ONTOLOGY DESIGN AND MAPPING FOR BUILDING SECURE E-COMMERCE SOFTWARE

Esmiralda Moradian and Anne Håkansson
KTH, Royal Institute of Technology, Electrum Kista, Stockholm, Sweden

Keywords: E-Commerce Security, Software Security, Security Ontology, Agent System, Mapping.

Abstract: Developers are struggling with the challenging task of producing secure e-commerce software. Nonetheless, software insecurity remains an issue for e-commerce organisations. Software engineers are expected to possess knowledge in the software engineering area, as well as, security. In addition, they are required to understand and correctly identify the relationships between the security concepts. However, developers commonly lack this knowledge and consequently, security is often omitted during the engineering process. To support developers to face the challenge, we use ontology based techniques for structuring and representation of security knowledge. Categorization according to the security properties of confidentiality, integrity, and availability is needed to provide a holistic view over the security requirements, assets, security threats, and security controls. Moreover, we propose mapping of different security ontologies to provide traceability. For this purpose, we use meta-agents and software agents in multi-agent system. We present a development scenario of electronic invoice presentment system, where we demonstrate how usage of ontologies in combination with multi-agent system can improve security of e-commerce software systems.

1 INTRODUCTION

The importance of secure software systems for e-commerce is unquestionable (McGraw, 2006). Many security threats arise due to poorly designed server and client software. For example, the complexity and size of software programs, lack of security knowledge has contributed to an increase in software vulnerabilities that threat agents can exploit (Laudon, Traver, 2008; Awad, 2007). Despite of different existing solutions, security of e-commerce systems is a growing issue and a challenging task that engineers are struggling with. Since it is difficult for e-commerce developers to sufficiently understand threats and security mechanisms, security needs to be implemented in an early stage of the development process. The lack of knowledge in security area and time constraints does not make it easier for developers. Consequently, insecure e-commerce software is, often, built.

To support engineers to develop more secure software systems for e-commerce, use ontology based techniques that enable security knowledge representation is proposed. The conceptual model where we define the terms, relationships, and dependability of security concepts is presented. Categorization of security expert knowledge and information according to security properties of confidentiality, integrity, and availability, is proposed. Moreover, mapping of different security ontologies, which is performed by the agents is demonstrated. Mapped ontologies result in integrated ontology and checklists. For ontology searching and mapping agents in multi-agent system are used. Multi-agent system enables mapping in an automatic way, which facilitates the developer's work and saves time. Two types of agents are used, namely, software agent and meta-agent. Mapping security threats and security controls to security requirements can provide traceability and holistic picture, which enables control over software security during the development process. Moreover, we present a development scenario of electronic invoice presentment system, where the process of checklist creation is demonstrated.

2 E-COMMERCE SECURITY

E-commerce systems attract attackers that can manipulate the backend of application where sensitive and often desirable data is stored – from credit card numbers to medical information (Moradian, Håkans-

Moradian E. and Håkansson A..
ONTOLOGY DESIGN AND MAPPING FOR BUILDING SECURE E-COMMERCE SOFTWARE.
DOI: 10.5220/0003938901670173
In *Proceedings of the 8th International Conference on Web Information Systems and Technologies* (WEBIST-2012), pages 167-173
ISBN: 978-989-8565-08-2

son, 2006). E-commerce security comprises following security properties: confidentiality, integrity, availability. Confidentiality refers to the ability to ensure that data and information is protected from unauthorized entities. Integrity ensures that no unauthorized alteration has been made to the data/information after the creation, i.e., information being displayed on the Web site, or transmitted or received over the Internet, has not been changed by unauthorized entities. Availability refers to the ability to use the resource desired, i.e., the ability of an e-commerce system or web service to function as intended. (Bishop, 2005; Laudon and Traver, 2008; Awad, 2007).

By using firewalls and/or Intrusion Detection Systems (IDS), e-commerce organizations are trying to protect themselves from attackers but are unaware that their assets are exposed even through firewalls and IDS's. E-commerce organizations must be able to share information with customers and vendors and, at the same time, protect the information from malicious threat agents.

Malicious threat agents can be a human or a software application that exploit vulnerabilities of e-commerce software system. E-commerce software can be exposed in many ways; therefore, developers of e-commerce systems need to decide how the software should react to illegitimate use, such as bad input or illegitimate request (McGraw. 2006). The ability to develop secure enough software for e-commerce depends on how well engineers can understand organization's goals and requirements, and the ability to align those goals and requirements to software development. They also have to understand the risks, which the intended system can be exposed to. Unfortunately, engineers commonly lack the security knowledge. Therefore, security ontology for structuring and representing a security knowledge is proposed.

