Aug 16th, 3:45 PM - 4:05 PM

Software Engineering: Security characterization for evaluation of software architectures using ATAM

Asad Raza
NUST-Military College of Signals, Pakistan

Haider Abbas
Royal Institute of Technology, Sweden

Louise Yngstrom
Stockholm University, Sweden

Ahmed Hemani
Royal Institute of Technology, Sweden

Security Characterization for Evaluation of Software Architectures using ATAM

Asad Raza, Haider Abbas, Louise Yngström and Ahmed Hemani

***Abstract*— Significant technological advancement in the current electronic era has influenced the work processes of private and government business entities. E-Government is one such area where almost every country is emphasizing and automating their work processes. Software architecture is the integral constituent of any software system with not only cumbersome modeling and development but require heedful evaluation. Considering this aspect we have highlighted in this paper, security evaluation of an ongoing e-society project ESAM using Architectural Tradeoff Analysis Method (ATAM). ESAM is a web based system intended to provide e-services to the Swedish community residents. ATAM is primarily used for architectural evaluation aligned with the quality goals i.e. performance, availability and modifiability of an organization. We present research analysis for characterization, stimuli, and architectural decisions to evaluate software architecture with respect to security measures using ATAM. This security characterization will serve as a tool to evaluate security aspects of a software architecture using ATAM. We believe that ATAM capability of evaluating software security will provide potential benefits in secure software development.**

***Index Terms*: Software Architecture, Security Characterization, Security Evaluation, Quality Attributes**.

I. INTRODUCTION

In this present net-centric age, electronic media is considered as a core need for a business organization, a government or even an individual. Almost every country in this present electronic era is emphasizing on providing governmental services electronically to the citizens. ESAM is a project launched by VERVA[1] to provide E-Services to the citizens and business organizations in Sweden. Software Architecture is the qualitative description of large and complex software systems [10] like ESAM. It is the building block of such systems and requires careful evaluation for its validation. It is crucial for understanding the high level relationships among the different components of a system and analyzing high level properties and tradeoffs of a large and complex system [1].

ATAM is a scenario based method used for the analysis of software architecture against its quality goals like performance, availability and security [2].

A. Raza is with the Military College of Signals (NUST), Pakistan (email: asadraaza@gmail.com)
H. Abbas is with the Royal Institute of Technology, Sweden (email: haidera@kth.se)
L. Yngström is with the Stockholm University, Sweden (email: louise@dsv.su.se)
A. Hemani is with the Royal Institute of Technology, Sweden (email: hemani@kth.se).

[1] Swedish Administrative Development Agency

978-1-4244-4609-4/09/$25.00 ©2009 IEEE

ESAM architecture evaluation required security to be considered as an essential quality attribute. ATAM provides the characterization for performance, availability and modifiability but we didn't find any details about security characterization, stimuli, response and architectural decisions. Characterization for a quality attribute like security provides basic rules and guidelines to carry out ATAM evaluation [2]. In this paper, first we will briefly describe ATAM and then security characterization, security scenarios examples and evaluation results from ESAM project using ATAM.

II. SECURITY EVALUATION IN SOFTWARE DEVELOPMENT PROCESS

Software systems are designed by considering non-functional quality attributes like performance, reliability, availability, maintainability etc. Security as a non-functional attribute is often addressed too late in the software development process [7]. The negligence of security aspect at design level leads to poor security of a software system and prone to many threats like information theft, unavailability of services, low productivity, increase in maintenance cost and many more. Best practices of secure software development [8] recommend to consider security as an integral part of the software development life cycle. The system architecture should explain how security issues are dealt at every point in the software system. The possible threats to the system and system response should be modeled in architecture and can be tested and verified in the testing phase of SDLC. Figure 1 shows the summary of how security is incorporated in each phase during secure software development. ATAM moves one step forward by validating the architecture with respect to the security requirements and security scenarios. It reduces the maintenance cost to a large extent by eliminating the threats and trade-offs at an early stage.

Security has always been considered as an overhead with respect to cost and time consumption in software development world. It is considered as a technical problem rather than a risk in achieving business goals. Evaluation methodologies like Common Criteria [13] is considered expensive and resource hungry procedure [9]. ATAM is designed as a cost effective evaluation procedure for software architectures with respect to non-functional quality attributes. This helps to motivate the validation of software architectures at primary stage. We have focused security evaluation using ATAM to promote cost effective security evaluation at architecture level in software development process.

