VULNERAPEDIA: SECURITY KNOWLEDGE MANAGEMENT WITH AN ONTOLOGY

Francisco J. Blanco, José Ignacio Fernández-Villamor and Carlos A. Iglesias
Departamento de Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid, Madrid, Spain

Keywords: Security knowledge, Knowledge management, Main ontology, Knowledge base, Semantic wiki.

Abstract: Ontological engineering can do an efficient management of the security data, generating security knowledge. We use a step methodology defining a main ontology in the web application security domain. Next, extraction and integration processes translate unstructured data in quality security knowledge. Thus, we check the ontology can perform management processes involved. A social tool is implemented to wrap the knowledge in an accessible way. It opens the security knowledge to encourage people to collaboratively use and extend it.

1 INTRODUCTION

Our main objective is to improve the security that is managed in organizations, encouraging people to use security knowledge in all the application lifecycle stages and in their daily work. The use of semantic in security data generates security knowledge that improves the security processes and strategies to follow. Then, the paper describes how to manage and deal with web security knowledge using ontology engineering.

Rich, full security knowledge repositories would reduce the mistakes and the lacks and necessities of knowledge. However, there is a lack of open security knowledge repositories and in general the few security existent semantic data are very diffuse. However there are large amounts of non-semantic security data stored in several, disparate communities.

This wealth of information is difficult to exploit. A powerful integration of available security information needs an efficient semantic content retrieval and a knowledge management system to wrap the extracted data. Semantic Web prove to be well suited for knowledge management such as integration, production, querying and maintenance (Antezana et al., 2009). From ontological engineering, we use a simple methodology to define a main ontology focused in web application security domain. This unified model is the key to carry out the management processes. We extend it to provide a rich security knowledge base to facilitate the security processes. It requires extraction from heterogeneous communities. The knowledge extraction and its integration try to check the viability of using ontologies under security knowledge.

People must interact with the knowledge to achieve the security objectives and should extend the contents with their own knowledge. So knowledge should be shared in a visible and accessible way. Therefore, we propose a tool that wraps the security knowledge management. The tool opens the contents so that users can apply more effectively the security. Security is an area in constant movement. So the extension of the knowledge base is a real need. However, annotation of security data is time-consuming and requires expert curators. The underlying ontology formalizes and manages new knowledge and the open tool encourages security experts and end-users to collaboratively add and interact with the semantic security contents.

This paper is structured as follows. In section 2 we study the background of knowledge management in software engineering and security area, focusing on ontologies. In section 3 we explain the methodology to carry out the security knowledge management. We detail the definition of the main ontology and present the extraction and integration processes. In section 4 we describe the open, collaborative platform that wraps the knowledge. Finally, in section 5, we expose the main conclusions and our future plans.

2 BACKGROUND

Organizations are changing their viewpoint about the security knowledge towards open systems where all

J. Blanco F., Ignacio Fernández-Villamor J. and A. Iglesias C..
VULNERAPEDIA: SECURITY KNOWLEDGE MANAGEMENT WITH AN ONTOLOGY.
DOI: 10.5220/0003718604850490
In *Proceedings of the 4th International Conference on Agents and Artificial Intelligence* (ICAART-2012), pages 485-490
ISBN: 978-989-8425-95-9

possible knowledge learnt can be accessible. Security engineering is extremely knowledge dependent and its management is a knowledge-intensive process (Papadaki et al., 2008). Any process that carries out knowledge management can mostly work through the individuals but this tacit knowledge is hard to communicate and reuse. The process of knowledge sharing and management conceptualizes the tacit knowledge into explicit knowledge that can be understood and used by the whole organization.

Knowledge acquisition, integration and sharing, belonging to knowledge management process, are significant but complex and time-consuming activities (Korkala and Abrahamsson, 2007). Therefore, specific mechanisms have been suggested to support knowledge processes (Ahlgren, 2011) and some of these solutions use Semantic Web technologies. Thuraisinghman (Thuraisingham, 2005) addresses the importance of the Semantic Web Domain in the security area. Mouratidis et al. (Mouratidis and Giorgini, 2006) show that ontologies are a current necessity and a common challenge in the security engineering.

Nowadays, security aspects are included in an inherent way in software applications (Fink and Koch, 2006) but however they are not integrated enough along the whole development process of such applications. Moreover, how organizations manage the information system security-related knowledge needs still to bel addressed (Aurum et al., 2008). Ergo the selection of a good method that brings effective knowledge management is essential. We offer the use of ontological engineering to manage the knowledge processes.

