Security in Requirements and Design Phases

Rajat Goel, M.C. Govil, Girdhari Singh
Department of Computer Science & Engineering, Malaviya National Institute of Technology Jaipur, India

Abstract: *Developing secure software is important in the light of increasing flow of sensitive information. Moreover, different kinds of users interact with the system. Integration of security in the development process is quite apt in this regard. More specifically, the early phases of requirements and design are the most appropriate for this. However, it is not easy due to several pressing issues and constraints. The existing techniques have one or the other limitations. This paper describes an attempt made in the direction of overcoming these limitations and enhancing the techniques.*

Keywords: *requirements engineering, modeling, functionality*

1. Introduction

Researchers [1][2] believe that development process is a major source of security problems in software and the solution lies in integrating security in the development process. The discussion presented in [3] reveals that such an integration in the first two phases of requirements and design phases will be most appropriate. Requirement elicitation is considered critical and ambiguous [4] and necessity of security in this phase has been stressed. An ideal approach is to freeze requirements initially but it is not practical. Similarly, designing is essential for better communication between developer and client. In this regard Unified Modeling Language (UML) is quite prevalent but researchers do not find it suitable for modeling security requirements due to varied reasons [5][6]. These issues have served as a motivation for the development of Security Requirements Elicitation, Assessment and Design (SecREAD) methodology. Several other software process models have been proposed earlier but all of them have certain limitations. These have been discussed elaborately in Goel et al. [7][8].

2. Methodology

SecREAD is a methodology that integrates of security in the software development life cycle. It is based on Assets (data items), functionalities and stakeholders. SecREAD is a well-structured process that first gathers requirements, then rates them on security parameters and finally, shows these ratings graphically in a meaningful way. It is kept in mind that only the relevant stakeholders rank the relevant assets and functionalities. The steps of this methodology are Identification, Refinement, Mapping, Ranking, Analysis and Design. Fig. 1 depicts the process flow.

https://doi.org/10.15242/IJCCIE.ER0317109

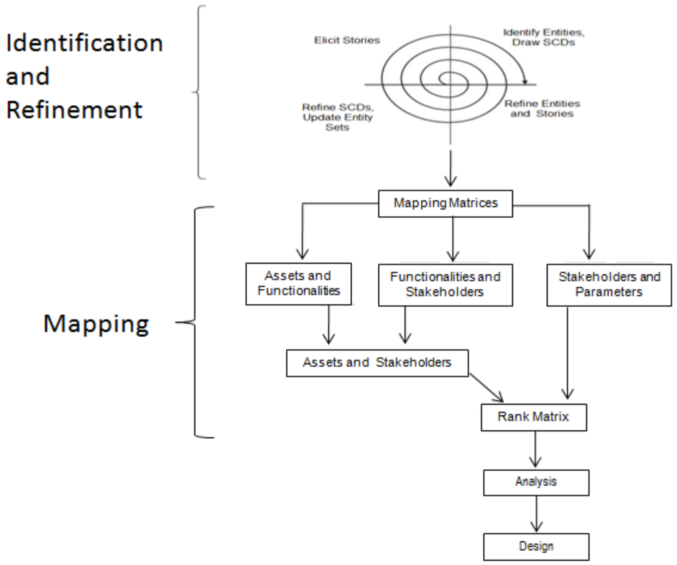
Identification and Refinement
Elicit Stories
Identify Entities, Draw SCDs
Refine SCDs, Update Entity Sets
Refine Entities and Stories
Mapping
Mapping Matrices
Assets and Functionalities
Functionalities and Stakeholders
Stakeholders and Parameters
Assets and Stakeholders
Rank Matrix
Analysis
Design

Fig. 1: SecREAD Methodology

3. Identification & Refinement

These two phases move spirally as seen in figure 1. The spiral activity is initiated by the representative(s) from the client and developer side and the expert group. After each spiral newer stakeholders, assets and functionalities are incremented, if any. The expert group includes the business experts and experienced technical people.

3.1. Stories and Story Conversion Diagrams

The client and the users narrate the requirements to the expert group. These requirements are called ‘Story’ and can be in natural language. Once a story is elicited the assets, stakeholders and functionalities are identified and put into three sets i.e. S, A and F respectively.

