A Knowledge-Oriented Approach to Security Requirements Engineering for E-Voting System

P. Salini
Department of Computer Science and Engineering
Pondicherry Engineering College
Pillaichavady
Puducherry-605014, India

S. Kanmani
Department of Information Technology
Pondicherry Engineering College
Pillaichavady
Puducherry-605014, India

ABSTRACT
In this paper, we introduce a knowledge-oriented approach for the Security Requirements Engineering phase for developing E-Voting System. The knowledge acquired through the process of eliciting and analyzing secure E-Voting System is represented in the form of UML models; which can be made available to future developers and the dependency towards security experts can be reduced. In this paper we present a set of security requirements and security requirements patterns that were developed based on the aforementioned approach. Security requirements for modelling have been identified by following the Model Oriented Security Requirements Engineering framework for web applications. The security requirements have been designed into security requirements patterns for creating security requirements ontology for an E-Voting System. The ontology allows all concepts of importance and their relationships to be identified. The paper also compares the approach with other relevant methods in the Security Requirements Engineering phase for developing secure applications.

General Terms
Software Engineering, Requirements Engineering, Security Requirements Engineering, Web Application Security

Keywords
Security Requirements, Security Requirements Engineering, Security Requirements Patterns, Ontology, E-Voting System, Knowledge-Oriented

1. INTRODUCTION
Security requirements engineering involves identification and specification of security requirements in an efficient and effective manner. This is a knowledge oriented security requirements analysis task for developing a web applications are typically performed by security experts. In most cases, however, solutions provided for a specific application is not documented in a form that would allow their application in other similar applications; thus, expert knowledge acquired through security analysis remains tacit.

Nevertheless, in applications where security is a critical factor, such as the E-Voting System, applying ad-hoc solutions may have hazardous effects on the trustworthiness of the applications. There is a need for a systematic method of recording and re-using security requirements in such critical application systems. This paper addresses the issue of identifying and fulfilling security requirements for critical applications, as those in the E-Voting System. The approach followed herein takes a knowledge management stance, using ontology as a vehicle for managing different threats, vulnerabilities and their corresponding security requirements. Specifically, an ontology is developed that can be applied by E-Voting application developers.

The goal of this paper is to show how a long-lasting difficulty in web application security development can be addressed through the use of security requirements ontology. Moreover, the approach described in this paper allows expert knowledge to be used by web application developers who otherwise would have to depend on security experts to develop security requirements. To achieve knowledge, the paper identifies set of security requirements; each encapsulating a specific security problem, security goals and other information concerning the web application. All the information allows developers to develop secure web applications. In this paper we pertain to applications in the Electronic Voting System. The paper is structured as follows: Section 2 provides an overview of related work, section 3 presents the security requirements analysis for E-Voting System. Based on this analysis, a set of common security requirements is identified. Section 3 also describes the mapping of a subset of these requirements to security Objectives, assets, threats and vulnerabilities in the form of a security requirements pattern. Section 4 describes the ontology that has been proposed in order to model security requirements and section 5 discussing about the approach proposed in this paper. The last section concludes the paper with suggestions for further research.

2. RELATED WORKS
Identifying and accommodating security requirements for applications have been the research issue for quite a long time now. Siponen [3] provided an in-depth analysis and classification scheme, underlines the difficulty to integrate security into the overall information system development process as a major drawback of modern information system security development methods. The Object-Oriented approach models are compiled in the Unified Modeling Language (UML). UML is used for modeling and specification of the requirements for software development. There are three kinds: Structural diagrams, Behavior diagrams, and functional diagrams. Since UML became a de facto standard for modeling, many efforts have been made to extend UML diagrams for security design. Such an extension of the UML can be done mainly in two ways UMLsec [5] and SecureUML [7]. An aspect-oriented framework for incorporating security requirements into software is the Acegi Security System. Acegi uses these aspect-oriented features to offer support for authentication and authorization services. Raskin et al. [4] advocate an ontological semantic approach to information system security design. Both approaches and their resources, the ontology and lexicons, are borrowed from the field of

natural language processing and adjust to the needs of the solutions to common security requirements. They are an effective method [4] for cataloguing and reusing existing security knowledge, for documenting software with security requirements. Security patterns are specialized design patterns [4]. Documentation as given in [8] may have following headings: pattern name, classification, applicability, motivation, "also known as", implementation, consequences, sample code, known uses and related patterns.

