I.J. Modern Education and Computer Science, **2014, 2, 27-32**
Published Online February 2014 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2014.02.04

An Analysis of Application Level Security in Service Oriented Architecture

Said Nabi
Shaheed Zulifikar Ali Bhutto Institute of Science and Technology (SZABIST), Islamabad, Pakistan
Email: saidnabi115@gmail.com

M. N. A. Khan
Shaheed Zulifikar Ali Bhutto Institute of Science and Technology (SZABIST), Islamabad, Pakistan
Email: mnak2010@gmail.com

***Abstract*—**In computing, the software elements like objects and components emphasize on reusability using design tools of abstraction and separation of concerns. Software architecture has appeared as an initial idea to develop huge, complicated and heterogeneous distributed systems successfully. Service Oriented Architecture (SOA) combines services together to make systems having a greater impact on the way software systems are developed. SOA addresses the need of standards-based, loosely connected, and distributed computing which is protocol independent. It is not easy to ensure the secure transaction of data, where the movement of data occurs through loosely connected services. A number of techniques have been proposed in the contemporary literature to guide the SOA implementation in distributed system. These techniques offer certain benefits, but pose some challenges alongside such as the use of meta-data as framework and standard, contract documents, security patterns and security adviser, etc. The objective of this research is to provide a comprehensive analysis of various approaches used to provide application level security to the web services in SOA. These approaches have been compared based on a number of parameters. In addition, we critically evaluate different security methods used in SOA. The study also discusses some future directions in this domain.

***Index Terms*—**Application Security, Cloud Computing, Service Oriented Architecture, SDLC, Agile Software Development.

I. Introduction

Requirement engineering is considered as one of the most critical phases in software engineering, specifically, in software design and development. If errors are introduced at the requirement stage, then they remain undetected till the later stages of software development process and [1]. Requirement engineering addresses the issues of requirement collection to design and develop the desired software. Requirement engineering has a direct impact on all the stages of software development including software design, architecture, implementation, testing and deployment. Software architecture deals with design and development of the abstract level structure of the software. It consists of a number of architectural elements like components and connectors, which are assembled in such a way to satisfy the functional and performance requirements [2]. There are a number of architectural styles or patterns used by the software architects including layered systems, event-based, object-oriented, data-abstraction and implicit invocation, etc. Although these styles provide sufficient space of architectural choices to the architects, but alongside pose challenge for the architects to realize the tradeoffs while selecting the best suitable style in a particular situation and environment [3].

Service Oriented Architecture (SOA) had been one of the most focused areas of research since the last decade for providing various solutions using the concept of services. Various researchers have proposed specific models to customize the service discovery, registry and composition for its effective working. A number of bugs have to be fixed throughout the life cycle of development of software products. In addition, new requirements always surface until the completion of the software product. These result in incurring significant cost on top of the maintenance of a product. In order to ensure that functionality of the developed software product works fine, service oriented architecture (SOA) is a solution to such problems due to its loosely coupled and interoperable services architecture.

The development of SOA application is different from traditional software development in term of analysis and design. SOA application requires undergoing analysis and design phases. Analysis phase produces candidate services from the business requirements. Business analysts and service architects emphasize on the usage of standards to refine the candidate services. Therefore, the formal definition of business processes is very important in SOA. The testing and development phase of SOA, however, are similar to the traditional development processes [4].

SOA is a set of services which are the collections of software component and carry out business process independently. Services must have the properties like loose-coupling, self-containment and should have well

defined independent interfaces. Self containment of the services mean it would be able perform their functionality independent of the other services. Loose-coupling mean services communicate with each other through sending messages and are not aware of technical details of other collaborator services [5].

Although the information security standards like identification, authentication, authorization, confidentiality, integrity and availability have same connotation in SOA based applications, but the agile implementation of SOA makes it difficult to ensure secure SOA implementation within the organizations. There is a lesser possibility for successful implementation of secure SOA without specific information security guidelines [7]. The principles behind the agile manifesto include:

- The highest priority is assigned to satisfy the customer through early and continuous delivery of software releases.
- Welcome the changing in the requirements, even late in development phase. Agile processes harness change for the customer's competitive advantage.
- Deliver functioning software frequently, i.e., in the time span ranging from couple of weeks to couple of months, with a preference to the shorter time-scale.
- Business people and developers should work together on daily basis throughout the project.
- Build projects team comprising the motivated individuals. Provide them a conducive environment and necessary support they need besides entrust them enough authority and empowerment to get the job done.
- The most efficient and effective method of conveying information within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers and user should be able to maintain a constant pace for an indefinite period.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity is essential.
- The best architectures, requirements and designs emerge from self-organizing teams.

