Int Jr of Advanced Computer Engineering and Architecture
Vol. 1, No. 1, June 2011

www.ijacea.yolasite.com

Tool to generate security template in design phase

RESHMA S. GAYKAR, DR.PROF. SHASHANK JOSHI, YOGESH S.TOTARE

Department of computer Engineering, Bharati Vidyapeeth University,
Pune , Maharashtra 4110043, India
reshma.gaykar@gmail.com
Department of computer Engineering, Bharati Vidyapeeth University,
Pune , Maharashtra 4110043, India
sdj@live.com
Sword Global India Pvt.Ltd.
Pune, Maharashtra 411014, India
ytotare@gmail.com

ABSTRACT

Providing a secure system is a difficult task. Faulty and incomplete design can result into providing unsecure system. Also there are chances of missing the requirements while passing it from analysis phase to the development phase. Most of the time if the requirements are not properly documented in analysis phase and if the same requirements are not considered as a part of design phase then in the development phase those requirements get lost. In case of dealing with security requirements same scenario will create security breach because of lost requirements and this will lead the system to be unsecure. The proposed system considers the security requirements in the design phase itself and generate a code template for the given security parameters. In the design phase, if the architect or the system designer knows what are the security parameters to be considered for the system then the proposed tool is helpful for them to generate the security template. Proposed system considered the Aspect-Oriented Risk-Driven Development (AORDD) methodology for developing secure systems, which considered few aspects like Authentication, RAD and SAS to generate security code template.

Keywords: UML, AORDD, SSL, SAS, RAD

1. Introduction

Secure software is a software development problem. Its solution is the responsibility of every member of the software development team–from managers and support staff to developers, testers and IT staff. Security must be on everyone's mind throughout every phase of the software lifecycle. A misstep in any phase can have severe consequences. However, finding a solution is not easy. The problems associated with application security are getting worse with time. Aging legacy software, which was never developed to be secure, is the foundation on which modern, highly connected and business-critical software is operating. The difficulty of patching these older systems and integrating newer applications has served to make the problem worse.

In a proposed system a tool that assists an end user to generate a code template in the design phase of SDLC based on the security parameters analyzed in the requirement phase is provided. The tool is based on the concept of AORDD (Aspect-Oriented Risk Driven Development). This methodology begins with designers defining system assets, identifying potential attacks against them, and evaluating system risks. When a risk is unacceptable, designers must mitigate the associated threat by incorporating security mechanisms methodically into the system design. Designers must specify where and how aspects should be incorporated into a system design. In the proposed tool we are considering the three security aspects First Communication level security which includes SAS and SSL protocols. Second, RAD (Resources Access decision) which specifies the access to the particular device on the system likes HDD, CPU, and ROM etc. Third, Security services which includes authentication roles and privileges on particular resources.

2. Background

AORDD Methodology is targeted toward the development of complex systems where there are competing project and security goals. Under these conditions, it can be difficult for a designer to determine how different parts of the system, designed to meet different goals, interact with each other. Performing security analysis in the context of the whole system can help a designer understand these interactions better. The AORDD Methodology has two steps that must occur prior to any analysis. First, system architects and designers must create system functional models. Since the Unified Modeling Language (UML) is the de facto software specification language used in the industry, our tool chain requires that these models be specified using the UML 2.0. Second, designers must perform a risk assessment of the system. Risk assessment begins with system stakeholders (e.g., end users, designers, developers, and management) identifying sensitive system assets such as system information or services. Different stakeholders can place different values on an asset, so the stakeholder and the value they assign to a particular asset are both needed in our methodology. Designers must develop security requirements for these assets and identify possible threats against them. Threats are attacks on the system with the goal of compromising assets. Designers and security experts must also rank the potential threats. Designers also identify potential security mechanisms that can mitigate specific risks, as part of the assessment process. Designers can be aided in a risk assessment by organizational experience or security experts.

