An Ontological Approach to Mitigate Risk in Web Applications

Marcius M. Marques, Célia G. Ralha

Departamento de Ciência da Computação – Universidade de Brasília (UnB)
Brasília – DF – Brazil

marcius@marciusmarques.com, ghedini@cic.unb.br

Abstract. *Information Security (InfoSec) is becoming a high priority asset to support business activities, as organizations struggle to assure that data is available and secure in web applications. However, security is not a concern from the beginning of the development process, mainly because developers are not security specialists. Consequently, vulnerable systems are designed and when attacked can compromise organization's data and operations, enclosing high financial losses. Because most attacks targets the application layer, we propose an intelligent approach based on ontology to mitigate risks in web applications. An ontological approach can contribute to InfoSec knowledge dissemination and reduce the burden of implementing secure web applications on organizations. The ontology is based on the OWASP Top 10 Project, applied to reduce the gap between the application developer and the security knowledge. The proposed model is employed in the development's design phase; with more secure web applications as the outcome. The extensible and reusable developed ontology is evaluated in a prototype scenario of a web application named 'SMS Broadcast'. The results show that vulnerabilities can be reduced by increasing the security awareness of web developers during the application development process.*

1. Introduction

The fact that organizations dependability on information systems (IS) to manage their business activities is increasing is a known and irreversible one [Weske 2007]. Global networking and Information Technology (IT) advances at high speeds allowing all types of organizations to take advantage and achieve better results, in order not to risk falling behind competitors. IT is a strategic area for organizations, with web-based systems playing a central role in the modern economy, where information needs to have instant availability. This needed feature is followed by an increase in the number and sophistication of attacks to web applications, and as a result, organizations faces a great challenge to keep information secure [da Silva and Ellwanger 2012].

Although Information Security (InfoSec) is a growing spending priority in most organizations, the vulnerability rates and losses numbers are very high. In a security assessment with more than 200 web applications (including e-commerce, on-line banking, credit cards companies, etc), vulnerabilities that could be explored were found in more than 90% of them [Only 10% 2005]. In Cyberattacks (2013), it is reported that companies losses due to hacking and cybercrime range from U$300 billion to U$1 trillion dollars, with hackers stealing more than one terabyte of data daily from vulnerable web applications. According to Key Findings (2013), the attacks on web

applications are a routine part of business and will be a part of doing business going forward. The worrying potential economic impact related to InfoSec for organizations can be found in details at Gordon and Loeb (2002).

A successful InfoSec program enfolds many layers, including software to detect viruses, firewalls, sophisticated encryption techniques, intrusion detection systems, automated data backup and hardware devices, to name a few. Organizations have then to decide what exactly needs to be protected, what is the level of protection that each resource requires and which tools can be used to achieve it all [Almeida 2007]. Achieving consensus regarding safeguards for an IS, among different stakeholders in an organization, has become more difficult than solving many technical problems [Dhillon and Backhouse 2000]. Moreover, the needed knowledge to apply a successful InfoSec project is available on technical standards (e.g. ISO 27001) or in the head of security specialists. Consequently, InfoSec projects tend to be complex, expensive and time consuming, with unclear and hard to measure benefits [da Silva et al. 2011].

However, researches showed that 75% of attacks are being deployed at the application layer vice infrastructure [Razzaq et al. 2009], giving opportunities for risk mitigation within the control of organizations. Because software developers are usually not security specialists, web applications are designed with minor or none security concerns. The neglect of good programming practices, including the simplest ones, is one of the main causes for the existence of vulnerabilities in web-based systems [Silva and Ellwanger 2012].

In this article, we propose an intelligent approach that uses an ontology to reduce the gap between web application developers and the needed security knowledge. The source of information is based on the OWASP (Open Web Application Security Project) initiative, specifically on the OWASP TOP10 Project [OWASP Top 10 Project 2014]. This project is constantly updated with the most critical web applications security flaws. When adopted by an organization, the main target is to change the software development culture in order to produce secure code.

One of the core OWASP project pillars that we agree with is that security in application development should be considered since the beginning of the development process, being included in all stages. Nevertheless, in this work we intend to focus on the design phase of the Software Development Life Cycle (SDLC). By increasing the security awareness of the web application developer, we believe the final product will be more secure as known potential risks will be mitigated.

The proposal is related to the risk management aspect of security activities, a very important process within InfoSec, required to be part of the organizations' security policy [Peltier 2013]. From the definition in Whitman and Mattord (2011), Risk Management (RM) is the process of identifying risk, represented by vulnerabilities, to an organization's information assets and infrastructure, and taking steps to reduce them to an acceptable level. RM involves three well-defined steps: identification, assessment and control, which will be the focus of the ontology we built to achieve our goal. By using this ontology, a web developer who is not a security specialist is able to identify and mitigate the risks related to the web application during the design phase, being this approach the main article's contribution.