Extreme programming puts more emphasis on the teamwork. All the stakeholders, particularly managers, customers, users and developers are all equal partners of a collaborative team. Extreme programming is much like a jigsaw puzzle as it consists of many small pieces — individual pieces do not make sense but when combined together, they depict a complete picture of the project. The amazing aspect of extreme programming is its simple rules; the rules may seem inelegant and perhaps even naive in the first place, but in fact, they are based on sound standards and principles. Extreme programming approach improves a software project in five different ways: communication, simplicity, feedback, respect and courage. In addition, there are five rules for extreme programming technique: planning, managing, coding, designing and testing. Each of these rules is further subdivided into small chunks of guidelines. In this paper, we only use planning and managing rules of extreme programming.

In contrast with the independent software development, extreme programming offers a magnitude increase in productivity. In addition, the combined understanding of the system by multiple people leads to improvements in the design. Further, the maintenance of this system is simplified to a great extent. To streamline different phases of the software lifecycle, first we need to model the requirement document so that we can analyze and integrate the software artifacts. Designers can ensure completeness and consistency of the system by generating models using the requirement documents. In RE, selection of pertinent tools and techniques in accordance with the type and complexity of the project is fundamental to eliciting requirement. This section outlines prominent RE tools and technique along with their role. These techniques are by and large classified into four main categories namely, classic/traditional techniques, cognitive techniques, modern and group elicitation techniques and contextual techniques. Each of these categories consists of a set of various techniques that are grouped together on the basis of their common characteristic and peculiarities.

Requirements are actually customer's statements of scope. In requirements finalization process, the stakeholders play an import role. A stakeholder can be defined as anyone who is directly or indirectly affected by the system being developed or deployed. Stakeholders are broadly categorized into two major classes — user and customer. User ordinarily uses the system and customer refers to those persons who have requested for the development of the system and are responsible for approving it. There may be a number of people who participate in the development of a system like business analysts, designers, coders, testers, project managers, deployment managers, use case designers, graphic designers etc. and are customarily considered as stakeholders.

The rest of this paper is organized as follows: the literature survey on application level security in SOA is provided in section II. A critical analysis of various techniques discussed in the literature is provided in section III. Finally, we conclude in section VI along with potential future directions to this research.

II. Literature Review

Imamura et al. [7] emphasize that the current tools used for the configuration of security assets of the web services presents a technology aspect, where user should fill the gap between configuration and security needs manually. This leads to mis-configuration problem and extra configuration costs.

Baghdadi [8] describes the need for web service architecture that can control all the levels of web service stack which may include: describing, registering, managing, monitoring, deploying, wrapping, and discovering fundamental software components to self-repeatedly composing software. The proposed framework has four components including:

(1) web services specification, which includes functional and non-functional requirements, communication style and its location on the web and are specified by machine readable language like WSDL;
(2) web server or application server is used to deploy web services;
(3) the candidate functionalities existed in the legacy systems are wrapped during the designing and development of the web services; and
(4) service as a component of composition of software supporting business processes shows that web services architecture (WSA) should be viewed from usage perspective to present adaptable software composition applying business process.

Chetty and Coetzee [9] proposed information security components for SOA environment based on ISO/IEC 270002-2005 security standards, SOA design principles and information security governance frameworks. The information security framework helps organizations to determine the security controls for SOA. SOA design principles provide guidelines for developing interoperable and agile service logic, which includes composition, discoverability, statelessness, independence, loose-coupling, abstraction, and reusability and service contract. Information security components like security services are used to minimize the level of exposure of information security approaches. These components include policy information framework, information security model, information security management and SOA information security governance.

Delessy and Fernandez [10] state that flexible and modular design of software applications have a negative impact on the security of software applications, and propose a pattern driven approach to develop secure SOA applications. Such models help solve security issues in SOA based applications and help create flexible and modular design that is imperative for usage in cross organization context.

Schnjakin et al. [11] emphasize that the complex nature of security policy languages leads to error-prone and inefficient creation of web services. Security patterns are used to present the expert knowledge and understanding related to the security techniques, which enables users to configure the fundamental framework. Security patterns cover the security domain expert knowledge and the local administrator of the system provides detail information about the relevant security of the underlying infrastructure. The local system admin also provides the key storage location and access criteria. The security advisor uses this information for generating the enforceable policy.

Pandey et al. [12] assert that the advent of the new software engineering models for SOA, Software as a Service (SaaS) and Service Oriented Software Engineering (SOSE) has brought new security challenges and risks. The authors propose a framework which authenticates the billing services and clients to minimize security risk by rapidly changing the encryption key. SaaS paradigm consists of modular architecture.

