CHAPTER-9
CONCLUSION

This chapter re-delineates the primary objectives of the research work and converses about the effectiveness by means of which the goals were assembled whilst unfolding the intrinsic restrictions built into the system. The survey of contributions made by this research work was summarized. Finally, this chapter summarised the discussion on results associated to SMOOD. Security is one of the most crucial aspects in the accomplishment of growth of software product. Organizations that want to improve their profits from firm and protected software product need to invest in software security approaches. However, secure software was not effortlessly attained. Investments in software development process improvement do not assure that software stands definite from attack.

9.1 Summary of Research Work:

There are nine chapters included in this thesis based on the research work concluded in present work. Brief chapter wise summary of current research work allied to the thesis. Each chapter of the report demonstrated the following:

Chapter 1: Introduction:

In this chapter, the brief discussion about the needs of secure development and secure modelling. This chapter has also presented the objectives, goals and comprehensive scope for carrying out the said study. Here, problem was well defined with general idea of the solution. How this research work can be made a challenging task was highlighted in the research and development challenges section. This also highlights the benefits of proposed modelling method. At the end, it has concluded with the organizational aspects of thesis.

Chapter 2: Background and Literature Survey:

This broad chapter has been divided into various sections like background concepts-basic concepts, definitions, background terminology related to security modelling technologies which have been used and referred in present research work. The review of literature survey has described the complete literature assessment which was embarked on to re-examine the diverse problem domains related to secure system development models, security approach, security consideration at design stage, susceptibility detection, analysis and threat etc. The significant conclusions derived
from the literature review have been summarized from more than 100 research papers and finally, the limitations of existing security approaches and methodologies have been discussed.

Chapter 3: Approach:
In this chapter, an approach towards the research work carried out has been related to propose security model for object oriented design (SMOOD).

Chapter 4: Requirement Engineering:
This chapter covers the various aspects of system engineering and requirement. This includes gathering, requirement validation and software requirement specification (SRS) was discussed in this chapter.

Chapter 5: Analysis:
This chapter covers identifying the data and the method used to deduce solution to the problem. Also, collection of information about security concept, security principles, security attributes and object oriented methodology with design properties were considered.

Chapter 6: System Modeling:
In this chapter, four case studies were demonstrated. This includes library management system, Bank management system, Hospital management system and Courseware management system. Multiple designs of each case study were prepared with respect to changes in design properties to estimate the combined effect on security. The calculations of security metrics and measurement of design properties on each diagram of case studies were given in the data collection.

Chapter 7: System Design:
In chapter seventh, the system design, the result which had been collected in chapter 6 and also the data collected in chapter 5 was used to develop security model for object oriented design (SMOOD).

Chapter 8: System Testing for Security Model:
This chapter has covered testing of different commercial open source software with proposed security model (SMOOD) and analysis of obtained results.

Chapter 9: Conclusion:
Chapters from 1 to 9 were briefly summarised. Then, survey of contributions made by this research work has been outlined. Finally, this chapter has concluded with the discussions on results and recommendations for continuity of research in the said area.
9.2 Survey of Contributions:
The contributions of this research work include:

1. Introduction to the concept of “secure by design”, which is an ideal approach in which security of design were considered.
2. Development of four different case studies with multiple diagrams.
3. Application of different earlier study findings on all case studies in data collection.
4. Finding the impact of individual design property on security for checking the complete design.
5. Finding the linkage between security principle, security attributes and object oriented design properties.
6. Development of SMOOD model which refers to checking of design from security point of view.
7. Testing of SMOOD model with commercial software in system testing phase.

9.3 Discussion on Results:
Currently security was considered after development of software and sometimes done throughout testing, preparation and maintenance phase. Security must be inbuilt to the software system from the start, which security activities ought to happen in the initial stage of software development lifecycle. Accomplishing this effectively and expeditiously wants an approach merging in depth understanding on what causes vulnerabilities, and how to prevent them. In this work, software security was accomplished by introducing secure by design approach. Secure by design refers to the point that the software has been premeditated in such a way that it was secure from the bottom level. This approach can not cover the application level security or emerging security polices in software life cycle. Focus was only on the design of software, testing the security of design and providing necessary data for designer to develop secure design.

We have used case study approach for preliminary investigation which will calculate the exact impact of each design property on software security. It was observe that the result after the analysis of single case study was not sufficient for checking the variations. Hence, multiple case studies were considered in the present research work. In object oriented development, UML diagrams were considered in design phase which gives multiple aspects of the software to be developed. From UML diagram a
class diagram has been used since the classes attributes and relationships were clearly mentioned in a class diagram. Inheritance, coupling, cohesion, abstraction, polymorphism and encapsulation can easily measured by class diagram. Security related information was specified by UML sec notation in class diagram.