The rest of the document is organized as follows. In Section 2 is presented the relation between ontology and InfoSec; in Section 3, the OWASP Top 10 Project used

to define the ontology is discussed; in Section 4 the solution proposal is showed, with the evaluation case results on Section 5. In Section 6, related work is listed and finally in Section 7 the conclusions and future work are presented.

2. Ontology and Information Security

Ontologies are being extensively used in different fields of study, primarily to organize information and formalize knowledge. It is receiving a special and growing attention from Computer Science professionals as experiences in IS development have shown to be related to long and expensive processes [Bai and Zhou 2011].

In Grubber (1993), there is an initial definition that is largely acceptable when ontology is related to Computer Science – "ontology is an explicit specification of a contextualization". This definition has been evolving over time. Another numerous times referenced definition is the one in Guarino (1998) – "an ontology refers to an engineering artifact, constituted by a specific vocabulary used to describe a certain reality, plus a set of explicit assumptions regarding the intended meaning of the vocabulary". Other definitions can be found, most of them complementing each other's meaning as the two mentioned before. For the purpose of this article, ontology will be a tool for InfoSec knowledge representation and organization.

An overview can be found at Almeida and Bax (2003), where ontologies are classified considering many different aspects like function, applicability, structure and contents. Depending on the level of abstraction, it can be divided in four groups: high-level, domain, task and application ontologies. High-level (or foundational) ontologies are based on very high generalization concepts, so it can be applied in different domains. Domain ontologies describe the vocabulary related to a defined domain, by specializing the concepts from foundational ontologies. In this article, we built a domain ontology, using InfoSec concepts as the domain, more specifically the OWASP Top 10 Project knowledge to secure web application development.

According to Raskin et al. (2001), the use of ontology in InfoSec can be summarized in one of two possible approaches: Based on Natural Language Processing - a reactive approach where an ontology is used to aid in the management of the huge volume of data provided by security logs and vulnerabilities alerts. The ontology is constantly updated with new information that is then used in attack analysis and prevention. In this method, the ontology is usually combined with other tools to provide a unified solution; and Based on Knowledge Representation – a proactive approach where an ontology is built to gather security domain concepts in order to help stakeholders to make security related decisions, according to organization's requirements. The ontology can be defined in different levels of abstraction and used for different objectives related to security activities.

Our proposal belongs to the second category – a proactive approach - information for risk mitigation in web applications will be modeled using domain ontology so application developers can use it during the design phase. The source of information for the ontology is a free open-source project that is detailed in Section 3 – the OWASP Top 10.

Considering the benefits of adopting the InfoSec domain ontology based on knowledge representation at the organization scope, we cite: i) creation of conceptual models to better understand security incidents; ii) support the interoperability between

different security tools; iii) creation of a standard for structuring security data, allowing terms to be mapped to the ontology; iv) use of automatic queries and inferences to filter ontology information. Moreover, the approach can also benefit from the basic ontologies capabilities of reuse and scalability [Almeida et al. 2010].

3. OWASP Top 10 Project

OWASP was established as an international organizational in 2004. It works as an open community dedicated to enabling organizations to conceive, develop, acquire, operate, and maintain applications that can be trusted [About OWASP 2014]. Because it is free from commercial and governmental relations, it is able to provide unbiased, practical, cost-effective information about application security, producing many types of materials in a collaborative and open way.

The InfoSec materials in OWASP are usually organized into independent open projects and currently there are almost 200 active projects. The TOP 10 Project is the most popular project at OWASP initiative, with the first version released in 2003. Subsequent releases happened in 2004 and 2007 with minor adjustments.

The 2010 OWASP TOP 10 version was the first to be prioritized by risks, as it is in the latest 2013 version. It lists the TOP10 security issues based on data from seven application security companies, which includes information from thousands of organizations and applications. The risks are rated using the rating scheme presented in Table 1, which is the basis for our ontology classes' definition.

Table 1. Rating scheme for OWASP TOP10 (from OWASP TOP 10 Project, 2014)

Threat Agents	Attack Vectors	Weakness Prevalence	Weakness Detectability	Technical Impacts	Business Impacts
App Specific	Easy	Widespread	Easy	Severe	App / Business Specific
	Average	Common	Average	Moderate	
	Difficult	Uncommon	Difficult	Minor	

The current security risks in the latest 2013 version are: A1 - Injection, A2 – Broken Authentication and Session Management, A3 – Cross-site Script (XSS), A4 – Insecure Direct Object Reference, A5 – Security Misconfiguration, A6 – Sensitive Data Exposure, A7 – Missing Function Level Access Control, A8 – Cross-site Request Forgery (CSRF), A9-Using Component with Known Vulnerability and A10 – Unvalidated Redirects and Forwards. Each risk is classified according to the rating scheme presented in Table 1, and the final document includes the following information: how to find out if applications are vulnerable to the risk, how to prevent the risk, risk examples and information references.

