IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 2, Ver. XI (Mar-Apr. 2014), PP 135-141
www.iosrjournals.org

Severe SOA Security Threats on SOAP Web Services– A Critical Analysis

Mohamed Ibrahim B[1], Dr. Mohamed Shanavas A R[2]
[1]*Software Solution Architect & Research Scholar, Camp: Malaysia*
[2]*Associate Professor, Jamal Mohamed College, Tiruchirappalli, India*

Abstract: *Enterprise Application Integration (EAI) involves several technologies; among them, the popular and recent one is Service Oriented Architecture (SOA). Mainly, SOA is used for developing loosely coupled distributed applications. Loosely coupled applications are a group of applications (to form EAI) which can operate independently of each other. The early SOA was achieved by using a number of architectures which include DCOM (Distributed Component Object Model), ORB (Object Request Broker), and RMI (Remote Method Invocation). However, these architectures work on their own defined protocols, and these protocols are specific to certain languages and compilers that do not permit the construction of distributed systems over heterogeneous platforms. Currently SOA provides remedies to these issues using common internet protocols in the form of Web Services. A service is a Software component that is well-defined, self-contained, and does not depend on the context or state of other services. The Web Service provides well-defined interfaces for distributed functionalities, which are independent of machine architectures, operating systems, and programming languages. In this way, Web Services has emerged as a dominant paradigm for constructing and composing distributed business collaborations over the web. As Web Services architecture is dynamic and loosely coupled, security aspects must be considered thoroughly at the time of designing, because Web Services require high security. In this paper, the authors critically analyzed few severe SOA security threats and their implications on SOAP Web Services based on the literature study and their real-time experience.*

Keywords: *EAI, Security Threats, SOA, SOA Attacks, SOAP, Web Services Security, WS Attacks*

I. INTRODUCTION

SOA is a model of components; providing an environment for building distributed systems [1], where a service in the distributed systems is a "course-grained, discoverable software entity that exists as single instance and interacts with other applications and services" [2, 3]. Thus, SOA exposures Software resources in the form of services, which can be accessed over network [4]. According to [5], the Service Oriented Architecture is defined as an open agile, extensible and federated architecture comprised of autonomous QoS capable, vendor diverse, interoperable, discoverable and potentially reusable services implemented as Web-Service for organizing and utilizing distributed capabilities that may be under the control of different ownership domains.

The basic architecture of SOA consists of three main components: (i) Service Provider, (ii) Service Registry, and (iii) Service Requestor [6] as shown in Figure-1. The service is a basic concept and core of an SOA. It is the technical representation and encapsulation of high-level business functionality [7]. Service Provider is an entity that creates and provides the services; it also makes a description of the services and publishes them in a central registry, called Service Registry (Universal Description, Discovery, and Integration - UDDI) [8]. Service Requestor is an entity that requires certain functions which are published by Service Providers, to perform its own tasks. The three core operations that are performed in basic SOA architecture are: (i) Publish, (ii) Find, and (iii) Bind [9].

The service provider has to publish the service description (Publish) in order to allow the service requestor to find it. In the discovery (Find), the service requester retrieves a service description directly or queries the service registry for the type of service required. The service requester invokes or initiates an interaction with the service at runtime (Bind) using the binding details available in the service description [10]. Web Services are published with Web Services Description Language (WSDL) interface and they use Simple Object Access Protocol (SOAP) as a communication protocol between Service Requestor and Service Provider [11].

