Volume 1, No. 2, July-August 2010

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

Security and Software Architectures

S. Ramamoorthy*
Professor in CSE
Dr MGR Uniersity, Chennai,
Tamil Nadu, India
srm24071959@yahoo.com

Dr. S. P. Rajagopalan
Professor Emeritus
Dr MGR University, Chennai,
Tamil Nadu, India
sasirekaraj@yahoo.co.in

S. Sathyalakshmi
Associate Professor
Hindustan University, Chennai,
Tamil Nadu, India
swamega@yahoo.com

***Abstract*:** Software Architecture plays a central role in developing software systems that provide basic functionality and satisfy critical properties such as reliability and security. Architectural modeling and risk management are invaluable to increase the security of a software system. However, the interplays between these principles and the side effects of the application of these secure design strategies on architectural qualities like maintainability have not been studied so far. Therefore, it is hard to make any trade-off decision between security principles and other qualities. The aim of this work is the enforcement of security principles at architectural level and identifying the best architectural pattern that incorporates maximum security.

Keywords: Software Architecture, Security patterns, Software Reliability, Software Maintainability, Risk Management, Optimization Model and Constraints.

I. INTRODUCTION

With the spread of the Internet and software evolution in complex intensive systems, software architecture often need be reconfigured during runtime to adapt variable environments and design objectives. Software security has emerged as a foremost concern for modern information enterprise. Several well-known security system architectures and models, including CORBA, EJB, and DCOM are cornerstones for designing scalable and flexible security systems. Despite these advances, however, how to analyze the design of security systems to ensure its consistency and integrity is still a largely open problem.

There is lack of rigorous and systematic ways in the literature to assess and assure critical properties in architectural composition of security systems. Although formal verification of security protocols has received increasing attention in recent years, these techniques are normally based on abstract computation or architecture of security systems. Many of these formal models or techniques are developed for a single security model and not scale well.

Security and reliability issues are rarely considered at the initial stages of software development and are not part of the standard procedures in development of software and services. Security patterns are a recent development as a way to encapsulate the accumulated knowledge about secure systems design, and security patterns are also intended to be used and understand by developers who are not security professionals.

Making incorrect assumptions during system development is a root cause of insecurity. Therefore, securing a software system implies eliciting the assumptions on which the correct operation of the system is based that may turn out to be invalid, and mitigate them. For this the relevant assumptions need to be found and assessed.

Most checkers can help in finding assumptions by formally showing what preconditions need to hold for a software system to fulfill its requirements. Additionally, they are able to provide counter-examples when the model contains inconsistencies or fails to uphold the intended requirements. On the other hand, modeling efforts should be focused on security-critical parts of the system. This is where risk comes into play.

Risk management usually elicits unwanted system behavior by means of misuse scenarios or checklists. The risks inherent to theses threats is then assessed and the results of the assessment are used to guide the security effort. Risk assessment focuses modeling effort on critical model parts and assures that no improbable assumptions are made in the context in which the system is being developed and deployed.

II. RELATED WORKS

In the paper, Component-Based Heterogeneous Software Architecture Reliability (COHAR) Modeling S. Ramamoorthy, Dr. S. P. Rajagopalan and S. Sathyalakshmi, [36] proposed an analytical model for component-based heterogeneous software architecture reliability and a method to find the solution for finding the optimal reliability of the overall software system according to the reliability of

each component, the operational profile, and the architecture of software. This approach was based on Markov chain properties and architecture perspectives to state view transformation in order to compute the reliability on heterogeneous software architecture consisting of various styles. In his work, Roshanak Roshandel [19] discussed the uncertainty of the execution profile is modeled using stochastic processes with unknown parameters, the compositional approach to calculate overall reliability of the system as a function of the reliability of its constituent components and their (complex) interactions and sensitivity analysis to identify critical components and interactions will be provided.

