DESIGN OF SECURITY TAXONOMY IN REQUIREMENT ENGINEERING

Tejas Shah

Department of ICT, Veer Narmad South Gujarat University, Surat, India

ABSTRACT

Non Functional Requirements (NFR) are important in all phases of software development and results in quality of software to be built. It is observed that security requirements are incorporated and identified later in the software development life cycle. Security as non functional requirements imposes new challenges in managing confidential data and preserving its integrity. The security requirements and related artefacts must be considered from Requirement Engineering (RE) phase to implementation phase. This paper focuses on new design of taxonomy of security in Requirement Engineering. The design covers the major properties of security which are required in developing any web based, secured, confidential and integrity oriented system.

KEYWORDS

Non Functional Requirement, Requirement Engineering, Security Taxonomy.

1. INTRODUCTION

1.1. Non Functional Requirement

Generally, requirements provide context and objective ways to measure progress and success of the system. What the system should do is stated by the functional requirements, whereas the non-functional requirement offers constraints on the system. The Requirement Engineering in real world projects majorly envisions and focuses on Functional requirements. The non-functional requirement is solution requirement that describes the characteristics customer recommends for the system. The non-functional requirement elaborates a performance characteristic of the system. Functional requirements outlines what a system should do? and non - functional requirements deals with how a system should be? Non functional requirements are often referred to as system "quality attributes" or “constraints” or “non-behavioral requirements”. These are referred to as the "ilities" informally because of characteristics such as security, availability, stability, and modularity.

Anton in [1] defines NFR as “The non behavioural aspects of a system, capturing the properties and constraints under which a system must operate”. The IEEE standard 803-1998 [2] defines the category like functionalities, external interfaces, performance, attributes (portability, privacy, security, etc.) and design constraints.

Several classification of NFR exists in the literature, but security and privacy requirement plays a vital role in designing the recent web based and distributed system. The service based software engineering treats NFR as Quality of Service (QoS) parameter of web services. Each NFR is further divided into sub-characteristic or attributes. The properties of security: confidentiality,

integrity, availability and non-repudiation have to be considered at RE stage. Early discovery of NFRs is highly desirable, because it allows system level constraints to be examined and incorporated into early development process. The NFRs can be effectively elicited, linked and detected in a structured manner from stakeholders through the use of checklist, questionnaire, prototyping and brainstorming techniques. The aspects of NFR must be specified from higher-level concern to lower level implementation list for the designer or programmer. E.g. if a security is an aspect of higher level, then confidentiality, integrity and availability properties are measured at intermediate level; and encryption algorithm belongs to implementation level.

1.2. Security Requirements

Information security is defined as "protecting information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction," according to U.S. law. The international standard ISO/IEC 27002 [3] defines information security as "the preservation of the confidentiality, integrity and availability of information".

A subset of NFR domains, security and privacy are extremely significant for distributed web based applications. The practise of preventing unauthorised access, use, disclosure, disruption, modification, inspection, recording, or destruction of information is known as information security. The authentication and authorization property is also the subset of security; however it can be part of privacy also. As indicated in Figure 1, the basic goal of information security is to balance between data confidentiality, integrity, and availability. (also referred as the CIA triad) [4].

Figure 1. CIA Properties

As per the CIA triad, the information security team is assigned different roles for implementing requirement engineering phase. The security requirement engineering has particular significance, because unless we know what is to be secured, against whom, and to what extent, it is very hard to develop a substantial secured system. Prior to move with design phase, the RE process must incorporate a coherent set of security requirement by identification of security goals from stakeholders.

1.2.1. Confidentiality

Confidentiality is close to, but not identical to, privacy. Confidentiality is a necessary component of privacy and refers to viewing sensitive and protected data with mechanism of authorization. Confidentiality is a concept that can be ensured by practices of role based access control at different level of RE.

1.2.2. Integrity

The ability to prevent data from being changed in an unauthorized or undesirable way is referred to as integrity. The change could be considered as unauthorized modification or deletion of our data or portions pertaining of our data. To maintain integrity, we requires prevention of unauthorized changes to our data and ability to reverse authorized changes that need to be undone. Integrity is utmost important when confidential data provides the basis for other process implementation and decisions of organization.