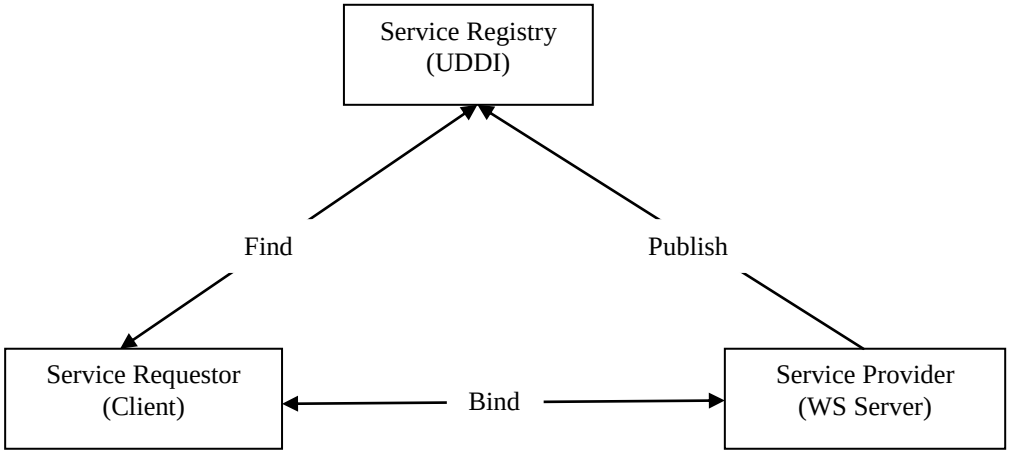

Fig. 1: Basic SOA Architecture

The WSDL is an XML document designed according to the standards specified by the World Wide Web Consortium (W3C) that describes exactly how a specific web service works [12]. The UDDI directory is used as a registry of web services that are available for use in a particular network. It would tell us where to find the service, also to examine the WSDL document. The SOAP specification provides standards for the format of a SOAP message and how SOAP should be used over HTTP. SOAP has been created to transport XML documents from one computer to another and it can be used with a number of standard transport protocols; it is the binding part of web services [13].

The Section II describes the possible severe SOA security threats on SOAP based web services and their harmfulness to destroy the throughput of web services. The Section III critically analyzes the essentiality of security for SOA in order to cater solutions for the SOA attacks. The Section IV concludes the paper and outlines the future work of the authors as an extension to the current research.

II. SOA SECURITY THREATS

Although an SOA can be implemented using different technologies, Web Services technology is commonly used [14]. Web Service infrastructures introduce new threats to web-based applications as well as new challenges when it comes to securing them [15]. Though the fundamental technology of Web Services, XML (eXtensible Markup Language), has given provided Web Services with many advantages but unfortunately it also caused many problems in security concerns too [16].

According to [17], a threat is a possible way a system can be attacked and the threats can be broadly categorized in four classes according to their consequences: (i) Disclosure –it is the unauthorized access to data, (ii) Deception –it is the provision of false data which is believed to be true, (iii) Disruption –it aims at preventing an asset from correct operation, and (iv) Usurpation –it leads to losing control of the asset to an unauthorized entity.

Loose coupling refers to modules of code that are independent of one another; a module can be changed without affecting the operation of other modules [14]. As Web Services architecture is dynamic and loosely coupled, security aspects must be considered thoroughly at the time of designing, because Web Services require high security [17]. Security measures in Web Services environment should be implemented to ensure that data can only be accessed by authorized users, or to provide a certain level of assurance on the identity of service processes when a client is about to pass sensitive information [18].

The threats on SOAP based Web Services include, Message alteration, Loss of confidentiality, Falsified messages, Man in the middle, Forged claims, Capture-replay of message, Replay of message parts, Denial of services, XML external entity attacks, XPath/Field/SQL injection, Harmful SOAP attachments, XML dereference attacks, XML recursion attacks, XML document size attacks, XML flooding, Dictionary attacks, Cookie poisoning, Data tampering, Message snooping, WSDL enumeration, Routing detour, Schema poisoning, Malicious morphing, Memory barrier breach, XML virus, Buffer overflows, Recursive elements, Resource hijacking, Cross site scripting, Eavesdropping, Spamming, IP spoofing, Phishing, Pharming, Malicious programs and Malicious file execution, Worms, Rootkits, Botnets, Identity theft, XML parser attacks, Jumbo payloads, and many more. The following sub-sections briefs few severe SOA threats with their impact on the web services.

