# A novel method: Ontology-based security requirements engineering framework

**Abstract:**

Software systems security is often exploited by threats through vulnerabilities of the systems. This reveals the necessity of eliciting and analyzing security requirements in the requirements engineering phase. Though some security requirements engineering methods are proposed by researchers for eliciting security requirements, the time and cost needed to use and implement the method are more than to elicit business requirements of the software systems. Moreover, requirements engineers are not trained in eliciting security requirements and they need to depend on security experts to identify security requirements. In order to solve these problems, we facilitate the security requirements elicitation process by security requirements reusability. This paper proposes an ontology-based security requirements engineering framework a novel method for developing secure software systems. Our method allows analysts to reuse existing security requirements ontologies when eliciting security requirements in security requirements engineering phase. We expect that the proposed method would be very helpful for requirements engineers to elicit and manage security requirements.

SECTION I.

## Introduction

Security Requirements Engineering (SRE) is an innovative research area in Software Engineering, with the consciousness that security must be analyzed early during the requirements phase [3]. Many researchers are working on this area; however, there is a lack in Security Requirements (SRs) management. One of the most important aspects to achieve secure software systems in the software development process is elicitation of SRs in the early phase of requirements engineering. SRs are considered non-functional requirements, which act as constraints on the functions of the system. An example of SRs is “The system shall not disclose any personal information about customers to the operators of the system”

The requirements engineers extract, analyze, specify, and manage business requirements and quality requirements such as performance, interoperability, usability, availability, reliability, and portability, but many are at a loss when it comes to SRs. This is because most of requirements engineerss are poorly trained on SRs and often confusing the SRs with the architectural security mechanisms that are traditionally used to solve security issues. Thus, they end up specifying architecture and design constraints rather than true SRs. The few trained requirements engineers are given only an overview of security mechanisms such as passwords and cryptography rather in actual SRs. This is because requirements engineering team lack knowledge about SRs and its specification. Either they need to depend on security experts or to be trained in the SRs elicitation process. This resulted in increase of time and cost to elicit and analyze SRs for software systems.

In this context, reuse comes out as an important factor in achieving efficient SRs elicitation and analysis. As coding, design and specification are reused, the SRs specifications can also be reused in the requirements engineering phase. By reusing the SRs, effective and high security of software systems can be achieved. Greater benefits are obtained when reuse is considered during the early phases of the Software Development Life Cycle (SDLC). In [5], it is said that “Requirements reuse has been pointed out as being one of the most pressing needs and grand challenges in requirements engineering research, whose solutions are likely to have the greatest impact on Software Engineering research and practice”. So, methods and technologies for facilitating requirements reuse are needed, including SRs [6].

In this paper, an Ontology-based Security Requirements Engineering Framework (O-SREF) is proposed and designed, which supports requirements engineers when eliciting and managing SRs. The proposed framework will reduce the time and cost needed in eliciting SRs. The fundamental challenge for our framework is the management of SRs knowledge of software systems.

The Web Applications (Web Apps) which is one of the important software systems is used for this research study, since they are more prone to security issues and vulnerabilities. So, we used Security Requirements Ontology (SRO) for developing secure Web Apps from [1], to implement O-SREF for eliciting SRs. In this paper, we propose to use SRO knowledge with the steps of Model Oriented Security Requirements Engineering (MOSRE) [2], [8], an SRE method for software system to elicit SRs. The remainder of the paper is organized as follows. Section 2 presents an outline of Security Requirements Engineering Framework (SREF) [4] and MOSRE. In section 3, presents about O-SREF, our proposed framework to elicit SRs and it also presents the comparison between the existing and the proposed frameworks. Finally, section 4 draws conclusions and suggests future perspectives.

SECTION II.