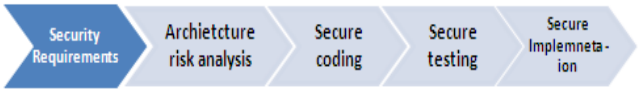

Figure1. Security in Software Development life cycle

III. ATAM PHASES

Duration of the ATAM evaluation depends on system size and available architectural details. ATAM consists of four phases and each phase comprises of sub activities as described below.

1. Presentation Phase

Presentation phase includes introduction of ATAM to stakeholders, presentation of business drivers, presentation of software architecture and how it addresses the business drivers.

2. Investigation and Analysis

Identification of architectural approaches, generation of quality attributes tree based on quality factors that comprise system and analysis of architectural approaches based on specified quality attributes. It also includes identifying architectural risks, sensitivity points, and tradeoffs.

3. Testing Phase

Brainstorming for prioritizing scenarios involving stakeholders and analyzing them again. This may uncover additional architectural approaches, risks, sensitivity points, and tradeoffs.

4. Reporting Phase

Complete documentation of the findings of architectural styles, scenarios, attribute specific questions, utility trees, risks, sensitivity points and tradeoffs.

IV. SECURITY CHARACTERIZATION FOR ATAM PROCESS

Software architecture has become a significant area of research in the field of software engineering [11] [12]. The evaluation of software architecture against the quality attributes calls for the precise characterization of those quality attributes [2]. This characterization serves as the starting point for evaluation of any architecture. During ESAM evaluation we needed to consider security attribute using ATAM and we could not found ATAM's characterization for security like it does for performance, availability and maintainability [2]. In the following sections we will define the security characterization for ATAM. We have followed the same structure of characterization for security as elaborated in the ATAM process for other quality attributes [2].

Quality attribute characterization consists of architectural decisions, external stimuli and responses. External events that affect the architecture directly or indirectly and cause the architecture to respond are known as external stimuli. Architectural responses are deterministic quantities to evaluate the architecture with respect to quality requirements, these quality requirements should be expressed in concrete terms which can be gauged. While the components, connectors and properties of architecture that are directly involved in achieving the attribute responses are called architectural decisions. For example, the external stimuli for Availability are hardware faults and software faults. The architectural decisions for availability are hardware redundancy, software redundancy, voting, retry and failover. Responses for availability include reliability, levels of service, mean time failure and availability [2].

Based on the characterization structure of performance, modifiability and availability described in ATAM process, we have characterized security attribute as shown in figure2.

