Jurnal Teknologi

Full Paper

CONSTRUCTION OF CUSTOMIZABLE SOA SECURITY FRAMEWORK USING ARTIFICIAL NEURAL NETWORKS

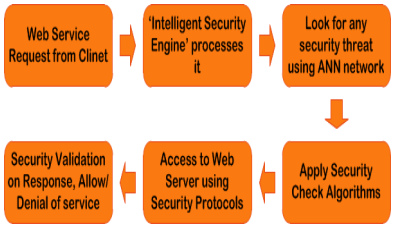
Mohamed Ibrahim B[a*], Mohd Fadzil Hassan[b]

[a]Research Scholar & Software Solution Architect, Malaysia
[b]Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Malaysia

Article history
Received
26 November 2015
Received in revised form
14 January 2016
Accepted
10 October 2016

*Corresponding author
bmdibrahim@gmail.com

Graphical abstract



Abstract

The Web Services technology for the implementation of Service Oriented Architecture (SOA) is the preferred choice in the current era of Enterprise Application Integration (EAI). As Web Services architecture is dynamic and loosely coupled, security aspects must be considered thoroughly at the time of designing. It is prone for attacks as it uses XML format for data exchange, which is a plain text. A novel security component named "Intelligent Security Engine (ISE)" is introduced into the proposed framework which incorporates Artificial Neural Networks (ANN) Learning Techniques for supervised knowledge acquisition on security threats of SOA. Thus, the proposed security framework is capable in the identification of future security vulnerabilities of SOA and can work effectively even for in-secured cross organizational EAI environment.

Keywords: SOA, web services, security, neural nets, machine learning, SOAP, WSDL

1.0 INTRODUCTION

The Loosely Coupled Architecture in Software development simplifies Enterprise Application Integration (EAI), which is an architectural style of integrating a set of interrelated computer applications that are running for an enterprise. The Loose coupling refers to application modules that are independent of each other; in such a way that a module can be changed without affecting the operations of other modules. The Service Oriented Architecture (SOA) is dynamic that is applied for development of loosely coupled distributed applications in the current era. Though early SOA was achieved using a number of architectural technologies such as RMI (Remote Method Invocation), ORB (Object Request Broker), and DCOM (Distributed Component Object Model), none of them provides a comprehensive architecture that works independent of programming languages, operating systems, and machine architectures. In this stage, Web Services is emerged as a dominant paradigm that eliminates to the dependency issues using the available Internet protocols such as HTTP and thus supports in the development of distributed systems over heterogeneous platforms.

Generally the services represent business functionalities at high level, which will not tie up to the context of other services. They are self-contained unit of code that runs on multiple servers [1]. As the data are stored in multiple servers and transferred over the Internet, the security aspects should be thoroughly considered from the design stage of web service application development [2].

The remaining sections of this paper are organized as, section 2 analyses the need of security for SOA, section 3 reviews the available security solutions and standards in the view of literature, the proposed security framework is briefed in section 4, and the section 5 concludes the paper.

1.1 Need Of Security For SOA: Background And Analysis

The SOA is a model of Software components, thus it exposes application resources in the form of services that can be accessed over networks. There are three components that a SOA consists of: (i) Service Provider, (ii) Service Registry, and (iii) Service Requester as shown in Figure 1. The service provider publishes the service description (Publish) into the Service Registry. In the discovery (Find) process, the service requester retrieves a service description directly or queries the service registry for the type of service required. The service requester initiates an interaction with the service (Bind) by using the binding details specified in the service description [3]. Thus the basic SOA architecture performs three different operations namely, Publish, Find and Bind.

The description of the web services such as web method names, parameters and their data types are described using Web Services Description Language (WSDL) standard by service provider and published to service registry. The service registry acts as a centralized repository for maintaining service information in terms of WSDL files, and provides directory services for web services. The service registry should adhere to Universal Description, Discovery, and Integration (UDDI) standard. The eXtensible Markup Language (XML) standard is used as data exchange format and Simple Object Access Protocol (SOAP) standard is followed up as communication protocol between service requester and service provider.