Lance Fiondella and Swapna S. Gokhale [20] considered the estimation of software reliability in the presence of architectural uncertainties and presented a methodology to estimate the confidence levels in the architectural parameters using limited testing or simulation data based on the theory of confidence intervals of the multinomial distribution. The sensitivity of the system reliability to uncertain architectural parameters was then quantified by varying the parameters within their confidence intervals. C. Smidts [4] presented an architecturally based software reliability model and underlines its benefits. The models based on an architecture derived from the requirements which captures both functional and non-functional requirements and on a generic classification of functions, attributes and failure modes. The model focuses on evaluation of failure mode probabilities and uses a Bayesian quantification frame work. Leslie Cheung and Leana Golubchik [21] discussed representative uncertainties which have identified at the level of a system's components, and illustrates how to represent them in the reliability modeling framework.

Through the work, Jun Han et.al.[35] gave an idea on the security characterization and integrity for Component-based Software. The work of Mark Moriconi, et. Al.,[34] gleaned the applications of incorporating security into software architectures. An Architectural foundation for security model sharing and reuse was analyzed and reported by Per Kakon Meland, et, at.,[32]. In his tutorial, Robert T. Monroe gave an idea on Modeling and analyzing software architectures with the emerging approaches. The Fail-Heterogeneous architectural Model was presented and also a discussion on applications of the model to DoS (Denial-of-Service) attacks mitigation and to group memberships by MarcoSeratini and Neeraj Suri [31]. A Software Architectural approach to Security by design was proposed by Arnab Ray and Rance Cleaveland [30] along with the security aspect so that an attacker can no longer take advantage of hidden assumptions. David G. Roasado et, al, [29] made a vast study of Security Architectural Patterns for measuring the security degree of the patterns, and indicating a fulfillment or not of the properties and attributes common to all security systems. Jungwoo Rayoo [28] and others gave presentation on the search of Architectural Patterns for Software Security. A Formal Approach to Designing Secure Software Architectures has been proposed by Huiqun Yu [24] and others.

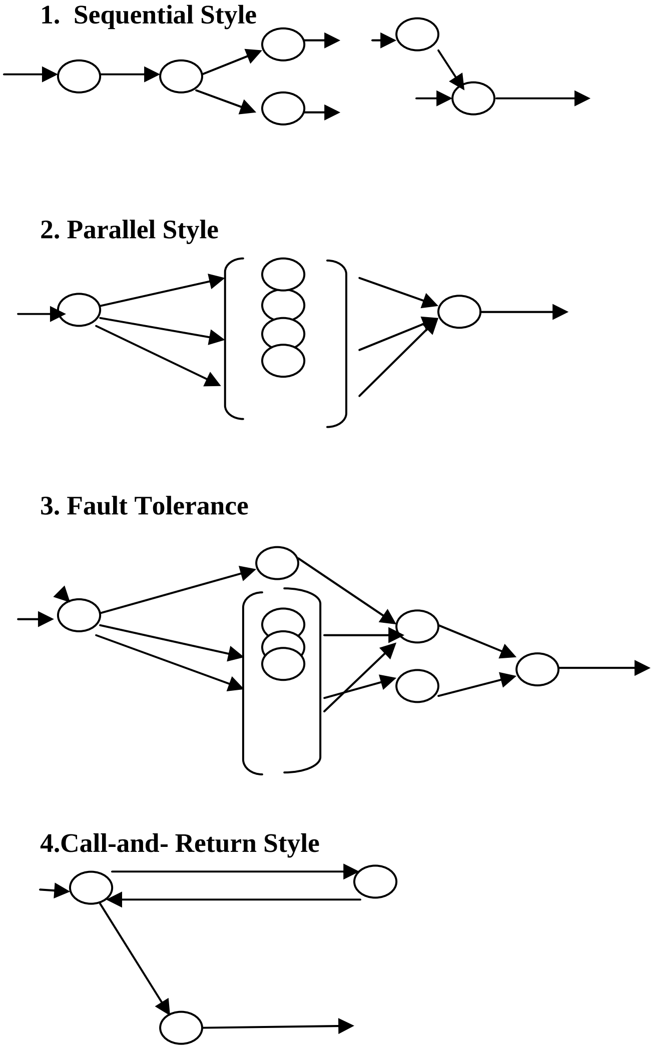
The rest of this paper is organized as follows: First the description on various architectural models followed by security patterns, then the proposed work is exhibited which is followed by a proposed algorithm and then the appropriate mathematical model and finally, suggestions and conclusion are depicted.