3 SECURITY ONTOLOGY DESIGN

Ontologies represent knowledge in a specific domain. The ontologies can define terminologies, describe domain, apply structures and build relationships between concepts (Håkansson, Hartung, Moradian, and Wu, 2010). Moreover, ontologies enable analysis and reuse of domain knowledge. The latter is an important factor for building secure e-commerce software. Knowledge about security threats and security controls is example that can facilitate the work of software engineers. Furthermore, possibility to integrate and map several ontologies, for example, security requirements to steering documents, or security threats to security controls, can provide traceability. Moreover, effort and time aspects are also important: time needed for manual integration and mapping of ontologies performed by humans differs significantly from an automated process (Moradian, Håkansson, Andersson, 2010).

The design of the security ontology is based on security process described in ISO/IEC 15408 standard (ISO/IEC 15408, 2009) and aims at structuring the security knowledge. The process involves a number of definitions and relationships between them. Figure 1 presents the security concepts and relationships.

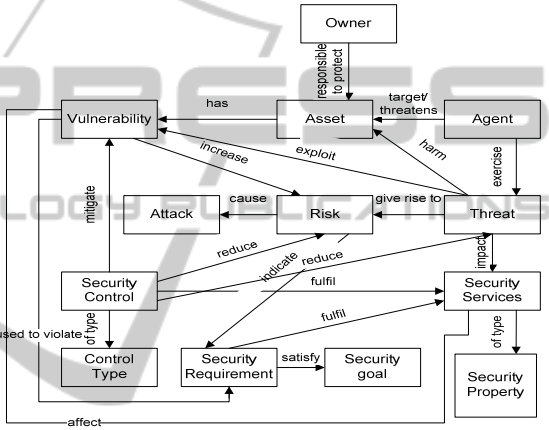

Figure 1: Security concepts and relationships.

In the process an owner is responsible for the asset and assigns a value on it. The asset can be threatened by an agent. The agent is an entity with malicious intention that aims to abuse assets, in order to achieve winnings. The agent exercises a threat, which is an indication of danger or harm. The threat is represents a potential danger to the asset with impact on the security properties, such as confidentiality, integrity, and availability. Threat exploits vulnerability, which is the weakness of the asset. The vulnerability of an asset implies loss of confidentiality, integrity, and availability. Vulnerability can be used to violate security requirement. Threat raise risks for the asset. An asset is an entity valuated by the owner. Risk can realize an attack to the asset. Risk is a danger that an event adversely affects the possibility of reaching the goal (Moradian, Håkansson, Andersson, 2010).

The process also includes security requirement, security goals, and security controls. Security requirements satisfy security property. These security requirements contribute to achieving security goals. Security controls are countermeasures to avoid, mi-

tigate or reduce security threats and risks and vulnerabilities. These security controls protect security properties, such as confidentiality, integrity, and availability.

Security concepts describe ontologies of the security domain. The taxonomy of the security ontology is generic, and thus, extendable as well as applicable on different organizations.

The concepts are defined as security classes where each class has at least one property and one dependency. Class ´Asset´ has properties, for example, ´hasVulnerability´, ´hasOwner´, and ´hasValue´. We can create an ontology rule stating: if an asset has vulnerability and the vulnerability is exploitedbyThreatAgent then the asset has threat. The rule can be expressed as follows:

$$Asset(a) \cap hasVulnerability(a,v) \cap exploitedbyThreatAgent(a,eta) \rightarrow Threat(a).$$

Class ´Threat´ has subclasses that can be designed in the following way: ´ThreatConfidentiality´, ´ThreatIntegrity´, and ´ThreatAvailability´.

The ontology is developed by applying a top-down approach. In a set of software security ontology, the core ontology is the Main Security ontology. The Main Security Ontology has a separate class ´SecServices´ that enables specification of the security objectives. The Main Security Ontology also has other classes, which among others are ´Threat´, ´SecurityControl´, ´Asset´, and ´SecurityServices´ classes. A partial overview of ´Threat´ ontology is presented in Figure 2.

