International Journal of Security and Its Applications
Vol. 11, No. 11 (2017), pp.31-40
http://dx.doi.org/10.14257/ijsia.2017.11.11.03

Security Problems of SOA Applications

Muhammad Qaiser Saleem

College of Computer Sciences and Information Technology, Al Baha University, Al-Baha, Saudi Arabia
muhammad.qaiser.saleem@gmail.com

Abstract

Service Oriented Architecture has gained the popularity in the market because workflows of a business process can easily be realized by the composition of services. Post-development stages express the real strength of the SOA paradigm when a new business process can be realized by just composing existing services and a new application is developed by just assembling existing reusable services. However; due to increase in the number of services, connectivity between the services also increases which causes the rise of the security risk exponentially. Security, which is one of the most important aspects of any software system, is often neglected or given less importance while developing SOA applications. This paper compiled the work of different researchers working in the area of SOA security. The focus of the paper is to present the security problems of SOA applications highlighted by the several authors. These security problems must be addressed for the development of secure Services Oriented applications.

Keywords*: SOA Application, SOA Security, SOA Security Standards, Model Driven Security*

1. Introduction

In today's market, the Information Technology (IT) environment is Network/Internet-centric. Cloud computing, Service Oriented Architecture (SOA), and Software as a Service (SaaS) *etc.* are popular architectural styles. This Internet/Network-centric IT environment provides the IT agility which is the hot demand of today's business [1, 2]. Currently, SOA is the best available IT environment for enterprises for achieving agility, interoperability, and other goals. SOA paradigm promises "*1) rapid application development to significantly enhance corporate agility 2) automated business processes and 3) multi-channel access to applications*" [3].

Business and IT domains can easily merge in SOA environment which facilitates the application development. Software applications are deployed over the Internet as a service. To develop business applications, these services are composed/integrated within or across organizations to form Internet-based systems [4]. The paradigm of SOA promises inter-operability and integration ensuring the availability of resources in the form of services over the network. The SOA paradigm utilizes services as a fundamental element for developing applications [5] and makes the application development easy by coupling services over the Internet or intranet [6]. A business application can be developed as a runtime orchestration of a set of services. Moreover, SOA has transformed the Internet from being a repository of data to a repository of services [7]. In addition to that, SOA is also a *design model* where application logic is encapsulated within service which interacts via a common communication protocol [8, 9]. Post-development stages express the real strength of the SOA paradigm when a new business process can be realized by just composing existing services and a new application is developed by just

Received (July 12, 2017), Review Result (November 6, 2017), Accepted (November 11, 2017)

ISSN: 1738-9976 IJSIA

assembling existing reusable services [3]. Furthermore, SOA is an *architectural style* in which software applications are developed by integrating the loosely coupled reusable services through their standard interface. Services are independent of platform, location, language, location. Furthermore, these services may be requested from the provider or locally developed.

Along with all the positive aspects of SOA, it is also a very complex environment where software applications are comprised of several distributed components such as web servers, databases, and storage nodes, computing nodes *etc.* and these components are distributed across different independent administrative domains. Furthermore, SOA is a loosely-coupled, complex, heterogeneous, distributed architecture which is composed of complicated firewalls topologies, and intermediate server where organizational resources and assets are visible by business services [6, 10]. Managing such a complex environment is not an easy task.

Security, one of the most important aspects, is often neglected or given less importance while developing SOA applications. With the increase in connectivity among the services, security risks rise exponentially. Business services located in various businesses are composed to form application, these businesses have their own security infrastructure [6]. These security infrastructures may work among each other or not, it is a big question. Furthermore, *loose coupling* nature of SOA paradigm also affects the security of business applications. Web services are re-configurable, re-useable and have to serve in several different scenarios. Same would be the case expected for security control *i.e.* it should be fit in the new situations and realizes the security objectives in the different environment [11], however, is security control work in the new scenario? it is a big question. Moreover, to enforce the *security goals*, several security implementations , security protocols, access control models are emerged during the past few years [6, 10]; however, focus of these SOA security protocols and security standards is towards technological level [12], *i.e.* they do not deal with the high level of abstraction.

The objective of this work is to highlight the security problems of the SOA environment highlighted by different researchers in their work. These security problems of the SOA environment must be addressed for a secure development of SOA applications.

