RESEARCH PAPER | Computer Science | Volume : 5 | Issue : 6 | June 2015 | ISSN - 2249-555X

Analaysisof Architecture Security Tips of Object-Oriented Software

KEYWORDS: Security, Software, Object Orientation, Architecture, Software Engineering,

Seyfali Mahini

Islamic Azad University, Khoy Branch, Khoy, Iran

ABSTRACT *To analyze architecture security tips of object-oriented software in which usual security issues are used, the present study tries to analyze security of an under designing system. Object-oriented systems are designed based on different architectures such as COM, DCOM, COBRA, MVC, and Broker. In an object oriented technology, the main component of the system is an object. Each component stems its risk in the system. Security and risk policies associated with this type of software can be calculated for a single component. The total risk can be calculated on the basis of risk factors in the architecture and field. Small risk factors gatherandcreate large riskinthesystemandcandamageit.*

1.INTRODUCTION

The software is a trillion dollar business, and the amount of effort and time spent to develop software is very large. In time of software developing, different architectures are used for the development and security is a major concern in software components. Security issues should be resolved in the early phases of development, and costs incurred to remove the objection if detected in the architecture can be minimum [1]. If the software development is performed in coding phase,deep diagnosis is needed becausefinding the design faults in the code is difficult. Risk analysis should be done for the architectures because the software security plays a very important role in the program. Risk identification is useful andcan be used to take thesoftware security measures in to consideration. Determination of risk quantifies and its impact should be calculated in software development environment because it can have direct concern in this trade.

One question that organizations usually face is how my system is safe. The answer to this question is often difficult. Most of system securities problems are rooted in software which are most vulnerable to the unexpected attacks [2]. In spite of the extensive researches in security engineering, security measurement is still a difficult problem. Though we do not have the security measure with full confidence, we mostly relay on risk measure to assess the security. The use of security measures to assess the risks of aggression is a common practice which is provided by a systematic mechanism for cost and resource optimization. The problem is to provide accurate information on the attacks and their probability. Because the systems are subject to constant changes, the related risks are also affected by these changes. However, risk assessments are not repeated often enough to change the system. Over the time, the initial estimates of the risks got obsolete and maybe result in less secure systems.

2. SOFTWARE ARCHITECTURE

The softwarearchitectureis an essential for the proposedpyramidanalysis method. Eachof the componentsmustbe identifiedand their interactions withothercomponentsof thesystemshouldbe consideredforDTMCmodel. Systemsoftwarearchitecturesshouldbe availablein thestandardform. Information of thecomponents interactionsshould beestimated using the experiencegained from thesimilarsoftware components. Astandardmethodforestimating theprobabilities is transitionof theinter-componentsflow control fromdifferentscenariosrunon theoperatingsystemprofile, reliability, efficiency, demand,and quantitative informationabouttheindividualcomponents. Asthe software development willcontinue, the estimation will improve and thus throughput the analysis accuracy will improve.

3. TEST STRATEGY

Architectureprovidesa structurethrough whichalargecomplexsystemcan beunderstood.Before thesystem construction starts, weaknesses are identifiable. Tocreatesuch a structure, system architecture selectsa set ofcomponents andconnectionsbetweenthemto displayand thus createsaparticularvision [3]. The architectureof asoftwaresystemdefinesthe system in terms ofcomponents andinteractions betweenthem. Differentoptions of the components or differentarchitecturesconnectionscreate different perspectivesofthe system. The testshould beanintegratedpartofthe complete cycle ofsoftware development. The exactmethodof testingprocessdepends on thesoftwaredevelopment life cycle. This cycle maybeadditive orrepetitiveand thus thetest software should bedeveloped using thesametechniqueswith productive software. The object-orienteddesign information hidingtechniquesrequiredifferenttests.Testingof object-oriented softwaresystem is more complicated than that oftraditionalsystem lifecycle. Certain areasofcodebecause of theinformationhiding techniques availableinobject-oriented languagesareunavailable, and this may preventsome of thetesting techniques. Testscorescan bededicatedtoaspecialclasswithoutintegrityis designedtobecompatible. It is possible to give specific access scores to a separate class of test without adjusting the design integrity. In the picture, different phases ofthe software developmentlife cycle(SDLC)are shown. If we let the requirements to change in the next phases except for the software requirements, thus requirements of one system users may change over the time and the requirements lied in the previous phases may be invalid. Inobject-oriented softwaredesign, easy maintenanceand reuse ofcomponentsisthe point focused on [4]. Software qualityattributessuch asintegrity, strength, develop abilityandcompatibilityshould alsobe consideredduring the design process.The main focusin thedesign phase of thetest is on theintegritytest. Theintegritytestof systematiccomponentsconstruction is in the sub-systems andsystems. The components stabilitytest-

Volume : 5 | Issue : 6 | June 2015 | ISSN - 2249-555X

sandwhetherthe components properlytransferandcontrolthe data guaranteesthesuccessful integrity ofthe dependent components.For integritytest, thefunctionaltest orblack boxandstructural test or white boxis frequentlyperformed toseparate thesystemcomponents anddetermine thecorrect behaviorofthe integrated units. Driversandspecific tools are needed forseparating thecomponents and testingthe individualcomponentsbecause theywere integrated. Errormalfunctioncanoccuratany timeofSoftwareRequirementSpecification)SRS(tomaintenanceevenwhen thesystemis working. Object-oriented approachismodular in nature therefore it helps in easy maintenanceandcorrecting the errors.

