CHAPTER-3
THE APPROACH

In the last chapter, literature review was carried out related to research work on different facets of secure software engineering with bird’s eye-view. In this chapter, we have illustrated the approach taken for research work and development of security model for object oriented design (SMOOD). This research work is divided into different phases. The research work on these phases is carried out in steps as follows:

• **Review of the Current Research Work**: Review was carried out with different activities such as problem definition, information gathering and literature review.

• **Requirement Engineering**: Various aspects of system engineering were broadly considered, like requirement gathering, requirement validation and software requirement specification (SRS) in this chapter.

• **System Analysis**: Analysis on development of security model (SMOOD), security attributes, security principles, object oriented methodology and finally a relationship between these mentioned properties were prepared with a view of content mentioned in books, research papers, articles and views of industry experts.

• **System Modelling**: Experimental works as case studies are prepared. For each case study, different diagrams were prepared with variation in design properties. Based on current researches, proposed algorithms/metrics were applied to all diagrams for each case study and results were collected.

• **Model Development**: Development of security model for object oriented design (SMOOD) by data collected in Chapter 5, (System analysis) and analysis of collected result from chapter 6,(System modelling).

• **System Testing**: In chapter 8, security model was tested using commercial open source software and results were evaluated.

• **Contributions and Conclusions**: Finally, in chapter 9. Conclusions were drawn along with the contributions made by this research work in the field of secure software engineering.

These phases were explained in subsequent sections with reference to the actual work carried out to accomplish the tasks related to these phases.
3.1 Review:
The review of “measuring the impact of software design on software security” is carried out to keep abreast with latest technological advances in the field of computer engineering and information technology in general and secure software modelling methodology in particular.

- Study of modelling and related concepts was covered in the initial stage.
- Summary of this activity with importance of modelling was represented in chapter 01. Collection of necessary information from the internet was also carried out in the present work.
- Various research papers and books related to this area were studied to formulate the problem definition of this research work.
- The problem was defined in section 1.3.
- How the defined problem can be a challenge for research and development was conveyed in section 1.4.
- Overview of the possible solution was discussed in section 1.5.
- Objectives, goals, and scope were defined in section 1.6.
- More than hundred research papers along with books and websites are referred for problem formulation of this research work as shown in references. These papers help in finding the limitations of existing security modelling methodologies.
- Limitations of methodologies were discussed in chapter 2.

3.2 Requirements Engineering:
This is mainly related to tracing requirements related to project. As a system engineer, for better product development melding the problem and process was important. First, the system engineer initiates the process related to software project. This is the first stage in software engineering where customer and developer come together and organise a joint meeting. After the meeting, the problem was recognised by both. This recognised problem can have various requirements. These requirements can be traced and partitioned as normal, expected and exciting requirements. These requirements were further genuinely evaluated before finalisation. Based on these final evaluated requirements, the prototype of model was created and given to the client for verification at initial stage. Finally, system requirement specifications (SRS) were made and handed over to client for review. SRS consist of different aspects related...
analytical, data and functional, behavioural, and architectural requirement of the system.

3.3 System Analysis:
Analysis is a process in which we firstly identify the problem, understand the processes, collect the factual data and propose the feasible suggestions that can improve the system functioning. This involves studying of the processes, data collection, understanding the data flow, finding out weaknesses and evolving solutions for overcoming the bottlenecks of the system. System analysis also includes subdividing of difficult processes involving the entire system, identification of manual processes and data store. The major objectives of systems analysis were to find answers for: What is being done? Who is doing it? When is he doing it? How is it being done? Why is it being done? How can it be improved? System analysis is more of a thinking process and involves the innovative skill of the system analyst. Better analysis process gives birth to a new efficient system that satisfies the current needs of the user and has scope for future growth within the organizational constraints. The result of this process was a logical system design. System analysis was an iterative process that continues until a preferred and acceptable solution emerges (Kunwar Deepak, 2011).

In this analysis work, initially all the object-oriented design properties were examined, and then the properties related to security are listed out. Next, the security factor was elaborated in terms of security attributes. Finally, major security principles were listed out. Once we have all this information, security principles were elaborated in such a way that they should relate with the design properties and security attributes, based on the earlier research findings, articles published on the internet, books and as on websites. By this approach, at the end of this chapter, we could find the relationship between design properties and security attributes.