1.2.3. Availability

The third major component of CIA triad is availability. Availability refers to the ability to access our data as and when we required it. The term "loss of availability" can refer to a variety of failures in the chain that prevents users to access their data. Such failures can result from power loss, OS failure or application problems, network attacks, compromise of a system, or other problems. Attacks on availability caused by an outside party are commonly known as a denial of service (DoS) attack.

1.2.4. Non-Repudiation

The term "non-repudiation" refers to a situation in which the author of a statement (the message sender) is unable to effectively contest the statement's authorship or the validity of an associated contract. Non-repudiation is the assurance that someone cannot conclude for denial of sending some content. Non-repudiation also known as the ability to ensure that a party to a contract or a communication cannot reject the authenticity of their signature on a document or transmission of a message that they initiated.

1.2.5. Authentication

Authentication is the process of verifying the truth of an attribute of a single piece of data claimed true by an entity. The identification indicates a claim allegedly attesting to a person, while authentication is the process of actually confirming that identity. Users are often recognized by a user ID, and authentication is accomplished when the user gives a credential, such as a password along with user name. There are three basic categories of authentication, based on what you know, what you have, or who you are in the system. Passwords and PINs fall under the "what you know" category, and they are low-reliability techniques since they can be lost, stolen, or guessed by an attacker. “What you have” technologies such as RFID cards and e-tokens can be used to authenticate but can be stolen or misused. A “two-factor” authentication scheme that pairs a what you-have technique with a what-you-know technique can be strongly recommended for use.

1.2.6. Authorization

Authorization is the property which is related to security to protect data with role based access control and linked with the privacy as it has to be be preserved with legitimate authorization process. In Authorization process, an administrator grants permissions to authenticated users to access resources of the system. The actor’s permissions, are either stored locally on the system or on the authentication server. The authentication is the process of verifying that "you are who you say you are", where as the authorization is the process of verifying that "you are permitted to do what you are trying to do". Role-based access control (RBAC) has become a popular alternative to traditional discretionary and mandatory access controls. The authorization procedure with role,

group and actor definition can be finalized at requirement elicitation level, which can be analysed and specified at later stage to enable authorized operations on functional and NFRs of system.

It is vital to identify and model distinct security aspects based on the system specific environment while developing software systems. The elicited security requirements should be implemented within the system, which should include all required measures for dealing with possible security breaches. This paper is organized as follows. Section II shows the related work of security requirements framework and some of taxonomies proposed in literature. Section III describes the actual design of security taxonomy and last section concludes the paper.

2. RELATED WORK

The application specific security requirements elicitation, elaboration, specification, analysis and documentation are the unexplored areas by requirements engineering researchers to date. The following section describes some of the security requirement engineering frameworks, techniques, some taxonomies related to security which covers either all or few phases of RE.

In the systematic review of the literature [5], various security requirements engineering methodologies are analysed effectively. It is essential to research different SRE methods to analyze and build security requirements for software systems and select methodologies for development of security features. In [6], Firesmith explored a taxonomy of security-related requirements which includes pure security requirements, security-significant requirements, security system requirements and security constraints.

A security ontology proposed in [7] includes cyber security attacks during early stage of requirement elicitation. The work includes knowledge from several security standards. The Comprehensive, Lightweight Application Security Process (CLASP) [8] approach describes Security Requirements Engineering as a life cycle process that proposes a number of activities to improve security. The core security services for CLASP are: authentication, authorization, confidentiality, integrity, availability, non-repudiation and accountability.

To describe security requirements, standard use case diagrams are not suitable and not widely used. Therefore, the authors in [9] [10] presented a systematic approach for eliciting security requirements by extending traditional use cases to cover misuse, which could be useful for non-functional requirements other than security.

The SecureUML approach based on RBAC authorization constraint is presented in [11]. The SecureUML is UML based process design language for formalizing access control requirements. SecureUML is UML based modelling language designed to integrate access control information into application models. This language defines a vocabulary for expressing different aspects of access control, like roles, role permissions and user-role assignments. UMLSec methodology explored in [12] focuses on specification of security requirements related to confidentiality and integrity in analysis models based on UML. The proposed models are multilevel security models that simulate the behaviour of a possible attacker. The majority of UML diagrams are used to model various security elements.

Haley et al. in [13] describes an iterative process which integrates RE and security requirements engineering discipline. The process incorporates steps of identification of functional requirements and security goals, generation of goal threat description, link CIA concerns to the assets and then construct satisfaction arguments which satisfy the security requirements.