Delessy et al. [13] maintain that the wider usages of computer and its applications demands for dynamic trust establishment and identity exchange protocols. To support this idea, authors propose an architectural pattern for identity management systems. This pattern is based on the language, which make patterns easy to be used in software development life cycle to develop secure software applications. To share the trust relationships, a circle of trust has been proposed which represents a federation of service provider. Identity federation pattern centralizes the administration of users of an organization and provides a uniform format for users to communicate identity information between different security domains.

Chetty and Coetzee [14] performed an evaluation of information security governance frameworks by comparing them with ISO/IEC 17799 (2005) governance standard to find the degree to which the governance frameworks solve the issues related to the information security in SOA. A closer look of the evaluation results shows that the governance frameworks are not able to addresses the information security issues in entirety.

Sidharth and Liu [15] describe that unknown consumers calling web services and SOAP API create security vulnerabilities. Authors propose a framework to handle the current problems of web services security (WS-Security). IAPF techniques are aimed to be the element of implementation structure and design of the web services at both the application and network level. To avoid and lower the effects of the Distributed Denial of Service (DDOS) and Denial of Service (DOS) attacks, IAPF techniques enable web services developers at application level to implement and design SOA producers.

Dikanski and Abeck [16] affirm that the existing security architectures are not compatible with the current security engineering approaches. This leads to the redundant development of security paradigms and do not provide relevant information to the security engineering process. The authors propose a view-based model for service oriented security architectures. The proposed model presents security information related to different architectures for various views like security engineering processes, security integration and security service view.

Nurse and Sinclair [17] focus on enhancing the existing approaches of web services security within e-businesses by proposing a complete and standard framework named Business-Oriented Framework for Web Services Security (BOF4WSS). This approach mainly concentrates on understanding and targeting the current inter-organizational problems that arise due to e-businesses.

Menzel and Meinel [18] articulate that business process paradigms explain the workflows at a high level to support business analysts and verify the business requirements. The authors propose a security meta model for SOA. Security Meta model give a base for exchange of information and model interactions. These models also illustrate the fundamental objects, associations and related roles in service oriented architecture. These models constitute participants and argue based digital identities to define the brokering of identity information. A policy model is proposed to combine and connect the security needs and sketch the effect of different entities e.g., data transfer entities and interactions. This model defines the security goals for representing the security requirements. For validating the applicability of the proposed model, a mapping to web services policy and web services security policy is described. Amir et al. [19] highlights different agile software development methods including scrum framework - a well known agile method. Mahmood et al. [20] highlighted various aspects regarding service composition in the context of SOA.

The field of software development is facing several challenges due to incorporation of nonstandard models and tools in the requirement engineering (RE) phase. Because of this, the failure rate of software projects is increasing rapidly. Proper emphasis on the requirement engineering process is considered as a key to the success of a software project [21]. A number of companies have employed Global Software Development (GSD) methodology as a useful tool for their software development practices. GSD is a contractual relationship between client and vendor organizations in which a client outsources all or some part of its software development activities to a vendor. The vendor in return provides the agreed services in lieu of certain amount of remuneration. The main reasons to select the GSD technique include reduced cost, faster development and access to skilled manpower. Though GSD is emerging as an effective technique, but it suffers from many challenges like poor communication, lack of trust and coordination. These challenges pose serious risk to the smooth execution of the GSD projects [22]. Rehman et al. [23] analyzed the requirement engineering processes, tools/techniques and methodologies.

According to Schummer and Lukosch [24], distributed pair programming is an agile software development methodology where two programmers located at different geographic locations jointly work using a collaborative real time editor. The key difference between pair programming and distributed pair programming is that the programmers in the latter technique are located at different geographic locations, and various communication means are required to be made available for practicing this methodology. The distributed extreme programming, especially distributed pair programming, is destined to failure unless proper tools are used that support social practices.

Khan et al. [25] reviewed requirement issues in software development. Ambiguous and unrealistic requirements are major source of failure in the software-intensive systems. Requirements engineering processes are complex as most of the requirements engineering documentation is written in natural languages which are less formal and often distract the designers and developers. Requirements management is a continuous process throughout the project lifecycle and relates to documenting, analyzing, tracing and prioritizing requirements and then finally controlling changes. The main issues related to requirements management are usually social, political and cultural. Software requirement engineers who gather the requirements generally consider that such issues are beyond the scope of their profession as they deem them within the project management ambit.

