Security-by-Ontology: A Knowledge-Centric Approach

Bill Tsoumas, Panagiotis Papagiannakopoulos, Stelios Dritsas, Dimitris Gritzalis

Information Security and Critical Infrastructure Protection Research Group
Dept. of Informatics, Athens University of Economics and Business
76 Patission Ave., Athens GR-10434, Greece
{bts, papajohn, sdritsas, dgrit}@aueb.gr

Abstract. We present a security ontology (SO), which can be used as a basis of security management of an arbitrary information system. This SO provides capabilities, such as modeling of risk assessment knowledge, abstraction of security requirements, reusable security knowledge interoperability, aggregation and reasoning. The SO is based on the exploitation of security-related knowledge, derived from diverse sources. We demonstrate that the establishment of such a framework is feasible and, furthermore, that a SO can support critical security activities of an expert, e.g. security requirements identification, as well as selection of certain countermeasures. We also present and discuss an implementation of a specific SO. The implementation is accompanied by results regarding how a SO can be built and populated with security information.

1 Introduction

The introduction of new technologies in conjunction with the dynamic character of Information Systems (IS) brings in attention several categories of information security risks, while in the same time underpins the importance of sound security management. Traditionally, the security controls requirements come up as a result of an IS Risk Assessment (RA) review, given the thorough intervention of security expert. This is an effort-consuming intervention, which has not yet been properly assisted by automated processes, especially in large and complex organizations, which are heavily IS-dependent. In such organizations "a security program in order to be successfully incorporated must be multi-dimensional...these include physical elements, people as well as computers and software" [1].

Our objective is to provide a management framework in order to support the IS security management, as defined with the PDCA cycle (Plan-Do-Check-Act) introduced in [2]. Our work is not directly related with RA approaches per se; nevertheless, it supports the security management process with the use of RA results providing automated support. The creation of such a framework was based in the research direction depicted in [3], that is a) the process in specifying safeguards, b) taking under consideration the nature of the organization's flexibility and c) the creation of adaptive safeguards. We propose a structured approach, in order to support the process leading from informal, high-level statements found in policy and RA documents, to deployable technical countermeasures. The outcome of this process

Please use the following format when citing this chapter:
Author(s) [insert Last name, First-name initial(s)], 2006, in IFIP International Federation for Information Processing, Volume 201, Security and Privacy in Dynamic Environments, eds. Fischer-Hubner, S., Rannenberg, K., Yngstrom, L., Lindskog, S., (Boston: Springer), pp. [insert page numbers].

will be a knowledge-based, ontology-centric security management system, eventually bridging the IS risk assessment and the organizational security policies gap with security management solutions. In order to achieve this, it is important to separate the *IS security needs* into two distinct parts: (a) security requirements ("*Controls*", or the "*What*" part), and (b) their actual implementation in a technical level ("*Technical Countermeasures*" - TC, or the "*How*" part).

We define our basic security knowledge container as a Security Ontology (SO). A SO formulates the basic concepts from the RA process, and in the same time extends the legacy DMTF CIM schema [4] with ontological support. We populate the SO with security information from various sources, ranging from infrastructure-related information to lexical analysis of the high-level statements; the latter stem from RA (security controls) using information extraction (IE) techniques. Wherever the requirements are deemed inadequate, a standards-based, security-best-practices database[1] (ready-to-use controls – refer to section 4) is used in order to fill the gaps.

In the sequel, the terms "Control" and "Countermeasure" will refer to the same concept, considered from a different view; the former is used in the Ontology part (security requirement - "What"), while the latter in the TC database part (technical implementation - "How"). Although in this work we focus on the RA domain, our approach can be applied equally to all domains of IS security management.

The paper is organized as follows; in section 2 we report on related work. In section 3 we define our SO, whereas in section 4 we focus on the attributes of security controls. We present our ontology-centered security management framework in section 5; we setup our case study and present practical results on control attributes extraction in section 6; finally, we conclude with further research in section 7.