$S = \{S_1, S_2,\dots S_n\}$, $A = \{A_1,A_2,\dots\dots A_m\}$ and $F = \{F_1,F_2,\dots\dots F_p\}$ $n \in W$, $m \in W$, $p \in W$
where, W is the set of whole numbers

A Story Conversion Diagram (SCD) is drawn for every story as and when it is elicited for graphical representation. There are different kinds of stories and each of them requires a different type of SCD. Stories may contain only one entity from stakeholder, asset or functionality. Other stories may contain two, three or multiple instances of entities. The SCD for a three-entity story is shown in figure 2. Here, $S_i \in S$, $A_j \in A$ and $F_k \in F$.

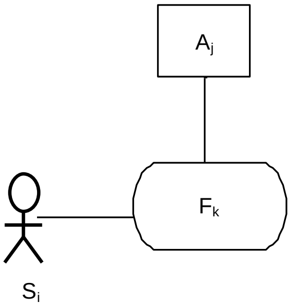

Fig. 2: Three entities

3.2. Refinement of Stories and SCDs

Up till now, the stories are elicited and converted as it is into SCDs. This information is refined by experts through the techniques of redundancy removal, and decomposition and aggregation. There may be more than one name (or similar names) for a single asset, functionality or stakeholder, which are removed. In the next cycle, the newly identified stakeholders sit together with the previous stakeholders to identify more entities. If new entities are not found, the process stops. At the end of the whole spiral final sets for asset, stakeholder and functionality are formed. In this paper, for explanation, following sets are used.

$F = \{F_1, F_2, F_3, F_4\}$
$S = \{S_1, S_2, S_3\}$
$A= \{A_1, A_2, A_3, A_4, A_5\}$

4. Mapping

4.1. Matrices

In this phase, the related entities are mapped through. The experts and the developers perform the mapping. These matrices are developed on the basis of the information collected. This process is performed by. The shaded cells in the matrices show relevance. Functionalities and assets mapped by matrix X shown in table I. Matrix Y (table II) maps stakeholders to functionalities. Matrix Z, shown in table III, relates stakeholders to assets and is obtained by the multiplication of matrices X and Y.

$X = (x_{ij})$
Where, $i = 1, 2, 3, \ldots\ldots.a$
$j = 1, 2, 3, \ldots\ldots.f$
$a = n(A), f = n(F)$

$Y = (y_{ij})$
Where, $i = 1, 2, 3, \ldots\ldots.f$
$j = 1, 2, 3, \ldots\ldots.s$
$f = n(F), s = n(S)$

Table I. Matrix X

F A	F_1	F_2	F_3	F_4
A_1			▨	
A_2		▨	▨	▨
A_3				▨
A_4		▨		
A_5	▨	▨		▨

Table II. Matrix Y

S F	S_1	S_2	S_3
F_1	▨	▨	
F_2		▨	
F_3		▨	▨
F_4	▨		▨

$Z = (z_{ij})$
Where, $i = 1, 2, 3, \ldots\ldots.a$
$j = 1, 2, 3, \ldots\ldots.s$
$a = n(A), s = n(S)$

Table III. Matrix Z

S A	S_1	S_2	S_3
A_1		▨	
A_2	▨	▨	▨
A_3	▨		▨
A_4		▨	
A_5	▨	▨	▨

4.2. Parameters

The assets are to be ranked by the stakeholders on the confidentiality, authentication, integrity, non-repudiation and authorization parameters. P is set of parameters. Stakeholders are mapped to parameters based on their expertise or technical awareness. A stakeholder, thus, can rank assets on his/her relevant parameters only. Matrix W (table IV) maps the two.

Table IV. Matrix W

P S	**Authentication**	**Integrity**	**Confidentiality**	**Non-repudiation**	**Authorization**
S_1	▨	▨	▨	▨	
S_2	▨	▨	▨	▨	
S_3		▨	▨		

4.3. Rank Matrix

R is a three dimensional matrix, developed using X, Y and Z. It has assets as rows, parameters as columns and stakeholders as depth or sheets. Every stakeholder has one version of the sheet on which he/she has to perform ranking.

$R = (r_{ijk})$

Where, $i = 1, 2, 3, \ldots\ldots.a$

$j = 1, 2, 3, \ldots\ldots.p$

$k = 1, 2, 3, \ldots\ldots s$

$a = n(A)$, $p = n(P)$, $s = n(S)$

5. Ranking

The applicable cells in the rank matrix are ranked as 1 (Low), 2 (Medium) and 3 (High) for the level of security desired. Always, the assets are ranked over authorization parameter by the core group only to avoid partiality. Goel et al. [9] describes elaborately how this parameter is dealt with and corresponding diagrams.