There are different initiatives to classify and discover vulnerabilities, most of them being supported by companies in the best interest to solve security issues. Organizations like SANS Institute (www.sans.org) and Mitre Corporation (www.mitre.org) provides the CWE (Common Weadness Enumeration) about the TOP 25 most dangerous software errors, listing the problematic practices in different categories [CWE/SANS TOP 25 2011]. We choose OWASP Top10 among other sources as it has a more didactic structure that can help developers and eases the ontology construction process. OWASP has more often updates and it is accepted by

many patterns as the minimal security requirements for web applications [PCI 2009]. The OWASP Top10 is also connected with other OWASP projects so they can be integrated in the future by using the ontology scalability property (example: OWASP Testing Guide and OWASP Risk Rating Methodology).

4. Proposal

This work proposal includes the OWASP TOP10 ontology that can be used by someone who is not a security specialist to evaluate what are the risks related to the web application being developed. For the ontology development, we use the method known as *101 Methodology* [Noy and McGuiness 2001]. The tool employed to build the ontology is the Protégé, a free open-source ontology builder from Stanford University (http://protege.stanford.edu/), and the language is the OWL (Web Ontology Language). The ontology uses definitions from the OWASP Top 10 scheme only, in order to have conceptual consistence, but it can be extended in the future to include new security aspects, like the ones related to network topology, for instance.

To illustrate the OWASP Top10 ontology, let us take the Risk A1 - Injection, where internal and external users are considered threat agents. Considering the rate scheme presented in Table 1, the attack vector is the own application interpreter that can receive malicious text-based attacks, classified to be easy to explore. The security weakness exists in SQL, LDAP, XPath and other technologies, prevalence is common and the technical impact for the organization is classified as severe.

For each of the terms presented in Section 3 for the OWASP TOP10 Project scheme, we created a class in the ontology, related to the super class 'Risk'. For each class, the data property and type presented in Table 2 were defined. Each data property is related to the scheme design presented in Table 1.

Table 2. OWASP TOP10 Ontology classes and data properties

Class	Data Properties	Type
Risk	riskName	String
	riskRange	String
ThreatAgent	threatAgentDescr	String
AttackVector	Exploitability	Easy / Average / Difficult
	attackVectorDescr	String
SecurityWeakness	Prevalence	Widespread / Common / Uncommon
	Detectability	Easy / Average / Difficult
	securityWeaknessDescr	String
Impact	technicalSeverity	Severe / Moderate / Minor
	technicalDescr	String
	businessDescr	String
Control	controlDescr	String

The relationships between classes were created using the object property feature of Protégé, based on InfoSec concepts for each of the classes retrieved from Whitman and Mattord (2011). A graphic view created with the *OntoGraf* built-in feature from the Protégé tool is shown in Figure 1.

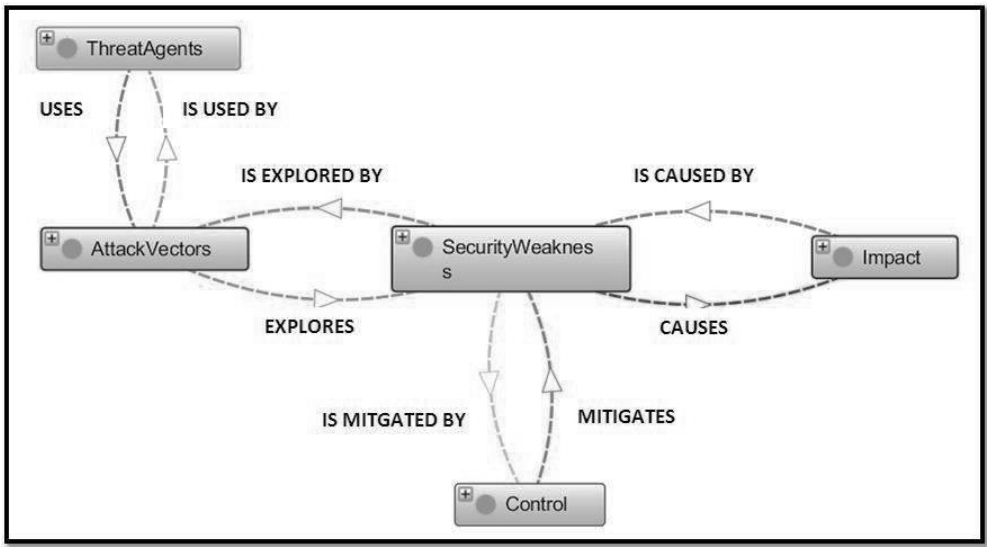


Figure 1. OWASP TOP10 Ontology classes diagram and relationships

Each individual of the ten-listed Risk (as presented in Section 3) is associated with an individual of each other class, according to the different object properties defined in Figure 1. For example, when Risk1 is instantiated as an individual, it is related to ThreatAgent1, AttackVector1, Control1, Impact1 and SecurityWeakness1. Following the model presented in Figure 1, ThreatAgent1 *uses* AttackVector1. AttackVector1 *explores* SecurityWeakness1. SecurityWeakness1 *causes* Impact1. Finally, Control1 *mitigates* SecurityWeakness1. An example of the individual AttackVector1 related to Risk A1-Injection is illustrated in Figure 2.