Initially, in section 2, the paper presents an overview of the SOA environment, its architecture, building blocks, and standards, so a reader can have an idea of the SOA environment. Later on, in section 3, the security problems, which SOA applications are facing in these days, are discussed in detail. These security problems are divided into nine groups with a suitable title.

2. Service Oriented Architecture

In SOA environment, software applications are comprised of reusable, loosely coupled services which are integrated through their standard interface. Services are independent of location, platform, and language; moreover, they may be locally developed or requested from the provider. A software application can be realized as a runtime orchestration of a set of services. Services are used but not owned by the user and they reside on the provider's side [13-15]. The SOA is also called a "*Find, bind and invoke paradigm*" [4, 16] as shown in Figure 1.

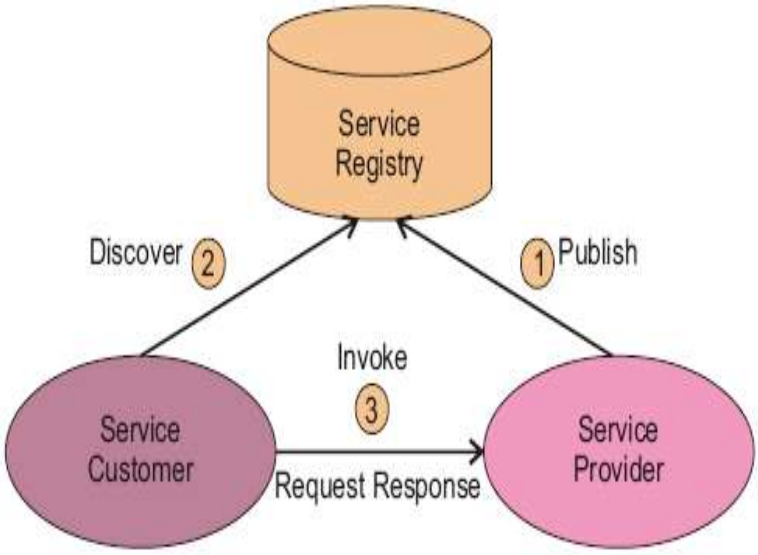

Figure 1. Collaboration of Services in an SOA Environment [4, 16]

Service provider, publish the description of service in a service registry. Service consumer/user search the service according to their description in the registry and use the services if found [4].

2.1. Building Block of SOA

A service is the basic building block of SOA paradigm. A service can be defined as "*A service is an implementation of a well-defined piece of business functionality, with a published interface that is discoverable and can be used by service consumers when building different applications and business processes*" [17]. These services are regarded as platform-independent, autonomous, computational elements that can be programmed, published, described, discovered, and orchestrated using standard protocols for building software applications [18]. In [15] authors describes different characteristics of a service such as discoverable, interoperability, self-contained *etc.* A service must contain these characteristics for an ideal SOA implementation; however, in practice, SOA application relaxes or limits these characteristics. In [17] authors called these characteristics as service-level-design principles and enlist many more properties for services.

Technically, a service consists of three parts: 1) Contract: It provides the formal and informal specification of the service. 2) Interface: It is a technical representation of the operations provided by the service which a client can invoke. 3) Implementation: This is the actual logic of service [16] as shown in Figure 2.