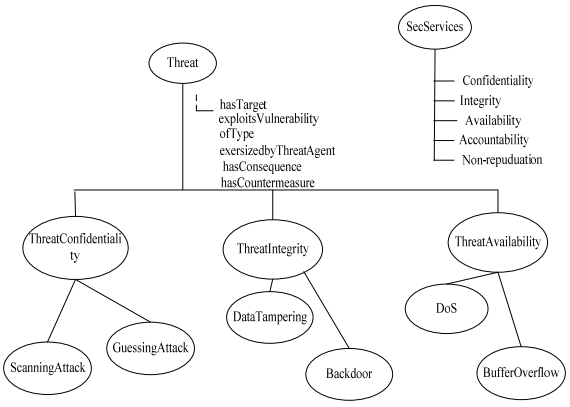

Figure 2: Partial view of threat ontology.

All threats have properties inherited from the parent class, such as hasTarget, exploitsVulnerability, ofType, exersizedbyThreatAgent, hasConsequence, and hasCountermeasure.

```
<owl:Class rdf:ID="ThreatAvailability">
  <rdfs:subClassOf
rdf:resource="#BufferOverflow"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty
rdf:about="hasConsequence"/>
      </owl:onProperty>
      <owl:someValuesFrom>
        <owl:Class
rdf:about="#DenialOfService">
      </owl:someValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

The above owl code shows that ´BufferOverflow´ is a subclass of the ´ThreatAvailability´ class, and that consequence of a Buffer Overflow attack can be Denial of Service attack.

4 AGENTS IN MULTI-AGENT SYSTEM

The purpose of the multi-agent system is to search for ontologies, analyze the content and combine the ontologies to create more complete solutions (Håkansson *et. al*, 2010). The multi-agent system consists of intelligent meta-agents and software agents. The agents are goal-oriented agents, and

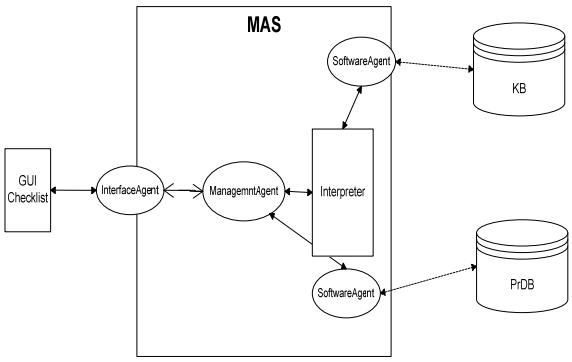

Figure 3: Ontology integration by agents.

perform search for information according to specified requests and satisfy goals. The agents are autonomous and perform tasks without human intervention. The environment that agents work in is observable, cooperative, accessible, and episodic. With observable environment, the agent can communicate with each other and cooperate. By making the environment accessible, the agents have access to information needed to accomplish task and satisfy

goal. In an episodic environment, the agents perform one task at the time (Håkansson *et. al*, 2010).

Agents are assigned different roles, such as search agents and mapping agents. The process is as follows: InterfaceAgent receives a request from the user and process it to the ManagementAgent. ManagementAgent manages and coordinates the activities of SoftwareAgents. ManagementAgent also assigns the task to the Software Agents, that search and retrieve data from repository. The ontologies are retrieved from the local knowledge base.

The search request can contain terminology, as well as, classes and/or properties. The ManagementAgent merges the result from the SoftwareAgents and passes it to the InterfaceAgent. Thus, the meta-agent perform mapping by comparing different ontologies. If there is a direct match the meta-agent can continue to work with the next part of the ontologies. The meta-agent uses the knowledge base to compare the ontologies and interpreter to match and execute rules. The process is illustrated in Figure 3.

The knowledge base contains facts and rules. Rules can be constructed by terms or rule with several different alternatives, such as synonyms. For mapping the ontologies, there must be related parts. (Håkansson *et. al*, 2010) The ontologies are examined against the knowledge base with tags like *owl:Ontology*, *owl:Class*, *rdfs:subClassOf*, *owl:onProperty* (Håkansson *et. al*, 2010).

5 E-INVOICE PRESENTMENT AND PAYMENT MODELS

Electronic Invoice Presentment and Payment Model (EIPP) is a process usually used by organisations to present invoices through the Internet. Ontology based techniques and agent systems are often used for implementation of the invoice presentment process. Presentment means taking the information that is available on a printed invoice and hosting it on an e-invoice presentment Web server. To assure the security of the process, security has to be considered at the early stage. There are three models of EIPP: buyer direct, consolidator, and seller direct (Turban, King, McKay, Marshall, Lee, and Viehland, 2008). Buyer Direct is the model where there is one buyer and many sellers. Seller post invoices to the buyer's EIPP for viewing. Consolidator, which is many-to-many model, acts as intermediary. Sellers and buyers register with the consolidator's EIPP system. Sellers generate and transmit invoice information to the EIPP system. The consolidator sends notification to buyers that invoices are ready for viewing. Seller Direct model implies that one seller is linked to many buyers (Turban *et.al*, 2008). The seller generated invoices and informs buyers that the invoices are ready for viewing. Byers can then log into the seller website to view the invoices. There are different options for making payments in an EIPP system among others the payment method that uses purchasing cards, and online banking. For the research in this paper, we focus on the Seller Direct model, only, and therefore, do not consider e-payment part.