III.ARCHITECTURAL MDOELS

Software architecture is defined as the structure of software at an abstract level, consists of a set of components, connectors and configurations. Modern software often embodies complex heterogeneous architecture to achieve multiple quality requirements, such as the use of a parallel architecture to increase performance and/or introduce a back-up component to provide fault tolerance.

Goseva-Popstojanova *et al.* classify the existing architecture-based models into three broad categories: state-based, path-based, and additive. State-based models use the control graph to represent
software architecture, and predict reliability analytically. Path-based models compute software reliability considering the possible execution paths of
the program. The execution paths may be determined
using simulation, by executing the application, or algorithmically.

The various architectural styles are incorporated in the following pictures. Any other model might be the combination and or the other way representation of these styles only.



1. Sequential Style

2. Parallel Style

3. Fault Tolerance

4.Call-and- Return Style

Figure1: Various Architectural Styles

IV .SECURITY PATTERNS

Security is a very important aspect of any computing system, and has become a serious problem since institutions have opened their databases to the Internet. It is important to develop systems where security has been considered at all stages of design and at all architectural levels, which not only satisfy their functional specifications but also satisfy security and other non-functional requirements. Security patterns are proposed as a means of bridging the gap between developers and security experts. Security patterns are intended to capture security expertise in the form of worked solutions to recurring problems.

A software pattern can be described through a set of properties (a template) such as i) Name, ii)Intent, iii) Context, iv) Problem, v) Description, vi)Solution, vii) Consequences, viii) Known uses and ix) Related Patterns. These templates allow authors to define new problems, but respecting this structure. Once the template has been defined, we present some of the most important security architectural patterns, analyzing characteristics and find out the degree of security that they supply to the systems that use them.

These patterns are as follows: 1) Authorization Pattern, 2) RBAC(Role-Based Access Control) Pattern , 3) Multilevel Security Pattern, 4) Reference Monitor Pattern, 5) Virtual Address Space Access Control, 6) Execution Domain Pattern, 7) SAP (Single Access Point) pattern, 8) Check Point Pattern, and 9) Session Pattern.

V.THE PROPOSED WORK

It is thus we proposed to design an algorithm and the appropriate mathematical model that reveals the concept of injecting security in the software architecture and also to identify the best pattern , so that a secured architecture will help the designer to design a good software system. Our algorithm is pictured below:

VI.THE ALGORITHM

A. Identify a candidate software project from popular repositories such as http://sourceforge.net etc.
B. Understand the architectural styles that are being used.
C. Analyze the software for its architectural weakness in security.
D. Identify some known security design patterns that strengthen security.
E. Generalize, from identified security patterns, create a list of security tactics, types of threats, and their known strengths, weakness and interactions with other tactics.
F. Formulate an optimization model of these patterns along with their constraints to find the optimal pattern that has maximum level of security in its architectural pattern.
G. Thus the result of step F gives the required answer to our problem.

VII.THE MATHEMATICAL MDOEL

The above discussion leads us to define a mathematical model as below:

$$\mathbf{Z = MAXIMUM\ (SECURITY\text{-}LEVEL\ (M`_1 , M_2 , M_3 , \ldots . , M_n))}$$

Subject to

$$\mathbf{C1, C2, C3, \ldots . C_n}$$

where M_1, M_2, M_3 … M_n are the different security design patterns of architectural models and C1, C2, C”3, …., C_n are the appropriate constraints for the respective secured architectural models. The solution Z is the optimal model pattern of software architecture with high level of security injected in design stage itself.

