An Approach to Predict SOA Security Vulnerabilities using Feed Forward Artificial Neural Networks

B. Mohamed Ibrahim* & Dr. A.R. Mohamed Shanavas**

*Software Solution Architect & Research Scholar, Tiruchirappalli, Tamil Nadu, INDIA.
E-Mail: bmdibrahim{at}gmail{dot}com
**Associate Professor, Jamal Mohamed College (Autonomous), Tiruchirappalli, Tamil Nadu, INDIA.
E-Mail: vas0699{at}yahoo{dot}co{dot}in

***Abstract*—**The Service Oriented Architecture (SOA) makes interconnection of related applications in an organization easy and the web services provide platform and language independent implementation of SOA as they are using the basic Internet protocols for data transmission. As the data is transferred between servers and clients in a plain text format, it adds more security vulnerabilities to the existing security threats. It is increasing due to the latest web development such as Web 2.0 and its upgraded version Web 3.0; the traditional transport layer security solutions are totally absolute. Also even if specific solutions will be developed for the known security attacks, it is not possible to create a comprehensive one as the security risks are increasing and the problem space is wide. In this paper, we propose an approach to predict SOA security vulnerabilities in an organization using Artificial Neural Networks by applying customized learning techniques.

***Keywords*—**ANN; Machine Learning; SOAP; SOA Security; Web Services.

***Abbreviations*—**Artificial Neural Networks (ANN); Service Oriented Architecture (SOA); Simple Object Access Protocol (SOAP).

I. INTRODUCTION

A service is a Software component that is developed for doing specific functionality and it does not depend on the context or state of other services. The Service Oriented Architecture (SOA) provides platform and implementation independent way of interconnecting interrelated applications in an enterprise. This is possible because SOA uses common internet protocols such as HTTP. However, it increases security vulnerabilities as these protocols works on data transmission in simple text format [Richard Welke et al., 11].

At present web service is the common and preferred choice in the industry for SOA implementation even though SOA can be implemented in number of ways such as RMI and CORBA. A Web Service provides well-defined interfaces for distributed functionalities that are independent of the platforms and implementing programming languages [Mustafa Bozkurt et al., 2; Michael Rosen et al., 3]. In this way, it has emerged as a prevailing method for constructing distributed applications using web technologies. Security is one of the major concerns when developing mission critical business applications and this concern motivates Web Services Security specifications [Subashini & Kavitha, 4].

We can use the mathematical model to find a solution for a problem if a problem space is well defined. If the problem space is bigger and cannot be well defined, then we have to use heuristic techniques in order to resolve the problems. The security attacks are huge and they cannot be concretely defined when they will appear and error accordance space is huge. The machine learning techniques can be applied in finding solutions to these kinds of problems, such one approach is Artificial Neural Networks (ANN) [Shah & Bhushan H. Trivedi, 5].

A complex problem may be decomposed into smaller elements in order to find solution for each of such small elements. Then, the solutions will be gathered to produce as a solution for the complex problem. The networks are one of the approaches in achieving this kind of divide & conquer technique. There are several types of networks, however all the networks have the common characteristics: (i) Nodes - vertices and (ii) Edges -Links between nodes. Consider the nodes as computational units, which receive inputs and produces the output after the defined functionalities are processed. The connections between the nodes determine the information flows. The connections can be unidirectional or bidirectional depending on the constructed network architecture [Gholami et al., 6; Devikrishna & Ramakrishna, 7].

The defined interactions of nodes though the connections lead to a global behaviour of the particular network and this global behaviour is said to be "emergent".

This paper is organised as, the Section II describes the architecture of ANN and how it can be used for solving complex problems, the Section III explores how ANN can be applied for intrusion detection, the Section IV outlines the potential attacks on SOA, the Section V briefs the related works of the proposed security solution, the Section VI explains the proposed model for predicting SOA Security vulnerabilities using ANN, and the Section VII concludes the paper.

II. PROBLEM SOLVING WITH ANN

Neural networks works on a different approach in problem solving than conventional computing techniques. Conventional computing techniques use the procedural and algorithmic approach which follows the processing of predefined set of instructions in order to solve a problem. That means, we should know exactly how to solve the problem and then we are feeding the computers how to automate the tasks. It will be very useful if the computers solve the problems that we don't exactly how to describe in steps. The Artificial Neural Networks is one of such kind of problem techniques.