## Security Requirements Engineering Methods

Software systems are becoming vital in every domain of the human society like finance, transportation, telecommunications, social networks, military, education and health care; the list goes on. These systems are used by individual users and across networks of society. The ample use of software systems results in a large quantity of critical information and processes, which certainly need to remain secure. So, it is important to ensure that software systems are developed according to the user needs, and equal efforts are given to ensure that the systems are secure. SRE is the process of eliciting, specifying and analyzing the SRs for software systems. It answers the question “what are SRs for software systems is?” in the SRs specification.

The SRE methods as given in [3] have been proposed, that facilitate this kind of requirements analysis and the development of SRs. We suggest that in the SRE methods, the SRs should be well thought-out as functional requirements and have to perform the important activities for SRE phase as stated in [3].

### A. SREF Method

First, SREF is an iterative process consisting of four stages that integrate requirements engineering and SRE [4]. Iteration between requirements engineering and design activities is an important part of the SREF. Fulfilling a security requirement might lead to new assets and thus resulting in new SRs. The different stages are given below:

* Stage 1 Identify functional requirements.

Stage 2 Identify security goals

At this stage three activities are done to identify the security goals they are:

1. Identify candidate assets
2. Generate threat descriptions
3. Apply management principles

* Stage 3 Identify security requirements.
* Stage 4 Construct satisfaction arguments

Haley et al. perform SRs elicitation and analysis, with their SRE framework. It covers all three Confidentiality, Integrity, and Availability (CIA) goals. They have evaluated the framework by applying it to a SRs analysis for an air traffic control technology project.

### B. MOSRE Method

The overview of the MOSRE framework for Web Apps [2] is depicted in Figure 1. It is a 16 step method, which are repeatedly performed throughout the incremental development cycle. Thus, MOSRE is iterative and the SRs evolve in the early phase of the SDLC. The MOSRE framework performs all the important activities of SRE like asset and threat modeling, risk analysis and specification of SRs.

This framework includes all the phases of requirements engineering from inception to specification. The process starts with identifying the objective of the web application developed i.e. to know the purpose of the system being developed. The next step is the identification of stakeholders of the application, who help the developers to develop a secure web application. The assets of the application are identified from all the layers, which are the treasure of the business. These assets are of two categories business assets and application assets. An eliciting technique is identified, which suits the application domain for eliciting the requirements. Then the architectural diagram of the web application in the environment is drawn with high level of abstraction. This helps to identify the level of security to be achieved in the web application. The next step is to identify the business goals and requirements, for the application developed.

The use case diagram of the business is generated with the business goals and the requirements, which gives a clean and complete view of the web application. The next step is the identification of security goals; CIA is the standard security goals, and other security goals depend on the domain of the web application developed.

The threats and vulnerabilities that affect the security goals have to be identified. The impact of the threats and vulnerabilities can be assessed by using any of the standard risk assessment methods, which is suitable to rate web application attacks. The next step is to categorize and prioritize the threats and vulnerabilities based upon the security goals it affects and the impact over the web application. The misuse case diagram is developed with the use cases and misuse cases of the application by the attackers.

The stakeholders give their suggestions and idea to identify the misuse cases. The next step is to identify the SRs for the web application. These are to be considered as the functional requirements as security is one of the important functionalities of the web application which will make the business available to the users. These SRs will be integrated with the business requirements of the Web Apps. The use case diagrams are generated considering the SRs for better clarity of the application usage. For elaborate and clear ideas about the requirements the structural, data flow and UML diagrams are drawn.

SECTION III.

## Security Requirements Ontology

In the recent past, a few researchers have been directed towards using ontologies in SRE. Lasheras et al. in [7] proposed an ontology-based framework for specifying and reusing SRs based on risk analysis. Lee et al. in paper [18] proposed a novel technique for systematically extracting, modeling, and analyzing SRs from multiple enforced regulatory documents. They build problem domain ontology from regulatory documents enforced by the department of Defense Information Technology Security Certification and Accreditation Process (DITSCAP). Chikh et al. [22] proposed an ontology based information SRE framework based upon the organization of knowledge in three corresponding domains - application, software requirements engineering, and information security. They integrated the three ontologies, to annotate the domain knowledge resources for building new information SRs.