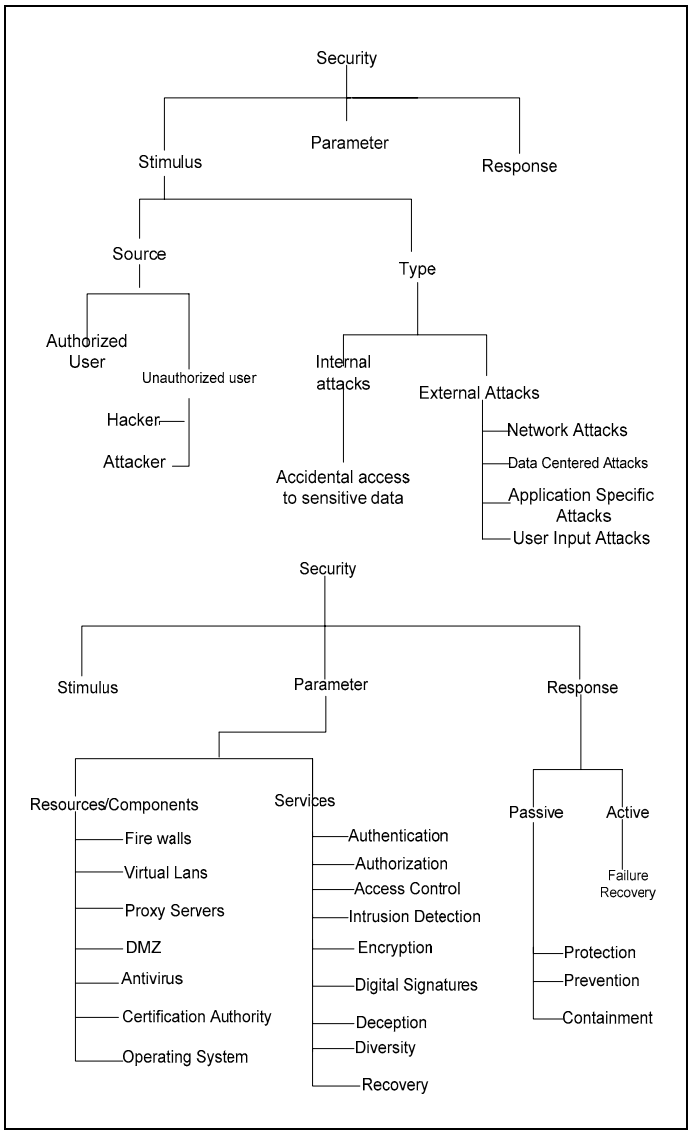

Figure2. Security Characterization for ATAM

The external stimuli for Security as we specified are authorized and unauthorized users. The architectural decisions include authentication, authorization, access control, intrusion detection, encryption, deception, diversity

and recovery. The architectural responses for security include protection, prevention, containment and failure recovery. This characterization will serve as the milestone for security analysis of software architecture. The parameters and response may vary from system to system depending on the nature of system and architectural decisions.

Characterization Description

In figure 2 if the stimulus is unauthorized user or disgruntled staff and the type of attack is data centered attack then the system response should be protection or prevention. The architectural parameters could be firewalls that restrict access to the exposed components, proxy application for filtering input/output data. If for the same stimulus the type of attack changes to user input attacks the architectural response is protection but the architectural parameters responding to this attack could be authentication, signing, access control for prevention of user impersonation etc. The characterization shown in figure 1 is given in a more comprehensible manner in Table 1.

TABLE I
Security Characterization

Stimulus	***Type of Attack***	***System Response***	***Parameters /Services***
Authorized user	Accidental Access to Sensitive Data	Prevention	Implementation of • Authentication • Authorization • Access control
Unauthorized user	Network	Prevention/Containment	• SSL communication • VLAN implementation • Incorporating firewalls that scans for open ports, keeps a check on network protocol vulnerabilities, controls access to exposed components like COTS ,prevents other network attacks like DOS, DDOS.
Unauthorized users / Disgruntled staff	Data Centered	Prevention /Protection	• Access control • Authentication • Encryption of stored data • Using hash functions • Emails and virus scans • Principle of least privilege • Filtering input/output data streams using application proxy • Incorporating firewalls
Unauthorized user	Application Specific	Prevention/Containment	• Limiting the available services • Use special services to protect application server • Isolation of exposed server using DMZ • Principle of least privilege • Backup services for availability • Maintaining services on different hosts
Authorized/Unauthorized users	User Input	Prevention	• Implementing signing, access control, authentication. • Escaping and filtering • Input validation

V. UTILITY TREE GENERATION

After successfully identifying stimuli, architectural parameters and architectural responses for security, the next significant phase in security evaluation is utility tree generation. Utility tree expresses the overall goodness of

a system [2]. It provides: 1) A method for making the system goals more specific and more tangible. 2) A comparison between the quality attributes related to each other. 3) Provides a comprehensive view to stakeholders for characteristics of the architecture that are critical to the success of system [5].

The utility tree for ESAM shown in Figure 3 has been generated on the basis of quality attributes identified by the evaluation team with mutual consensus to VERVA and other stakeholders. In this paper we have presented utility tree only for security only (See Figure 3).