The neural networks process the data in the similar way that the biological brain does. A large number of interconnected neurones (the processing elements) work in parallel in solving a specific problem. Thus, neural networks learn by examples (training). It is contradict to algorithmic step by step instructions. With ANN, the humans do not need to interpret the produced results. The disadvantage of neural networks is its operation can be unpredictable because of the network finds out how to solve the given problem by itself [Sethi et al., 8; Shifei Ding et al., 9; Chirag Modi et al., 10; Emilio Corchado & Álvaro Herrero, 11].

The ANN techniques are applied to solving complex problems in the domains: Modelling (functional approximation, regression analysis), Data Processing (clustering, filtering, mining, compression), Forecasting (prediction, sampling), Classification (pattern recognition, decision making), Computational Neuroscience (neuro-hydrodynamics), and Estimation (control).

III. APPLYING FEED FORWARD ANN FOR INTRUSION DETECTION

An artificial neuron is a computational model inspired in the biological neurons. The natural biological neurons receive signals through synapses located on the dendrites of the neuron. When the received signal is strong enough which surpass a certain threshold, then the neuron is activated and then emits a signal though the axon. The basic structure of a biological neuron is shown in Figure 1. A neuron has a cell body called as "Nucleus", a branching input structure named "Dendrites" and a branching output structure named "Axon". Axons connect to dendrites via synapses. The electro chemical signals are propagated through nucleus from dendrites to axon, which will be passed to other neurons as dendrite. A neuron only fires if its input signal exceeds a certain amount (called threshold) in a short period of time.

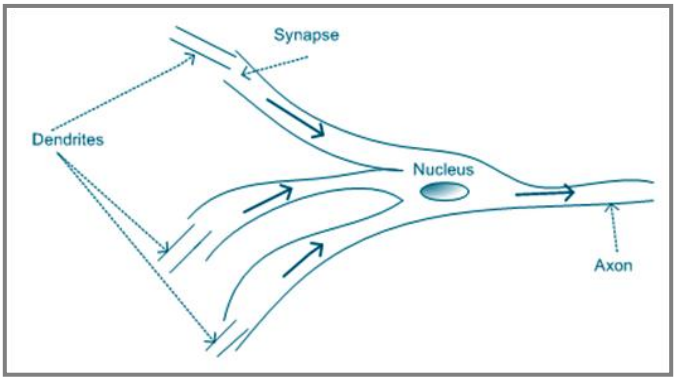

Figure 1: Biological Neuron

The artificial neuron basically consist of inputs, which will be multiplied by weights i.e. the strength of the respective signals, and then computed by a mathematical function which determines the activation of the neuron. If the weight is high, then it means the corresponding neuron is strongly connected. The Figure 2 outlines basic structure of an artificial neuron.

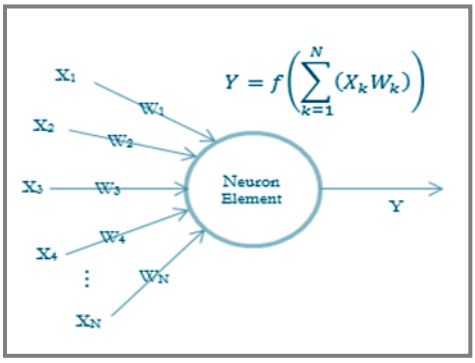

Figure 2: Artificial Neuron

Artificial Neural Networks incorporate the two fundamental components of biological neural nets: Neurons as nodes and Synapses as weights. The weights can be negative, zero, or positive. The computation of the neuron will differ depending on the weights. We can obtain the output that we want for specific inputs to a neural network by adjusting the weights of the artificial neurons in the network. For the neurons in a small neural network, we can manually adjust the weights. It is very complicated if the number of neurons in a network is huge. But we can find algorithms which adjusts the weights in order to obtain the desired output from the neural network. This process of adjusting the weights is called learning or training.

The sum of the input signals (x) multiplied with their corresponding weights will be feed to the transfer function which will determine whether to output 1 or 0. We can choose any adaptable transfer function based on our problem space such as Linear, Hyperbolic Tangent Sigmoid. The

mathematical forms of these functions are shown in Table 1.The back-propagation algorithm is used in layered feed-forward ANNs where the artificial neurons are organized in layers, and send their signals "forward", and then the errors are propagated backwards. The neural network receives input from input layers and produces the output to the output layer and the processing can be performed in hidden layers. There should be only one input and output layer, but there might be any number of hidden layers.