The standardized SREP (Security Requirements Engineering Process) proposed in [14] deals with the security requirements during the early phases of software development in a systematic and intuitive way. It is based on the reuse of security requirements, by providing a Security Resources Repository, along with the integration of the Common Criteria (ISO/IEC 15408). It also conforms to ISO/IEC 17799:2005 with regard to security requirements. The approach begins with iterative software development and includes a micro-process that consists of nine security RE activities that are completed in an iterative and incremental manner.

SQUARE [15] is a comprehensive methodology and framework for security requirements engineering which incorporates step by step prioritized security work carried out jointly by requirement engineers and stakeholders. The authors propose a process that establishes security requirements elicitation, categorization and prioritization. This methodology focuses on building security concepts into the early phases of the software development life cycle.

Based on the re-use of security requirements, conforming to Common Criteria (ISO/IEC 15408) standard, Toval et al. defined SIREN (SImple REuse of software requiremeNts) process in [16]. SIREN describes a spiral process model, some basic guidelines, techniques and tools which includes requirements elicitation, requirements analysis, negotiation, requirements specification and validation phases. A requirement repository is created based on classification of domain and profile.

Several security standards (such as ISO/IEC 15408, ISO/IEC 27001, and so on) and requirements approaches (such as UMLsec, security use cases, and so on) have been established in recent years to aid in the development of secure information systems. However, it is very difficult to develop a methodology / process that integrates all the artefacts of security. The security requirements are frequently developed independently from the rest of the RE activity and therefore not integrated into the mainstream of the RE phase. It can be observed from the preceding literature review that the RE requires a new design of security taxonomy.

3. DESIGN OF SECURITY TAXONOMY

There are different taxonomies for non functional requirement, security and privacy in different domain, system and model. In this paper, focus is given to the design of security requirements with their properties and artefacts. The security attributes are considered at system level and functional requirement level, where the linkages of security properties are implemented.

Based on the article presented in [17] for NFR categories and other literature review, a novel, modified, integrated taxonomy is designed for the RE framework. The major components of the taxonomy are presented in Figure 2.

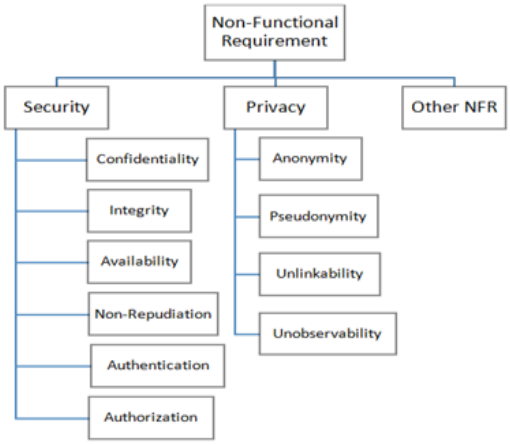

Figure. 2. Design of NFR Taxonomy

As shown in Figure 2, the security requirement is further divided into confidentiality, integrity, availability (CIA triad), non-repudiation, authentication and authorization. In privacy category, we considered anonymity, pseudonymity, unlinkability and unobservability for design of privacy taxonomies. This paper focuses on a novel security taxonomy that has been adapted from current taxonomies found in the literature on Requirement Engineering.

Main Properties and Attributes of Information Security are mentioned below.

- Confidentiality
- Integrity
- Availability
- Identification
- Authentication
- Authorization
- Non repudiation

3.1. Confidentiality at FR Level

The confidentiality property links the functional requirement with the reference contents in terms of questionnaire as shown in Table 1.

Table 1. List of Questionnaire for Confidentiality at FR Level

No	Question	Answer	Method
Q1	Which requirement (module) should be kept confidential?	List of Requirements	-
Q2	Which actors provided with access permissions to requirement?	List of actors	-
Q3	How to implement confidentiality?	Encryption	Encryption Algorithm e.g. Symmetric Key and Asymmetric Key
		Access Control at FR level	Role Base Access Control (RBAC) CRUD operations for accessing confidential requirement or module

3.2. Confidentiality at System Level

This property requires access control at System level to implement confidentiality. The business analyst can ask to stakeholders for fine grain level of confidentiality implementation for their system. Functionalities for each access control requirements required for confidentiality at system level are shown in Table 2.