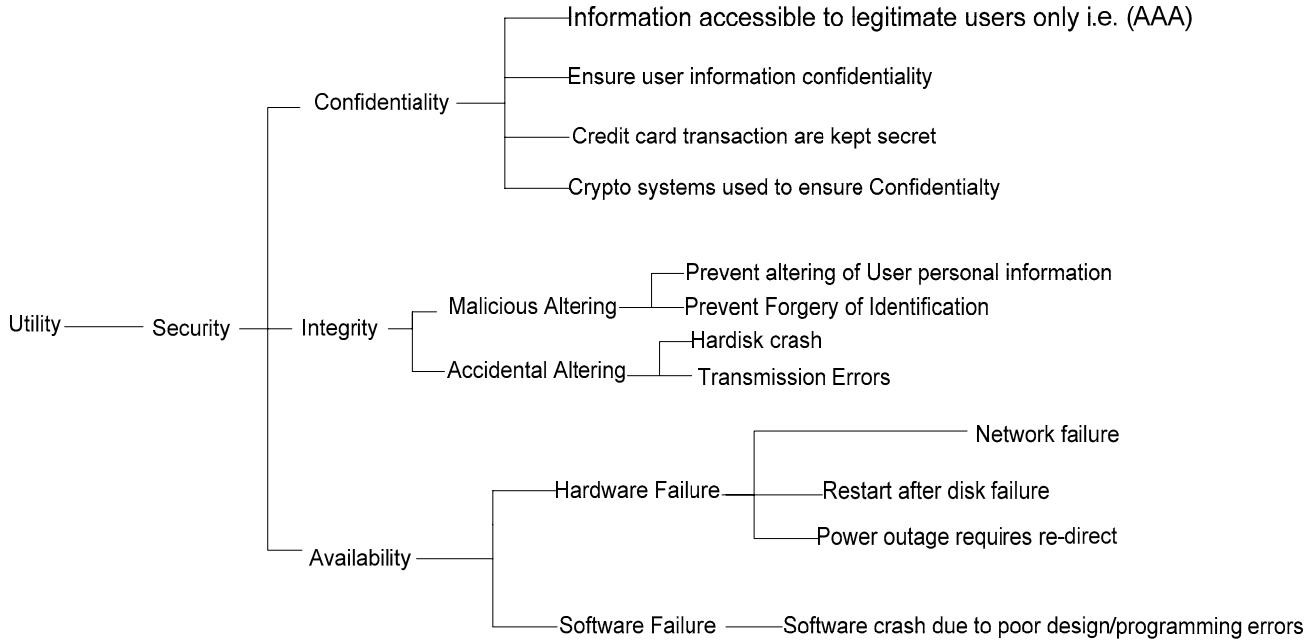

Figure3. Security Utility tree for ESAM

VI. SECURITY SCENARIOS ELICITATION

Scenario elicitation is the most significant activity while carrying out ATAM evaluation. A scenario not only helps to determine whether the architecture meets a specific functional requirement, but also predicts the system qualities like security, performance and modifiability [6]. Scenarios are the main drivers of the ATAM process and the analysis depends on scenario elicitation. A well-written scenario leads to an accurate result. Scenarios can be elicited in different ways depending on the type of the system architecture and evaluators.

We classified scenarios for ESAM in the following two types

1. Scenarios with respect to each quality attribute.
2. Scenarios based on E-Services (Use case)

Relevant and high priority scenarios were selected through voting of stakeholders. Some of the high priority security scenarios elicited with the help of security characterization chart and utility tree are presented below.

Security Scenarios for ESAM

1. In ESAM a sender and receiver may wish to be able to determine 1) if a message has been modified in transit 2) if the point-to-point encryption is not appropriate because of intermediaries or system architecture choices. In this scenario the system either signs the arbitrary portion of document or uses digital signature to guarantee that the message has not been modified in transit.

2. If an authorized user enters the wrong user name password more than 5 times, the user account is blocked and the administrator is notified for appropriate action.

3. There could be a failure of service due to E-ID service, Forms, SHS nodes, SHS packaging services, SHS forwarding etc. If there is a service failure at any point in the ESAM architecture, it is expected to be discovered and fixed in minimum time.

4. All the agreements of stakeholders are stored in secure central repository/SHS[2] nodes.

5. The intermediary SHS server forwards a message to the ultimate receiver on behalf of an initiator. The sender/ultimate receiver wishes to

[2] Spridnings- och Hämtningssystem

enforce the non-repudiation property of the route. The intermediate message service handler that appends a routing message must log the routing header information. Signed routing headers and the message sender/readers must be logged at the message handler that passes the message to the ultimate sender/receiver to provide the evidence of non-repudiation.

VII. SECURITY EVALUATION RESULTS

We presented some of the security scenarios of ESAM in aforementioned section. The complete evaluation results of ESAM architecture is beyond the scope of this paper. But some of the evaluation results are summarized in this section to provide an insight about the plug-in.