Mouratidis et al. [6] have presented extensions to the well-known Tropos ontology to enable it to model security issues of agent-based systems. They [6] have introduced the concept of security constraints that allow functional, non-functional and security requirements to be defined together and clearly distinguished. They [6] argue that it's easy to identify security requirements at the earliest requirements stage and propagate them up to the implementation stage using their method. In [12] Andreas Ekelhart et al. simulated threats to corporate assets, considering infrastructure. They also [12] state that effective countermeasure and costs can be calculated quickly without expert knowledge. The main objective of [13] is to amalgamate and extend the security ontologies proposed in Problem Frames and Secure Tropos. A. Chikh et al. In [14] proposes a framework for building a part of Software Requirement Specification related to information security requirements using ontologies. Such a framework [14] allows ensuring information security requirements traceability and reuse.

Karyda et al. [15] uses OWL to propose security ontology with which to develop secure applications. It captures the security knowledge of experts to support the communication between security experts, users and developers and developers use it to include security requirements as well as to support design choices. The proposed ontology is formed of "assets", "countermeasures", "objectives", "persons" and "threats. They validated the defined ontology using nRQL queries. In [16] security ontologies have been proposed. They [16] proposed an analysis and a typology of existing security ontologies and their use for requirements definition. Though many works have been done in developing security ontology, the ontologies are not prepared for reusing and extending. There is also no ontology for security requirements engineering and for security requirements.

3. IDENTIFYING SECURITY REQUIREMENTS FOR E-VOTING SYSTEM

All Electronic Voting System, or E-Voting System, aims to improve the election process in India and to achieve real security system. A typical E-Voting System, as the one considered for the purpose of this paper, provides a secure electronic voting solution for political elections in India. The E-Voting System should allow the voters to cast their vote on chosen candidate, check that only valid users are logging into the voting system, create a database to store votes, provide user information on the system, enable the system to tally votes cast according to candidate voted for, create administration to manage the election system effectively and generate voting results for the administrator to analyze and declare the results. In E-Voting System, security covers the protection of data integrity, anonymity, availability, disclosability authenticity uniqueness, accuracy, transparency, Non-coercibility, confidentiality and privacy. In compliance with personal data protection regulations, the voters must identify themselves to be entitled to vote, admin are identified and have functional access on the system; votes must not be security domain. Security patterns [4] represent standard associated with voter identity; the vote is secret; each vote is recorded as intended, system cannot be re-configured during operation and operator with records of good behavior; are critical factors towards achieving users' trust and acceptance of E-Voting System. A set of security requirements identified using our improved version of [1], MOSRE Framework for web applications [2] are presented in the following paragraph; these security requirements should be fulfilled in order to develop, reliable and secure E-Voting System.

3.1 Eliciting Security Requirements for an E-Voting System

In the E-Voting System, it is important to ensure that the data like vote, voter's details are kept secret and available to them who are authorized to access it. This is often not easy to achieve since many data related to a voter are delivered or processed through public networks, particularly through the Internet. In our method of analysis we consider threats and vulnerabilities that originate from deliberate actions and aim to violate the fundamental security objectives like confidentiality, integrity and availability as well as secrecy. Adopting our MOSRE Framework for web applications [2], the security requirements that should be considered in the process of developing a secure E-Voting System is identified.

To achieve real security, the entire Software Development Life cycle must be security conscious. A Secure Software Development Life Cycle process (SSDLC) takes threats, vulnerabilities and mitigation into consideration from the initial requirements analysis and through development, testing, deployment and maintenance. We will apply our framework to Elicit Security Requirements for an E-Voting System. Our framework follows the spiral process model which is iterative and covers all phases of Requirements Engineering. The Model Oriented Security Requirements Engineering (MOSRE) Framework for Web Applications covers 16 steps. The client with respect to this project is election department. Fig 1 depicts the high level Network Architecture diagram for an E - Voting System.