Recent technological advancement and development of new standards have lead to the creation of new methods for designing and development of web applications. These web applications are linked through independently published web service components. These web applications are also called web services. A system that has the tendency of integrating multiple web services automatically in a transparent way is considered as part of the web service oriented system. Service Oriented Architecture (SOA) is an architectural method that is used for the creation and usage of business services in the form of web services.

III. Critical Evaluation

A critical evaluation of the different approaches, frameworks, techniques and policies proposed for addressing security issues in SOA is provided in this section. The critical evaluation is based on the following parameters: proposed method, model or standard used, key area addressed and merits/benefits. The critical evaluation is presented in Table-I.

Table 1. Critical Evaluation of Security Issues in SOA.

Ref	Proposed Method	Model or Standard Used	Key Area Addressed	Merit
[7]	Model driven security configuration for WS	MDA and SOA viewpoints	Refining the security requirements to bridge the gap between security requirement and configuration.	Make easy the configuration of security requirements for web services.
[8]	Meta data framework	UML, WSDL	Guidelines for SOS development process.	Reduce effort, time and cost for developing web services.
[9]	Comparison of security frameworks with ISO/IEC 27002:2005	ISO/IEC 27002:2005	Framework provides an initial point for implementing controls and developing guidelines. Provide effective information security components for developers and managers.	Protects vulnerable.
[10]	Security pattern based approach	MDA view points	Reduce security challenges.	The proposed system can be applied at specific architectural style.
[11]	Pattern driven security advisor	Rampart Policy	Helps generate a security configuration for web services. Automate policy generation for the security domain.	Reduces the dependency of web services developer on the security domain expertise.
[12]	Encryption algorithm	Java, PHP, VS	Frequent change in session to avoid brute force attack.	Reduce the chances of brute force attack but can affect performance.
[13]		UML and Use Case	A uniform format for users to share identity information. Provide a pattern language, which is easy to use in developing secure software	It helps define a common format and standard for implementing security in web services.
[14]	Comparison of NES with ISO/IEC 17799 (2005) controls	ISO/IEC 17799 (2005) controls	Evaluate the SOA governance framework.	The comparison shows that the existing frameworks cannot handle the security issues holistically.
[15]	UDDI, WSDL and SOAP	SOAP API, XML	IAPF framework for preventing DOS and DDOS attacks. Guidelines for securing web services.	-
[18]	Meta model for model driven approach	WSDL, SOAP and UML	Provides high level policy pattern and describe security constraints and map these constraints and policy patterns to WS-policy and WS-security policy.	-

IV. Conclusions

Software development, as a whole, is a complex process and on top of it, the requirements keep changing during the development phase. Software configuration management happens to be the most critical part as it necessitates doing considerable modification in the software design and code. Agile software development process provides a solution to such a changing environment. Agile methods use an incremental approach to develop high quality software within time, cost and other associated constraints through several iterations. There are some prominent factors in software project management e.g., scope, cost, time and quality. Software engineering explores constructive and dynamic ways to manage the entire project lifecycle. Change in the requirements is a critical phase in any of the software development process and managing the requirement change is an open issue in the literature for many decades.

Due to the agile nature of SOA, it is difficult to ensure the secure implementation of SOA within the organizations. In this study, we have made an effort to provide a review of the concepts related to different techniques, practices and SOA security frameworks. On the basis of this study, we conclude that the security standard like identity, authentication and authorization are more challenging to implement at the application level. Most of the techniques proposed in the literature focus on the use of security patterns, which provide some basic concept and understanding about the security domain needed for the web services developers. But, it is still a challenge for the developers of the web services to implement security modules for the web services. To enable developers to implement security patterns successfully there is a need for further automation and improvements in the model.

REFERENCES

[1] A. Chakraborty, M. K. Baowaly, A. Arefin, A. N. Bahar. The Role of Requirement Engineering in Software Development Life Cycle, Journal of Emerging Trends in Computing and Information Sciences, ISSN: 2079-8407, Vol. 3, No. 5, pp: 723-729, 2012.

[2] P. Kruchten The 4+ 1 view model of architecture. Software, IEEE, 12(6), 42-50, 1995.

[3] D. Garlan, & M. Shaw, "An introduction to software architecture," 1994.

[4] N. A. Delessy, A Pattern-Driven Process for Secure Service-Oriented. In Workshop on Security in Object-oriented Systems, Florida Atlantic University. Vol. 70, p. 79, 2008.

[5] D. T. Sanders, J. A. Hamilton Jr,., & R. A. MacDonald, "Supporting a service-oriented architecture, "Proceedings of the 2008 Spring simulation multiconference. Society for Computer Simulation International, 2008.