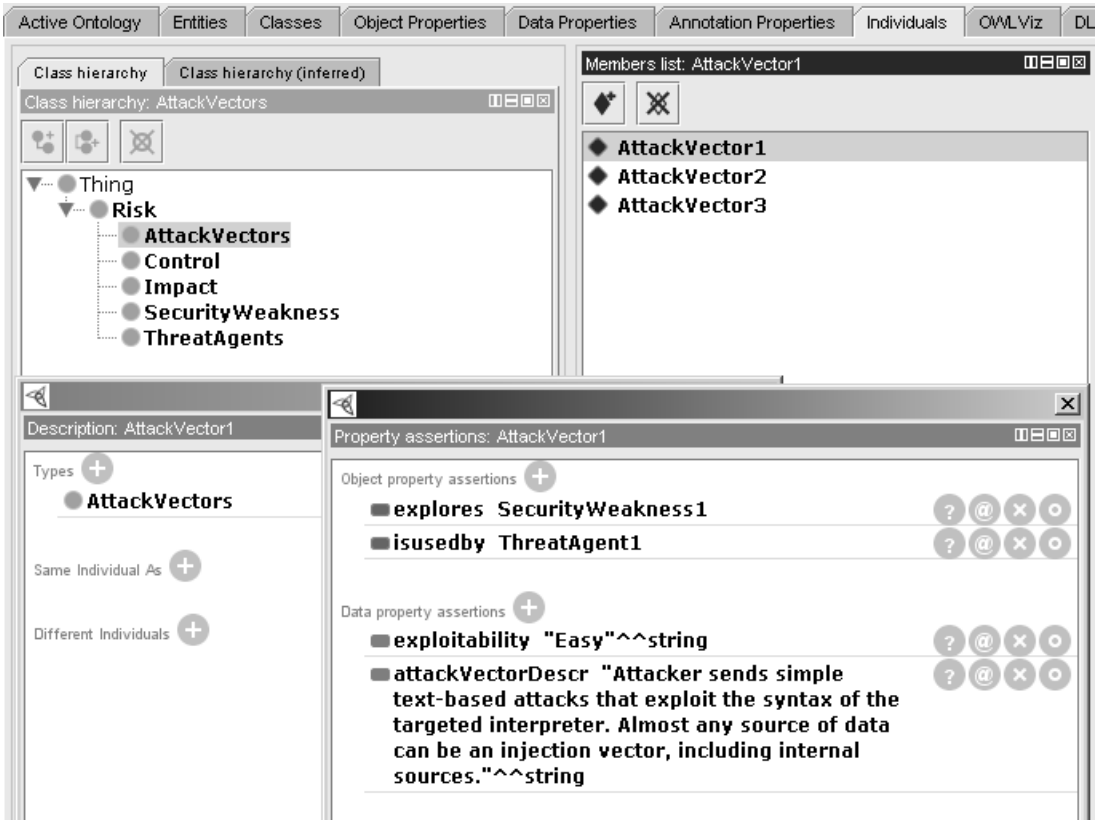


Figure 2. OWASP TOP10 Ontology AttackVector1 individual

The OWASP TOP10 ontology consistency was verified by using the Protégé Pellet 2 plugin [Clarck&Parsia 2011] as a reasoning engine. It checks for hierarchies, domains, ranges, conflicting disjoint assertions, and others issues through all ontology.

Once the ontology is consistent, information can be recovered through the use of a query language based on RDF (Resource Desciprtion Framework) such as SPARQL (SPARQL Protocol and RDF Query Language) [Pérez et al. 2006]. RDF is a query language designed to be applied on a set of "subject-predicate-object" triples, very similar to SQL (Structured Query Language). SPARQL is widely used as a powerful and complementary tool to ontologies, especially when used in the Semantic Web for knowledge representation.

The TOP 10 ontology is aligned with the basic preliminary criteria that must be taken into consideration before building an ontology - clarity, consistency and extensibility, as stated in Grubber (1993). Consider that the individuals defined in the ontology are the ten risks listed in Section 3, with data properties completed according to the OWASP TOP10 Project available information. It encompasses all the concepts available from the data used as the source of information in order to be applied in risk management activities of secure web application development as presented in Section 5.

5. The development of secure web application

The OWASP TOP10 Ontology was tested in a real case scenario during the development of a web application named ‘SMS Broadcast’. This application is designed to send text messages to registered organization’s employees based on selection filters, in an eventual emergency situation that requires employees to be notified immediately and at once. The text message is sent to a third-party bulk SMS provider that delivers it for the users according to the system’s configuration. The chosen application includes most of the commonly found classes of web-based systems like authentication, parameters communication and privilege level to name a few.