In the next chapter, we present a scenario where security of e-commerce software is considered from the requirement phase of the development process.

6 ELECTRONIC INVOICE PRESENTMENT

In this section a development scenario of electronic invoice presentment software system is described. The identification of security threats and countermeasures is demonstrated by mapping ´Threat´ ontology to ´SecurityControl´ ontology. The search for ontologies is performed by agents that use a knowledge base to compare the ontologies and an interpreter to match and execute rules.

6.1 Scenario

Consider that a Telecom Corporation (TC) develops an electronic invoice presentment software system. The system will be used to provide invoice presentment service for TC's customers through the web. The TC works according Seller Direct model and the process involves producing the invoices, monitoring, archiving, and presenting invoices to the customers. The steps of the invoicing process are depicted in Figure 4.

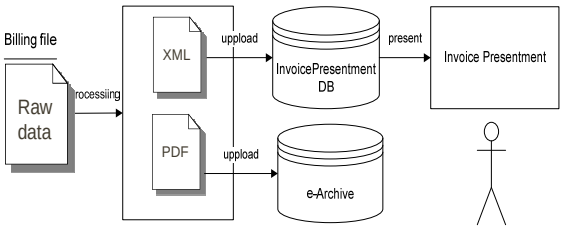

Figure 4: EIPP system.

The first step is to prepare the invoice data. Invoice data is the raw billing data. Next step is processing, which implicate converting raw data into xml format used for presentment of the invoice on the webpage. XML invoices are stored in Web server. A PDF

invoice is produced and saved in electronic archive (e-Archive). The PDF invoice is saved in e-archive due to legal requirements as well as can be viewed and printed by customer. The invoice issuer sends e-mail notification to the customer about e-invoice. Customers' data is stored in customer database.

Customers/buyers log into the TC's Web site to view the invoices. After viewing an invoice the customer can pay the bill. TC organizational requirements are as follows. The customer should be able to view only its own invoices. Customer information and invoice data should be protected. The availability of the system should be 98%. Information presented in an electronic invoice originates from the billing file.

External users, i.e, outsiders, are non-customers, competitor organizations, and suppliers should be able to view the marketing information that resides on the web server. External users must not access the development environment. TC's customers are managed by the admin users/developers via the admin tool. Hereby, following user classes are identified: outsiders, developers, corporate managers, and customers.

The goals of the TC security policy are:

1. Corporate data that is classified as restricted must be protected and available only to authorized users and only to those who need to know. One example is data involved in developing of new software.
2. Customer data and all the information about customer, which involve personal information, order info, and invoice information must be protected and can only be accessed by those who provide the information/fill the order, pay the invoices, i.e. only by the customer himself. Developers and order managers can only obtain the information that is necessary to process the orders and manage customers' information.

The identified corporate assets are: information, IT-systems, physical assets, and staff. In this research we only consider the asset, such as information.

Information is classified as: public data, development data, corporate data, and customer data. Public data is available to everyone (outsiders). It includes product information and specification, price information, and marketing information. Development data, such as product and software specifications is available only to developers. Corporate data, such as legal information, corporate sensitive data is available only to corporate managers and lawyers. Customer data is the data provided by customers. TC needs to identify threats that possibly can target the system and countermeasures that can eliminate or mitigate those threats. To identify threats and countermeasures, security ontology in combination with the multi-agent system is applied.

6.2 Security Checklist Creation