Specific and global processes of security management and their heterogeneous resources and requirements need a data definition without ambiguities (Huner and Otto, 2009). Ontologies provide a solid, unified data modeling and a shared terminology that solve these issues. So IS security management issue can be done following the ontology perspective of knowledge management (Guo, 2010). Knowledge management using ontologies does aware and sensitive to all users about security implementation in their work processes.

Security engineering needs consensus decision-making activities. Knowledge integration and sharing should explicitly support collaboration between different users, allowing to reach agreements (Debruyne et al., 2010). This property enables the active involvement of all important stakeholders in the decision making process (Papadaki et al., 2008). We strengthen the integration and sharing of ontologies with its wrapping in a social tool.

There are some works that deal with security aspects in an ontological way. Most of them are focused in very specific domains (Blanco et al., 2008). For example, Tsoumas et al. (Tsoumas and Gritzalis, 2006) implement a security ontology that supplies reusing and interoperability of the security knowledge and the aggregation and reasoning of this knowledge from several sources. Wang et al. (Wang and Guo, 2009) present a closed and private application around a security ontology that contains the main concepts about security information. Their ontology is focused on the software vulnerability management.

The work that we take as starting point is provided by Fenz et al. (Fenz and Ekelhart, 2009). They expose an open general security ontology that allows corporations to implement an integral security proposal in IT. Their main architecture can be summarized as follows. A threat represents a potential damage in the assets of an organization and affects to specific security attributes. This threat exploits a particular vulnerability. The defined assets are associated to those vulnerabilities which can be exploited in them. Controls, counter-measures, must be implemented to mitigate or solve the identified vulnerabilities.

3 SEMANTIC SECURITY MANAGEMENT

The no-existence of a common model definition to build well-suited security knowledge is a great problem. Focusing in our web application security domain, we need an ontology that is adjusted to this area. This security ontology allows to carry out the management processes. The model defines the semantic base, a set of well-structured general concepts. After that, we can wait for users to extend the ontology themselves with their learnt security knowledge. However, a knowledge base without initial security contents has great disadvantages for users:

- Slow addition of knowledge: few people use the ontology because it does not contain specific contents.
- An empty knowledge base discourages to users to add knowledge.
- To allow that not-expert users can build the initial security data of the ontology is a mistake.

Therefore, we have to extend the ontology with specific security contents. This addition also contributes to research in two goals. The first is to check the ontology can manage these security contents and all the processes involved, integrating them in a compact security knowledge base. The latter is to try to mitigate the existent lack of open security knowledge with the generated knowledge base.

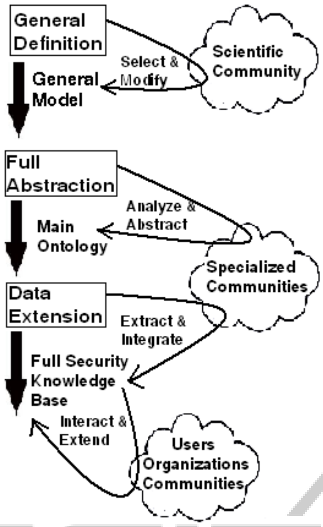

Figure 1: Security Semantic Knowledge Methodology.

To carry out the goals commented, we follow a simple step methodology (see Figure 1) that provides a set of tasks:

- **General Model.** It provides a starting point where the main concepts and their general relationships are defined. Thanks to it, we can do a better search and selection of relevant security communities.
- **Full Abstraction.** The main ontology is defined, extending the general definition. It provides the abstract elements and their possible connections that are used or derived by the security data.
- **Data Extension.** Security contents of the communities selected are integrated in the knowledge base. Security data are wrapped as security knowledge through semantic annotations, using the main ontology.

General model serves as semantic base to define the model focused in the web application security. All next security knowledge should be properly added and integrated in this model so we put special emphasis on its well-definition. Thus, we use Fenz's ontology (Fenz and Ekelhart, 2009). It provides an excellent general (simple and intuitive) security model. Other security ontologies such as (Elahi et al., 2010) (Wang and Guo, 2009) (Tsoumas and Gritzalis, 2006) (Herzogand et al., 2007) have the same basic schema changing few details. However, the Fenz's complete ontology focuses the security management around tangible resources, far from our intangible software application security.