2 Related work

Although the need for a SO has been recognized by the research community [5],[6], only partial attention has been drawn for a common solution. The legacy DMTF approach (i.e. the root of our SO), lacks: (a) the security management aspect, (b) the centralized management of security management information, and (c) the domain knowledge perspective. The modelling of CIM with OWL has been proposed by Clemente et al. in [7]. Work in [6] deal mainly with access control issues; standards discussed include XML Signatures and integration with SAML [8] and XACML [9]. Research on KAON [10] focuses mostly on the managing infrastructure of generic ontologies and metadata, whereas in [11] authors present a policy ontology based on deontic logic, elaborating on delegation of actions. The CIM-Ponder mapping is discussed in [12][13], while Raskin et al. presented an ontology-driven approach to information security [14]. With respect to Semantic Web languages, the design of the KAoS [15] policy ontology suggests the use of a description logic inference engine to analyze policy rules and the Rei [11] policy ontology uses F-Logic to compute the policy restrictions and constraints. The policy analysis mechanism in the e-Wallet system [16] exploits the XSLT and JESS technologies, and the SOUPA [17] policy language is similar to Rei but the specific policy ontology has limited support for

[1] The details of this database and its detailed structure, is out of scope of this paper.

meta-policy reasoning. Most of these approaches are related with specific aspects of security and specific application domains, while our approach is suitable for every IS. Furthermore, all aforementioned approaches lack the security standards support, which we use for modeling the security requirements.

3 A BS7799-based Security Ontology for RA

The kernel of our approach is the formulation of an adequate container of the IS security requirements. This container has to fulfill the following high-level attributes: (a) the containment of IS security requirements in such a way, that it is possible to combine them and draw conclusions, (b) the linkage to a global information management framework, and (c) the adherence to globally accepted information security management standards. At a later stage, these security requirements can be used for querying repositories of TC to formulate the proper *Actions* to mitigate the risks.

We model our security knowledge container across the Common Information Model (CIM) [3], a conceptual information model developed by Distributed Management Task Force (DMTF) and ontologies ("*an explicit specification of a conceptualization*" [18]), which have been widely used as an effective means to support knowledge sharing and reuse. Thus, we combine the engineering view of CIM with the knowledge representation world of ontologies.

Extending the modelling of CIM with OWL (Clemente et al. [7]) into the security management domain, we define a generic Security Ontology (SO)[2], as "*an ontology that elaborates on the security aspects of a system*". The SO is formulated as a CIM *Extension Schema* enriched with ontological semantics, modelling the security management information stemming from the RA process ("*What*" part of security needs); in addition, SO is linked with the legacy CIM concepts in order to access the already modelled IS information. While there is no standard method for ontology development [19], we followed the collaborative approach for ontology design [20], building an ontology by a group of people improving the ontology in every iterative round, following the 3-phase approach of [21]:

1. *Phase 1: SO conceptual modelling* is done by using the overall framework in [20] and the security standards ISO/IEC 17799 [22], BS 7799 Part 2 [1], AS/NZS 4360 [23] and the CRAMM Method [24], for extracting the necessary security concepts and their underlying relations;
2. *Phase 2: Linking with CIM as an Extension Schema* is done by introducing the *SecurityManagedElement* concept which inherits from CIM_ManagedElement, populated with certain attributes from [22], [1],[23],[24] and is the sub-root for all security-related sub-ontologies;
3. *Phase 3: Implementing the SO in OWL* is done by using Protégé and its' embedded OWL plugin [25]. The resulting SO is depicted partially in Fig. 1.

[2] In the sequel, the terms "Security Ontology" and "Ontology" will be used interchangeably and refer to the RA sub-ontology.

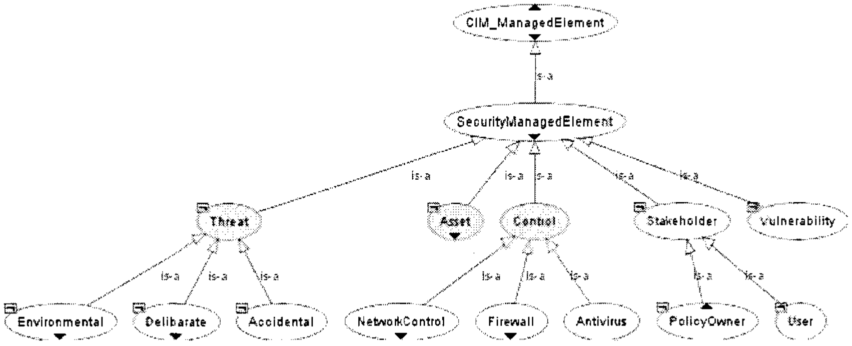

Fig. 1. Security Ontology for Risk Assessment.