To consider the security in design stage, quantitative valuation of security was essential. Quantitative estimation of quality has been stated in many studies. Security is one of the major factors of quality therefore; there is likelihood to quantify security based on quality quantification methodologies for object-oriented software. For gauging design properties and their effect on security, various researchers have suggested different algorithm and metrics. And a few of them had to attain a little quantification echelons on the security by co-operating the influence of the individual design property. For that some of them have developed some metrics and algorithms which were fitted and appropriate as per the impact of individual design property. Although all of these were good but none of them have tried to see the combined impact and inter-dependency between these different design properties. So, in this research work, we have considered all important research findings. All research work were combined and made the basis of our research work and applied these works on case studies as a part of data collection. By this approach we have investigated the impact of all important design property on the security.

Here, the security models consists of three primary entities: a set of high level security attributes, a set of security principles and a set of security carrying design properties. In the process of adapting this model, we had considered following primary constructive directed relations which may be used to assist us in the building security software.

- The linkage between object oriented design properties and security design principles.
- The linkage between security design principles and security attribute relation.
- The linkage between security design principles and object oriented design properties.
- The linkage between security attribute and security design principles.

Overall, these relations allow us to sight the job of creating high-level security attributes into software in bottom-up manner by assuring that specific product
properties were contented. This viewpoint was very valuable to those (i.e., to programmer) who have the responsibility of putting into practice security software. In SMOOD the goal is to find the weakness that will turn out to be susceptible which become basic indicators for the level of trustworthiness of the software. To achieve this goal the quantification of security at design stage is necessary. Our suggested model was founded on general security design ethics which were in exercise in the security engineering processes. Our model can be, thus used to monitor the pursuit of security metrics. The most common method of devising a model for software product security aims firstly the identification of a small set of high level security attributes. Next, it aims to decompose these attributes into sets of subordinate attributes in a top-down manner. Develop the security model for object oriented design (SMOOD) by weighting the relationships. In this way, it is anticipated to inspect the safety in software building level. It allows the designer to do the necessary action to deliver the software that is secure by design. Despite the fact that the model used modest and straightforward assumption, the correlations with the limited set of project were high. This provides indication that the model of this type can be efficiently used in monitoring security of the software product. It illustrates diverse views which have never been conversed earlier which give a new and meaningful methodology towards security measurement. This model has a flexible style in providing security so in the future it can easily accommodate any new research findings in the related area. The benefits of suggested work were to identify and eliminate flaws at earlier stage, which in turn decreases development time and cost.

Many adaptive secure systems are large and complex in nature. These systems work or function in distributed and dynamic environment, which involve complex timing constraints. To meet security challenges, a good and efficient secure modelling approach is compulsory. By SMOOD model at design stage, it is possible to develop such optimal secure systems. As it can be understood from the literature review, designing and modelling of adaptive secure system is a challenge. Therefore, in this research work, it was proved that if the design is secure itself, it will explore properly, without any loopholes in it. Further, the SMOOD model was tested with open source software. It is not possible to compare results, as there are no standard benchmarking methods available in the market. So based on the software releases trend, the hypothesis were set and the obtain results were compare against the hypothesis. And the results obtain from the
SMOOD matches the expected trend. However, the proposed modelling method effectively captures the behaviour of both soft and firm secure systems by use of SMOOD model.

9.4 What’s Next?

In this research work, we have proposed the concept of “secure by design” for object-oriented software, which could take care of the security of the design by eradicating its flaws in the same stage. This ultimately enhances the security of the software. Similar type of approach or kind of security awareness should be introduced in the other phases of SDLC.

The proposed SMOOD model was designed for object-oriented software based on the current research findings. So, it may be appropriate for current threats and vulnerabilities. It may be possible that it is not suitable after some passage of time due to presence of new threats evolving with time. As SMOOD model is flexible, it can easily accommodate the new research findings. So, it should be updated with latest research findings with respect to design properties and their impact on security.

Security is a multi-faceted concept. It means different for different applications e.g. for military purposes, it is confidentiality, for media, it is integrity and for business, it is availability. We can consider SMOOD model as a basic model in this domain. It is not specific to any particular software or application. Further, it can be extended to specific application.

With augmented software security incidents, compliance and regulatory necessities were altering the scene of security by diminishing the chance of releasing vulnerable software. Mutilation to status affected by a security rupture, and the subsequent damage of consumer trust and assurance, might be certainly irretrievable. A formal, secure and structured software development methodology, along with enforceable and apposite dogmas must develop a part of any organization’s operations. Employees should be endowed with the understanding of in what way to use software security controls, balance intimidations and counter measures, and balance trade with technology.

It is essential for the researchers associated with software development not only to be mindful of the requirement for security, but also educated, trained and qualified to apply it fittingly as well. Further, a security mindset needs to be revived. Furthermore to being capable to design, develop, and deploy software, they ought to be capable to
do harmonizing coercions with counter measures and business with technology. Awareness, training, and education programs ought to be custom-made to distribute processes, secure coding policies and technologies.