At the time of writing this paper, the latest version of WSDL is 2.0 which is a W3C recommendation. The SOAP 1.2 is the latest version which is also a W3C recommendation and it operates on transport protocols such as HTTP, TCP and UDP. The Organization for the Advancement of Structured Information Standards (OASIS) supports the format of WSDL and it uses W3C and Internet Engineering Task Force (IETF) Internet standards such as XML, HTTP, and DNS protocols.

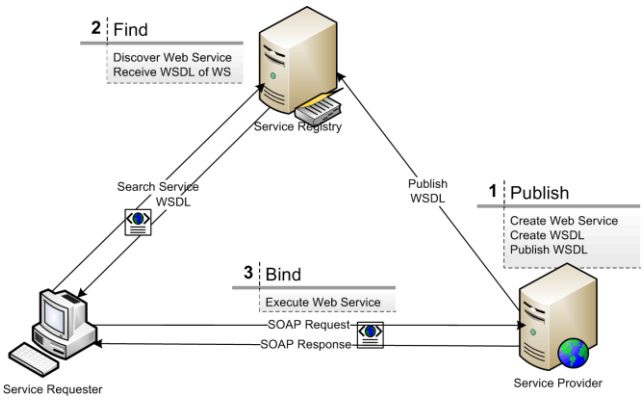


Figure 1 Service oriented architecture to form EAI

Though the WSDL and SOAP contents are represented in XML format which provides many advantages but unfortunately it also causes many security issues in SOA. The architecture of Web Service adds new threats to the service applications and they turn into challenges when considered for applying security solutions. In SOA security terms, the asset is an entity in SOA applications which is worth for protection such as sensitive data and critical operations. The threat is any possible event that could harm the assets. Vulnerability is the weakness of the SOA applications which may result in threat and an action that exploits vulnerability is called attack. Thus the threats are the possible ways that the SOA system can be attacked. SOA threats generally falls under four categories: (i) Disclosure – unauthorized access to the data, (ii) Deception – making false data believed to be genuine, (iii) Disruption –making denial of service of an asset, and (iv) Usurpation –an unauthorized entity that gains control of the asset [4].

The latest generation of web service and web based technologies such as Web 2.0 and Web 3.0 introduce additional severity to the SOA threats where the traditional security such as Secure Socket Layer / Transport Layer Security (SSL/TLS), and Virtual Private Network (VPN) become obsolete. So that it became mandatory for the organizations to adopt for a comprehensive security solutions that caters for new technologies [5] and security in SOA context is an active and core topic in research and industry. Unsecure SOA applications cannot become the preferred choice, even though applying security requires extra cost [6]. Hence, security became the core concern of SOA based enterprise computing, especially in inter-organizational environment [7].

In practice, the security aspects for SOA applications are not defined at the time of design, left to the developers and incorporated in ad-hoc manner. For the application developers, it is difficult for them to understand the security requirements of SOA and the way to implement them as they are not security experts. Furthermore security is a cross-domain concern that all of the security requirements cannot be defined and available to the downstream phases [8]. Security as a factor should be applied at the application architecture level. The security experts should look for architectural security solution that should be well-planned, well-defined and well-implemented [9].

Thus, the basic SOA architecture does not have sufficient security solution within it and the available security solutions for SOA are tightly coupled to the concerned proprietors. Also there are several security standards and specifications are developed, but they have own pros and cons as they depend on specific computing environment such as machine architectures, operating systems and implementation technologies. In recent times, many research works had been conducted for SOA security and many researchers have also proposed various frameworks and models, but cannot achieve any landmark in the construction of a comprehensive SOA security framework, as they have own pros and cons.

2.0 METHODOLOGY

The following sub-sections review the SOA architectural requirements, available security standards and related research works in the view of literature.

2.1 Functional and Non-Functional Requirements

As a matter of fact, SOA brings several additional security issues. In order to overcome, the various functional and non-functional security requirements should be considered properly before constructing SOA security solution.

The possible and severe attacks on SOA SOAP Web Services are listed in Table 1. This is not a complete list as the number of vulnerabilities is increasing over the years. According to NVD (National Vulnerability Database, US) survey [10], an average of 19 operating systems (OS) vulnerabilities per day were reported in 2014.

The major security requirements for SOA computing, particularly on SOAP based web services are briefed in Table 2. These of the functional and non-functional requirements that any of the SOA based implementations should support to the Information Technology industrial standard in term of web oriented and service oriented security.