Therefore, we select the most-general part of their ontology and do some adjusts to make it more appropriate to software applications. We also simplify some terms, providing synonymous links so that formalized data using Fenz's ontology can be automatically added to our ontology.

In the next step, we must find relevant contents and specialized information in Web security to provide a rich security encyclopedia. Due to the lack of open security knowledge, we search security-specialized Internet communities that have well-reputed and trustable security data.

The information presented in the communities selected is rich. However, knowledge inside them is not explicitly labeled and we must do an abstraction process to label them, defining the main ontology. The main ontology should cover almost all possible semantic abstractions of the security data existing in the communities. The ontology can be considered as a semantic wrapper of these communities.

After inspection and analysis of each community, we do the full abstraction process to identify, define and integrate the main concepts/abstractions and properties/links enclosed in their security data. This conceptualization process has to solve the different terminology used in each community to a proper integration. Thus, we obtain the main ontology, having extended the general model.

Moreover, we add some concepts from other ontologies. Then users can use them to apply new functionalities in a controlled structure. The connections are:

- Connection to baetle[1]. Now, our security management opens the possibility of semantically managing and tracking bugs and enhancements in assets.
- Connection to doap[2]. Now, the security management can link the assets to their projects, obtaining relevant semantic data from the specification of the project in the doap ontology.
- Connection to foaf[3]. Now, the security management allows associating the provider of the security data with its foaf profile to collect semantic personal and contacts information.

The main ontology is shown in the Figure 2. It defines the semantic abstractions that will be used to annotate the security data to extract. Security data use these unified concepts as tree roots to be extended in a taxonomy way, forming hierarchical structures without divisions.

Data extension is defined by the linkify process. It translates the implicit knowledge to explicit knowledge, linking the contents generated between them. This process should identify the specific resources