2.1 Threats on Identity of Data

These threats are targeted on 'identity' data. The common attacks on identity data are, (i) Dictionary Attacks, where the attacker's goal is to obtain sensitive identity data such as username and password; the attacker systematically tests all possible word in the dictionary to perform this dictionary attack [19], (ii) Brute Force Attacks, where the attackers try with different permutations of any variable that could expose a security hole, for example, if the attacker became to know that the password is 8 characters long, then he tries 256x8 combination of characters to break the password [20], (iii) IP Spoofing, where the attacker fakes original client IP address with attacker's malicious IP address to deceive the web server that the request is coming from original authorized sender [21], (iv) Network Eavesdropping, where the attacker can capture the traffic on the network and obtain sensitive information as web service data are transferred in plain text format [22], and (v) Data Tampering, where the attacker modifies legitimate data with illegal data while it passes over the network [23].

2.2 Threats on Session with Web Server

The Web Service Provider (Web Server) maintains session with its clients for its operations and keeps the 'state' information. These session threats are targeted on capturing web service messages and/or insert false instructions. The severe attacks which are targeted by intruders on accessing web service session include, (i) Replay Attacks, where the attacker captures input messages, and sends it repetitively, which results in overloading of Web Server [24], (ii) Man-in-the-middle Attacks, where there are more possibilities of attacks in the intermediary systems as Web Service SOAP messages pass through multiple intermediate systems before reaching web server [25], and (iii) Session Management Attacks, where the attacker manipulates session data, which may be a session ID, Cookie or URL Parameter, in order to identity vulnerabilities in the application [26].

2.3 Threats on Parsing Web Service Data

There are possibilities of attacks at parsing XML data in web server, where Web Services input and output data are exchanged in XML format. Some of the possible threats related to parsing data include, (i) Recursive Payloads, where the attacker tries to send too deeply nested data to the web server so that the XML parser will be heavily stressed [24], (ii) Oversized Payloads, where the attacker attempts to send unlimited size of input data which will result in Denial-of-Service (DoS) of web server [24], (iii) Schema Poisoning, where the attacker compromises and modifies the original XML Schema, which provides formatting instructions for parsers at server when interpreting XML documents [21, 27].

3.4 Threats on Web Service Implementation Logic

These threats are targeted on finding any possibility of vulnerability of security hole in the deployed web service application. Successfully performed code implementation attacks give the opportunity for attackers, for example to extract the whole XML database. Some of the threats on the code implementation of the web service, which results in security vulnerabilities to attackers include, (i) SQL Injection, where the malicious SQL statements are inserted into XML in order to disrupt the back-end system, for example trying to receive data that it is not authorized to access, or even destroy the data [28], (ii) XPath Injection, as an XML document has no access control or privilege system associated with it, the attacker can to extract even the whole XML database [29], (iii) Cross-site Scripting, where the attacker inspects the web service applications and chooses the one web method that does not filter the input and at which the user is authenticated; the attacker inserts a malicious code in the request and this will be returned to the victim by web server. Then the malicious script will run at client with the privileges of a legitimate script originating from the legitimate web server [30].

3.5 Threats on Scanning WSDL Content

It is apparent that WSDL document includes all of the operations that are available for a web service, for example, with the list of web-methods, the parameters for those methods, and types of input/output. Also WSDL contains detailed data about the available ports for a web service and binding information to web service client. It is straightforward for an attacker to run through all of the operations with different message request patterns until a breach is identified. Lindstrom [24] points out that through scanning the WSDL document an attacker may reveal sensitive information like invocation patterns, types, messages, operations, port types, bindings, and guess other methods. If the attackers successfully get the WSDL file, they may exploit XML wrapping to bypass authentication [31]. It is also possible by an attacker that he/she can cause a DoS condition by submitting special characters/malicious content to a web method, further the attacker may attempt to call other web methods by guessing their names.

3.6 Overflow Threats

This kind of threats targeted on crashing the web server, which handles the web service requests and responses. When performing a buffer overflow attack, an attacker put a larger amount of data than expected into a program variable. The amount of memory, reserved for the operation becomes smaller than the amount of data written to the memory which result in unexpected behaviors to the web service application [32]. Normally, a buffer overflow occurs when a program tries to store more data in an allocated buffer than it was intended to hold. Since the buffers are intended to contain a finite amount of data, the extra information (which exceeds the size of the allocated buffer, and this also should be stored somewhere else) can overflow into adjacent buffers, corrupting or overwriting the valid data held in them. Imagine if the 'extra' data contains malicious code, then it will become security vulnerability for an attacker through which an attacker can trigger some specific harmful actions on the web server.