The SO concepts have been populated with axioms and relevant semantic constraints, resulting to more than 150 SO concepts, with more than 400 properties. Figure 2 depicts the defined restrictions from the Control concept related to *Asset* protection.

```
Class(Control complete restriction(Protects allValuesFrom(Asset)))

SubClassOf(Control restriction(AcquisitionCost cardinality(1)))
SubClassOf(Control restriction(Subject minCardinality(1)))
SubClassOf(Control SecurityManagedElement)
SubClassOf(Control restriction(LevelOfAssurance minCardinality(1)))
SubClassOf(Control restriction(Effectiveness minCardinality(1)))
SubClassOf(Control restriction(OperationalCost cardinality(1)))
```

Fig. 2. User-defined restrictions for the Control concept.

4 IS asset control semantics

In the case of any risk management methodology, every IS asset is associated with certain threats, which can be then mitigated in an acceptable level by applying specific security controls. Thus, our first task is to define the basic attributes and properties to adequately define a control (depicted in Table 1).

Table 1. Control definition.

Control Structure	
Control Identifier	Unique identifier
Target *	The IS asset that this control is going to be applied (IP address, operating system, open ports & services, etc.)
Subject *	The entity that is going to apply the control to the Target
Control Group *	Categorizes the control in a group
Control Subgroup *	Categorizes the control in a subgroup (further)
Action *	Action(s) to be taken for the control to be applied
Constraints [] *	Time, place, and subject constrains
Type	[Managerial | Procedural | Technical]
SecurityAttributes2Preserve	[Confidentiality | Integrity | Availability | Non-Repudiation]

Control Structure	
Type Of Control	[Protective | Detective | Corrective]
Risk Mitigation Factor	[High | Medium | Low]
Control Purpose	[Security | Audit]

In our SO, every Asset is associated with a set of Threats and every Threat is mitigated by a set of Controls. Thus, every Asset contains this information in the form of a *Threats-Controls[]* array, with each row representing a single threat for the specific asset, along with an array of controls that mitigate the specific threat. This two-dimensional array is shown in Fig. 3; at the ontology implementation level, we dynamically create a series of individuals, which are linked with the respective threats. The attributes *Control Group* and *Control Subgroup* follow the related CRAMM taxonomy for Controls [24]. Using mainly RA information sources, we try to give values in each Control attributes facilitating the binding of each control with appropriate and specific TC by narrowing the search space in the database of collected TC, using the Control attributes as query parameters. We also implement a layered control refinement at the TC database side, resulting in a set of concrete technical actions, which have to be followed in order to implement the initial control. We use the JESS tool [26] providing for different rules in each distinct refinement layer.

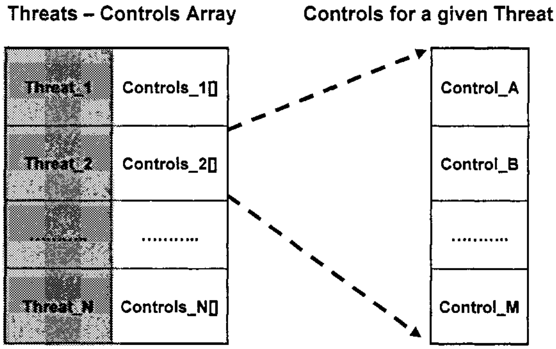

Fig. 3. Decomposition of Threats-Controls array for a given Asset.

5 Framework description

In this section, we extend the generic architecture for IS security management, based on an ontology-centric approach defined in [19]. The aim is to associate the security requirements ("What"), stemming from the security knowledge sources, with the appropriate actions ("How"), and eventually deploy them to the IS.

To accomplish these tasks, four main phases are proposed: (a) Building of SO, in order to simulate the underlying IS, (b) Security Requirements Collection and Evaluation, capturing the IS security requirements ("What") from high-level policy statements into appropriate instances of the SO concepts, (c) Security Actions Definition, matching every security requirement with the appropriate technical security controls ("How"), eventually producing a set of Actions for every IS device instance, and (d) Security Actions Deployment and Monitoring, which can be accomplished by piping the necessary data to a policy-based management platform, such as Ponder [26]. Our approach is modular enough, in such a way that enhancements in any given component(s) can be applied with a minimal overhead to the architecture.

The necessary steps, in order to establish the proposed IS security management framework, are briefly presented at Fig. 4 (the numbers in this figure denote the sequence of main actions in each phase).