The architecture was analyzed for observing the response for each scenario and it exposed some potential issues for ESAM system related to documentation, security and other quality attributes.

a) Architectural Documentation

A detailed architectural document is the prerequisite for a successful evaluation of software architecture. ESAM was lacking of a well-written architectural document. After holding regular meetings with all the stakeholders and organizing workshops it was managed to get a deep insight of the architectural decisions. This helped to cope the issue of architectural documentation.

b) Single Point of Failure (SPOF)

In ESAM, the exchange of messages and communication is dependant on SHS server and all the E-Services are provided through SHS system. If SHS system fails due to some reason then all the E-Services will become unavailable. Therefore ESAM architecture is prone to single point of failure threat and it does not provide any evidence of dealing with SPOF.

c) Error Location

ESAM is a distributed architecture with multi party involvement and this requires precise error tracking. ESAM architecture is lacking of a suitable mechanism to exactly locate the point of failure in architecture. Error/failure detection could take undefined time period to be located / fixed. The services could become unavailable for unspecified period of time that may cause huge business losses.

d) Unidentified Error Reports

There is no authorization required for end users to use error-reporting service in ESAM. Anonymous error reporting can lead to disinformation and wastage of resources.

VIII. CONCLUSIONS

Our goal in introducing the security characterization is not to claim that we have created an exhaustive taxonomy for security, but we have tried to suggest a framework that will help in writing effective security scenarios during the architectural evaluation of a software system using ATAM method. We believe that our research in using ATAM as a security evaluation process will encourage security evaluation at architecture level in software development process as it is aimed to be a cost/time-effective solution as compared to other security evaluation methods like Common Criteria. ATAM evaluation can be carried out with minimum input due to its interactive nature. Security risks and trade-offs are identified in the early phase of development process hence reducing the maintenance cost and avoiding business goals.

ACKNOWLEDGMENT

We pay our sincere thanks to Christer Marklund, project leader for SHS development project from VERVA (Swedish Administrative Development Agency) for his help and valuable suggestions during this research.

REFERENCES

[1] David Garlan and Mary Shaw, “An Introduction to software architecture”, CMU-CS-94-166, January, 1994.

[2] Rick Kazman, Mark Klein, Paul Clements, “ATAM: Method for Architecture Evaluation” Technical Report CMU/SEI-2000-TR-004 ESC-TR-2000-004, Software Engineering Institute, Carnegie Mellon University, August 2000.

[3] Kurt Helenelund, Anders Bremsjö, Stephan Urdell, Bo Sehlberg, Jan Lundh, Christer Marklund “SHS Version 1.2 Architecture” *The Swedish Agency for Public Management*, Oct 2003.

[4] Rick Kazman, Len Bass, Gregory Abowd, and Mike Webb, "SAAM: A Method for Analyzing the Properties Software Architectures", *Proceedings of the 16th International Conference on Software Engineering*, Sorrento, Italy, May 1994, pp. 81-90.

[5] Mildred N.Ambe, Frederick Vizeacoumar, “Evaluation of two Architectures Using ATAM”, April 29, 2002, p: 4

[6] Rick Kazman, Mario Barbacci, Mark Klein, S.Jeromy Carriere, Steven G. Woods, “Experience with Performing Architecture Tradeoff Analysis Method”, Software Engineering Institute, Carnegie Mellon University.

[7] Adam Sachitano, Richard O. Chapman, John A. Hamilton, “Security in Software Architecture A Case Study” *Proceedings of the 2004 IEEE Workshop on Information Assurance United States Military Academy*, West Point, NY June 2004.

[8] Razvan Peteanu“Best Practices for Secure Development” v 4.01, Oct 2001.

[9] W. Jackson,” Under attack Common Criteria has loads of critics, but is it getting a bum rap?”, *Government Computer News,* August 10, 2007.

[10] Mary Shaw and Paul Clements, “The Golden Age of Software Architecture: A Comprehensive Survey”,

CMU-ISRI-06-101, *Institute of Software Research International School of Computer Science*, Carnegie Melon University Pittsburg, February 2006, pp: 4.

[11] Hofmeister, Christine; Nord, Robert; and Soni, Dilip *Applied Software Architecture*, Addison-Wesley, 1999.

[12] Bass. L, Clements. P.and Kazman, R. *Software Architecture in Practice*, 2nd ed. Prentice-Hall, 2003.

[13] Common criteria for Information Technology security evaluation, Version 3.1, CCMB-2006-09-001.