3 Proposed System

In this system we are proposing a tool that will generate a source code template for a given correct UML diagram. This tool will take three aspects into consideration. First, communication level security which includes SSL (Secure Socket Layer) and SAS (Secure

Authentication Services) is applied to the use case diagram. In this scenario we draw a communication line between the Actor and the Case and also provide the communication type either SSL or SAS. According to the type selection code is generated in a package which can be exported later to the real time applications. As shown in Fig.1 communication link between actor and the case has a SSL protocol enabled, which will generate the template code shown in Fig.2

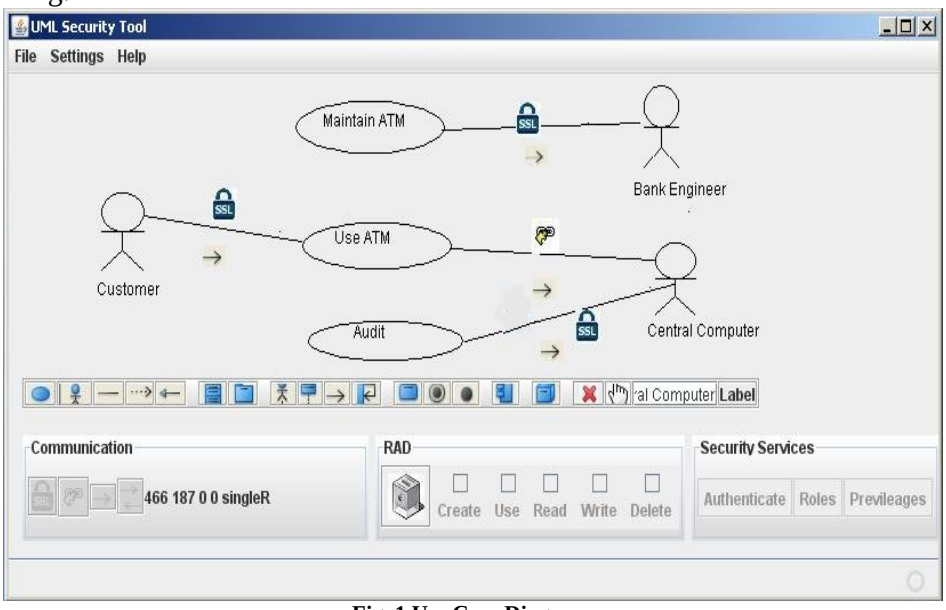

Fig. 1 Use Case Diagram

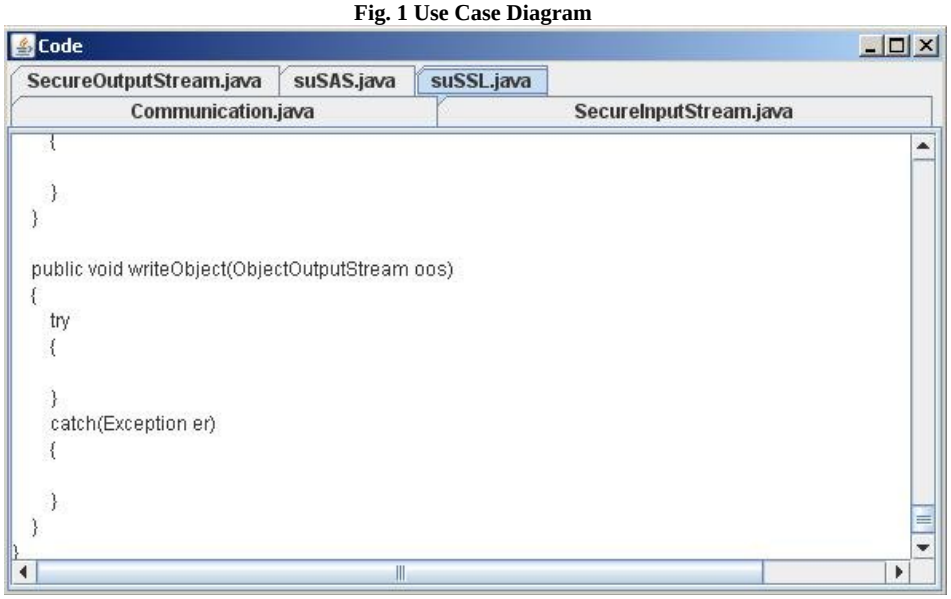

Fig. 2 Template Files After Applying SSL and SAS

Second, RAD (Resources Access decision) which specifies the access to the particular device on the system likes HDD, CPU, and ROM etc. We are providing create, read, write and delete operation for the specified devices. According to the selection criteria this toll will generate the template with all the operation as methods. Third, Security services which includes authentication roles and privileges. Second and Third aspect will need a class diagram to generate the code template associated with the particular class. As shown in Fig.3 Class has the RAD and Security services associated with it, which will generate the template code shown in Fig.4