The authentication and authorization are the mandatory security requirements for SOA security, where the others are additive to SOA based on the organizational security policies.

Table 1 Sample attacks on SOA SOAP web services

Category	Group	Attacks
WSDL	WSDL Attacks	WSDL Scanning, XML Wrapping, Parameter Tampering
SOAP	Identity Attacks	Dictionary Attacks, Brute Force Attacks, IP Spoofing, Message Eavesdropping, Data Tampering
	Session Attacks	Reply Attacks, Man-in-the-middle Attacks, Session Management Attacks
	Parsing Attacks	Recursive Payloads, Oversized Payloads, Schema Poisoning
	Code Attacks	SQL Injection, XPath Injection, Cross-site Scripting, Attacks on improper code
	Network Attacks	Denial of Service (DoS) Attacks, UDDI Attacks, Malicious Attachments

Table 2 Major security requirements of SOA

Security Requirements	Purpose
Authentication	It deals with verification of identity. In SOA, this is to find out who is calling which service.
Authorization	It deals with the allowed actions of an identity. In SOA, this is to check whether the client is allowed to consume the service or not.
Access Control	While the services are consumed by the clients, the SOA application should do access control by enforcing certain constraints.
Federation	Federation is used when the service provider has to establish a trust between its security domain and an external domain; it is an extension of authentication to external systems.
Service Usage	The service provider can limit the clients on the actions of service.
Confidentiality	Only authorized clients can access the services and contents. The goal of network security is to encrypt data packets transferred between service requester and service provider.
Integrity	Integrity confirms that the data should not be altered during transmission between service requester and service provider by an unauthorized person.
Availability	It ensures the service availability according to the Service Level Agreement (SLA), as it is possible to make a service inoperable by flooding attacks such as Denial of Service (DoS) and message replay attacks. The availability is achieved in network layer.
Privacy	It deals with the protection of private data and what can be disclosed in what circumstances.
Non-repudiation	It is the understanding and confirmation that the sender and the receiver cannot deny what they performed.
Accounting	Accounting is to keep track of service consumption.

2.2 SOA Security Standards for Web Services

The W3C and OASIS consortiums have standardized several specifications across the years for web services security. Recently a set of security specifications for SOA are emerging in both open-sourced and product based. Some well-known security specifications that may be used as part of securing web services are listed in Table 3. These are the selected security specifications that are widely practiced in the SOA security world, even though there are other security specifications exist in a good number.

Table 3 Major SOA security specifications

WS-Security It specifies an abstract security model for web services using security tokens with digital signatures to protect and authenticate SOAP messages.
WS-Trust It provides the extensions to WS-Security, specifically dealing with issuing, renewing, and validating of security tokens in the process of secure message exchange.
SAML The Security Assertion Markup Language (SAML) provides an XML-based specification for the exchange of authentication, entitlement, and attributes information.
WS-Federation It defines how the trust relationships are managed and brokered in a heterogeneous federated environment.
WS-Secure Conversation It defines an extension to the WS-Security standard, and provides a framework for establishing and sharing security contexts and session key derivation.
WS-Security Policy WS-Policy is framework used for expressing security constraints as policies using policy assertions. The WS-Security Policy is an addendum to WS-Security that specifies the policy assertions for WS-Policy which applies to WS-Security.
WS-Provisioning It facilitates interoperability between provisioning systems that supports provisioning vendors to provide provisioning facilities in a consistent manner.
SPML The Service Provisioning Markup Language (SPML) provides support for managing service provisioning data such as identity information and system resources at the organizational level.
XML-Encryption It is used for encryption of data defined in XML format.
XML-Signature It specifies XML digital signature processing rules and syntax.
XACML The Extensible Access Control Markup Language (XACML) standard is used to evaluate authorization requests according to the defined rules in organizational security policies.
XrML The Extensible Rights Markup Language (XrML) is an XML based standard that describes rights and conditions together with message integrity and authentication information.
XKMS The XML Key Management System (XKMS) is a framework that enables secure inter-application communication using Public Key Infrastructure (PKI).
SSL The Secure Sockets Layer (SSL) is a protocol that is used to transmit private data securely over Internet.