4. TESTING OBJECT-ORIENTED SYSTEM

Software testing techniques have been evolved over the years and for management of unique issues of object-oriented software testing, traditional software testing techniques must be adapted and new techniques have to come into existence. Object-oriented software architecture is different from traditional software architecture. Features such as hiding, inheritance, polymorphism, and reusability are specific for object-oriented software. Therefore, some of the issues related to object-oriented software testing are different from traditional software testing issues. The main advantage of the object-oriented paradigm is that it provides a uniform structure for all components of the interface process [5]. The focus class of the unit test is in object-oriented software. Resolving the problems at runtime complicates the testing for dynamic annexing. In addition, the paradigm shifts from waterfall model of software development to iterative and incremental model of software development, and this leads to the fact that the object-oriented software testing becomes repetitive and increasing in nature. The error may appear anywhere in the code. Object-oriented class methodology helps to detect errorsby providing check point from both type of interpretation (compile) and run-time errors for class objects. Check type of the runtime will detect large number of errors because lie invalid objects (a lot of errors cause) are automatically detected and reported.For better system performance, the programming language should avoid leak memory problem. Leak memory logically is allocation of dynamic storage which leads the program to fail in the second request. Namely,the objects which are not used any more do not correct. On the contrary, a large number of samples may lead to leak memory. If the leak memory is severe, it may lead to failure of the program due to memory reduction. Leak memory is occurred by objects that help other objects to continue holding the references, accordingly,they prevent the garbage collecting torequest the held in objects. Table of resource objects can be used to identify such resources.

5. OBGECT-ORIENTED SOFTWARE ARCHITECTURE

Theobject-orientedsystemsmaybebasedonthefollowingarchitectures:

5.1Componentobjectmodel(COM)

COM has Application Programming Interface(API) medium which support component creation for use in a particular application integration. COM has several components to interact during API design. Nevertheless, to interact the components should resort to a binary structure specified by Microsoft. Until the components resort to this binary structure, the written components can work with different languages.

5.2 Distributed COM (DCOM)

DCOM is an extension of COM which causes the interaction between network-based components. While COM processes can be run on one machine but in different address spaces, DCOM allow the processes to expand on a network. As DCOM environment will be available, the components running on different platforms can interact withDCOM. To obtain aCOM / DCOM architecture, component-level testing is important [6]. Three types of property should be included in the definition of the functional part of the componentof which the application state of the test can be made. Everyoperation, in terms of the restrictions,is defined on its input and output which have been shown in pre-conditions and post-conditions form.

5.3 Common Object Request Broker Architecture (COBRA)

Object management group creates the COBRA to make it possible for the objects to connect with each other and supports the layer server client from the distributed objects, and data processing needs no extra processing on PC to complete the transaction. Objectshave platforms which are dominant in terms ofspace and sharing because they share the resources. All errorsor faults should be tested before the component is integrated in the distributed system. It should be tested aside from others so that it can find and remove the system errors and fault. This test should be done with unite test and integration test. All the uncertainties and implementation of interface inconsistencies should be found before administration of medium identification.

5.4 Model-View-Controller (MVC)

The data model, user medium, and control logic have been separated into three distinct components in MVC; changes in one component can bedone with minimal effect on other parts of the MVC.

6. SECURITY CONSIDERATIONS

Software security considerationsappear from the first phase of the software development even before the emergence of the problem domain-specific architecture. Almost all models and known architectures such as COBRA, EJB, COM, and DCOM take the security as the primary consideration in software architecture. Because COM / DCOM copponents haveaccess to API and Microsoft Windows, bad actors can damage the user's computing environment. To consider this problem, Microsoft applies authoritative code which uses common coding for digital signature of the components. Independent operator ofcertification issuance such as VeriSign are used for identity confirmation of component source, but even the certified code may include some orders which can affect the user environment either by accident or malignantly.

7. SECURITY RISK ANALYSIS

We use the term risk analysis to refer to risks identified and ranking activities in a particular stage in the life cycle of software development. Risk analysis particularly when used in architectural and product design is popular. Mostdescriptions ofrisk analysisemphasizesthat identification, ranking, and risk reduction are continuous process and not a single step in the development stage which is completed in the life cycle. Risk analysis results and risk categories thus drive both into requirements (early in the lifecycle) and into testing (where risk results can be used to define and plan particular tests).An architectural risk analysis method contains several main activities which most include the following result:

Volume : 5 | Issue : 6 | June 2015 | ISSN - 2249-555X

Discussion
Learnas muchas possibleabout the objectiveanalysis.