Most of the existing ontologies [9]–[10][11][12][13][14][15] are for security mechanism and design rather for SRs. There are very few research works for requirements engineering phase [16]–[17][18][19][20] and they are in a primitive stage. These security ontologies are not considering the important concepts such as assets, threats and security objectives and vulnerabilities. As a solution SRO [1] was constructed with concepts such as assets, threats and security objectives, security requirements, security issues with its impact of risk and vulnerabilities.

In paper [1], the authors elaborate on the process of construction of ontology for Web Apps SRs. Ontology was developed in four phases: Knowledge acquisition, Creation of SRO for Web Apps, Knowledge retrieval - query processing and Security requirements specification, each phase with different activities to identify assets, threats and vulnerabilities [1]. The SRO can be reused in developing secure web applications.

The need for complete security ontology has been identified as an important challenge and research branch between the security community [21]. As there is no standard for SRs specification and reusability, we propose that the ontology can be used to represent SRs for software systems and reused. In this context, an ontology-based SRs elicitation and analysis can be adopted in SRE phase as stated in [22]. To implement O-SREF, the SRO developed in [1], can be reused to elicit for Web Apps as it is the most security sensitive critical software systems.

In SRO [1], the assets for the Web Apps are employees, media, search, files, records, client, server, computer communications and communication lines. Threats such as SQL injection attacks, cross site scripting attacks, data transmission threats, time bomb, virus, authentication threats, denial of service, information gathering and sniffing are included in the security ontology.

### A. The Proposed Framework-Ontology-Based Security Requirements Engineering Framework (O-SREF)

One of the goals of requirements analysis is to develop a requirements specification document of high quality. In order, we propose SRs analysis and specification method based on ontology technique. In this section, we discuss the proposed O-SREF for designing Secure Web Apps, which present an integrated view of the knowledge and best practices related to SRs for the software development project.

SRO can be used with SRE methods, since the requirements engineers depend on security experts to elicit SRs, and they are not trained on SRs. The Figure 2 shows how the SRO is used with SRE framework - MOSRE to elicit SRs. The SRO can be used in some steps of MOSRE, for e.g. to identify stakeholders, asset, security goals, threats and vulnerabilities.

This will help the requirements engineers to identify a number of assets, threats and vulnerabilities which in turn help to identify SRs. The retrieved knowledge can be analyzed, and high-security levels can be achieved. Using SRO during the development stage enables designers to practice and reuse higher level of knowledge than is usually the case in Software Engineering. The time and cost can also be reduced to elicit SRs using an SRO because of the reusability property of ontology.

### B. Comparative Analysis of O-SREF with other SRE Methods

The Table 1 shows the comparison of O-SREF with other SRE methods and it is clear that the use of SRO in the MOSRE framework reduced the overhead of developers and requirements engineers, and their dependency on security experts.

The time to elicit, analyze, verify and specification of SRs is affordable if the SRO is reused and very high level of security can be achieved, despite of the fact that more time is spent to create and populate SRO.

SECTION IV.

## Conclusion

Ontologies can provide an effective mechanism to specify, describe and exploit knowledge in the evolving area of SRE. This paper presents Ontology-based Security Requirements Engineering Framework to develop secure software systems. To develop and maintain SRs ontologies are very hard, but if developed, it can be used and applied in different software system contexts. Moreover, it can assist requirements engineers and developers in eliciting SRs with less security knowledge and dependency on security experts.

It also helps to save time and make better choices in applying SRs, since it allows requirements engineers to exploit accumulated knowledge. So we conclude that the O-SREF are more efficient and allow us to build more secure software systems.

In future work, we plan to develop ontologies to represent SRs for domain specific applications such as power systems, product designs that can be used with CAD software. We also plan to make them available online so that the developers can access SRO to develop different secure software systems.