2.3 Related Works in the View of Literature

In the literature of SOA Security on SOAP based Web Services, many researchers had done potential researches and proposed a number of approaches in the form of various security models, frameworks, and architectures in addition to the industry standards. Still there is no comprehensive security solution available for SOA. It is also true that there is no standard and formalized information security framework that has been adopted for service-oriented architectures.

Few researchers which include [11-14], introduced and proposed security add-on solutions to the existing security standards, which are also promising remedies to the security threats, however they are totally inadequate to the security needs of SOA in industrial perspective. From the reviewed literatures, the authors found the following eight research works which are more relevant to the proposed framework.

a) Shah et al. [7] proposed the security architecture for global SOA that they use message interceptor design pattern to protect message transfer on SOA.

b) Deepti Sisodia et al. [15] evaluated the Public Key Infrastructure (PKI) on enforcing security for SOA and proposed an inbuilt security module named "SecSOA" which is based on PKI.

c) Shahgholi et al. [16] proposed a new framework which aims to protect Web Services against WSDL attacks.

d) Wei She et al. [17] proposed an enhanced security model to facilitate the control of information flow through service chains i.e., composed services, for example service-1 calls service-2, which in turn calls service-3.

e) Tao Xu et al. [13] proposed a security processing model named SIMSA (Security Interactive Model based on SOAP and Authentication) based on SOAP and authentication in order to solve the security issues of heterogeneous platforms.

f) Kou Hongzhao [18] introduces a concept, Security Token Service, into the WS-Security and presented Security Token Service based security architecture, named STS-WS.

g) Kharat et al. [19] introduces Single Sign On (SSO) certificate based authentication for Web services security, which provides a unified authentication mechanism for accessing web services where the user did login one time, then he/she will get accesses to all the web service providers in the enterprise.

h) Navya Sidharth et al. [20] proposed a new framework named "Integrated Application and

Protocol-based Framework (IAPF)" that mainly prevents Denial-of-Service (DoS) attacks.

In reviewing literature, we can see a huge interest is shown in the development of solutions for SOAP message level attacks; however, there is not much on WSDL attacks. In fact, WSDL attacks are severe in nature, which can even halt the entire web services.

2.4 Proposed SOA Security Framework

In the proposed SOA security framework for SOAP web services, a new pluggable security component named "Intelligent Security Engine (ISE)" is introduced at the Service Provider end in the standard SOA architecture. The component "ISE" adheres to the organizational standards for security. The proposed security framework is tested for the following two categories of security threats in SOAP based web services environment.

- WSDL Attacks
  (Service Provider → Service Registry ←→ Service Requester)
- SOAP Message Level Attacks
  (Service Requester ←→ Service Provider)

Each WSDL 2.0 document may contain up to seven segments: description, types, interface, binding, service, documentation and import. Out of which, the mandatory segments interface and types are describing the offered service names (web methods) and the data types of their input/output parameters respectively. These critical sections of the WSDL document will be encrypted using Symmetric key and published in to Service Registry. The hashing technique is applied to derive digest at service provider and service requester side to ensure message integrity, authentication and non- repudiation where the WSDL content should not altered during its transmission over intermediate nodes. The Public Key Infrastructure (PKI) and XML Key Management Specification (XKMS) standards are applied in transferring the symmetric key from service provider to service requester in order to decrypt the encrypted content of received WSDL. The Digital Signatures with Certificate Authority (CA) are applied in service provider and service requester end to ensure the validity of each other in obtaining WSDL and keys. The algorithm for preventing WSDL attacks is given in Figures 2 and 3 that should be applied respectively at web service server and client side.

```
Algorithm: WSDL_Attack_Prevention_Server_Side
Input: WSDL file
Variable: wsdl: WSDL file;
          PriKSP: Private Key of Service Provider;
          DSignSP: Digital Signature of Service Provider;
          Digest: Hash of WSDL content;
          SymK: Symmetric Key;
          encWSDL: Encrypted WSDL file
Output: Encrypted WSDL
Begin
  Step I:   /* Service Provider sends WSDL to ISE */
            wsdl = WSDL(SP)
  Step II: /* Generates values for Keys and Hash */
            Call XKMS, KeyGen, Hash modules
            Digest = Find digest of WSDL for message integrity
            PriKSP = Private key of generated asymmetric key
                     pairs [SP]
            DSignSP = Derive Digital Signature of SP
            Register with Certificate Authority
  Step III: /* Encrypt WSDL */
            encWSDL = [wsdl+ [Digest]PriKSP + DSignSP]SymK
                      + Custom_Security_Elements
  Step IV: /* Publish encWSDL */
            Call UDDI Directory Service API
            Service Registry ← encWSDL
End
```