- Read andunderstand the definitions, architecturedocuments and otherdesigning issues.
- Playwith the software(if thereareexecutable).

- Getagroupdiscussion about thegoal.
- Specify the boundaries between systemandsensitivity/ criticality of thedataset.

- Read thecodeandothersoftware products(includingcode analysistool).

- Identifythreatsandagree on the attackssources. Discuss about the security issuesthat

Surroundthesoftware
Talkabouthow the productusesanddetermine areas ofdisagreementorambiguity.

-Identifythe areaswhich are at possible risk, sometimesuse at risktoolsorlists.

-Identifythebehaviorsanddiscussabout the possiblecorrections.

-Gain an understandingofcurrent securitycontrolsandplanning.

During theprocess ofarchitecturalrisk analysis, wefollowthe basicsteps. Risk managementisthefractaltheory. In other words,the wholeproceeding processcan beapplied inseveraldifferentlevels. The mainlevel is that ofthe project. Eachstageofvalidationcycle mayhave a display withthe requirements, design, architecture, andthe otherthingsused fortestingprograms.

8. COST-BENEFIT ANALYSIS

Architecture selection and analysis is performed with cost benefit analysis and can help in the economic aspects of software molded to help determine the optimum. Methods of cost-benefit analysis (MCBA) help to determine suitability of software architecture for the question and answer application environment stems. CBAM has many steps, and each step is checked and then implemented in detail and thus justify the cost of software development.While selection of scenarios and strategies are considered for the systems security, architecturestrategyshouldbe chosenfora particularscenario. For example, ifthere was ascenariothatcallsforincreasedaccess, thenaspecialstrategyshouldbe chosenwhichaddsto thesystemsomeredundancyandthe ability toovercomeover failure. The quality assuranceand quantity determinationof architecturestrategiesextract the benefit information fromcommercial conceptsand thesystem conditions (what is supposed to be better than all the commercial conceptsis the changes in the system and how it functions) to aid architecture strategy. Benefits better understand that a strategy reaches a favorable level of qualityfeaturesthan the architects. In the fourth stage, costs and quantify pick architectural strategy concepts are quantified. The costs are extracted and information is planned from shareholders. There is no special technique for this extraction. It is assumed that in the organization there is a cost estimation and plan method to achieve this goal. Based on the extracted values, the desirability degree is calculated for architecture. One measure of desirability degree (divided by the cost-benefit ratio) is performed for each architecture strategy. Inherited uncertainties in each of these values which can also help in the final stages of decisionmaking are calculated. The above sixsteps help tocalculate theextracted amountsas an acceptablebasisfordecisionmakingprocess which include technicalmeasuresandtradecriteriafordetermining whetheraparticularchangein thesystemprovidesa high returnon investmentandthesuitabilityor not (ROI).

9. ARCHITECTURE SECURITY MODEL

Creating a component-level security and security architecture model of the software is not easy because there should be a tool to provide information for the applications and IT systems to extract, categorize and prioritize security requirements and should enter its security needs at the architecture level. This methodology focuses on security considerations entering in to the early stages of the development life cycle. The model may be useful for documenting and analyzing the security aspects of the respective systems and may be used to guide future developments and changes in these systems. This model is elaborate on the development of the security architecture.

10. COCLUSION

Security issues are discussed in detail. All software architectures used for software development are discussed in this article. Security is essentially necessary for software architecture because remove theloopholesin thesystem and it can be done with risk analysis. In this method, commercial objectives determine the risks, the risks guide the methods, the methods gain the measurement, the measurement leads the decision support, and decision support directs the back work and program quality. We used the usual notions of security engineering to create a model to assess security. The model was risk-based, which is widely accepted as a measure of security. Our approach emphasizes that each part of the system own its risk, no matter how small it is. Software components, people, and communication channels provide security risks. Only one comprehensive analysis of all of these provides real results on the system security. Security requirements of a system should be finalized in the design phase.

REFERENCE]1[.R. Allen, D. Garlan, Software Testing Approach for Detection and Correction of Design Defects in Object Oriented Softwarel Journal of Computing, Volume 4, Issue 2, April 2011, ISSN 2141-9617, Page No. 44-40. |]2[. M. Allen, D. Ronua, —A Formal Basis for Software Architectural, ACM Trans. Software. Engineering. Methodology, 1999. 6(3): ps. 210-241 |]3[.Gelperin and M. Mahaw, —An introduction to software development, Advances in Software Engineering and Knowledge Engineering, edited by F. Gelperin and E. Allen, World Scientific Publishing Company, 2003. |]4[. B. Aelperina, E R etzel,"The growth of software testing" Commun. ACM 31(6), Jun. 2008, ps.387-692 |]5[.M. Shaw, "Managing Software Security Risks", IEEE Computer, 30(4), March 2012, sp. 91–101. |]6[.ShawSudbury —Proactive Cyber Defense and Reconfigurable Framework for Cyber Securityl International Review on computer andSoftware (IRCOS) Vol.2. No.2. March 2010, Pages 88-94, India, sp.62-70. |