To comply with organization requirements, the chosen language for the application is classic ASP (Active Server Pages) and the publishing web tool is Microsoft SharePoint®. The web application was designed to have two modules:

- Module 1 – Users Registration – all organization’s users must be able to apply to receive text message in case of emergencies. When applying, they need to provide information for the following self-explanatory fields: *lastname*, *firstname*, *city*, *section* and *phonenumber*
- Module 2 – Message Broadcast – selected organization’s employees will have access to this page where they can see all registered users and select the ones they want to receive the message. For example, it should be possible to select all employees from *financial* section in a specific city and send a message to them only.

The evaluation of the proposed model uses two different scenarios.

Scenario 1 - the task to develop the web application was assigned to two web developers – DA1 and DB1. Both have similar work experience and knowledge, except that web developer DA1 has attended security courses that included OWASP awareness activities in the last year. As a result, two different web applications were created, one coded by developer DA1 and one coded by developer DB1.

Scenario 2 - two others developers, DA2 and DB2, are assigned the same task. Exactly as in Scenario 1, both have similar work experience and knowledge, except that web developer DA2 has attended the same security courses as web developer DA1. The main difference is that in this scenario, during the design process, web developers DA2 and DB2 are required to answer a questionnaire with twenty questions about the application requirements they will develop, and only after receiving the questionnaire output with potential risks they should be worried about, they should start the development process.

The questionnaire has only application specific questions that developers must know how to answer at the design phase. For example: (i) System contains information in transit coded in XML language, (ii) User can recover login information by using 'forgot password/forgot login' feature, (iii) There is sensitive data (PII – Personally Identifiable Information) stored in the system's database. For each question the answer can be "Yes", "No" or "N/A". The questionnaire was prepared by three application developers that are also security specialists with extensive hands on experience on InfoSec. They have been delivering OWASP and InfoSec training for others developers with different levels of experience and knowledge around the world in the last three years within the organization. Due to space limitation, it was not possible to include the complete questionnaire in the article, but it is available in *marciusmarques.com/owasp*.

Based on DA2 and DB2 answers to the questionnaire, an interface that was developed using Apache Jena (https://jena.apache.org/) informs the risks associated with the web application to the developer. The interface will use ARQ, a SPARQL processor for Jena, to query the Top 10 Ontology using SPARQL language. The objective is to advise the web developer about how to mitigate the risks in the design phase, in order to build a more secure code from the beginning of the development process. The proposal architecture is presented in Figure 3.

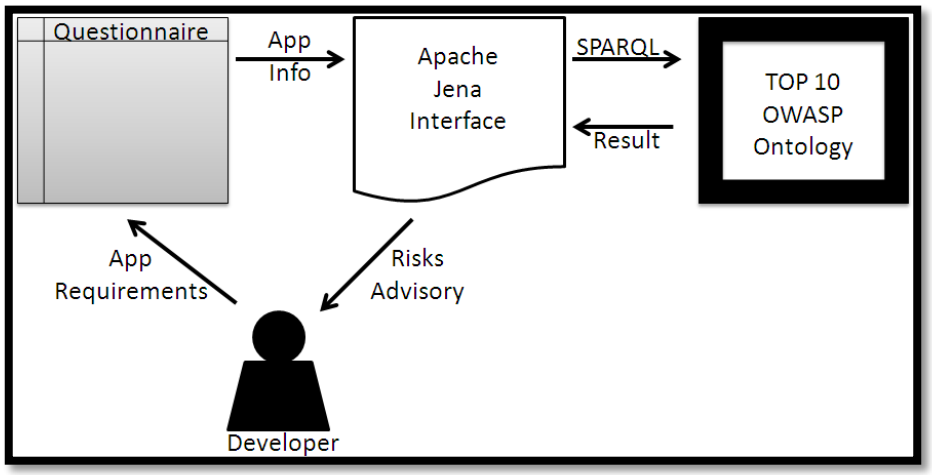


Figure 3. Proposal architecture

To illustrate the proposed model, assume the question "User can click a 'remember me' box so they don't have to re-authenticate" receives a "Yes" answer. The SPARQL query will associate the key-terms (remember me, session, authentication) with the risk A2 (Broken Authentication and Session Management) in the OWASP TOP 10 ontology class, as this information is available mainly at the data property 'securityWeaknessdescr' from risk A2, as presented in Table 2. As a result, control

measures related to this risk A2 available at the data property 'controlDescr' from the OWASP Top 10 ontology will be presented for the developer in the end of the questionnaire which can be used during the development process. Once concluded, web application security of each scenario are evaluated as detailed in Section 5.1.