Table 1: Transfer Functions Representation

Function Name	Mathematical Form
Linear	$f(x) = x$
Hyperbolic Tangent Sigmoid	$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$
Logistic Sigmoid	$f(x) = \frac{1}{1 + e^{-x}}$

The back-propagation algorithm uses supervised learning. The supervised learning accepts inputs and the desired outputs; then adjusts the weights of the network until the network produces the desired output for the given specific inputs. For that we provide the algorithm with examples of the inputs and outputs we want the network to compute, and then the error (difference between actual and expected results) is calculated. The core idea of the back-propagation algorithm is to reduce this error, until the ANN learns the training data. Usually the training begins with random weights, and the learning goal is to adjust them so that the error will be minimal.

The goal in training a neural network is to find the gradient of each weight with respect to the output:

$$\frac{\partial E}{\partial w_{ij}}$$

where E is the sum of all input values multiplied with their corresponding weights, and w_{ij} represents the weight assigned to the edge connecting i and j nodes. The back-propagation algorithm does this action so that we can update the weights incrementally using stochastic gradient descent:

$$w_{ij} = w_{ij} - \eta \frac{\partial E}{\partial w_{ij}}$$

where the adjustment of each weight (w_{ij}) will be the negative of the constant eta multiplied by the previous weight on the network.

IV. Literature Review on SOA Security Attacks

The basic architecture of SOA consists of three main components: (i) Service Provider, (ii) Service Registry, and (iii) Service Requestor. The service is a basic concept and core of an SOA. It is the technical representation and encapsulation of high-level business functionality. 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. Service Requestor is an entity that requires certain functions which are published by Service Providers, to perform its own tasks.

The service provider has to publish the service description (Publish) in order to allow the service requester 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.

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. SOAP has been created to transport XML documents from one computer to another. SOAP can be used with a number of standard transport protocols; it is the binding part of web services.

Although an SOA can be implemented using different technologies, web services technology is commonly used [Gholami et al., 6]. The threats on 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.

V. Related Works

In the literature of SOA Security on SOAP based Web Services, so many researchers had done research and proposed a number of approaches in the form of various security models, frameworks, and architectures. This section gives a state-of-the-art overview of the frameworks and other solutions provided by the researchers in this field.

Navya Sidharth & Jigang Liu [12] introduced a framework named "Integrated Application and Protocol-based Framework (IAPF)" for enhancing Web Services Security. The framework works on the four sequential operations: (i) Protection against UDDI attacks, (ii) Protection against WSDL attacks, (iii) Protection against SOAP attacks, and (iv) Protection against attacks on openly available web services. Deven Shah & Dhiren Patel [13] worked on Global SOA security. They used message interceptor to apply security before sending and after receiving the SOAP request and response.

Deepti Sisodia et al., [14] developed an inbuilt security model named "SecSOA" which uses Public Key Infrastructure (PKI). Shahgholi et al., [15] proposed a security solution which prevents WSDL attacks such as WSDL scanning, Parameter Tampering and WSDL Wrapping. Nafise Fareghzadeh [16] presented a comprehensive method for Web Services Security guaranty in SOA. Wei She et al., [17] proposed an effective security framework for web services chain, for example client calls service-1 which calls service-2, which in turn calls service-3.Tao Xu & Chunxiao Yi [18] proposed a security model named SIMSA (Security Interactive Model based on SOAP and Authentication) which solves security issues due to cross platform web service call.

Kou Hongzhao [19] proposed a token based security service for accessing web services. However, it only works on closed environment of web services implementation. Prachi M. Kharat et al., [20] proposed Single Sign On (SSO) certificate based authentication for Web services security. Nils Gruschka and Norbert Luttenberger & Norbert Luttenberger [21] introduced a system for protecting Web Services from Denial-of-Service (DoS) attacks.

Among Web service's XML attacks, most of them appear in SOAP messages. So, in many previous studies, those attacks have been analyzed and related solutions also have been offered. But very few researches only have done works on preventing WSDL threats.

VI. Designing ANN Model for Securing SOA

The design of the system for predicting SOA attacks using ANN follows a number of systematic procedures as outlined in Figure 3.