[6] A. Arsanjani," Service-oriented modeling and architecture," http://www.ibm.com/developerworks/ webservices/ library/ws-soa-design1, 2004.

[7] T. Imamura M. Tatsubori Y. Nakamura, C. Giblin, "Web Services Security Configuration in a Service-Oriented Architecture,n" WWW, pp. 1120-1121.ACM, 2005.

[8] Y. Baghdadi, "A metadata for Web services architecture: A framework for service- oriented software development," GCC Conference & Exhibition, 2009 5th IEEE Issue, On page(s): 1 – 6. March 2009.

[9] J. Chetty, M. Coetzee, "Towards An Information Security Framework For Service-oriented Architecture," IEEE 2010.

[10] N. Delessy and E. B. Fernandez. "A pattern-driven security process for SOA applications," Proceedings of the 3rd Int. Conf. on Availability, Reliability, and Security (ARES 2008).Barcelona, Spain, 2008.

[11] M. Schnjakin, M. Menzel, and C. Meinel. "A pattern-driven security advisor for service- oriented architectures," Pro 6th Workshop SWS (in conjunction with 16th ACM CCS), ACM Press, Chicago, USA, pages 13–20, 2009.

[12] T. Pandey, D.S. Kushwaha, B. Singh, Authentication and billing framework for service oriented architecture, in Proc. Int. Conference on Systems, (ICONS 09), pp.91–95, 2009.

[13] N. A. Delessy, E. B.Fernandez, & M. M. Larrondo-Petrie, "A pattern language for identity management," In Computing in the Global Information Technology (ICCGI), International Multi-Conference on (pp. 31-31), IEEE, 2007.

[14] J. Chetty & M. Coetzee, "Evaluating Information Security Controls Applied By Service-oriented Architecture Governance Frameworks," ISSA2009.

[15] N. Sidharth and J. Liu, "IAPF: A framework for enhancing web services security," in 31st Annual International Computer Software and Applications Conference (COMPSAC), Beijing, China, pp. 23–30, 2007.

[16] A. Dikanski and S. Abeck, "A View-based Approach for Service-Oriented Security Architecture Specification," in The Sixth International Conference on Internet and Web Applications and Services, St. Maarten, The Netherland Antilles, 2011.

[17] J. R. Nurse and J. E. Sinclair, "BOF4WSS: A Business-Oriented Framework for Enhancing Web Services Security for e-Business," in 4th International Conference on Internet and Web Applications and Services (ICIW). IEEE Computer Society, pp. 286–291, 2009.

[18] M. Menzel and C. Meinel. "A security meta-model for service-oriented architectures," In Proc. SCC, 2009.

[19] Amir, M., Khan, K., Khan, A., & Khan, M. N. A. (2013). An Appraisal of Agile Software Development Process. International Journal of Advanced Science & Technology, 58.

[20] Mahmood, A., Ibrahim, M., & Khan, M. N. A. (2013). Service Composition in the Context of Service Oriented Architecture. Middle East Journal of Scientific Research, 15(11).

[21] Khalid, M., ul Haq, S., & Khan, M. N. A. (2013). An Assessment of Extreme Programming Based Requirement Engineering Process. International Journal of Modern Education and Computer Science (IJMECS), 5(2), 41.

[22] ul Haq, S., Raza, M., Zia, A., & Khan, M. N. A. (2011). Issues in Global Software Development: A Critical Review. JSEA, 4(10), 590-595.

[23] Ur Rehman, T., Khan, M. N. A., & Riaz, N. (2013). Analysis of Requirement Engineering Processes, Tools/Techniques and Methodologies. International Journal of Information Technology and Computer Science (IJITCS), 5(3), 40.

[24] T. Schummer and S. Lukosch, ―Supporting the Social Practices of Distributed Pair Programming, CRIWG, LNCS 5411, pp. 83–98, Springer-Verlag Berlin Heidelberg. (2008).

[25] Khan, M. N. A., Khalid, M., & ul Haq, S. (2013). Review of Requirements Management Issues in Software Development. International Journal of Modern Education and Computer Science (IJMECS), 5(1), 21.

Said Nabi is pursuing for MS in Computing (Software Engineering) at Shaheed Zulfikar Ali Bhutto Institute of Science and Technology (SZABIST), Islamabad, Pakistan. His research interests include Information Security and Service Oriented Architecture.

M. N. A. Khan obtained D.Phil. degree in Computer System Engineering. His research interests are in the fields of software engineering, cloud computing, cyber administration, digital forensic analysis and machine learning techniques.