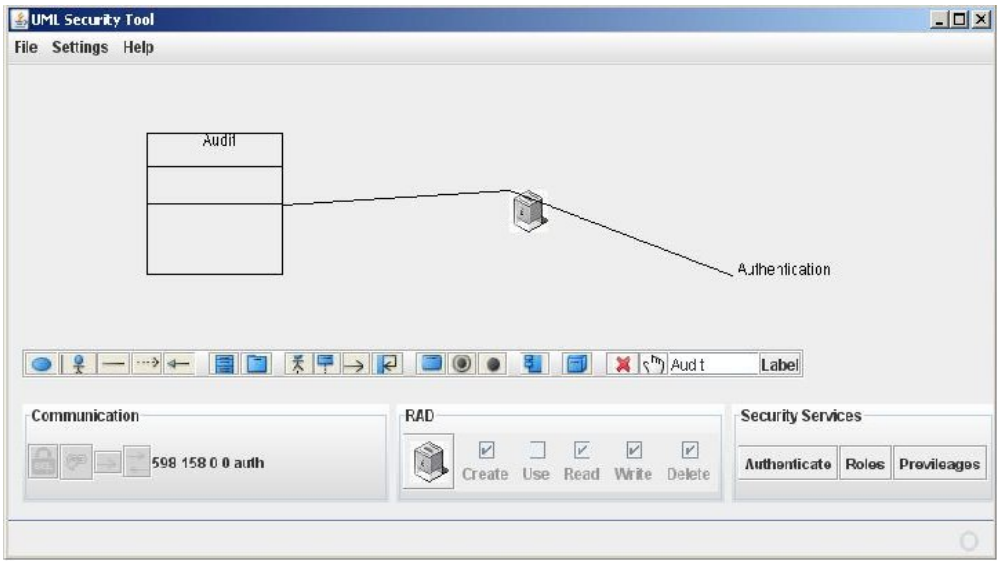

Fig. 3 Class Diagram with RAD Mechanism

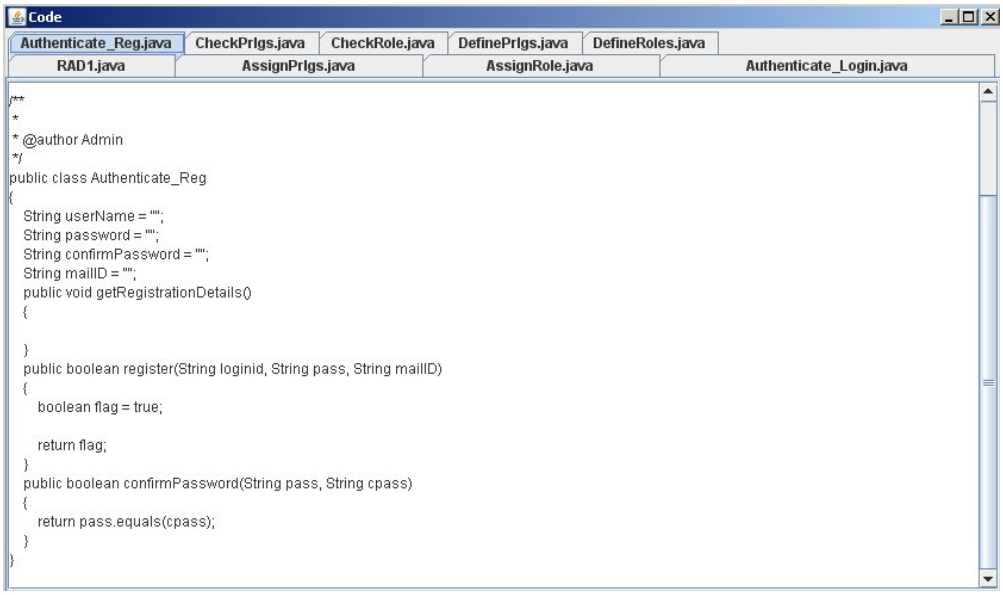

Fig. 4 Template Files After Applying RAD Mechanism

5. Conclusion

With the help of the proposed tool developer's life became easy. One of the major problem in the with the secure software is, if the security stuff is missed in design and development phase then that stuff takes much more efforts to recover later stage. Because of the complex roles and privileges in the complex software systems there are much chances of missing of such complex requirements. So if the security parameters are identified in the design phase then in the design phase itself with the help of those parameters we will be able to generate the code. So there are very few chances of missing out the security code in the development phase. With the help of proposed system using the security requirements for the role based application we can minimize such problems.

6. References

[1] Geri Georg, Kyriakos Anastasakis, Verification and Trade-Off Analysis of security Properties in UML System Models (IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 36, NO. 3, MAY/JUNE 2010.)

[2] Siv Hilde Houmb, Geri Georg, The Aspect-Oriented Risk-Driven Development (AORDD) Framework, February 16, 2005