TC system is based on client/server software application which interacts with users or other systems using HTTP for untrusted users and HTTPS for trusted users. Software is responsible for management of the TC's customers, presenting of invoices for those customers and for processing customer/user input. Each component of the system has vulnerabilities and potential threats. Security policy of TC can be categorized according to the security properties of confidentiality, integrity, and availability and, hence, following security requirements can be identified.

The confidentiality aspect of TC security policy requires that application software must have high assurance. Therefore, authentication function must authenticate the users. Moreover, authorization function shall prevent unauthorized subjects and objects from accessing the system. Web server and databases have to be secured, so that only authorized individuals can access the information.

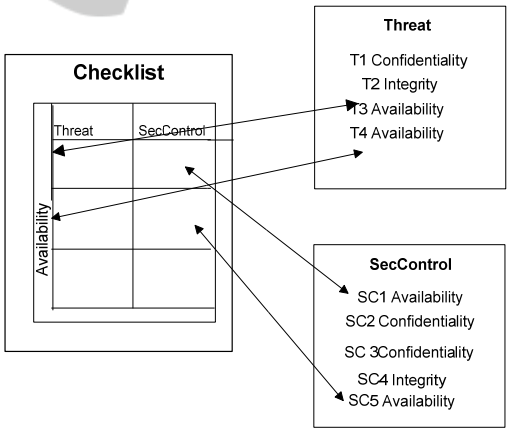

Figure 5: Checklist example.

The integrity aspect of the TC security policy requires that data is protected from modification. Therefore, improper or unauthorized change of data/information shall be prevented and data origin shall be assured. Following security events shall be logged together with the userID, date and time for transaction: login/logout attempts in order to detect any attempts to change the data in unauthorized way.

The availability aspect of the TC security policy requires that the software system is available to the

public, developers and customers. The needed information should be available to authorized users depending on the access right.

To enable mapping between security requirements, threats, risks and countermeasures, a multi-agent system that uses security ontologies is applied. Security ontologies contain knowledge about the security requirements, threats, risks, and countermeasures, which are categorized according to security properties of confidentiality, integrity, and availability. The agents check ontology by terms, classes or properties. The knowledge base contains rules as well as has equivalent terms, properties or classes. If equivalent term is not found an indirect mapping is performed. A threat ontology is used to represent knowledge about threats as described in section 3. To identify potential threats TC engineers provide system with keywords about asset, and security requirement. The system searches for ontologies and examines them as described in section 4. After performed search and examination of ontologies the system can present a threat list from the ´Threat´ ontology. Relationship between threat and security control enables identification and mapping threats to security controls, which are countermeasures.

Figure 5 demonstrates mapping of two ontologies into a checklist. The checklist is created by the agents as a result of mapping between threats and countermeasures.

Table 1 shows a small example over TC's security requirements and corresponding threats and countermeasures that can be generated as a result of ontologies mapping. The list provides the information about how TC's requirements are fulfilled but can also be used for traceability purposes.

Table 1: Merged list from ontologies.

Sec. Property	Sec.Req	Security Threats	Security Control
Confidentiality	User authentication	Eavesdropping	Encryption during transmission i.e. encrypted communication channel. SSL and IPSec should be used
		Dictionary attack	use strong passwords
		Brute force attack	use strong passwords
Integrity	prevent improper / unauthorized change of data / information	Man-in-the-middle	cryptographic techniques
	Security functions shall assure the data origin	Data tampering	Access control
Availability	Information should be available to authorized users depending on the access right	Denial of Service	All data input should be validated. Exception handling should be used. Harden the TCP/IP stack against denial of service

Usage of ontologies in combination with multi-agent system can provide TC engineers with the information about specific threat, vulnerability that specific threat exploits, as well as, what security properties the threat impacts. Engineers can also obtain information about countermeasures that can be implemented in order to avoid or minimize the threats. Figure 6 presents a simple user interface where the information about a BufferOverflow threat is presented to the engineer.

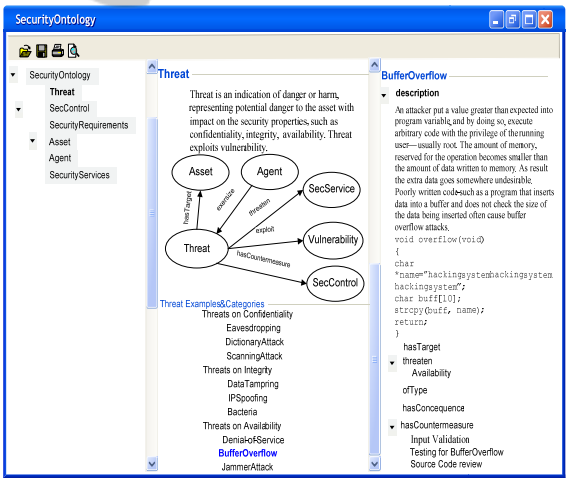

Figure 6: User interface.