Phase A (Step 1): Building of Security Ontology

I. *Get IS asset infrastructure data*; vital data concerning the IS Assets (for example, network topology, technologies used, servers, wireless access points, services and active ports) are located through the use of scanning tools such as Nmap [28];

II. *Generate ontology concepts' instances from infrastructure data*; ontology instances are generated and populated with data (step I) via Protégé API calls [25].

Phase B (Steps 2, 3, 4): Security Requirements Collection and Evaluation

III. *Extract security knowledge from the IS RA and policy documents*; information is extracted from the RA and policy statements, by using IE tools such as GATE [29], and populates the SO concept instances. Eventually fill the gaps (if possible) in the instances from step II.

IV. *Justify with organization managers and discuss business decisions*; management input may influence dramatically the security requirements of the IS, since it might affect network topologies, active services and open ports (e.g. "*salesmen with wireless laptops must have access to the Sales system during the weekend*").

V. *Present the security requirements to management and security experts for evaluation*; if necessary, perform adjustments and/or corrections to security requirements. The database of security and assurance standards may be used for enriching the SO, in case the information gathered so far is deemed insufficient.

Phase C (Step 5): Security Actions Definition

VI. *Associate the security requirements ("What") with technical security countermeasures ("How");* using the information from steps I-V, a matching algorithm performs the linking of security requirements (from the SO) with deployable TCs (from the Database of TCs), customized for the concept instance under question. TC refinement is performed, resulting to a set of N tuples of the form (*IS Asset*$_i$, *Action*$_1$...*Action*$_{m}$,), where *N* is the number of IS Assets identified in the RA and *m* the number of Actions realizing the security requirements of the specific IS Asset$_i$.

VII. *Transform the actions identified into a Ponder-compatible input;* conversion of the Actions specified in step VI into a form that can be piped into Ponder rules or a similar framework through an appropriate interface.

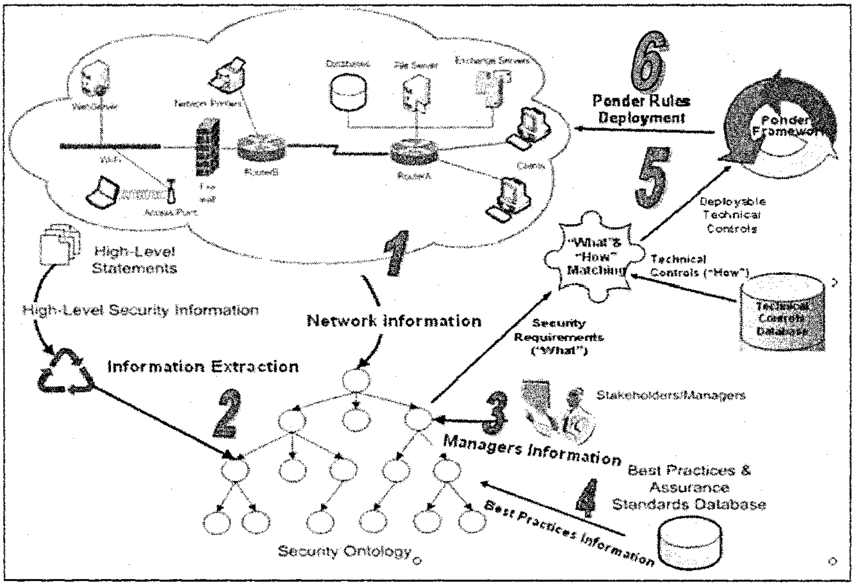

Fig. 4. Ontology-based IS Security Management framework.

Phase D (Step 6): Security Actions Deployment and Monitoring

VIII. *Deploy the Ponder rules over the IS infrastructure;* employ Ponder management framework in order to apply the set of *Actions* over the IS devices.

IX. *Iterate from step I in a timely basis;* stay current with the IS and policy changes.

6 Case-study: Control attributes acquisition

In this section, we present a case study focused on the implementation of the first three steps (i.e. I-III), as part of a RA exercise. We utilize security knowledge from: (a) network-level data referring to the IS infrastructure, and (b) high-level control statements from RA, in order to identify the control requirements.

Having defined and implemented our model SO, the next steps are: (a) to create the relevant ontology concept instances for IS assets, and (b) to populate each Threats--Controls[] array with the control characteristics (defined in section 4).