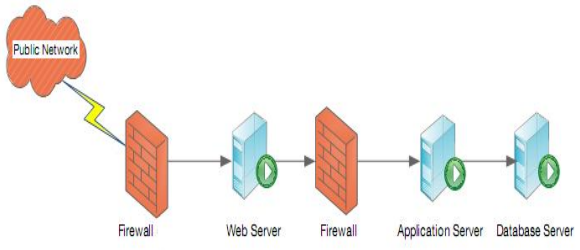

Fig 1: High level Network Architecture diagram for E-Voting System

With the details given by the client and with Architecture diagram, the requirements analyzing team sketched the workflow for the E-Voting process in 3 phases. Each of them consists of one or more processes. In each process we have identified threats, vulnerabilities and assets. With the architecture, threats, vulnerabilities and assets we have identified security objectives and security requirements.

Phase 1: Pre-Voting
- Voters Registration Process
- Submission of Candidates Details
- Processing Candidates Details

Phase 2: Voting
- E-Voting

Phase 3: Post-Voting
- Approval of votes and ballots
- Registration of P-votes results
- Counting E-votes & P-votes
- Declaration of Results

In this section we consider only phase 2 process of E-Voting System and identified the following security requirements using MOSRE framework:

SR 2.1.1 It should not be possible to insert, delete or modify any votes without authorization by the Election department personnel.

SR 2.1.2 It should be ensured that the election system presents an authentic ballot to the voter.

SR 2.1.3 The solution for voting in an uncontrolled environment should issue a message to inform the voter whether the vote has been successfully cast.

SR 2.1.4 The Election System should provide the e-voter with 'end-to-end' proof that the cast vote is received and recorded.

SR 2.1.5 The election system should ensure that the voter's choice is accurately represented in the vote and that the sealed vote is successfully stored.

SR 2.1.6 To allow for a delay in messages, the election system should remain open for a configurable period of time after the end of the polling.

SR 2.1.7 The voter can, at any time cast vote, abort his polling process till he submit his vote without losing his right to vote due to timeout or errors during communication.

SR 2.1.8 A voter should only be able to vote in contests that he/she is entitled to vote in.

SR 2.1.9 The E-Voting components of the election system should be configurable to authenticate for every vote or session.

SR 2.1.10 The voter authentication should expire after an idle period. The length of the idle time-out period should be configurable.

SR 2.1.11 The E-Voter's decision or the display of the e-voter's choice should be destroyed after the vote has been cast.

SR 2.1.12 It should not be possible during transfer in the network, to alter, delete or add vote records.

Table 1 gives a set of security requirements with regard to E-Voting System mapped to the security objectives. The set has been built using the threat and asset analysis.

Table 1. Security Requirements mapped to Security Objectives

Security Requirements	Security Objectives
SR 2.1.1	Integrity
SR 2.1.2	Availability
SR 2.1.3	Accuracy
SR 2.1.4	Accuracy
SR 2.1.5	Reliability
SR 2.1.6	Availability
SR 2.1.7	Uniqueness
SR 2.1.8	Identification
SR 2.1.9	Authentication
SR 2.1.10	Authentication
SR 2.1.11	Confidentiality
SR 2.1.12	Integrity

With this set of security requirements, regard to web-based E-Voting Systems, we will build ontology for E-Voting System, which can be used by the developers to model and implement security requirements for an E-Voting System.