[1] http://code.google.com/p/baetle/

[2] http://trac.usefulinc.com/doap

[3] http://xmlns.com/foaf/spec/

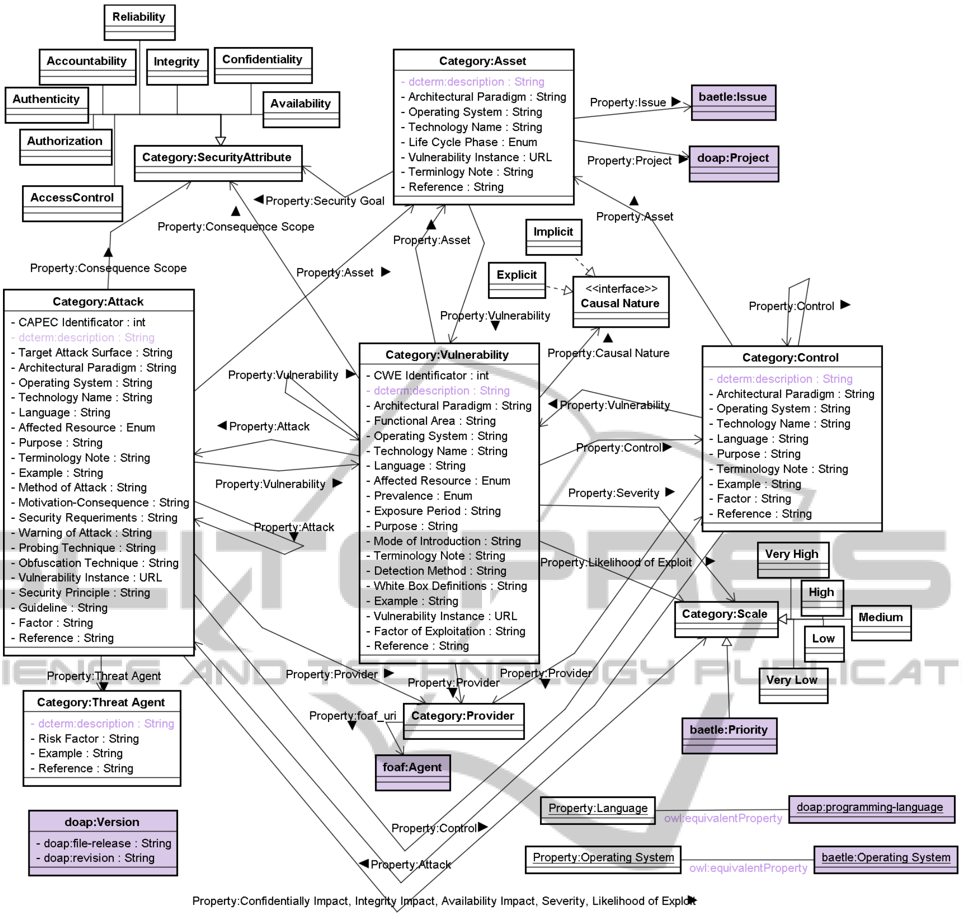
Reliability
Accountability
Integrity
Confidentiality
Authenticity
Availability
Authorization
AccessControl
Category:SecurityAttribute
Category:Asset
- dcterm:description : String
- Architectural Paradigm : String
- Operating System : String
- Technology Name : String
- Life Cycle Phase : Enum
- Vulnerability Instance : URL
- Terminology Note : String
- Reference : String
Property:Issue
baetle:Issue
Property:Project
doap:Project
Property:Security Goal
Property:Consequence Scope
Property:Consequence Scope
Property:Asset
Property:Asset
Property:Asset
Implicit
Explicit
<<interface>>
Causal Nature
Property:Vulnerability
Property:Control
Category:Attack
- CAPEC Identificator : int
- dcterm:description : String
- Target Attack Surface : String
- Architectural Paradigm : String
- Operating System : String
- Technology Name : String
- Language : String
- Affected Resource : Enum
- Purpose : String
- Terminology Note : String
- Example : String
- Method of Attack : String
- Motivation-Consequence : String
- Security Requeriments : String
- Warning of Attack : String
- Probing Technique : String
- Obfuscation Technique : String
- Vulnerability Instance : URL
- Security Principle : String
- Guideline : String
- Factor : String
- Reference : String
Property:Vulnerability
Property:Attack
Property:Vulnerability
Property:Attack
Category:Vulnerability
- CWE Identificator : int
- dcterm:description : String
- Architectural Paradigm : String
- Functional Area : String
- Operating System : String
- Technology Name : String
- Language : String
- Affected Resource : Enum
- Prevalence : Enum
- Exposure Period : String
- Purpose : String
- Mode of Introduction : String
- Terminology Note : String
- Detection Method : String
- White Box Definitions : String
- Example : String
- Vulnerability Instance : URL
- Factor of Exploitation : String
- Reference : String
Property:Causal Nature
Property:Vulnerability
Property:Control
Property:Severity
Property:Likelihood of Exploit
Category:Control
- dcterm:description : String
- Architectural Paradigm : String
- Operating System : String
- Technology Name : String
- Language : String
- Purpose : String
- Terminology Note : String
- Example : String
- Factor : String
- Reference : String
Very High
High
Medium
Category:Scale
Low
Very Low
baetle:Priority
Property:Threat Agent
Category:Threat Agent
- dcterm:description : String
- Risk Factor : String
- Example : String
- Reference : String
Property:Provider
Property:Provider
Property:Provider
Property:foaf_uri
Category:Provider
foaf:Agent
doap:Version
- doap:file-release : String
- doap:revision : String
Property:Control
Property:Attack
Property:Language
owl:equivalentProperty
doap:programming-language
Property:Operating System
owl:equivalentProperty
baetle:Operating System
Property:Confidentially Impact, Integrity Impact, Availability Impact, Severity, Likelihood of Exploit

Figure 2: Main Ontology.

and associate them into class taxonomies and property values. The process must integrate all resources using the main ontology.

Our work is to linkify a greater number of security knowledge to provide a full knowledge base. Firstly, we must collect the data from the communities. The sites selected are: OWASP (www.owasp.org), CWE (cwe.mitre.org) and CAPEC (http://capec.mitre.org). Later, we extract the enclosed knowledge following the main ontology. Finally, we integrate all resources in the knowledge base.

In order to extract the knowledge that is present in OWASP, it is necessary to deal with unstructured data. Often, web pages have information that is understandable by humans, but which is not targeted to automatic agents. Thus, the knowledge cannot be automatically processed or extracted, and therefore the security data that are present cannot be retrieved and stored for further processing.

The extraction process is a tedious and time-consuming task (Ahlgren, 2011). The approach followed is modeling the mappings between the unstructured data of the web sites and the semantically annotated data represented there. To achieve so, we use the Scraping Ontology (Fernández-Villamor et al., 2011). The mappings defined in this ontology are sequences of fragments with the RDF data that they represent, and define what elements inside a web resource represent RDF nodes and/or predicates.