Figure 2 Algorithm for preventing WSDL attacks (server side)

```
Algorithm: WSDL_Attack_Prevention_Client_Side
Input: Encrypted WSDL file
Variable: wsdl: WSDL file; digest: Hash; SymK: Symmetric key;
          encWSDL: Encrypted WSDL file
Output: WSDL
Begin
  Step I:   /* Service Requester gets WSDL from Service
               Requester */
            Call UDDI Directory Service API
            Service Requester ← ServiceRegistry[WSDL]
  Step II:  /* Check whether received WSDL is encrypted */
            If Custom_Security_Elements found  Then
              encWSDL = Received WSDL
            Else
              Return "WSDL is not encrypted"
              Go to Step V
            End If
  Step III: /* Get Symmetric Key from SP using PKI */
            Provider: Encrypt Symmetric Key using
                      Public Key of Requester
            Provider: Transfer encrypted Symmetric Key to SR
            Requester: Decrypt it using Private Key of SR
            SymK = Call Transfer Key using PKI module
  Step IV: /* Decrypt the received WSDL content */
            wsdl = [encWSDL]SymK
  Step V: /* Bind Web Service Call */
           Start invoking web service methods
End
```

Figure 3 Algorithm for preventing WSDL attacks (client side)

The proposed security framework uses Artificial Neural Networks (ANN) for knowledge acquisition process on security threats in an organizational perspective. The supervised learning on ANN is applied so that the ISE can detect any similar kind of security threats automatically in future.

3.0 RESULTS AND DISCUSSION

In the computational problem solving technique, the set of instructions of the defined algorithms is executed as the problem space along with expected result is clearly known. Neural networks works on a different approach in problem solving; they process the data and produce the result in the similar way that the neurons in biological brain do. The basic structure of a biological neuron is shown in Figure 4. Each neuron has

a cell body named as "nucleus", a branching input structure called as "dendrites" and a branching output structure termed as "axon". One neuron's output (axon) will be connected to input (dendrite) of another neuron via synapse; it depends on how strong synapses are for the concerned neurons. The nucleus of a neuron will emit electro chemical signal as output only if its input signals are strong enough which surpass a certain threshold in a short period of time [21].

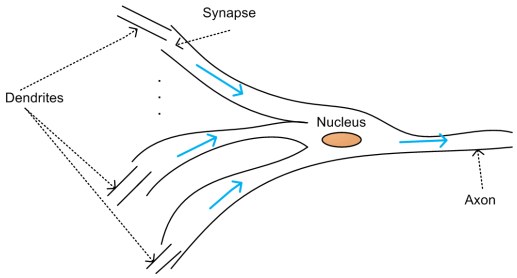


Figure 4 Structure of biological neuron

The artificial neuron basically consists inputs (x_i), which will be multiplied by weights (w_i) i.e. the strength of the respective signals (synapse), and then computed by a mathematical function (sigmoid) which determines the activation (Y = 1: activated; or 0: deactivated) of the neuron with the comparison of defined threshold value. We can try to obtain the desired output by adjusting the weights to a neuron. This process of adjusting weights is called ANN Learning or Training the ANN. The Figure 5 sketches the basic structure of an artificial neuron. The number of neurons should be defined for an ANN to work in parallel varies based on the nature of concerned problem. The ISE can be placed at the web server, proxy server or even on Enterprise Service Bus (ESB) in order to support security for all the SOA based services. Based on the organizational security policies and constraints, the ANN learning can be optimized. The ISE can also be configured to skip security verification for defined set of SOA threats permanently or for a period of time.

