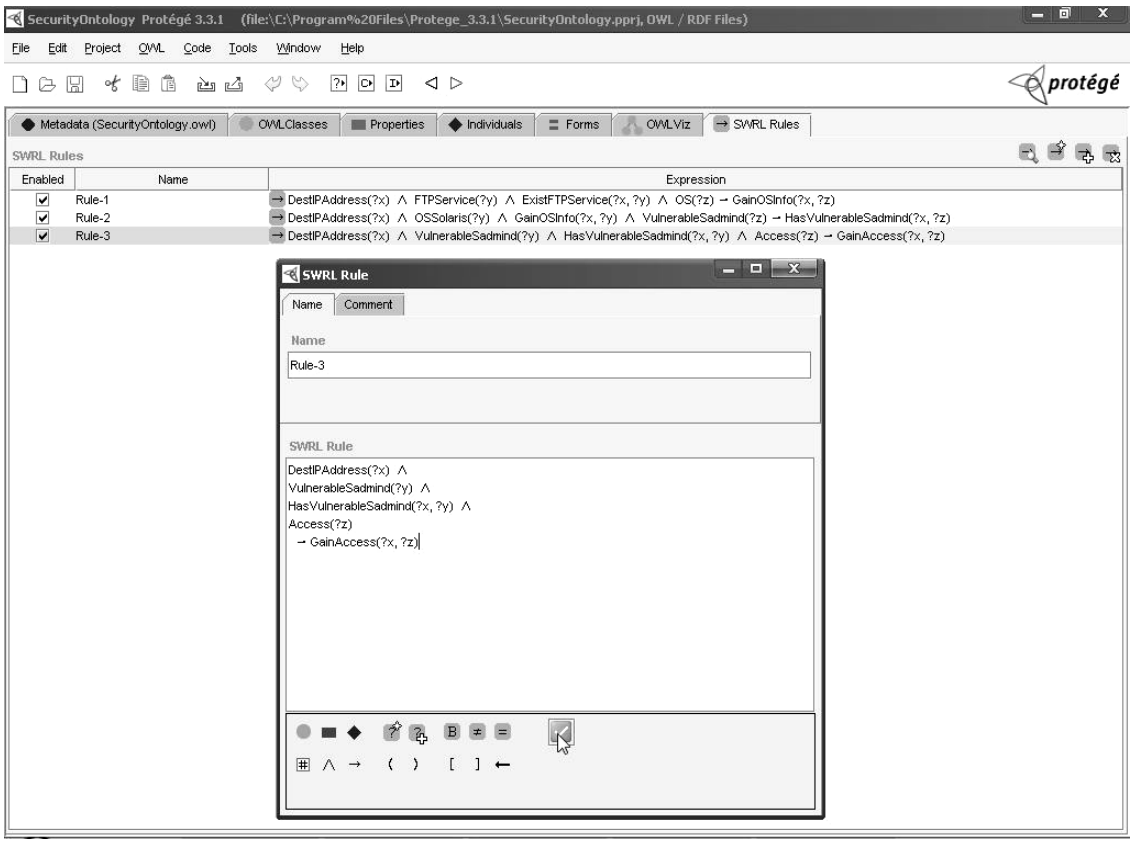

Figure 5. Building of SWRL expressions for some DDoS correlation rules by the SWRLTab tool of Protégé-OWL editor.