We use Scrappy[4], an Open Source Semantic scraper that makes use of the Scraping Ontology. Scrappy allows defining mappings for each resource and automatically performs some crawling across the whole site, building an RDF graph that contains all the information.

From the pages collected, it extracts the data according to the mappings defined using the Scraping Ontology. Scrappy provides the RDF triples, organizing the resources extracted into taxonomies using the OWASP structure. The mapping elements follow the main ontology terms so the generated RDF graph presents the extracted security semantic data accord-

[4] http://github.com/josei/scrappy

ing to the main ontology formalization.

With OWASP knowledge, we have a rich security encyclopedia. But an effective management in the diversified field of the security involves a solid and coherent understand of the existent data in multiples communities and organizations (Huner and Otto, 2009). The more security knowledge from variety of sources is visible, people and organization can make better informed decisions. Moreover, the more quality linked data, the better to contribute to the lack of open security knowledge. Thus, we complete the security knowledge with contents of CWE and CAPEC communities.

CAPEC and CWE provide their security contents downloadable in XML format. The XML format provides a more or less structured data, defining a terminology used in the whole document. The extraction data process is done by a script. Data are grouped in taxonomies using xml relationships and its terminology is automatically translated to main ontology terms.

Now we have the semantic security data from each community. These data must be integrated to merge the contents in a full, compact, unified security knowledge base. A simple integration can be directly performed in most of the data because all the knowledge is formalized over the same main ontology. For further integration, the main ontology defines properties that allow to connect different communities. For example, attack patterns of CAPEC have references to vulnerabilities in CWE. More integration can be done less automatically using mappings between OWASP and CWE vulnerabilities[5] or OWASP-CWE-CAPEC[6].

The security data combination gives advantages. For example, thanks to CAPEC and OWASP attack contents, communities can be put in the attacker perspective and watch approaches that are used to exploit the software flaws. This knowledge contributes to identify and mitigate the relevant existent vulnerabilities in the software designed that are presented by CWE and OWASP contents.

4 SEMANTIC WIKI

The security knowledge generated should be easily accessible by heterogeneous people to apply them in the whole application lifecycle and fulfill the daily security processes. There is a need of interaction between people and security knowledge. In security, people have another need of interacting between them to discuss about security processes and the knowledge involved. Moreover, continuous revisions are essential so security knowledge must be maintained, updated and extended.

Community-based tools can manage these interactions and specifically wiki systems facilitate the quick collaborative acquisition of knowledge. The wiki allows sharing and exchanging the knowledge inside. Due to the openness and collaborative aspects, the wiki motivates the people's collaboration and learning in developing and maintaining the security data. However, openness could involve loss of quality and correctness. Semantic Web Technologies address these challenges though semantic wikis. They extend the flexibility of a wiki to address structured data.

Their simple and easy-to-use methods make knowledge management accessible to all users of the organization and reduce the learning time (Kasisopha and Wongthongtham, 2009). Semantic wikis allow different formalization levels and provides improvements in the visualization of semantic data. The social perspective of semantic wikis makes a collaborative environment where all users can proactively maintain, enrich and update together the contents. The wiki provides a social space where people can discuss and reach a consensus about the contents, locating each item in its context. So the semantic wiki generates an interoperable, collaborative management service to be used in an internal or external way (Garcia and Gil, 2010).

We select the Semantic MediaWiki solution that supports metadata over semantic notations. They follow the semantic link network (SLN) principle (Zhuge, 2003). Each class, property and individual forms its own wiki page. The relationships with other pages (object properties) and their attributes (datatype properties) make up the page contents.

The wiki wraps our full generated knowledge base, particularly the underlying main ontology. The security knowledge is open to everybody in the URL: http://lab.gsi.dit.upm.es/semanticwiki. Everyone can manage his own security wiki, downloading it from SourceForge: http://sourceforge.net/projects/vulneranet/files/Wiki. Using the export RDF wiki functionality, the last edition of the full knowledge base expects to be included in the Linked Open Data initiative[7]. An open access and the inclusion in LOD is the springboard to reach to a great and active community.