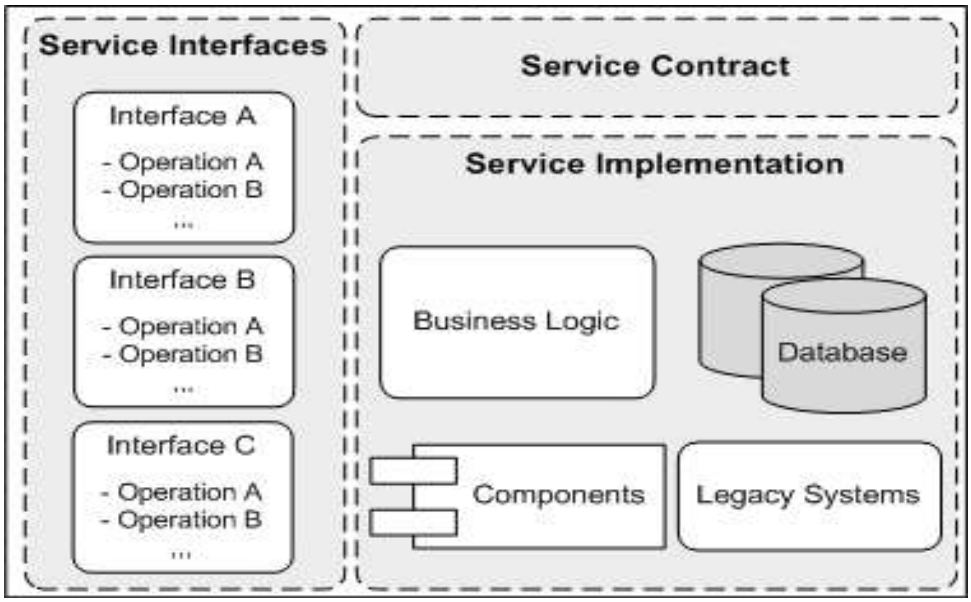

Figure 2. Essential Service Elements [16]

An SOA paradigm can be implemented with different technologies like the Common Object Request Broker Architecture (CORBA), Web services and JINI (pronounced GEE-nee; loosely derived from the Arabic for magician) *etc*. as represented in Figure 3 [15]. However, Web services technology is a widespread instantiation of an SOA [15, 19]. Web Services has high interoperability, well-defined abstraction layer approach, and broad industry support [20]. Due to broad industry acceptance of Web services, web services and SOA names are used interchangeably [3] which is a common misunderstanding.

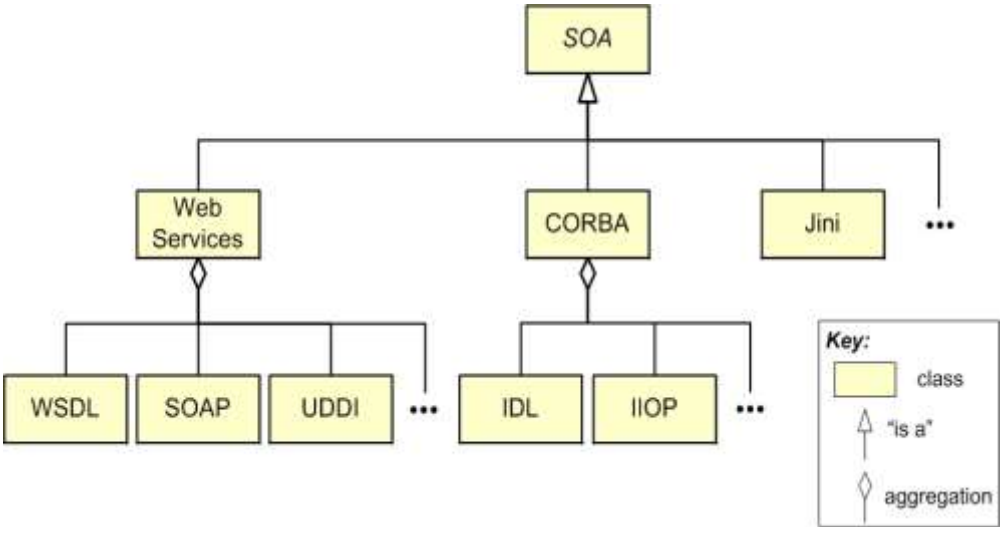

Figure 3. SOA and SOA Technologies [15]

2.2. SOA Standards

There are several XML based standards which lies the foundation of the Web Services technology e.g. SOAP (Simple Object Access Protocol), UDDI (Universal Description, Discovery, and Integration), WSDL (Web Services Description Language) *etc*. [11, 15] as can be seen in Figure 4.

Figure 4. Protocol Stack of Service Discovery, Description, and Invocation [11]

WSDL is used for the service interface description, SOAP messages are used for the communication between services and UDDI is used for the description and discovery of services into/from the service registry. Service provider; publish the description of service in a service registry. Service consumer/user search the service according to their description in the registry and use the services if found [4].

3. Security Problems of SOA Applications

After having a brief discussion about the SOA environment, the core of this paper is presented in this section *i.e.* the security problems of SOA environment. Different authors, working in the area of SOA security have highlighted several security problems of SOA application. For the sake of simplicity, these security problems are grouped into nine separate groups and given a suitable title. A detailed discussion is provided below.

3.1. Security is not Unified with Software Engineering Process

Ideally, security must be unified along the software engineering process for SOA application development. However; in practice, it is implemented in an ad-hoc manner *i.e.* during the implementation phase or during the system administration phase or sometimes it is outsourced [21].