The ANN can be compared as just an electronic networks of artificial neurons based on the structure of the biological neurons. The ANN processes one record at a point of time, and it learns through the comparison of the learn classification of the record with the actual classification of the record. The error difference of the first record's initial classification will be feed back into the ANN which is used to modify the network behavior for the further iterations. The neurons in ANN are organized into three layers, (i) Input Layer, (ii) Hidden Layer, and (iii) Output Layer. The Input Layer does not contain all the neurons in the network, but contains the record's values which will be feed to next layer as input. Normally only one input and output layer exist in a neural nets, but there will be multiple hidden layers present based on the scope of the defined problem that should be solved through ANN.

The proposed security section is tested with the WSDL parameter scanning as input layer and the scanning prevention as the output layer, where the applied hidden layer improves the security protection [22, 23].

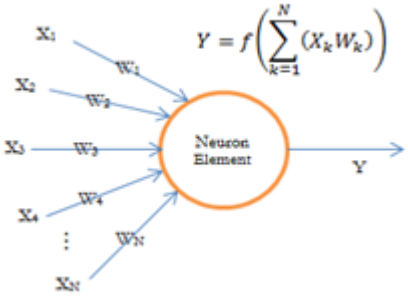


Figure 5 Structure of artificial neuron

The cross-cutting view of ISE is as shown in Figure 6, which uses rule based engine, organizational security policies, batch job –mass operations, ANN learning and decision engine for the process of identifying security attacks and preventing them. When a SOAP request is sent from service requester to service provider, it will be interpreted by ISE for intrusion detection. If the ISE encountered any security issues on the SOAP request (such as oversized message, deeply nested message, message replay, and request from fraud requester), then it will be blocked by ISE and then the configured neural networks will update the knowledge database. The same kind of security check is performed on SOAP response too. Thus the maintained knowledge database on security threats is the key resource of the proposed framework to prevent SOAP and WSDL attacks. For Proof-of-Concept (POC), the proposed framework is implemented using Java & related technologies and tested in a large-scale web service environment with real time logistic data.

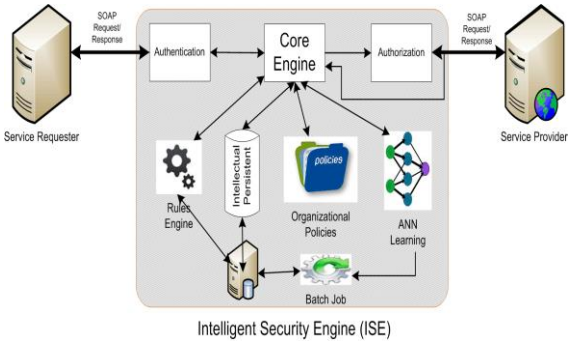


Figure 6 Cross-cutting view of ISE

4.0 CONCLUSION

Service Oriented Architecture is the modern trend in programming where the web services are choice of organizations to implement EAI (Enterprise Application

Integration) which provides platform and language independent way of developing distributed applications. However, it brings additional security threats as it uses simple text format for information exchange. There should be a comprehensive security mechanism implemented in organizations that uses SOA for EAI, especially in organizations where their data are very sensitive. This paper analyzed the architecture of SOA and its security needs especially for SOAP based web services, it also reviewed the available security solutions as in the form of vendor products and research works in the view of literature. Even though there are a good number of security solutions available for SOA in terms of standards/specifications/research proposals and products of vendors, there is no comprehensive security solution derived yet for SOA especially which works for in-secured environment. This paper also describes the gaps between the existing solutions and the need of SOA security with relevant works. An intelligent component named "Intelligent Security Engine (ISE)" is adopted at the web service provider side of SOA, aimed to prevent WSDL and SOAP message level attacks; and it uses Artificial Neural Networks (ANN) learning techniques for knowledge acquisition of security threats on SOA in an organizational perspective. For Proof-of-Concept (POC), the proposed security framework is developed using Java technologies and tested with real-time data on the SOAP Web Services in a large-scale logistic domain.

References

[1] Fonseca, J., Abdelouahab, Z., Lopes, D., & Labidi, S. 2009. A Security Framework for SOA Applications in Mobile Environment, *International Journal of Network Security & Its Applications (IJNSA)*. 1(3)

[2] Fernandes, D. A., Soares, L. F., Gomes, J. V., Freire, M. M., & Inácio, P. R. 2014. Security Issues in Cloud Environments: A Survey, *International Journal of Information Security*. 13

[3] Oldooz Karimi. 2011. Security Model For Service-Oriented Architecture, *Advanced Computing: An International Journal (ACIJ)*. 2(4).