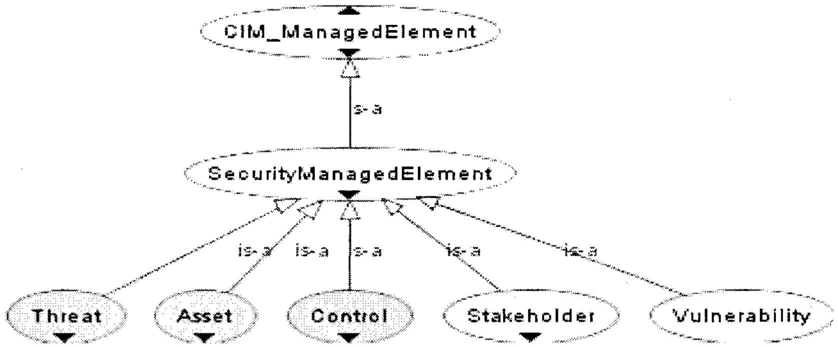

Fig. 5. Case Study: Security Ontology for Risk Assessment.

In order to keep it simple, a cut-down SO version is employed (Fig. 5), consisting of T*hreat, Stakeholder, Vulnerability, Control*, and *Asset.* We specifically deal with the identification of a subset of the control attributes at Table 1 (marked with an "*"): *Target, Subject, Control Group, Control Subgroup, Action*, and *Constraints.* Our conventions and heuristics related to the input data and IE process are as follows:

<u>Input Data</u>

- We are concerned only for the technical controls part of the RA output, which (after their translation to TCs) they can be directly applied to IS devices
- Network information is considered to be accurate and precise

<u>IE process</u>

- If the *Target* cannot be identified, Target defaults to *Information* as a resource
- If the *Subject* cannot be identified, Subject defaults to a predefined group of users (e.g. administrators or network operators).
- We rely on syntactic patterns of the control description, such as
  - <Noun> <to> <Verb> <Something>
  - <Verb> <Something> <Preposition> <Something>
  - <Verb> <Something> <to> <Something> <Preposition><Something>
  - <Verb> <Something> <to> <Something> <List of Prepositions> <Something>, where the word "Between" exists in the <List of Prepositions>

6.1 Testbed IS description

Our test network is depicted in Fig. 6. Following the methodology outlined in section 5, we used an Nmap scanner/parser [28] to isolate the necessary information for creating the relevant IS asset instances in our ontology - i.e. four SO instances are going to be created, one for the router *Alcatel Speed Touch*, one for the *3COM* router and two for the laptops.

The next step is to fill each instance with the retrieved information (e.g. OS, its version, open ports, etc.), using the Protégé OWL Java API [25]. Finally, we feed the control statements (from RA output, Table 2Table 2), to our IE program.

We implemented the IE in Java using the GATE API [29] and JAPE [30], annotating the IE results on the analyzed texts. Apart from the pattern recognition, a number of heuristic rules operate on these annotations as well. Finally, *Target Scope* is also provided – i.e. it is specified whether the specific control applies *only* to the IS asset under question, or to *a set of IS assets* (*Scope of Control Application*).

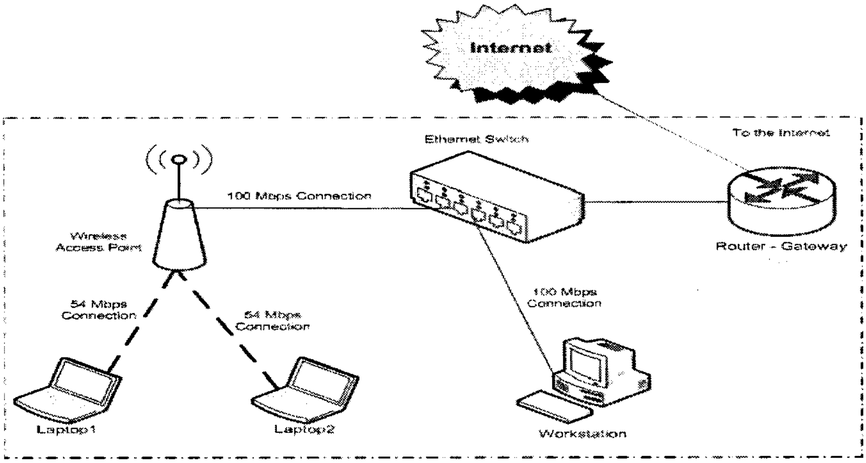

Fig. 6. Case-Study: Network Topology.

Table 2. Control statements from RA output.