In many cases, security is left to the developer and added after the implementations of functional requirements. The development of SOA system is very complex; which a developer alone cannot meet anymore. Sometimes security is added at the time of integration of distributed applications. SOA applications are coupled over various protocols and network technologies, just adding security to software applications is not a realistic approach [6, 22] because all the required security information is not available at the downstream phases [6, 23]. This approach degrades implementing and maintaining the security of the system [24]. Security goals are very complex, which would make the SOA application easily prone to error. Moreover, finding, removing or repairing security defects towards downstream would be a tedious task and it would increase the cost as well [6].

3.2. Security Objectives are not Defined at "Business Process Modelling" Level

Business process modeling is defined as "*a set of activities and execution constraints between these activities*". It is considered as a foundation of an organizational work-flow [12]. Several well defined and standardized notions are available in the market to model a business process UML is one of the examples of these notions. These notions are used by the business domain expert to model the business process depicting certain business logic.

To model the security objectives, business process modeling is the most appropriate layer [12]. Empirical studies reveal that the security requirements can be specified by the Business Domain Experts at high levels of abstraction [21]. However, while

modeling the business process, the focus is towards modeling of functional requirements of the software system and the "*notion of security*" is often neglected. It is happening due to several reasons e.g. no currently available business process modeling language has the ability to model the security objectives [25] and the business domain expert may not be a security expert [21, 26]. Furthermore; two different and disjoint models are created one for security model and the other for system models, moreover, these models are expressed in different ways *i.e.* a security model is represented as a structured text whereas a system a model is graphically represented in a modeling language like a UML [21].

It is obvious that the security expert and business domain expert, both should define their requirements *i.e.* business requirements and security requirements collaboratively during business process modeling. The business domain expert is facilitated with the several standardized and well-defined notations for service orchestrations which are available in the market. However, the problem is the currently available business process modeling standards are unable to capture security goals as the first-class citizen during the business process modeling. As a result, security experts only able to specify security goals at a very technical level causing loss of valuable security domain knowledge [10].

3.3. Differences on "*Notion of Security*"

Business analyst (business domain experts) at one side and technical people (Security experts and developer) at another side; have differences on the notion of security. Business people define security goals at very abstract level *i.e.* in the context of business logic to achieve some business goal. Whereas, technical people want to realize these security goals through some technical means *i.e.* through some implementation or algorithm or protocol [11]. For example "*authorization*" is defined by a business analyst for accessing an application. Technical people have several choices for the implementation of "*authorization*" *i.e.* four-eyes-principle or a certificate based authorization mechanism *etc.* There may be a different implication for implementing a particular authorization mechanism, maybe from a juristic point of view or any other point of view. It may require the re-organization of the whole business process [11].

3.4. Security Requirements are Specified in "*Non-Formalized*" Way.

In practice, while modeling the business process modeling, security objectives are specified in "*non-formalized*" way. Normally these security objectives are specified by the business department as an unstructured text. As natural language is ambiguous, one can derive several meaning from it. If these security specifications are misunderstood by the IT security department, an error-prone and a complicated coordination process between both departments arise. This result in a loss of security requirement sovereignty by the business department which is the owner of the application [27].

3.5. Problems in Current Security Standards for Web Services based SOA Applications

To build a business application, Web Services located in various businesses domains are connected. These businesses domains have their own security infrastructure [6]. To enforce the security goals, several SOA security access control models, protocols, and security implementations have been developed during the last few years [6, 10]. However, the problem is, the focus of these security access control models, protocols, and security implementations are towards the technological level, they do not provide a high level of abstraction. [12, 28]. This approach will lead to security vulnerabilities. It justifies spending more effort in defining security objectives in pre-development phases, where finding and removing an error/bug is easy and cheaper [29]. Furthermore, mastering these

security access control models, protocols, and security implementations is also a daunting task [12, 28].

Currently, plenty of standards like WS-Security, XACML, and WS-Trust *etc.* are available in the IT market to fulfill the security requirements of SOA applications based on Web Services. However, these Web Services-based SOA security standards mainly have two kinds of limitations. Firstly; they only provide the technical details of these security requirements and do not provide low-level abstractions about them. Application design is not their focus, their concern is to abstract the heterogeneity of the middleware platform. It means; business domain expert cannot utilize them during defining the functional requirements; he has to adopt some other means to represents the security requirements. Secondly; enormous growth in these security standards and their inter-dependencies issues make it daunting task for developers for mastering and utilizing them [28].