5.1. Results analysis and discussion

To evaluate the web applications created, the four ASP codes developed (DA1, DB1, DA2 and DB2) for both modules were submitted to the organization's CnA (Certification and Accreditation) process. During the CnA, security specialists analyzes the system using many different criteria related to security aspects. For the purpose of this article, we will present the code review and penetration test score, which are part of the CnA process. Due to the organization's security policy, the score methodology and tools used cannot be detailed in the article. Based on the result of the CnA, the web application can be authorized to go to production when it attends the minimum security requirements. If it does not attend, it needs to be reviewed in a new submission after the changes suggested in the CnA report are performed.

The combination of the code review with penetration test is considered one of the most effective methods to be used during the assessment of the security of an application [Curphey and Arawo 2006]. A comparison and benefits of both can be found in an OWASP conference presentation - The Strengths of Combining (2009).

The metric used by the organization requires that the system achieve a maximum score of six in the CnA security assessment in order to have its implementation authorized. The higher the security risks, the higher the score. The results for the codes submitted and a summary of the scenarios can be found in Table 3.

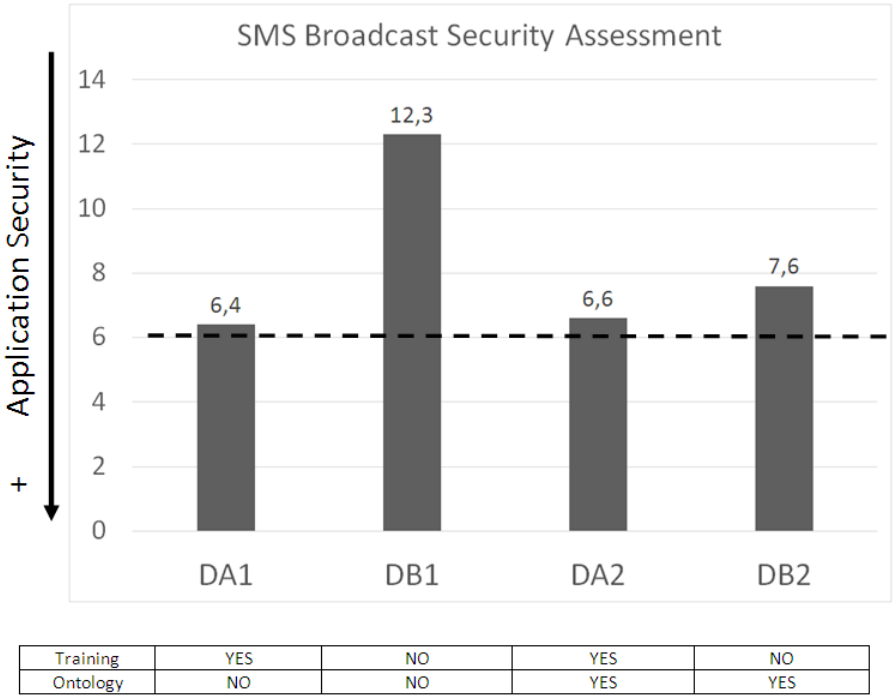
Table 3. Proposed model evaluation scenarios and results

Web Developer	DA1	DB1	DA2	DB2
Development Experience	High	High	High	High
Security awareness	High	Low	High	Low
TOP 10 Ontology use	No	No	Yes	Yes
Development Outcome	DA1 code	DB1 code	DA2 code	DB2 code
Risk assessment CnA score	6,4	12,3	6,6	7,6

From the results, the most vulnerable application was the one developed by the developer DB1 that did not have specific security training about web application development and did not use the proposed OWASP TOP 10 ontology to evaluate the risks associated with the application requirements. Web applications from developers DA1 and DA2, although need to be reviewed as neither reached the minimum score of six, are the most secure ones. Both were developed by someone with a security awareness background, having DA2 using also the OWASP Top 10 ontology during the design phase. Finally, DB2 developer had no security awareness training and by means of the proposed ontology achieved an acceptable score of 7,6 in his web application.

Web developer DA2 and DB2 reported that it was useful to use the questionnaire output before the development coding step had started, as it was helpful to identify in advance security issues they were not considering in the beginning. Both agreed that the information provided by the ontology influenced in the development process in a positive and efficient way. Positive because risks were avoided and

efficient because it was not required going through long reading and analysis activities to identify application related issues. A graphic view of the result is presented in Figure 4.



Training	YES	NO	YES	NO
Ontology	NO	NO	YES	YES

Figure 4. Results overview

6. Related Work

The use of domain ontologies to aid InfoSec related activities can be found with distinct objectives under both approaches: reactive and proactive (Section 2).We analyzed some related work where InfoSec domain ontologies are created with different levels of details using different source of information, aiming to build a knowledge repository that can be useful for the organization. The main challenge of all proposals is how to use the security information efficiently in the organizations.

For instance, in Almeida et al. (2010) and de Azevedo et al. (2007), the ontologies were created mainly to classify the information, in order to point out what needs to be protected and how. This approach is in line with business management activities, where SPARQL can be used to identify more high-level security concerns.