6. Analysis

The final rank of the entities is calculated and diagrams are drawn. Taking matrix R, the mode of all values stored in the same cell in all the sheets is calculated and this consolidated value is stored in a matrix C. In this way a 2-dimensional matrix is obtained with assets as rows and parameters as columns. For every row, mode of all values is calculated. These values serve as the consolidated asset rank. Similarly, consolidated parameter rank is obtained by calculating the modes of all columns. To find the rank of any functionality, the assets relevant to it only are considered as in matrix X (table I).

7. Design

A rank diagram is drawn for every functionality of the system. Figure 7 is an example rank diagram for functionality F_4 as per matrices X, Y and Z. It shows that F_4 is a medium security functionality with two stakeholders S_1 and S_3. Three assets are associated with it. Security requirement for asset A_2 is low, for A_3 is medium and for A_5 is high. Ranks are denoted by concentric rectangles for assets and concentric ovals for functionality. Fig. 8 summarizes the complete system.

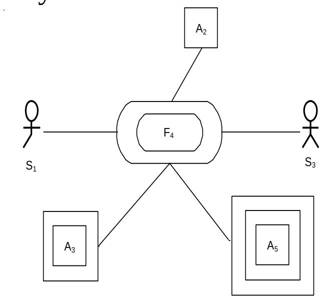

Fig. 7: Rank diagram for functionality F_4

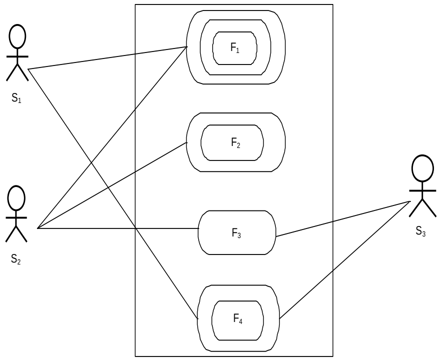

Fig. 8: System Rank Diagram

8. Conclusion

The proposed methodology is unique as it involves all stakeholders. Functionalities and assets are ranked by the stakeholders that are related to them only. The diagrams proposed grow with the increasing information obtained. A single diagram illustrates a large amount of information in a manner which is simple to comprehend.

9. References

[1] M. Lindvall et al. “Empirical Findings in Agile Methods,” in 2nd XP Universe and First Agile Universe Conference on Extreme Programming and Agile Methods, 2002, pp. 197–207.
https://doi.org/10.1007/3-540-45672-4_19

[2] D. Shreyas, “Software Engineering for Security - Towards Architecting Secure Software,” in ICS 221-Seminar in Software Engineering, 2001, pp. 1–12.

[3] R. Goel, M. C. Govil, and G. Singh, “Security Requirements Elicitation and Assessment Mechanism (SecREAM),” in 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2015, pp. 1862–1866.
https://doi.org/10.1109/ICACCI.2015.7275889

[4] R. Breu et al. “Key Issues of a Formally Based Process Model for Security Engineering,” in 16th International Conference on Software & Systems Engineering and their Applications, 2003, pp. 1–15.

[5] E. Woods, “Harnessing UML for Architectural Description —The Context View”, IEEE Softw., vol. 31, no. 6, pp. 30–33, 2014.
https://doi.org/10.1109/MS.2014.139

[6] C. Choppy and G. Reggio, “Requirements capture and specification for enterprise applications: A UML based attempt,” Proc. Aust. Softw. Eng. Conf. ASWEC, vol. 2006, pp. 19–28.
https://doi.org/10.1109/aswec.2006.43

[7] J. Chanda, A. Kanjilal, S. Sengupta, and S. Bhattacharya, “Traceability of requirements and consistency verification of UML use case, activity and Class diagram: A Formal approach,” Proc. Int. Conf. Methods Model. Comput. Sci., 2009.
https://doi.org/10.1109/icm2cs.2009.5397941

[8] R. Goel, M. C. Govil and G. Singh, “Modeling Software Security Requirements through Functionality Rank Diagrams,” in International Conference on Computational Science and Its Applications, 2016, pp. 398–409.
https://doi.org/10.1007/978-3-319-42092-9_31

[9] R. Goel, M. C. Govil and G. Singh, “Security Requirements Elicitation and Modeling Authorizations,” in International Symposium on Security in Computing and Communication, 2016, pp. 239–250.
https://doi.org/10.1007/978-981-10-2738-3_20