3.6. Unclear Security Objectives for SOA Applications

Numerous security objectives for the SOA environment can be found in the literature e.g. Confidentiality, Integrity, Availability, Auditing, Non-Reputation *etc.* These SOA security objectives may be different for different stakeholder like business process experts, vendors, consultants, security experts *etc.* Moreover, these SOA security objectives can be related to some specific deployment, technology, business case, governance *etc.* Unclear SOA security objectives result in unclear security implication which is mentioned as one of the topmost issues that limit the SOA benefits and hence slow down the adaption of it in the IT market [1].

Furthermore, current Model Driven Security (MDS) approaches do not describe the consistent selection of security objectives for SOA environment. In MDS approach, a business processes model is enhanced with security objectives [12].

3.7. Inability of the current Business Process Modelling Languages to model the Security Objectives

Currently, business process analysts express the business logic of the SOA applications with the help of a general purpose modeling languages e.g. Business Process Modeling Notions (BPMN) or UML [30]. A general-purpose modeling language has a broader scope which results in several limitations. For example, there may be a situation where these languages are not appropriate for modeling the specific domains e.g. real-time, security, *etc.* Moreover; there may be a situation where the syntax and semantics of the elements of these languages are not able to express some specific concepts of particular systems. Furthermore; elements of these languages are normally too abundant and too general, there may be a situation when these elements may be restricted or customized [31].

Currently it is possible to express some characteristics of SOA applications as a part of business process models such as service orchestration and choreography; however, it is not possible to express the security goals as a part of business process model [10]. "*Currently available business process modeling languages do not have the ability to capture security goals* [25]". Similarly, currently available business process modeling tools such as MagicDraw do not cater all the characteristics of a software system; security is one of them [20].

3.8. Security is Not Defined during Services Composition Modeling

SOA applications are basically a composition of services which are scattered across the Internet. Several web services composition frameworks/methods are proposed [32-35]; however, the notion of security is neglected in almost all of them *i.e.* security objectives are not defined during the services composition modeling. These frameworks just

describe the different combinations of steps/phases for services composition, however; they do not define the security objectives during the business process modeling of SOA applications. People [36] are presenting framework, where they are trying to define the security objectives along the services composition modeling; however, they only define the security along the business process modeling and do not provide any information regarding the security implementation.

3.9. General Security Problems of SOA Applications

Few general security problems of the SOA Applications are discussed in the following paragraphs.

a) In practice, security requirements are implemented in the system with a programming language dependent handcrafted fixed code. Such inflexible code cannot meet the unforeseen challenges of an SOA environment such as patchy platform technologies, change in business logic or workflow variations *etc.* [37].
b) Basically, SOA applications are *virtual-organizations*; the security challenges of SOA applications cannot be met by relying on Proven Patterns and Best Practices [11].
c) "*Loose Coupling*" is one of the prominent characteristics of SOA paradigm which also affects the security of SOA applications. Web services are re-configurable and re-useable and they have to serve in multiple different scenarios. Same would be expected from security control *i.e.* it should be served in the different environment, which is not practical [11].
d) The security objective, "*Authorization*" is the only focus of the current security modeling approaches which is not feasible to enable a comprehensive verification of security policies [12]

4. Conclusion

This paper has compiled the work of several researchers working in the area of SOA security. Several security problems of SOA environment are presented which are highlighted by the several prominent authors working in the area. We believe our efforts will benefit IT practitioner as well as the researcher as well. It will facilitate the IT practitioners in understanding the security problems faced by SOA applications to develop secure SOA applications. Furthermore, our efforts will also facilitate the researchers in the area of SOA security to get the ideas about the research focuses; it will serve as a basis for understanding and further improvements in the said areas.