Table 2. List of Functionalities for Access Controls at System Level

No.	Access Controls Categories	Functionality
1	Admin Controls	Security policy, Monitoring and supervising, Information classification, Personnel procedures, Investigations, Testing, Security-awareness and training
2	Physical Controls	Fences, Locks, Badge System, Security Guard, Biometric System, Mantrap Doors, Motion Detectors, Closed-circuit TV, Alarm, Backups, Safe storage area of backups
3	Technical Controls	ACLs (Access Control List), Routers, Encryption, Audit Logs, IDS, Antivirus Software, Firewalls, Smart Cards, Alarms and alerts

At the system level, there can be various access control requirements may arise which can be defined as different models as shown in Table 3.

Table 3. Access Control Models at System Level

No.	Access Control	Description
1	Discretionary Access Control (DAC)	The admin or business analyst can choose the resource access control for different system subjects based on the customer's preferences. The ACL (Access Control List) is with user or group identity.
2	Mandatory Access Control (MAC)	Classification of subjects as secret, top secret, confidential for accessing resources. Security labels are assigned to objects.
3	Nondiscretionary Access Control - RBAC	This ACL is based on role based access control by interaction of subject and object with central administration. User roles plays important role in overall system

3.3. Integrity at FR Level

To preserve integrity of specific requirements of system, following points are considered in integrity taxonomy. This property connects functional requirements to aid the system designer in maintaining the integrity of important requirements of the system. Business analyst can provide list of questionnaire to stakeholders for implementing integrity at FR level as shown in Table 4.

Table 4. List of Questionnaire for Integrity at FR Level

No	Question	Answer	Method
Q1	Which requirement required prevention from unauthorized modification?	Requirement Name	-
Q2	Who performs authorized modification to which requirement?	Actors	-
Q3	How to implement integrity?	Hashing	Hashing Algorithms e.g. MD5, SHA
		Digital Signature	Public key signature Private key signature

3.4. Integrity at System Level

This security property conforms the prevention of unauthorized modification and disruption of information available in the system. The successful requirement elicitation and specification at system level facilitates trustworthiness, correctness and completeness.

Implementation of integrity at system level can be obtained by two mechanisms: Preventive and Detective. Detective mechanisms includes various methods as shown in Table 5.

Table 5. Detective Mechanism for Integrity Implementation

No.	Method
1	Least Privilege
2	Separation of concern
3	Rotation of duties
4	Cryptographic checksums

3.5. Availability at FR Level

This property ensures that the information is readily accessible on demand and any time to authorized users. The stakeholders can perform various operations on requirements and it should be available and accessible. Questionnaire for such requirements is shown in Table 6.

Table 6. List of Questionnaire for Availability at FR Level

No.	Question	Specification
1	Which requirement is available to the user on demand?	List of Requirements
2	Which module is available and accessible to the user?	List of Modules

3.6. Availability at System Level

To design the availability of requirements at system level following parameters of Table 7 can be considered.

Table 7. Parameter List for Availability at System Level

No	Parameters
1	24*7 Operational Time
2	Downtime
3	Maintenance slack day
4	Location of Operation

3.7. Authentication at System Level

Authentication establishes that the subject identification is correct for system level access. There can be the exhaustive list of authentication property in taxonomy of security. But here few methods are included for authentication methods at system level as shown in Table 8.

Table 8. Sample List of Authentication Methods

No	Type	Methods
1	Biometrics	Finger Print, Palm Scan, Hand Geometry and Topography, Retina Scan, Iris Scan Signature Dynamics, Keyboard Dynamics, Voice Print, Facial Scan
2	Passwords	Password Requirements, Password Generator, Password breaker, Encrypted and hashed password, Password aging
3	One Time or Dynamic Password	Token based (synchronous, asynchronous)
4	Cryptographic Keys	Private keys and digital signature
5	Pass phase	Sequence of characters transferred to virtual password
6	Memory Cards	Swipe card, ATM card
7	Smart Cards	Contact, Contactless (hybrid combination)

3.8. Authorization at FR Level

The authorization property defines access rules for users with combination of Create, Update, Delete and Read operations on functional requirements which are defined by business analyst. These operations' combinations are used to manage various functionalities of a system.

3.9. Non-Repudiation at System Level

The Non-repudiation property shall be implemented through digital signature and other methods. Here the actor (sender) cannot deny of sending the message (Requirement). This property is designed for system level access only which is shown in Table 9.