[4] Moradian, E., Håkansson, A., & Andersson, J. O. 2012. Ontology Based Patterns for Software Security Engineering, *Advances in Knowledge-based Intelligent Information and Engineering Systems*, IOS Press,

[5] Baby, Anu Soosan, Deepu Raveendran, and Aswathy Josephine Joe. 2012. A Study on Secure and Efficient Access Control Framework for SOA, *International Journal of Computer Science and Telecommunications*. 3(6).

[6] Miede, A., Nedyalkov, N., Schuller, D., Repp, N., & Steinmetz, R. 2010. Cross-organizational Security –The Service Oriented Difference, *Service-Oriented Computing*, Springer

[7] Shah, D., Agarwal, M., Mehra, M., & Mangal, A., Global SOA: RSS-based Web Services Repository and Ranking, *IEEE Fifth International Conference on Internet and Web Applications and Services (ICIW)*, 2010

[8] Saleem, M., J. Jaafar, and M. Hassan. 2012. A Domain-Specific Language for Modeling Security Objectives in a Business Process Models of SOA Applications, *AISS*. 4(1)

[9] Kassou, M., & Kjiri, L. 2012 A Goal Question Metric Approach for Evaluating Security in a Service Oriented Architecture Context, *International Journal of Computer Science Issues*. 9(4): 1

[10] https://web.nvd.nist.gov

[11] Upendra Kumar & Rao. 2011. Designing Dependable Agile Layered Web Services Security Architecture Solutions, Indian *Journal of Computer Science and Engineering (IJCSE)*. 2(3)

[12] Arezoo Mirtalebi and Mohammad Reza Khayyambashi. 2012. A new Security Framework for Protecting WSDL File of Web Service, *International Journal of Computer Science and Network Security (IJCSNS)*. 12(9)

[13] Xu, Tao, and Chunxiao Yi. 2011. SOAP-Based Security Interaction of Web Service in Heterogeneous Platforms, *Journal of Information Security*.

[14] Pankaj Choudhary, Rajendra Aaseri, and Nirmal Roberts. 2013. HTTPI based Web Service Security over SOAP, *International Journal of Network Security & Its Applications (IJNSA)*. 5(3)

[15] Sisodia, Deepti, Lokesh Singh, and Sheetal Sisodia. 2012. Web Based Secure SOA, *International Journal of Computing Algorithm*. 1(2)

[16] Shahgholi, N., Mohsenzadeh, M., Seyyedi, M., & Qorani, S. H. 2011. A New SOA Security Framework Defending Web Services against WSDL Attacks, *Proceeding of IEEE 3rd International Conference on Social Computing.*

[17] She, Wei, I. Yen, and Bhavani Thuraisingham. 2008. Enhancing Security Modeling for Web Services using Delegation and Pass-on, *IEEE International Conference on Web Services (ICWS)*, China.

[18] Kou Hongzhao, 2010. A Study on the Security Mechanism for Web Services, *Proceedings of the World Congress on Engineering and Computer Science*, San Francisco, USA. I.

[19] Kharat, Prachi M., Prachi A. Deshpande, and Aaditya P. Bakshi. 2013. Single Sign On Certificate Based Authentication for WS-Security, *International Journal of Advanced Research in Computer Science*. 4(6)

[20] Navya Sidharth and Jigang Liu, IAPF: A Framework for Enhancing Web Services Security, *31st Annual International Conference on Computer Software and Applications*, 2010

[21] Sen, Anand Swarup, and Pritesh Jain, Technique of Intrusion Detection based on Neural Network –A Review, *IEEE Conference on IT in Business, Industry and Government (CSIBIG)*, 2014

[22] Grzonka D, Kołodziej J, Tao J, Khan SU. 2014. Artificial Neural Network Support to Monitoring of The Evolutionary Driven Security Aware Scheduling in Computational Distributed Environments, *Future Generation Computer Systems*. 51: 72-86.

[23] Rohani, M.F.A., Maarof, M.A., Selamat, A. and Kettani, H. 2007. Uncovering Anomaly Traffic Based on Loss of Self-Similarity Behavior Using Second Order Statistical Model, *International Journal Computer Science and Network Security*. 116-122