References

[1] R. S. Ulrich Lang, "Top SOA Security Concerns & OpenPMF Model-Driven Security", Object Security white-paper, Topics Cloud Computing and Security Management, **(2009)**.
[2] D. G. Firesmith, "Engineering Security Requirements", Journal of Object Technology, vol. 2, no. 1, **(2003)**, pp. 53-58.
[3] M. Alam, "Model-Driven Realization of Dynamic Security Requirements in Distributed Systems", Ph.D. Dissertation, University of Innsbruck, Austria, **(2007)**.
[4] D. Y. Xie, S. Zhang, T. Jia, X.-Y. Liang, Z.-Q. Yao and J.-Feng, "An Approach for Describing SOA", in International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2006, **(2006)**, pp. 1-4.
[5] Microsoft, "The Future of Information Technology: Growing the Talent Critical for Innovation", Microsoft white paper, July 2006, http://research.microsoft.com/en-us/um/redmond/events/fs2006/papers/TheFutureofInformationTechnology.pdf (Date Accessed 13-11-2011).
[6] M. T. Yuichi Nakamura, T. Imamura and K. Ono, "Model-driven security based on a Web services security architecture", in IEEE International Conference on Services Computing, vol. 2005, **(2005)**, pp. 7-15 vol.1.
[7] S. Hanna and M. Munro, "Fault-Based Web Services Testing", in Fifth International Conference on Information Technology: New Generations, 2008. ITNG 2008, **(2008)**, pp. 471-476.

[8] T. Erl, “Service-Oriented Architecture: Concepts, Technology, and Design”, Prentice Hall PTR Upper Saddle River, NJ, USA © 2005, **(2005)**.
[9] T. Erl, “SOA Principle of Service Design”, Prentice Hall, **(2008)**.
[10] M. M. Christian Wolter, C. Meinel, A. Schaad and P. Miseldine, “Model-driven business process security requirement specification, J. Syst. Archit., vol. 55, **(2009)**, pp. 211-223.
[11] R. B. Michal Hafner, “Security Engineering for Service-Oriented Architectures”, Springer-Verlag Berlin Heidelberg, **(2009)**.
[12] I. T. Michael Menzel and C. Meinel, “Security Requirements Specification in Service-Oriented Business Process Management”, in International Conference on Availability, Reliability, and Security, ARES '09., **(2009)**, pp. 41-48.
[13] G. Lewis, A. Morris, E. Simanta, S. and L. Wrage, “Common Misconceptions about Service-Oriented Architecture”, in Sixth International IEEE Conference on Commercial-off-the-Shelf (COTS)-Based Software Systems, 2007. ICCBSS '07, **(2007)**, pp. 123-130.
[14] P. N. Asit Dan, “Dependable Service-Oriented Computing”, IEEE Internet Computing, March/April 2009, **(2009)**, pp. 11–15.
[15] R. K. Philip Bianco and P. Merson, “Evaluation of Service-Oriented Architecture”, Software Engineering Institute/ Carnegie Mellon, vol. Technical Report, CMU/SEI-2007-TR-015, September 2007, **(2007)**.
[16] M. P. Papazoglou, “Service-oriented computing: concepts, characteristics and directions”, in Proceedings of the Fourth International Conference on Web Information Systems Engineering, WISE 2003, **(2003)**, pp. 3-12.
[17] L. B. Liam O'Brien and P. Merson, “Quality Attributes and Service-Oriented Architectures”, Software Engineering Institute/ Carnegie Mellon, vol. Technical Note: CMU/SEI-2005-TN-014, **(2005)**.
[18] W. T. Tsai, “Service-oriented system engineering: a new paradigm”, in IEEE International Workshop on Service-Oriented System Engineering, SOSE 2005, **(2005)**, pp. 3-6.
[19] S. G. Antonio Bucchiarone, “A Survey on Services Composition Languages and Models”, International Workshop on Web Services Modeling and Testing (WS-MaTe 2006), **(2006)**.
[20] M. Jensen and S. Feja, “A Security Modeling Approach for Web-Service-Based Business Processes”, in Engineering of Computer-Based Systems, 2009. ECBS 2009. 16th Annual IEEE International Conference and Workshop on the, **(2009)**, pp. 340-347.
[21] F.-M. E. Rodríguez Alfonso and P. Mario, “A BPMN Extension for the Modeling of Security Requirements in Business Processes", IEICE - Trans. Inf. Syst., vol. E90-D, **(2007)**, pp. 745-752.
[22] ORACLE, “Web Services Security: What’s Required To Secure A Service-Oriented Architecture”, An Oracle White Paper, **(2008)**.
[23] Y. N. Fumiko Satoh, N. K. Mukhi, M. Tatsubori and K. Ono, “Methodology and Tools for End-to-End SOA Security Configurations”, in IEEE Congress on Services - Part I, 2008, **(2008)**, pp. 307-314.
[24] J. D. David Basin and T. Lodderstedt, “Model-driven security: From UML models to access control infrastructures”, ACM Trans. Softw. Eng. Methodol, vol. 15, **(2006)**, pp. 39-91.
[25] M. M. Christian Wolter and C. Meinel, “Modelling Security Goals in Business Processes”, Proc. GI Modellierung 2008, GI LNI 127, Berlin, Germany, vol., **(2008)**, pp. 197 - 212.
[26] A. Rodriguez, E. Fernandez-Medina and M. Piattini, “Security requirement with a UML 2.0 profile”, in The First International Conference on Availability, Reliability and Security, ARES 2006, **(2006)**, p. 8 pp.
[27] W. C. Klarl Heiko and E. Christian, “Identity Management in Business Process Modelling: A Model-Driven Approach”, in Konzepte, Technologien, Anwendungen: 9. Internationale Tagung Wirtschaftsinformatik, Wien, 25 -27. February 2009. Teil 1, H. R. Hansen, Eds., ed Wien: Ã– sterreichische Computer Gesellschaft, **(2009)**, pp. 161-170.
[28] M. Alam, “Model Driven Security Engineering for the Realization of Dynamic Security Requirements in Collaborative Systems”, MoDELS 2006 Workshops, LNCS 4364, Springer-Verlag Berlin Heidelberg, **(2007)**, pp. 278–287.
[29] A. Rodríguez, Fernández-Medina, Eduardo, Piattini and Mario, “Towards a UML 2.0 Extension for the Modeling of Security Requirements in Business Processes”, in Trust and Privacy in Digital Business, ed, **(2006)**, pp. 51-61.
[30] W. M. P. Van der Aalst, M. Dumas and A. H. M. Ter Hofstede, “Web service composition languages: old wine in New Bottles?”, in Euromicro Conference, 2003. Proceedings, **(2003)**, pp. 298-305.
[31] A. V.-M. Lidia Fuentes-Fernández, “An Introduction to UML Profiles”, UPGRADE, the European Journal for the Informatics Professional, vol. V, no. 2, **(2004)**.
[32] B. Y. Orriëns, J. Papazoglou and Mike, “Model-Driven Service Composition”, Service-Oriented Computing - ICSOC 2003, Springer Berlin / Heidelberg, pp-75-90, vol. 2910, **(2003)**, pp. 75-90.
[33] I. S. Roy Grønmo, “Towards Modeling Web Service Composition in UML,” INSTICC Press”, Presented at The 2nd International Workshop on Web Services: Modeling, Architecture and Infrastructure, Porto, Portugal, (2004).
[34] D. Skogan, “Web service composition in UML”, in Eighth IEEE International Enterprise Distributed Object Computing Conference, EDOC 2004, Proceedings, **(2004)**, pp. 47-57.