3.4 System Modelling:
Case study is an ideal methodology when a holistic, in depth investigation is necessary. Case study research excels at bringing us to an understanding of a complex issue or object and can extend experience or add strength to what is already known through previous research contribution. Investigators have used the case study method
for many years across a variety of disciplines. In this research work, case studies were carried out to accomplish goals which were essential for preliminary investigation, needed in identification, and determining requirements. In this research work, we have used case study approach for preliminary investigations were used. Four case studies have been taken for this work namely, Courseware management system, Bank management system, Hospital management system and Library management system. From UML diagrams, a class diagram has been used because the classes’ attributes and relationships were clearly mentioned in a class diagram. Inheritance, coupling, cohesion, abstraction, polymorphism and encapsulation were easily measured by class diagram. Security related information was specified by UML sec notation in class diagram. For each case study, one base diagram has been prepared and from base diagram rest of the diagrams was drawn by changing design properties in each diagram, so that we could see the changes caused by design properties and can measure the changes in security calculation caused by design property changes.

Historical data collections have probable benefits. It can provide base for validating proposed capacities and approaches, recognizing whether measurements were prominent or sheathing for useful pointers. Ancient data assemblage could also be used as orientation for guiding or ranking the measurements. Historic data collected about the characteristics of other similar types of software and the vulnerabilities they experienced can be useful as predictive estimates. Empirical verification also mention that traits associate with susceptibilities and that components comprising of previous susceptibilities can be employed to forecast, with rational exactness, susceptible mechanisms, centered on their traits. Available case studies results can serve as a preliminary point for upshot assortment. Some metrics and algorithms were mentioned by researchers to identify the impact of design properties on security. In this research, an attempt was made to identify those metrics and algorithms and applied them to the case studies diagrams. It can aid as a resource to inspect novel approaches of spotting and determining both probable and unforeseen associations for us as estimators, and to grow novel and better-quality scientific and computational practices. With a heavy data collection, consistent estimation of the overall efficiency of design properties can be made that might be employed in security model development (Wayne Jansen, 2009).
3.5 Model Development:
There were two types of approaches: qualitative and quantitative. Some properties will lend themselves to a quantitative analysis, while others will demand for a more qualitative approach. Certain qualitative properties were vague and incapable to be seized through reliable measurement. For example, characteristic like beauty can be extremely particular in nature, fluctuating broadly amongst entities. Qualitative projects can be employed to characterize quantitative methods of security properties (e.g., low means no susceptibilities found; medium refers to susceptibilities found between one and five; and high refers to susceptibilities found more than five). More habitually, numeric values were used to characterize positions that may be qualitative (e.g., 1, 2, and 3 versus low, medium, and high). While the numeric difference between ranked values may be significant for some metrics, it might not be for others, which was frequently the circumstance with security metrics. Quantitative evaluations of numerous security properties might also be subjective and joint to stem a complex value (e.g., rating = 0.25 x ranking A + 0.75 x ranking B) (Wayne Jansen, 2009) such configurations were used in Common Vulnerability Scoring System (CVSS). In this research work, the quantitative approach for designing security model based on information collected in chapter 05 and data collection in chapter 6. More detailed discussion on security model design was covered in chapter 7.

3.6 System Testing:
Software testing is a step for verification and validation of proposed work. Verification is the collection of actions, which make sure that software exactly implements a specific task. Validation is a different set of actions, which guarantee that the software that has been built meets the requirements (Andy JuAn Wang, 2005). Boehm states this in another way (Roger Pressman, 2004):

- Verification: “Are we building the product right?”
- Validation: “Are we building the right product?”

Hence the proposed security model (SMOOD) was tested for different open source software. Different test data were developed for each model. We have, here collected security data from selected open source software (OSS) implemented with C++. The selected target OSS is categorized into three types - application software, framework and library.
3.7 Discussion:
Software security is often being considered after the software is complete or in nearly completion phase like during testing, deployment and maintenance. Software security needs to be considered in the software from the beginning, and security activities need to take place in the earlier stage of software development lifecycle. For achieving this goal, an effective and efficient approach, which combines with the detailed understanding of what causes vulnerabilities and how to prevent them, is required. This approach was accomplished by introducing “secure by design” approach. Focus was only on design of software, testing the security of design and providing necessary data to designer to develop secure design.