3.2 Security Requirements Pattern in Requirements Engineering Phase

In security requirements engineering phase, we first decide what to protect called assets, then we analyze the security goals, threats and vulnerabilities to the assets. There are several analysis process patterns in Requirements Engineering phase [9]. We calculate the risk and categorize and prioritize the security requirements and finally we specify these security requirements into security requirements pattern. Security requirements Patterns use existing, well-proven pattern in software development and help to promote effective secure software development practices. The use of security requirements patterns increases the security awareness and assists software developers in incorporating security mechanisms and techniques into the software development process, by using known practices in recurring security problems. Hence, security requirements patterns assist developers in adopting effective security solutions and in using them in the intended way.

The structure of the security requirements patterns that is proposed herein comprises the following attributes: a) name of the security requirements pattern, b) overview of the security problem the pattern addresses, c) general description of the security requirements that meets the identified problem, d) security objectives it addresses, e) assets protected when the security requirements is applied, f) threats addressed, g) vulnerabilities encountered, h) the impact of the threats and vulnerabilities to the web based applications and i) related security requirements patterns in conjunction with the main security requirement pattern. We propose to build security requirements ontology to derive important security requirements with attributes and the relations among them are described in the next section.

3.3 Security Requirements Patterns for E-Voting System

The Tables 2 and 3 present the security Requirement patterns based on the structure given in the subsection 3.2. The developed Security Requirements patterns have all possible security requirements [10] that can be used during the process of designing and developing an E-Voting System. They form a representative set of possible requirements that illustrates what security requirement patterns should be implemented by the application developers. In this section we will give the Security Requirements Patterns for some of the security Requirements given in the subsection 3.1.

Table 2. Security Requirement Pattern 1

Pattern Name	Vote Confidentiality
Problem	Secure Vote exchange over public networks.
Security Requirements	The E-Voter's decision or the display of the e-voter's choice should be destroyed after the vote has been cast
Security Objectives	Confidentiality, Integrity, Privacy
Assets	Vote, Voters credentials, Election System, E-Voting application, Voters System
Threats	A malware accesses, Man-in-the-middle, Voter Impersonation

Vulnerabilities	File Manipulation, inadequate protection of user accounts
Impact	High
Related Patterns	Authentication, Integrity

Table 3. Security Requirement Pattern 2

Pattern Name	Voter Authentication
Problem	Each actor have access to each component of the e-voting application based on their roles and specific privileges
Security Requirements	The E-Voting components of the election system should be configurable to authenticate every vote or session.
Security Objectives	Confidentiality, Authentication, Privacy, Integrity
Assets	Vote, Voters credentials, Election System
Threats	Authentication Token Forgery, Man-in-the-Middle Voter Impersonation, Identity spoofing, Unauthorized access
Vulnerabilities	Weak authentication method, insecure credential transfer, weak passwords
Impact	High
Related Patterns	Confidentiality, Integrity

In Table 2 and 3 we have given the Security Requirements Patterns for vote Confidentiality and Voter Authentication with security requirements, assets, threats, vulnerabilities, and impact. These patterns can be used to develop the ontology for the specification of security requirements in security requirements engineering phase. In the next section we discuss on our proposal for security requirements ontology.

4. SECURITY REQUIREMENTS ONTOLOGY FOR DESIGNING SECURITY REQUIREMENTS PATTERNS

This is a proposal to initiate the design of the ontology for Security Requirements Patterns. We use Ontology for specification of Security Requirements in Security Requirements engineering phase.

Security Requirements patterns can be used by developers to fulfill the security requirements of an E-Voting System. The security Requirements patterns can be developed by security experts for different web applications and can be applied for different domain. The design of the security requirements patterns is followed with the development of security requirements ontology. The development of the security Requirements ontology can be carried out in the following phases:

In the first phase, a set of questions were found, the answers and information we would like to receive when using the security requirements ontology are prepared. Our focus was to build the security requirements ontology in the area of E-Voting System. This process enabled us to identify the important concepts and terms within the e-voting domain. The second phase is the formation of ontology classes. The terms used as ontology classes are the following: Stakeholder, Security Objective, Threat, Security Requirements, Assets, Vulnerabilities, Security Requirements Pattern, impact, severity and web application. The third phase involved drawing the relations among the ontology classes and the properties used to represent the relations are use, have, requires, is vulnerable to, implemented in, protects, mitigated by, provide, damage, affects, exploited by, addresses, assessed and part-of. These phases can be repeated several times, and the ontology can be validated using questions formed earlier in phase one. The iterations can be ended only when the system could provide valid answers. With the idea of [10, 11] we propose our security requirements ontology for E-Voting System.