3.7 Threats Resulting in Denial-of-Service (DoS)

This is another kind of threats that is targeted on crashing the web server to stop responding. Performing DoS the attacker may attack a router, firewall, or proxy server with the goal of making them unusable. Also by attacking proxy server, for example, an attacker can redirect his malicious traffic for own benefit [33, 34]. With DoS, an attacker attempts to 'flood' a network to prevent legitimate network traffic, an attacker attempts to disturb connections among selected computer systems by preventing network service, an attackers attempts to prevent a particular web server in providing service, or an attacker attempts to disrupt service to a specific web service client. In addition to make a web service unavailable, the DoS attacks can also target on tangible system resources which include computational resources (such as bandwidth, disk space, and processor time), configuration information (such as routing information, web service port configuration), and state information (such as session data).

3.8 Threats due to Broken Access Control

These threats are severe and they result in vulnerabilities that occur when restrictions on what authenticated users are authorized to do are not properly enforced. An authenticated, but unauthorized user of the same organization can become an attacker who can exploit broken access control to access unauthorized data and functions. Through Access Control, the web server grants access to specified web methods to the web service clients. Access control governs what an 'authorized' user is allowed to do. A web service provider's access control model is closely tied up into the content and web methods that the web service provides. In addition, the web service clients may fall into a number of groups or roles with different abilities or privileges. Practically, the authentication must precede authorization. Authentication verifies the identity of a user, in our case it is a web service client. Authorization is then used to determine what that user is allowed to access. To ensure proper access control, a web application must ensure both that it has proper authorization checks, and that it is using reliable and secure authentication that can distinguish privileged users from others.SS

3.9 Threats on HTTP Header Manipulation

HTTP header consists of control information that is passed between the client and server during web service call/return, where the web service client sends 'request' header and web server sends 'response header' in the format of SOAP standard. An attacker can write his/her own program to manipulate these headers, for example by handling requests which results the target service is attacked. The not validated malicious data may be inserted by attackers into a HTTP response header in order to attack the web service clients. Header manipulation can also result in cache-poisoning, cross-site scripting, cookie manipulation, page hijacking and open redirect. Normally Header Manipulation vulnerabilities occur when the data which enters to the web service server through HTTP request header from an untrusted source and the data which will be sent to web service client in an HTTP response header without being validated.

3.10 Threats on Bypassing Firewalls

By knowing the service port of web service, the attacker may try to bypass the security firewalls. Since Web Services often are implemented using port 80, most firewalls will happily pass the data through without any inspection of the traffic being made. Poorly implemented services can then be exploited to compromise other systems behind the firewall [36]. The most important factor in ensuring a firewall provides maximum protection is to ensure it is configured appropriately. A firewall is dumb entity in the sense that one must configure what does it to allow through/block. A poorly configured firewall will leave gaping holes in the attack surface. Depending on the feature-set of the firewall, it will only restrict access in certain ways. Although some penetration techniques might try to exploit vulnerabilities in the firewall's software, the majority of techniques are focused on exploiting poorly configured firewalls.

III. NEED OF SECURITY FOR SOA

In order to enable service-based cross-organizational collaboration, the security of the participating systems, exchanged messages, and used communication channels has to be ensured. Achieving and guaranteeing basic IT security goals such as confidentiality, authentication, authorization, non-repudiation, integrity, and availability is an absolute must in this context and still an active and core topic both in research and industry. Although security introduces additional costs and has an impact on the Quality of Service, unsecured business transactions are not an option in most scenarios [7].

The success or failure of any non-trivial system depends heavily on its overall representation commonly known as Software Architecture [37]. Software architecture of a system has shown to be very important in the realization of system wide qualities such as security, performance, etc. [38]. Security, as a factor, should be applied at the architectural level. It is true that Web services provide significant new benefits for SOA based applications, but they also expose significant new security risks. There are huge number of WS security standards and processes. At present, there is still a lack of a comprehensive approach which offers a methodical development in the construction of secure Web Service based SOA [39]. It is also true that there is no standard and formalized information security framework that has been adopted for service-oriented architectures.