Table 9. List of Methods for Non-Repudiation at System Level

No.	Methods
1	Digital Signature Algorithms
2	RSA-based signature schemes, such as RSA-PSS
3	DSA and its elliptic curve variant ECDSA
4	ECDSA
5	Rabin signature algorithm
6	Pairing-based schemes such as BLS
7	Undeniable signatures
8	Aggregate signature
9	Signatures with efficient protocols

The description of above taxonomy is not abstracted to the user. But business analyst can guide the user to select a specific method and its use.

4. CONCLUSION

Security properties, artefacts and attributes are required to be elicited in RE phase and same will be applied in whole software life cycle. The integrated security taxonomy is required to be designed for requirement engineering. This paper unveils the design of novel taxonomy of security in Requirement Engineering. This design is based on several questionnaires which are formed at RE level by keeping in mind business analyst, requirement engineer and customer. This new taxonomy of security unveils the major properties of security like confidentiality, integrity, availability, authorization, authentication and non-repudiation which can be taken care with respect to requirement of a system.

REFERENCES

[1] A. Anton, "Goal identification and refinement in the specification of software-based information systems," Georgia Institute of Technology, USA, (1997).

[2] Institute of Electrical and Electronics Engineers, "IEEE 830-1998 - IEEE Recommended Practice for Software Requirements Specifications," New York, (1998).

[3] ISO/IEC, "ISO/IEC 27002:2013 - Information technology -- Security techniques." (2013)

[4] J. Andress, The basics of information security : understanding the fundamentals of InfoSec in theory and practice

[5] D. Mellado, C. Blanco, L. E. Sánchez, and E. Fernández-Medina, "A systematic review of security requirements engineering," Computer Standards & Interfaces, vol. 32, no. 4, pp. 153–165 (2010).

[6] Firesmith, D.G.: A taxonomy of security-related requirements. In: RHAS 2005, Paris (2005)

[7] Amina Souag, Camille Salinesi, Raul Mazo, and Isabelle Comyn- ´ Wattiau. A security ontology for security requirements elicitation. In Engineering Secure Software and Systems, volume 8978 of Lecture Notes in Computer Science, pages 157–177. Springer, (2015).

[8] J. Viega, John, Viega, and John, (2005), "Building security requirements with CLASP," in Proceedings of the 2005 workshop on Software engineering for secure systems---building trustworthy applications - SESS '05, vol. 30, no. 4, pp. 1–7.

[9] G. Sindre and A. L. Opdahl, (2005) "Eliciting security requirements with misuse cases," Requirements Engineering, vol. 10, no. 1, pp. 34–44.

[10] G. Sindre, D.G. Firesmith, A.L. Opdahl, (2003), A Reuse-Based Approach to Determining Security Requirements, Proc. 9th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'03).

[11] T. Lodderstedt, D. Basin, and J. Doser, (2002), "SecureUML: A UML-Based Modeling Language for Model-Driven Security," Springer, Berlin, Heidelberg, pp. 426–441.

[12] J. Jürjens, (2002), "UMLsec: Extending UML for Secure Systems Development," Springer, Berlin, Heidelberg, pp. 412–425.

[13] C. B. ; Haley, R. ; Laney, J. D. Moffett and B. Nuseibeh, (2008), "Security Requirements Engineering: A Framework for Representation and Analysis," IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, vol. 34, no. 1.

[14] D. Mellado, E. Fernández-Medina, and M. Piattini,(2007) "A common criteria based security requirements engineering process for the development of secure information systems," Computer Standards & Interfaces, 29(2) , pp 244–253.

[15] N. R. Mead, E. D. Hough, and T. R. Stehney,(2005), "Security Quality Requirements Engineering (SQUARE) Methodology".

[16] A. Toval, J. Nicolás, B. Moros, and F. García,(2002), "Requirements Reuse for Improving Information Systems Security: A Practitioner's Approach," Requirements Engineering, vol. 6, no. 4, pp. 205–219.

[17] J. M. L. Chung, B. Nixon, E. Yu, (1999), "Non-Functional Requirements in Software Engineering," Dallas.

AUTHORS

Dr. Tejas R Shah is working as an assistant professor in Department of ICT, VNSGU. He has 15 years of teaching experience. He has published 12 research papers in various national and international journals. Requirement Engineering is the topic of research interest of him.