VIII.SUGGESTIONS AND CONCLUSION

Software security as a particular non-functional requirements of software systems is often late in the software development process. Modeling and analyzing of these concerns and especially security in the software architecture facilitate detecting architectural vulnerabilities, decrease cost of the software maintenance. In this paper, we described a formal approach for constructing and identifying a secured software architecture. This may certainly help the software developer for designing a security injected architecture for software.

IX. REFERENCES

[1] Nourchene Elleuch, Adel Khalfallah, and Samir Ben Ahmed *Software Architecture in Model Driven Architecture* , IEEE 1-4244-1158/2007pp 219-223.

[2] Bo Yang, and Xiang Li , *A\Study on Software Reliability Prediction Based on Support Vector Machine* , IEEE , 1-4244-1529-2/2007, pp 11761180

[3] Hans Van Vliet, *Software Architecture Knowledge Management,* 19th Australian Conference on Software Engineering, IEEE Computer Society, 2008.

[4] C. Smidts, D. Sova, G. K. Mandela, *An Architectural Model for Software Reliability Quantification,* IEEE 1071-9458/97 pp 324 – 335.

[5] Noel De Palma, Konstantin Popov, Nikos Parlavantzas, Per Brand, and Vladimir Vlassov , *Tools for Arhitecture Based Autonomous Systems,* Fifth International Conference on Autonomic and Autonomous Systems 2009.

[6] Hamid Bagheri, Vajih Montaghami, Gholameraza SaFI, Seyed-Hassan Mirian-Hosseinbadi , *An Evaluation Method for Aspectual Modeling of Distributed Software Architectures,* IEEE 2008.

[7] Odd Petter N. Slyngstad, Reidar Conradi, M.Ali Babar, Viktor Clerc, and Hans van Vliet *Risks and Risk Management in Software Architecture Evolution: An Industrial Survey,* 15th Asia-Pasific Software Engineering Conference 2008.

[8] Richard C. Holt , *Grooking Software Architecure,* 15th Working Conference on Reverse Engineering 2008.

[9] Xiao Xiao and Tadashi Dohi , *On Equilibrium Distribution Properties in Software Reliability Modeling,,* IEEE International Conference on Availability, Reliability and Security 2009.

[10] Tirthankar Gayen , *Analysis and Proposition of Error-Based Model to Predit the Minimum Reliability of Software ,* International Conference on Education Technology and Computer IEEE Computer Society 2009.

[11] Tomotaka Ishii and Tadashi Dohi *A New Paradigm for Software Reliability Modeling – from NHPP to NHGP,* IEEE Pacific Rim International Symposium on Dependable Computing 2008.

[12] Hingguo Li, Xiaofeng Li and Yanhua Shu , *An Early Prediction Method of Software Reliability Based on Support Vector Machine,* IEEE 2007.

[13] Zuzana KRAJCUSKOVA , *Software Reliability Models* , IEEE 2007.

[14] Shiyi Xu , *An Accurate Model of Software Reliability* , 13th IEEE International Symposium on Pacific Rim Dependable Computing 2007.

[15] Shiyi Xu , *Recondiseration of Software Reliability Measurements,* 16th IEEE Asian Test Symposium 2007.

[16] Shinji Inoue, and Shigeru yamada , *Generalized Discrete Software Reliability Modeling with effect of Program Size,* IEEE Trasnactions on Systems, Man, and Cybernetics – Part A : Systems and Humans, Vol 37, No:2, 2007.

[17] Alaa Sheta , *Parameter Estimation of Software Reliability Growth Models by Practice Swarm Optimization – AIML* Journal, Volume (7), Issue (1), 2007.

[18] S. Chatterjee, S. S. Alam and R. B. Misra , *Sequential Baysian Techniques: An Alternative Approach for Software Reliability Estimation,* Sadhana Vol 34, Part 2, 2009.