SPARQL querying domain InfoSec ontologies was also used in da Silva et al. (2011) and Martimiano and Moreira (2005). On the later, security incident data was used as the source of information to validate the ontology, which can be combined with other approaches to produce interesting results. This type of approach was proposed in Silva and Ellwanger (2012), where *CODI* ontology is presented. It groups an InfoSec ontology with influence diagrams to create a methodology that could facilitate the dissemination of information and the accumulation of knowledge among stakeholders, but the implementation is suggested as a future work.

A reactive approach that uses ontology can be found in Razzaq et al. (2009). It presents an intrusion detection system developed using an InfoSec ontology, aiming to detect zero-day attacks in the application layer, by comparing the event information with a knowledge base that is updated constantly. The idea presented in Rosa et al. (2011) is similar, where an ontology is suggested to detect XML Injection attacks via web services. In both works, the ontology is applied for scanning and assessing the vulnerabilities when it already exists in the web applications.

The contribution of our work is different from previous mentioned in regards to three main concepts. First, our target audience is well defined, it is focused on web application developers; second, the timing of applying the ontology is the design phase of the software development cycle; third, it uses OWASP Top10 Project as the source to build the knowledge about InfoSec. It is similar to others in the sense that it is based in the knowledge representation approach discussed in Section 2 and it uses SPARQL to query the ontology. However, it does not require any pre-knowledge about SPARQL from the user in order to achieve the results.

7. Conclusion and Future Work

In order to be more efficient, effective and responsive to remain competitive, organizations need to be up to date with networks and computer based IS. Due to the features provided by web applications, it is becoming the favorite choice for organizations in order to offer customers a modern solution to perform business. However, this solution charges a price related to needed security concerns, which are often not in the top priority list of organization's stakeholders. This happens not as a deliberated choice, but because InfoSec is a complex and expensive business process.

The main objective of InfoSec in web applications is to reach a balance between accurate access and secure information. IS must hold data that has to be available for authorized users. On one side, it is possible to grant access to everything to everyone. On the other side, it is possible to remove all computers from the network to prevent unauthorized accesses, losing of course all benefits from information sharing. InfoSec is somewhere between these two utopic realities, with business processes that exists to attend users and system's needs without giving up to all defined security requirements at the same time.

Application developers play an important part in the line of defense of web-based systems, as the majority of current explored vulnerabilities are a consequence of development activities executed without the necessary security concerns. "The security issues are not being adequately considered during the development process, both by lack of knowledge as by the pressure caused by tight delivery schedules" [Uto and de Melo 2009].

Based on this scenario, this article presented an ontological approach to produce more secure web applications. It relies on the application developer common knowledge about the system he is about to develop, in order to provide information on the security risks related to this application. Our goal is to identify the good security practices necessary to be applied in order to mitigate the risks related to the application that is being developed. The burden of training and research is abstracted from the web developers as an ontology is used to provide information about security concerns.

The proposal is tested in a real case scenario, where four developers are assigned the task to build a simple web application with commonly used features for this type of system. They have different knowledge about security aspects and two of them are using the proposed OWASP Top 10 Ontology. After being submitted to the organization's risk analysis CnA process, it was found that the use of the ontology was useful to produce more secure web application. Similar risk scores were achieved by a web developer (DA1) with many hours of security awareness courses when compared to a web developer (DB2) with no security training but making use of the ontology for the same system.

However, none of the four applications could be implemented in the first version, as the minimum score required by the organization was not obtained. This emphasizes the importance of risk assessment activities in the process of web application development, which is in line with the presented proposal of executing it during the design phase. Moreover, both developers that used the ontology reported the benefits of it to mitigate risks in web applications.

The proposed ontological approach does not require that the application developer becomes a security specialist in order to produce a more secure system. It can also benefit from the advantages offered by knowledge representation using ontologies.

As future work, we suggest the use of different inference tools compatible with the OWL standard to query the ontology; the integration of the TOP 10 ontology with other InfoSec domain ontologies to produce other results and the use of other sources of information for ontology instantiation according to the organization's requirements (e.g. ISO27001). In the meantime, we are executing a similar test with another more complex web application to compare the results with the ones from this article. We also intend to expand the ontology by using the information provided in other OWASP Project – SAMM (Software Assurance Maturity Model) [OWASP SAMM Project 2013]. We believe a more comprehensive model can have even more significant results.

8. References

About OWASP. (2014). Retrieved June 10, 2014, from https://www.owasp.org/index.php/About_OWASP

Almeida, M. B. (2007). Aplicação de ontologias em segurança da informação. Diretoria da Prodemge. Revista Fonte, 4(7),75-83.

Almeida, M. B., and Bax, M. P. (2003). Uma visão geral sobre ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção. Ciência da Informação, Brasília, 32(3), 7-20.

Almeida, M. B., Souza, R. R., and Coelho, K. C. (2010). Uma proposta de ontologia de domínio para segurança da informação em organizações: descrição do estágio terminológico. Informação & Sociedade: Estudos, 20(1).