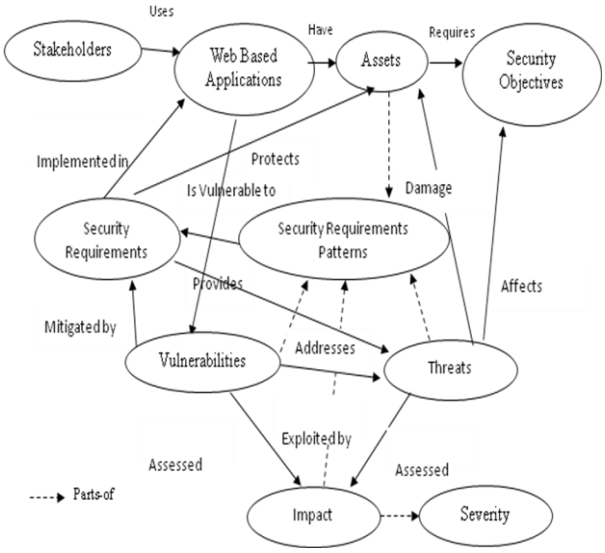

Fig 2: Security Requirements Ontology

In the ontology presented in the Fig 2, the concept of a Security Requirements Patterns is a representation of the Security Requirements Patterns and is connected with the concept of Security Requirements with a provide relationship: each security Requirements pattern provides a specific set of Security Requirements. In practice, each Security requirements pattern is matched with a set of Security Requirements during the ontology instantiation. A Security Requirements Pattern is defined as a set of Asset, Vulnerability, threats and impact. In this way, one can start from the security objectives, find the Security Requirements Pattern that match them and, thus, choose specific Security Requirements. In this way, the high level security requirements and objectives can be fulfilled by implementing in the web applications by the developers. The ontology is used to model higher level class diagram and used in the design phase to reach a low level of abstraction of a class diagram. The Protégé tool can be used for developing the security requirements ontology.

5. DISCUSSION

The approach described in this paper aims to exploit accumulated knowledge and expertise in the field of security requirements for the benefit of the application developer. Our knowledge-Oriented approach uses the idea of Siponen [3] since the use of patterns allows security solutions to be incorporated in the application development process. Compared to other approaches, the use of security requirements patterns coupled with the development of security requirements ontology provides better flexibility. Furthermore, this approach can provide solutions for all types of security requirements that may be relevant to an e-voting application.

The Acegi system, for example, is oriented towards addressing security requirements related solely to authentication and authorization. The approach described in this paper provides the features like it captures the knowledge

of security experts and other system stakeholders and aims to use it to address the needs of the software developer. Other approaches, such as those proposed in Jurjens [5] and Raskin et al. [4] are meant to be used by security experts, not software developers. It employs ontology to model the related concepts and the relationships among them. The developed ontology includes concepts, such as the security requirements patterns, at a higher abstraction level, than that of other approaches. The security requirements ontology can be instantiated in different contexts, besides the e-voting domain described in this paper. Ontology can model details that are more generic, such as the one proposed by Raskin et al. [4].

6. CONCLUSION

The aim of this paper is to illustrate the way security requirements patterns can facilitate the process of identifying security requirements. We have proposed to develop and validate an ontology that includes the major related concepts and the relationships that connect them. Based on this ontology, we can design and develop set of security requirements for applications that provide electronic voting. Not all identified security requirements have been mapped to the security objectives and we need to develop security requirements ontology. These works are objective for the future. Moreover, further elaboration is needed with regard to standardize the security requirements patterns for web applications.