ID	Control from Risk Assessment output
1	Use asymmetric algorithms for signatures
2	Use filters to restrict the level of access between internal and external hosts
3	Use filters to control which systems are permitted connections with the Internet
4	Passwords to be at least 6 characters long

6.2 Results and discussion

In order to evaluate our approach, we compared the results of our method with security experts' opinion, using the human expertise as a benchmark. The results from the proposed method are summarized in Fig. 7, with a success detection ratio of up to 71% in comparison with security experts' output. The criteria for success or failure were the ability of the IE to identify correctly the network elements in the text and further classify them in the correct Control Group/SubGroup categories of [24].

As a more detailed example, we focus on the control No 2 (see Table 2) which is related with the router depicted at Fig. 6. In Table 3, we compare the results from the automated extraction (2^{nd} column), against the experts-derived output (3^{rd} column).

Table 3. Automated Control attributes extraction - comparison with expert output.

Control: "*Use filters to restrict the level of access between internal and external hosts*"

Control attribute	IE output	Expert output
Target	10.0.0.138, All_Assets[3]	10.0.0.138, *Routers*
Subject	Administrators	*Administrators*
Group	NetworkAccessControls	*NetworkAccessControls*
Subgroup	Firewalls	*Firewalls*
Action	Use filters	*Use filters*
Constraints	Between internal and external hosts	*Between internal and external hosts*

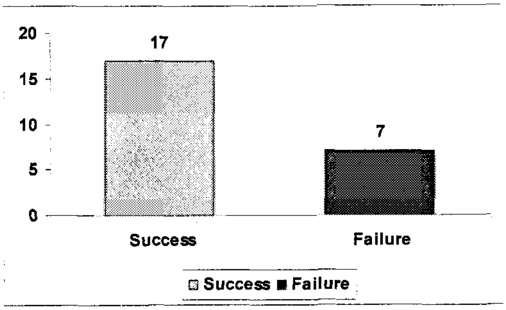

Fig. 7. Automated information acquisition results.

Our IE approach is based on pattern recognition of high-level control statements. Such a method could provide satisfying results, as long as it can be combined with implicit knowledge. Our method was successful into recognizing the network elements and categorizing the controls into the correct *Control Group/Subgroup*, facilitating more accurate identification of *Actions* during the TCs database querying.

This new approach towards the formal representation of security requirements can effectively support security expert's work in an automated way. The modular nature of the framework makes it flexible and tolerant to any changes to both network topology and high-level statements.

On the other hand, our approach can be improved in terms of IE process accuracy, to support more effectively the Control Scope feature (e.g. the IE program identified incorrectly the scope of the control as "All_Assets", i.e., applicable to every IS asset), as well as the identification of constraints (apart from isolation of text describing the possible constraint). *Implicit security knowledge* must be taken into account, e.g. in the control presented above, this control should not be applied to all assets (as stated by the IE output), but only to network assets that are connected to the Internet (i.e. routers).

[3] An example of the *Control Scope* feature, locating (falsely) "All_Assets" as the control scope.

7 Conclusions and further research

In this work we proposed a centralized framework for security knowledge acquisition and management, using a knowledge-centric approach gathering security information from a variety of sources, separating security requirements from their technical implementations. Furthermore, we defined and implemented a standards-based knowledge container (Security Ontology), which: (a) models the Risk Assessment domain and extends the CIM model with ontological semantics; (b) abstracts the security management requirements of a CIM-based domain from the actual implementation, therefore reducing the complexity of controls management; (c) defines a structure for the abstraction of security control attributes; (d) can be used for reusable knowledge interoperability, aggregation and reasoning, using security knowledge from diverse and (already modeled) sources. Finally, we demonstrated the feasibility of security information extraction from RA statements, using IE techniques.

Regarding future work, we envisage the enhancement of heuristic rules so as to produce more concrete and accurate results, as well as the development of a standards-based, best-practices database with implicit security knowledge, in order to support the information extraction and decision making process; further work on countermeasures refinement is necessary, while the evaluation of the results will be assisted by the further enrichment of the ontology with more semantic rules.

References

[1] National Research Council: Computers At Risk: Safe Computing in The Information Age, System Security Study Committee/ Nat.ional Academy Press, Washington (1991).

[2] British Standard 7799, Part 2 (1999), Information Technology - Specification for Information Security Management System, BSI.