One of the major design issues of SOA is meeting its security requirements, since it affects interaction of services and applications in SOA environment [40]. Security must be unified with the software engineering process [41] and security engineering [42] should be implemented for SOA applications at design phase itself, instead of applying as an ad-hoc requirement. Successfully implemented SOA security has to be well-defined, well-planned, and well-implemented [43].

Transport layer security mechanisms such as TLS or SSL are commonly used to provide data integrity and confidentiality between two communicating entities. The problem arises when the concept of intermediary nodes are introduced. The point-to-point principle implies that a message has to be decrypted at the intermediary node and then re-encrypted before it is forwarded to the final destination. This model cannot guarantee that the intermediary service does not manipulate the data either intentionally or unintentionally, thus end-to-end security is not ensured [44].

The applicability of the security protocols, such as WS-Security, WS-Trust, WS-SecureConversation, WS-Federation, WS-Authorization, and WS-SecurityPolicy, is limited as they only protect SOA communication between two trusted parties with an established security association. The pervasiveness of Web services and SOAP API that can be invoked especially by anonymous consumers introduces security vulnerabilities are not addressed by the existing standards [8]. In the same way, digital signatures alone do not provide message authentication as the attackers can record and resend signed message. Unfortunately, XML Encryption and XML Signature do not solve all web services security problems, mainly due to the data are available in plain text format. Bhavani Thuraisingham [45] states in her book while talking about Secure Services that the Web services and Service Oriented Architectures are at the heart of the next-generation web. However, one of the major problems in Service Oriented Architectures is ensuring a secure infrastructure environment [46] as service communications are made by machine to machine. With the introduction of Web 2.0, Web 3.0 and second generation internet based services, the traditional security approaches such as Secure Socket Layer / Transport Layer Service (SSL/TLS), and Virtual Private Network (VPN) become obsolete. Hence the organization should extend SOA security framework to adopt with these new generation technologies [47, 48].

IV. CONCLUSION

Security is one of the major concerns of SOA-based implementations, especially when it spans outside the boundaries of enterprise. The basic SOA framework does not possess any security and the available implementation of SOA security depends on the respective proprietor of the framework/implementation. Several standards for SOA security have been developed that are for a specific platform, specific technology, and specific implementation constraints. Many SOA vendors researched and supported security to SOA in their framework level and the security implementation has dependency with the core system of their products. In recent times, many research works had done for SOA security. Researchers have also proposed various frameworks and models which try a lot, but cannot achieve any landmark in the construction of a comprehensive framework, as they have own pros and cons.

In this paper, the authors critically analyzed the severe SOA security threats that are possible on SOAP based web services, and they explained the security essentials for SOA with knowledge sharing of their real-time work experience and literature study. Despite the existing security standards, some very skilled attacks are still discovered in SOAP Web Services especially after the introduction of Web 2.0, Web 3.0 and new generations of Web Services. It is not possible to provide total or absolute security. However, it is possible to identify the severity of SOA threats and the attackers' move on those threats in advance for each individual SOA applications as analyzed in this paper, and this gives an opportunity for security experts to design and

implement security for those SOA applications. When SOA security standards are properly leveraged, there is a potentiality to create entirely new and robust service-oriented security architectures, and that is the future work of the authors.

REFERENCES

[1] Srirama, S. N., Jarke, M. and Prinz, W., "Security Analysis of Mobile Web Service Provisioning," International Journal of Internet Technology and Secured Transactions, 2007

[2] Controneo, D., Graziano, A., and Russo, S., "Security Requirements in Service Oriented Architectures for Ubiquitous Computing," Proceedings of the 2nd Workshop on Middleware for Pervasive and Ad-hoc Computing, ACM, 2004

[3] Noor A. Altaani, Ameera S. Jaradat, "Security Analysis and Testing in Service Oriented Architecture," International Journal of Scientific & Engineering Research, Volume 3, Issue 2, 2012