[19] Roshanak Roshandel Computer Science Department University of Southern California Los Angeles, CA 90089- 0781 U.S.A. *roshande@usc.edu alculating Architectural Reliability via Modeling and Analysis.*

[20] Lance Fiondella and Swapna S. Gokhale Dept. of Computer Science and Engineering Univ. of Connecticut, Storrs, CT 06269 {lfiondella,ssg}@engr.uconn.edu ,*Software Reliability with Architectural Uncertainties.*

[21] Leslie Cheung, Roshanak Roshandel, Nenad Medvidovic, Leana Golubchik , *Early Prediction of Sfotware Component Reliability.*

[22] Fan Zhang, Xingshe Zhou, Junwen Chen, Yunwei Dong School of Computer Science and Engineering, Northwestern Polytechinical University, Xi'an, China{zhangfan, zhouxs, yunweidong, *junwenchen}@nwpu.edu.cn ,A Novel Model for Component-Based Software Reliability Analysis.*

[23] Leslie Cheung, Leana Golubchik, Nenad Medvidovic, Gaurav Sukhatme {lccheung,leana,neno,gaurav} usc.edu ,*Identifying and Addressing Uncertainty in Architecure-Level Software Reliability Modeling.*

[24] Huiqun Yu, Xudong He, Yi Deng, and Lian Mo, *A Formal Approach to Designing Secure Architectures,* Proceedings of the Eighth International Symposium on High Assurance Systems Engineering IEEE 2004.

[25] Hamid Bagheri, *Injecting Security as aspect able NFR into Software Architecture,* Proceedings of the 14th Asia-Pacific Software Engineering Conference , IEEE Computer Society, 2007.

[26] Thomas Heyman, Riccardo Scandariato and Wouter Joosen, *Risk-driven Architectural Decomposition,* Proceedings of International Conference on Availability, Reliability and Security, IEEE Computer Society, 2009.

[27] Koen Bayens, Riccardo Scandariato and Souter Joosen, *Measuring the Interplay of Security Principles in Software Architectures,* Third International Symposium on Empirical Software Engineering and Management, IEEE 2009

[28] Jungwoo Ryoo et,al. *In Search of Architectural Patterns for Software Security,* IEEE Computer Society 2009.

[29] David G. Rosado, et.,al,. *A Study of Security Architectural Patterns,* Proceedings of First International Conference on Availability, Reliability and Security, IEEE Computer Soceity, 2006.

[30] Arnab Ray and Rance Cleaveland, *A Software Architectural Approach to Security by Design,* Proceedings of the 30th International Computer Science and Applications Conference IEEE Computer Society, 2006.

[31] Marco Seratini and Neeraj Suri, *The Fault-Heterogeneous Architectural Model,* 26th IEEE International Symposium on Reliable Distributed Systems, 2007.

[32] Per Hakon Meland, Shanai Ardi, Jostein Jensen, Erkuden Rios, Txus Sanchez, Nahid Shahmehri and Inger Anne Tendel, *An Architectural Foundation for Security Model Sharing and Reuse,* International Conference on Avaialbility, Reliability, and Security, IEEE Computer Society, 2009.

[33] Stephen Bode, Anja Fischer, Winfried Kuhnhauser, Matthias Riebisch, *Software Architectural Design meets Security Engineering,* 16th International Conference and Workshop on the Engineering of Computer Science Systems, IEEE Computer Society 2009.

[34] Mark Moriconi, Xiaolei Qian, R. A. Riemenschneider, Li Golng, *Secure Software Architectures,* 1081- 6011 / 97 IEEE.

[35] Jun Han and Yuliang Zheng, *Security Characterisation and Integrity Assurance for Compoennt-Based Software.* 0-7695-0903-7/00, IEEE 2000.

[36] S. Ramamoorthy et. al. *Component-Based Heterogeneous Software Architecture Reliability (COHAR) Modeling* , (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 04, 1280-1285, 2010.