[5] http://cwe.mitre.org/data/graphs/711.html , /629.html , /809.html ; https://www.owasp.org/index.php/CWE_ESAPI

[6] http://www.codeimmunizer.com/coverage.html

[7] http://linkeddata.org/

5 CONCLUSIONS

Semantics in security data improves their management and exploitation. Ontology engineering is designed to face the challenges of management and formalization of knowledge. We define a simple methodology to get security knowledge properly managed. Finally, we obtain a ontology-based system that does an efficient management of the security knowledge.

Due to the lack of an already reference model, we must define the necessary main ontology to our web application security domain. The defined ontology is the base to provide security knowledge that will be efficiently managed. Among other advantages, all stakeholders can now share their security knowledge in an understandable way.

A semantic scraper that uses the main ontology overcomes challenges associated to extraction and integration processes. We demonstrate heterogeneous and unstructured information from various communities can be formalized, organized and merged with a main ontology. The generated full and integrated security knowledge base provides a rich encyclopedia and specific guidelines so that people can apply security knowledge in their daily works from the beginning. With a large amount of open and quality security knowledge, people and organizations can make better informed decisions, building active communities. We check the ontology can manage the management processes involved in the security knowledge generation. Moreover, the knowledge base mitigates the lack of open security linked data.

The security knowledge management system provided by the semantic wiki facilitates intelligent access to knowledge. The wiki facilities reduce the time that users have to spend in security and knowledge management processes, providing everyone can contribute to the knowledge extension using the underlying main ontology. This social platform brings to reality the need of a collaborative knowledge system. Inside it, people can discuss and reach a consensus in security activities with knowledge awareness

Future works include researching techniques and means to make people more awareness to apply security knowledge in their processes. In this sense, methods to enhance people' understanding and to improve the content quality will be investigated. A further enhancement of the knowledge management provides the possibility of tracking security processes over explicit semantic processes. By the way, users add controls that try to solve the vulnerabilities in mitigation processes. By using a reputation system, users can be reputed to indicate what controls are a priori better suited according to the user that references it.

REFERENCES

Ahlgren, R. (2011). Software patterns, organizational learning and sotware process improvement.

Antezana, E., Blonde, W., and more (2009). Biogateway: a semantic systems biology tool for the life sciences.

Aurum, A., Daneshgar, F., and more (2008). Investigating knowledge management practices in software development organizations - an australian experience.

Blanco, C., Lasheras, J., and more (2008). A systematic review and comparison of security ontologies. *Availability, Reliability and Security*, 0:813–820.

Debruyne, C., Reul, Q., and more (2010). Gospl: Grounding ontologies with social processes and natural language. In *Information Technology: New Generations*.

Elahi, G., Eric, Y., and more (2010). A vulnerability-centric requirements engineering framework: analyzing security attacks, countermeasures, and requirements based on vulnerabilities. *Requir. Eng.*, 15:41–62.

Fenz, S. and Ekelhart, A. (2009). Formalizing information security knowledge.

Fernández-Villamor, J. I., Blasco, J., Iglesias, C. A., and Garijo, M. (2011). A Semantic Scraping Model for Web Resources – Applying Linked Data to Web Page Screen Scraping. In *Third International Conference on Agents and Artificial Intelligence*.

Fink, T. and Koch, M. (2006). An mda approach to access control specifications using mof and uml profiles.

Garcia, R. and Gil, R. (2010). Semantic wiki for quality management in software development projects.

Guo, K. H. (2010). Knowledge for managing information systems security: Review and future research directions.

Herzogand, A., Shahmehri, N., and more (2007). An ontology of information security.

Huner, K. M. and Otto, B. (2009). The effect of using a semantic wiki for metadata management: A controlled experiment.

Kasisopha, N. and Wongthongtham, P. (2009). Semantic wiki-based ontology evolution.

Korkala, M. and Abrahamsson, P. (2007). Communication in distributed agile development: A case study.

Mouratidis, H. and Giorgini, P. (2006). Integrating security and software engineering: Advances and future vision.

Papadaki, E., Polemi, D., and more (2008). A holistic, collaborative, knowledge-sharing approach for information security risk management. In *Internet Monitoring and Protection, 2008*, pages 125 –130.

Thuraisingham, B. (2005). Security standards for the semantic web.

Tsoumas, B. and Gritzalis, D. (2006). Towards an ontology-based security management.

Wang, J. A. and Guo, M. (2009). Ovm: An ontology for vulnerability management.

Zhuge, H. (2003). Active e-document framework adf: model and tool.