[35] C. Dumez, "Approach dirig´ee par les mod`eles pour specification, la verification formelle et la miseen oeuvre de services Web compos´es", Ph.D. Dissertation, Universite de Technologie Belfort-Montbeliard, France, **(2010)**.
[36] M. Q. Saleem, J. Jafreezal and M. Fadzil Hassan, "A Framework for Model-Driven Development of Secure Web Services Composition", Advances in Information Sciences and Service Sciences (AISS), an International Journal of Research and Innovation, vol. 4 no. 9, **(2012)**, pp. 67-78.
[37] M. H. Mukhtiar Memom and R. Breu, "SECTISSIMO: A Platform-independent Framework for Security Services", MODSEC08 Modeling Security Workshop, **(2008)**.

Author

Muhammad Qaiser Saleem, he is currently a working as Assistant Professor in College of Computer Science and Information Technology, Al Baha University, Al Baha, Saudi Arabia since August 2013. He has completed his PhD. in Information Technology from Universiti Teknologi PETRONAS (UTP), Malaysia in 2013. He has completed his MS in Computer Sciences from Malardalen University, Vasteras, Sweden in 2006. He has obtained his first Mater in Computer Sciences from International Islamic University, Islamabad, Pakistan, in 1998. During his professional career, he remained involved in academia as well as IT industry and being involved in various development projects related to Databases and Data Warehousing environment. Currently, his main research interest is modeling of security requirement during Business Process Modelling in SOA Application.