[4] Deven Shah and Dhiren Patel, "Architecture Framework Proposal for Dynamic and Ubiquitous Security in Global SOA," International Journal of Computer Science and Applications, Vol. 6, No. 1, pp. 40-52, 2009

[5] Dirk Krafzig, Karl Banke, Dirk Slama, "Enterprise SOA Service Oriented Architecture Best Practices," Pearson Education, Inc, USA, 2005

[6] Johnneth Fonseca, Zair Abdelouahab, Denivaldo Lopes and Sofiane Labidi, "A Security Framework for SOA Applications in Mobile Environment," International Journal of Network Security & Its Applications (IJNSA), Vol.1, No.3, pp. 90-107, 2009

[7] Andr´e Miede, Nedislav Nedyalkov, Dieter Schuller, Nicolas Repp, and Ralf Steinmetz, "Cross-organizational Security – The Service-oriented Difference," International Conference on Service Oriented Computing, Springer (ISBN: 978-3-642-16131-5), pp. 72-81, 2010

[8] Navya Sidharth and Jigang Liu, "IAPF: A Framework for Enhancing Web Services Security," 31st Annual International Computer Software and Applications Conference (COMPSAC 2007), 2007

[9] Hassan Reza, and Washington Helps, "Toward Security Analysis of Service Oriented Software Architecture," Proceedings of the 2011 International Conference on Software Engineering Research and Practice, Vol. II, 2011

[10] Oldooz Karimi, "Security Model For Service-Oriented Architecture," Advanced Computing: An International Journal (ACIJ), Vol.2, No.4, pp. 48-58, 2011

[11] Alaeddin Kalantari, Anahita Esmaeili, and Suhaimi Ibrahim, "A Service Oriented Security Reference Architecture," International Journal of Advanced Computer Science and Information Technology (IJACSIT), Vol. 1, No.1, pp. 25-31, 2012

[12] http://www.w3.org/TR/wsdl

[13] http://www.w3.org/TR/soap/

[14] Jacqui Chetty and Marijke Coetzee, "Towards An Information Security Framework For Service-oriented Architecture," Information Security Conference, South Africa, IEEE ISBN: 978-1-4244-5494-5, 2010

[15] Vorobiev, A. and Han, J., "Security Attack Ontology for Web Services," Proceedings of the 2nd International Conference on Semantics, Knowledge and Grid (SKG'06), Guilin, China, 2006

[16] M. B. Juric, A. Sasa, B. Brumen, and I. Rozman, "WSDL and UDDI extensions for version support in web services," Elsevier at The Journal of Systems and Software, vol. 82, pp.1326–1343, 2009

[17] Esmiralda Moradian and Anne Hakansson, "Possible attacks on XML Web Services," IJCSNS International Journal of Computer Science and Network Security, Vol. 6, pp. 154-170, 2006

[18] Hatem Hamad, Motaz Saad, and Ramzi Abed, "Performance Evaluation of RESTful Web Services for Mobile Devices," International Arab Journal of e-Technology, Vol. 1, No. 3, pp. 72-78, 2010

[19] Amrit Tiwana, "Web Security," Digital Press, USA, 1999 (ISBN: 9781555582104)

[20] MSDN Library, "Chapter 2: Threats and Countermeasures for Web Services", Patterns & Practices, Microsoft (Referred on Nov 2011)

[21] Matthew Tanase , "IP Spoofing: An Introduction," White-paper, SecurityFocus, 2003

[22] Garfinkel, S. and Spafford, G., "Web Security, Privacy & Commerce," 2nd Edition, O'Reilly Media Inc., 2002

[23] McClure, S. and Shah, S., "Web Hacking: Attacks and Defense," Pearson Education Inc., 2002.