Identified potential threats to the TC's system as well as countermeasures allow TC engineers' implement security controls during the development process. The security controls can avoid threats and increase probability that more secure software system is developed.

7 RELATED WORK

Tsoumas, Dritsas, and Gritzalis (Tsoumas, Dritsas, and Gritzalis, 2005) present an ontology-based approach that manages security of information systems. The authors argue that the approach is structured to support the process from policy and Risk Analysis (RA) documents to technical controls. The authors state that the result is a knowledge-based, ontology centric security management system that may bridge the IS risk assessment and organizational security policies with security management (Tsoumas, Dritsas, and Gritzalis, 2005).
Gorodetski *et al.* (Gorodetski, Popyack, Kotenko, and Skormin, 1999) propose a multi-agent model of an information security system that is based on ontology. Authors use the ontology to structure the distributed knowledge. Nodes and relations defined in ontology are used by an agent that aims at solving the entire multitude of problems related to particular tasks.

In our work, we propose the usage of ontology based techniques for security knowledge representation and multi-agent system that enables mapping between security requirements, security threats and security controls. We develop the security ontology that supports security management during the development of new software systems and maintenance of the existing systems. The ontology is general and can be applied on different types of organizations. For mapping of ontologies we apply multi-agent system.

8 CONCLUSIONS

We have proposed security ontology that aims to support engineers during software engineering process and improve the security of software systems. The security ontology provides holistic view over the security concepts and relationships between those. We presented taxonomy of the security ontology and discussed the necessity, importance and usefulness of using security ontologies. Furthermore, we presented electronic invoice presentment and payment models and demonstrated electronic invoice presentment scenario. In the scenario, the possibilities of utilizing multi-agent system with ontologies for automatic mapping between threats and countermeasures, which can provide traceability and facilitate to the development of more secure software systems is demonstrated.

ACKNOWLEDGEMENTS

This research is supported by Sweden-Korea re search Cooperation Programme funded by STINT, The Swedish Foundation for International Cooperation in Research and Higher Education. http://www.stint.se

REFERENCES

Awad, E. M. 2007. *Electronic Commerce From Vision to Fulfillment.* Pearson Prentice Hall 3rd Ed., ISBN 0-13-173521-7

Bishop, M. 2005 *Introduction to Computer Security*. Pearson Education.

Gorodetski, V. I., Popyack, L. J., Kotenko, I. V, Skormin, V. A. 1999. *Ontology-Based Multi-Agent Model of an Information Security System.* Springer-Verlag Berlin Heidelberg. LNAI 1711, pp. 528-532

ISO/IEC 15408:2009 Common Criteria for Information Technology Evaluation, Part 1: Introduction and general model. V.3.1 Revision 3, CCMB-2009-07-001

Håkansson, A., Hartung, R., Moradian, E., Wu, D. 2010. *Comparing Ontologies Using Multi-Agent System and Knowledge Base*. Proceedings of the 14th international conference on Knowledge-based and intelligent information and engineering systems: Part IV, Springer-Verlag Berlin, Heidelberg ©2010

Laudon, K. C., Traver, C. G. E-Commerce business. Technology. Society. 4th Ed. Pearson InternationalEdition. 2008. ISBN-10: 0-13-500932-4

McGraw, G. 2006. *Software Security Building Security in.* Addison-Wesley Pearson Ed., ISBN 0-321-35670-5.

Moradian, E., and Håkansson, A. 2006. *Possible attacks on XML Web Services.* (IJCSNS) International Journal of Computer Science and Network Security. Journal ISSN: 1738-7906 Volume Number: Vol.6, No.1B,

Moradian, E. Håkansson, A., Andersson, J-O. 2010. *Multi-Agent System Supporting Security Requirements Engineering.* Accepted in The 9th International Conference of Software Engineering Research and Practice (SERP 10)

Tsoumas, B., Dritsas, S., and Gritzalis, D. 2005. An *Ontology-Based Approach to Information System Security Management.* LNCS, 2005, Volume 3685/2005, pp. 151-164

Turban, E., King, D., McKay, J., Marshall, P., Lee, J., Viehland, D. 2008. *Electronic Commerce A Managerial Perspective.* Pearson Education. Upper Saddle River, NJ