[3] Baskerville, R.: Research Notes: Research Directions in Information Systems Security, International Journal of Information Management, 14 (5), 385-387, 1994

[4] DMTF CIM Policy Model v. 2.9, available at http://www.dmtf.org

[5] Donner, M.: Toward a Security Ontology, IEEE Security and Privacy, Vol. 1-3, (2003).

[6] Denker, G.: Access Control and Data Integrity for DAML+OIL and DAML-S, SRI International, USA (2002).

[7] Clemente, F., et. al: Representing Security Policies, in Web Information Systems, in Proc. of the Policy Management for the Web Workshop (WWW 2005), Japan (2005).

[8] OASIS Security Service TC, SAML, available at http://www.oasis-open.org.

[9] XACML Specification v. 1.1, available at http://www.oasis-open.org.

[10] Bozsak, E., Ehrig, M., Handschub, S., Hotho, J.: KAON – Towards a Large Scale Semantic Web, in Proc. of the 3rd EC-WEB Conference, Bauknecht, K., et al. (Eds.), France (2002).

[11] Kagal, L. et al.: A policy language for a pervasive computing environment, 4th IEEE International Workshop on Policies for Distributed Systems and Networks, Italy (2003).

[12] Lymberopoulos, L., Lupu, E., Sloman, M.: Ponder Policy Implementation and Validation in a CIM and Differentiated Services Framework, in Proc. of the 9th IEEE/IFIP Network Operations and Management Symposium, Seoul, South Korea (2004).

[13] Alcantara, O., Sloman, M.: QoS policy specification - A mapping from Ponder to the IETF, Dept. of Computing, Imperial College, United Kingdom.

[14] Raskin, V. et al.: Ontology in Information Security: A Useful Theoretical Foundation and Methodological Tool, in Proc. of the New Security Paradigms Workshop, V. Raskin, et al. Eds. USA (2001).

[15] Uszok, A. et al.: KAoS: A Policy and Domain Services Framework for Grid Computing and Semantic Web Services, 2nd Intl. Conference on Trust Management, UK (2004).

[16] Gandon, L., Sadeh, N.: Semantic web technologies to reconcile privacy and context awareness, Web Semantics Journal, Vol. 1, No. 3 (2004).

[17] Chen, H. et al.: SOUPA: Standard ontology for ubiquitous and pervasive applications, in Proc. of the 1st International Conference on Mobile and Ubiquitous Systems: Networking and Services, USA (2004).

[18] Gruber T.: Toward principles for the design of ontologies used for knowledge sharing, in Formal Ontology in Conceptual Analysis and Knowledge Representation. Kluwer Academic Publishers (1993).

[19] Noy N., McGuiness D., "Ontology Development 101: A Guide to Creating Your First Ontology", Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.

[20] Holsapple C., Joshi K., "A collaborative approach to ontology design", Comm. of the ACM, 45(2):42-47, 2002.

[21] Tsoumas, V., Dritsas, S., Gritzalis, D.: An Ontology-Based Approach to Information Systems Security Management, in 3rd Intl. Conference on Mathematical Models, Methods and Architectures for Computer Network Security (MMM-2005), Russia (2005).

[22] ISO/IEC 17799, Information technology - Code of practice for information security management, ISO (2000).

[23] Australian/New Zealand Standard for Risk Management 4360 (1999).

[24] United Kingdom Central Computer and Telecommunication Agency. CCTA Risk Analysis and Management Method: User Manual, v. 3.0, UK CCTA (1996).

[25] Protégé Ontology Development Environment, at http://protege.stanford.edu/

[26] Ernest Friedman-Hill, "JESS – The Rule Engine for the Java Platform", Sandia National Laboratories, http://herzberg.ca.sandia.gov/jess/index.shtml (Nov. 2005)

[27] Damianou, N. et al.: The Ponder Policy Specification Language, in Proc. of the Policies for Distributed Systems and Networks Workshop, Lecture Notes in Computer Science, Vol. 1995. Springer-Verlag, (2001) 18-39.

[28] Nmap scanner, available at http://www.insecure.org/nmap

[29] Cunningham, H. et al.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications, in Proc. of the 40th meeting of the Association for Computational Linguistics (ACL'02), USA (2002).

[30] Cunningham, H., Maynard, D., Tablan, V.: JAPE: a Java Annotation Patterns Engine, (2nd edition), Dept. of Computer Science, Univ. of Sheffield, United Kingdom (2000).