[24] Lindstrom, P., "Attacking and Defending Web Services", White-paper, Spire Security, 2004

[25] Fitzgerald, J. Dennis, A. "Business Data Communications and Networking," John Wiley and Sons, 2002

[26] Shema, M. "HackNotes: Web Security Portable Reference," McGraw Hill Professional, 2003

[27] Fengyu Zhao, Xin Peng, and Wenyun Zhao, "Multi-Tier Security Feature Modeling for Service-Oriented Application Integration," 8th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2009), Shanghai, China, 2009

[28] WG Halfond, Jeremy Viegas, and Alessandro Orso, "A Classification of SQL-injection Attacks and Countermeasures," Proceedings of the IEEE International Symposium on Secure Software Engineering, Arlington, VA, USA, 2006

[29] Nuno Antunes, Nuno Laranjeiro, Marco Vieira, and Henrique Madeira, "Effective Detection of SQL/XPath Injection Vulnerabilities in Web Services," IEEE International Conference on Services Computing (SCC '09), 2009

[30] Philipp Vogt, Florian Nentwich, Nenad Jovanovic, Engin Kirda, Christopher Kruegel, and Giovanni Vigna, "Cross-site scripting prevention with dynamic data tainting and static analysis," Citeseer, 2007

[31] Danish Jamil and Hassan Zaki, "Security Implication of SOAP and Web-Service Interface to the Cloud Computing System," International Journal of Engineering Science and Technology (IJEST), ISSN : 0975-5462 Vol. 3 No. 4, 2011

[32] Stuart McClure, Joel Scambray, and George Kurtz, "Hacking Exposed: Network Security Secrets & Solutions," 7th Edition, McGraw Hill Professional, 2012

[33] Candolin, C. and Kiviharju, M., "A roadmap towards content based information security," The 6th European Conference on Information Warfare and Security, Shrivenham, UK, 2007

[34] Torry Harris Business Solutions Inc., White-paper, "Migration and Security in SOA", University of Leeds, 2009

[35] Eric Pulier and Hugh Taylor, "Understanding Enterprise SOA," Manning Publication, 2006

[36] Kalantari, A., Khezrian, M., Esmaeili, A. and Taherdoost, H., "Enabling Security Requirements for enterprise Service Oriented Architecture", International Journal of Recent Trends in Engineering and Technology, pp. 75-81, 2011

[37] Taylor, Richard N., Nenad Medvidovic, and Eric M. Dashofy, "Software Architecture: Foundations, Theory, and Practice," Wiley Publishing, 2009

[38] Reza, Hassan, and Emanuel Grant, "Quality-oriented Software Architecture," Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC), Vol. 1, IEEE, 2005

[39] Nafise Fareghzadeh, "Web Service Security Method to SOA Development," World Academy of Science, Engineering and Technology, 2009

[40] Jeremy Epstein, Scott Matsumoto, Gray McGraw, "Software Security and SOA: Danger", IEEE Security & Privacy, Vol. 4, Issue 1, 2006, pp. 80-83
[41] Yuichi Nakamura et al., "Model-driven Security based on Web Services Security Architecture," IEEE International Conference on Services Computing, Vol. 1, 2005
[42] Premkumar T. Devanbu and Stuart Stubblebine, "Software Engineering for Security: A Roadmap," Proceedings of the Conference on the Future of Software Engineering, ACM, 2000
[43] Tipnis, A., and Lomelli, I., "Security: A Major Imperative for a Service-Oriented Architecture – HP SOA Security Model and Security Assessment", HP Viewpoint Paper, 2009
[44] Jostein Jensen and Asmund Ahlmann Nyre, "SOA Security – An Experience Report," Proceedings of the Norwegian Information Security Conference (NISK), Trondheim, Norway, 2009, pp. 185-196
[45] Bhavani Thuraisingham, "Secure Semantic Service Oriented Systems," Auerbach Publications (Taylor & Francis Group), USA, ISBN: 978-1-4200-7331-7
[46] Anu Soosan Baby, Deepu Raveendran, and Aswathy Josephine Joe, "A Study on Secure and Efficient Access Control Framework for SOA," International Journal of Computer Science and Telecommunications, Vol. 3, Issue 6, pp.71-76, 2012
[47] Yamany, H.F., and Capretz, L.F., "Use of Data Mining to Enhance Security for SOA," Proceedings of the 3rd International Conference on Convergence and Hybrid Information Technology (ICCIT), IEEE, Vol. 1, 2008
[48] HP, "Securing web 2.0: Are your Web Applications Vulnerable?" White-paper, Hewlett-Packard Development Company, L.P, 2007