Bai, X., and Zhou, X. (2011). Development of Ontology-Based Information System Using Formal Concept Analysis and Association Rules. In Advances in Computer Science, Intelligent System and Environment (pp. 121-126). Springer Berlin Heidelberg.

Clark and Parsia (2011). “Pellet: OWL 2 Reasoner for Java”. Retrieved June 18, 2014, from http://clarkparsia.com/pellet/protege/

Curphey, M., and Arawo, R. (2006). Web application security assessment tools. Security & Privacy, IEEE, 4(4), 32-41.

Cyberattacks. (2013). Retrieved June 10, 2014, from http://www.cnet.com/news/cyberattacks-account-for-up-to-1-trillion-in-global-losses/

CWE/SANS TOP. 25 (2011). Retrieved June 11, 2014, from http://www.sans.org/top25-software-errors/

da Silva, B. A., Ellwanger, C. (2012). CODI Methodology for Managing Security in Web Application Development.

da Silva, P. F., Otte, H., Todesco, J. L., and AO, F. (2011). Uma ontologia para gestão de segurança da informação. In: IV Seminário de Pesquisa em Ontologia no Brasil (p. 141).

de Azevedo, R. R., Almeida, M. J. S., and Barros Filho, C. (2007). Uma Ontologia Genérica de Segurança Aplicada a Gestão de Processos de Negócios. In: I Workshop Brasilieiro em Gerenciamento de Processos de Negócios (WBPM).

Dhillon, G., and Backhouse, J. (2000). Information System Security Management in the New Millennium. Communications of the ACM, 43(7), 125.

Guarino, N. (Ed.). (1998). Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy (Vol. 46). IOS press.

Gordon, L. A., and Loeb, M. P. (2002). The economics of information security investment. ACM Transactions on Information and System Security (TISSEC), 5(4), 438-457.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. Knowledge acquisition, 5(2), 199-220.

Jacobson, I., Booch, G., Rumbaugh, J., Rumbaugh, J., and Booch, G. (1999). The unified software development process (Vol. 1). Reading: Addison-Wesley.

Key Findings. (2013). Retrieved June 11, 2014, from http://www.pwc.com/en_US/us/increasing-it-effectiveness/publications/assets/us-state-of-cybercrime.pdf

Martimiano, L. A., and Moreira, E. S. (2005). Using ontologies to assist security management. In Proceedings of the 8th International Protégé Conference.

Noy, N. F., and McGuinness, D. L. (2001). Ontology development 101: A guide to creating your first ontology, from http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html

Only 10%. (2005). Retrieved June 10, 2014, from http://www.prnewswire.com/news-releases/only-10-of-web-applications-are-secured-against-common-hacking-techniques-58703902.html

OWASP TOP 10 Project. (2014). Retrieved June 10, 2014, from https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

OWASP SAMM Project. (2013). Retrieved June 10, 2014, from https://www.owasp.org/index.php/Category:Software_Assurance_Maturity_Model

PCI (2009). Payment Card Industry (PCI) Data Security Standard. Requirements and Security Assessment Procedures, version 1.2.1. PCI Security Standards Council.

Peltier, T. R. (2013). Information Security Policies, Procedures, and Standards: guidelines for effective information security management. CRC Press.

Pérez, J., Arenas, M., and Gutierrez, C. (2006). Semantics and Complexity of SPARQL. In The Semantic Web-ISWC 2006 (pp. 30-43). Springer Berlin Heidelberg.

Raskin, V., Hempelmann, C. F., Triezenberg, K. E., and Nirenburg, S. (2001). Ontology in information security: a useful theoretical foundation and methodological tool. In Proceedings of the 2001 workshop on New security paradigms (pp. 53-59). ACM.

Razzaq, A., Ahmed, H. F., Hur, A., and Haider, N. (2009, February). Ontology based application level intrusion detection system by using bayesian filter. In Computer, Control and Communication, 2009. IC4 2009. 2nd International Conference on (pp. 1-6). IEEE.

Rosa, T. M., Santin, A. O., and Malucelli, A. (2011). Uma Ontologia para Mitigar XML Injection. In: XI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg), p. 1-14.

The Strengths of Combining Code Review with Application Penetration Testing. (2009). Retrieved June 12, 2014, from https://www.owasp.org/index.php/The_Strengths_of_Combining_Code_Review_with_Application_Penetration_Testing

Uto, N., and Melo, S. P. (2009). Vulnerabilidades em Aplicações Web e Mecanismos de Proteção. Minicursos SBSeg.

Weske, M. (2007). Concepts, Languages, Architectures (Vol. 14). Berlin: Springer-Verlag. New York, Inc., Secaucus, NJ, United States.

Whitman, M., and Mattord, H. (2011). Principles of information security (3rd ed). Course Technology Press, Boston, MA, United States.