7. REFERENCES

[1] P. Salini and S. Kanmani. "A Model based Security Requirements Engineering Framework applied for Online Trading System". In *Proceedings of the IEEE International Conference on Recent Trends in Information Technology (ICRTIT 2011),* India, pp. 1195-1202, June 3-5, 2011.

[2] P. Salini and S. Kanmani. "Model Oriented Security Requirements Engineering (MOSRE) Framework for Web Applications". In *Proceedings of the Second International Conference on Advances in Computing and Information Technology (ACITY 2012),* India, July 13 - 15, 2012, Vol.2 and in *Advances in Intelligent and Soft Computing book Series*, Vol.177, pp.341-353.

[3] Siponen M. "Analysis of modern IS security development approaches: towards the next generation of social and adaptable ISS methods", *Information and Organization*, Vol. 15 (4), pp. 339-375.

[4] Raskin V., Hempelmann C., Triezenberg K., and Nirenburg S. "Ontology in Information Security: A Useful Theoretical Foundation and Methodological Tool". In *Proceedings of the New Security Paradigms Workshop*, 2001, New York, USA, ACM.

[5] Jurjens J. "Towards development of secure systems using UMLsec". *Lecture Notes in Computer Science*, Vol.2029:187, 2001.

[6] Mouratidis H., Giorgini P., and Manson G. "An Ontology for Modelling Security: The Tropos Project". In *Proceedings of the KES 2003 Invited Session Ontology and Multi-Agent Systems Design (OMASD'03)*, 2003, University of Oxford, United Kingdom.

[7] Basin D., Doser J., and Lodderstedt T. "Model driven security for process-oriented systems". *In Proceedings of the Eighth ACM Symposium on Access Control Models and Technologies (SACMAT '03)*, Como, Italy, ACM.

[8] Gamma E., Helm R., Johnson R., and Vlissides J., Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995.

[9] N. Yoshioka, H. Washizaki and K. Maruyama. "A survey on security patterns". *Progress in Informatics*, No. 5, pp. 35-47, 2008.

[10] S. Dritsas, L. Gymnopoulos, M. Karyda, T. Balopoulos, S. Kokolakis, C. Lambrinoudakis and S. Katsikas. "A knowledge-based approach to security requirements for e-health applications". *The electronic Journal on Emerging Tools and Applications,* In the Special Issue: "*Emerging Security Paradigms in the Knowledge Era"*, Volume 2, issue 1, 2006.

[11] Andrew Simmonds, Peter Sandilands and Louis van Ekert. "A Ontology for Network Security Attacks". *Lecture Notes in Computer Science*, 2004, Vol.3285/2004, pp. 317-323.

[12] Andreas Ekelhart, Stefan Fenz, Markus D. Klemen, and Edgar R.Weippl. "Security Ontology: Simulating Threats to Corporate Assets". In *Aditya Bagchi & Vijayalakshmi Atluri, ed.,'Information Systems Security (ICISS'06)*, Springer, Kolkata, India, pp. 249-259.

[13] Fabio Massacci, John Mylopoulos, Federica Paci, Thein Thun Tun and Yijun Yu. "An Extended Ontology for Security Requirements". *Advanced Information Systems Engineering Workshops, Lecture Notes in Business Information Processing,* 2011, Vol.83, Part 10, pp.622-636.

[14] Azeddine Chikh, Muhammad Abulaish, Syed Irfan Nabi and Khaled Alghathbar. "An Ontology Based Information Security Requirements Engineering Framework". *Communications in Computer and Information Science,* 2011, Vol.186, Part 1, pp.139-146.

[15] Karyda, M., "An ontology for secure e-government applications". In *proceedings of first International Conference on Availability, Reliability and Security (ARES'06).* IEEE Computer Society, 2006: p. 1033-1037.

[16] Amina Souag, Camille Salinesi and Isabelle Wattiau. "Ontologies for Security Requirements: A Literature Survey and Classification". In *2nd International Workshop on Information Systems Security Engineering - WISSE'12 conjunction with the 24th International Conference on Advanced Information Systems Engineering CAiSE 2012*, 2012, pp. 1-8.