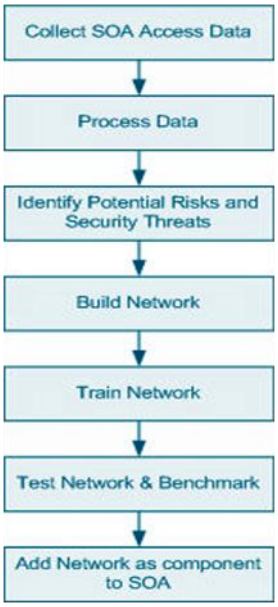

Figure 3: Design of the Proposed ANN Model

The following are the core modules of designing an Artificial Neural Networks model with supervised learning for predicting security attacks on Service Oriented Architectures.

- Data Collection on SOA Security Threats
- Data Processing
- Data Analysis
- Construction of Neural Nets
- Training the ANN
- Interpret the Result

An ANN model with back-propagation learning algorithm is designed and implemented for web services in a banking environment. This ANN based model produces reliable results than other forecasting techniques.

A separate security component is attached at the server side, which is the entry and exit point for web service access. For example, if a web service client tries to call a web method that is provided by a web service on the web server, then the web service request will be caught by the security component which will decide whether to allow for the service or not by checking vulnerabilities on the request by client. In the same way, when the web server tries to send the response to the requested web service client, then the security component will check the response for vulnerabilities before sending it to the requested client. Thus the security component acts as a gateway for request and response of a web service.

The security component gets its knowledge on the security vulnerabilities through the ANN learning techniques periodically. However, a bulk number of security risks and their security severity levels can be configurable through batch job as learning process through ANN is time-consuming and requires additional cost of programming. The data on the security threats are collected through data mining, and the data is processed to categorize the security threats. The categorized data will be analyzed for interpreting security threats, the required set of neural networks are created and trained to identify the SOA security vulnerabilities.

VII. 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 development of applications. However, it brings additional security threats as it uses simple text format for information exchange. There should be a comprehensive forecasting mechanism should be implemented in organizations who uses SOA for EAI, especially in organizations where there data are very sensitive. In this paper, we proposed a forecasting technique to predict SOA attacks using neural networks technology. This approach is prominent one and produces reliable results when the security attack problem space is wide. The construction of algorithms for predicting known security attacks is the future work of our project.

REFERENCES

[1] Richard Welke, Rudy Hirschheim & Andrew Schwarz (2011), "Service Oriented Architecture Maturity", *IEEE Transaction on Computers*, Vol. 44, No. 2, Pp. 61–67.

[2] Mustafa Bozkurt, Mark Harman & Youssef Hassoun (2013), "Testing and Verification in Service-Oriented Architecture: A Survey", *Software Testing, Verification and Reliability*, Vol. 23, No. 4, Pp. 261–313.

[3] Michael Rosen, Boris Lublinsky, Kevin T. Smith & Marc J. Balcer (2008), "Applied SOA: Service-Oriented Architecture and Design Strategies", *John Wiley & Sons Book Publication*,

[4] S. Subashini & V. Kavitha (2011), "A Survey on Security Issues in Service Delivery Models of Cloud Computing", *Journal of Network and Computer Applications*, Vol. 34, No. 1, Pp. 1–11.

[5] B. Shah & Bhushan H. Trivedi (2012), "Artificial Neural Network based Intrusion Detection System: A Survey" *International Journal of Computer Applications*, Vol. 39, No. 6.

[6] M. Gholami, N. Cai & R.W. Brennan (2013), "An Artificial Neural Network Approach to the Problem of Wireless Sensors Network Localization", *Robotics and Computer-Integrated Manufacturing*, Vol. 29, No. 1, Pp. 96–109.

[7] K.S. Devikrishna & B.B. Ramakrishna (2013), "An Artificial Neural Network based Intrusion Detection System and Classification of Attacks", *International Journal of Engineering Research and Applications (IJERA)*, Vol. 3, No. 4, Pp. 1959–1964

[8] K. Sethi, Ishwar & Anil K. Jain (2014), "Artificial Neural Networks and Statistical Pattern Recognition: Old and New Connections", Vol. 1, *Elsevier Publications*.

[9] Shifei Ding, Hui Li, Chunyang Su, Junzhao Yu & Fengxiang Jin (2013), "Evolutionary Artificial Neural Networks: A Review", *Artificial Intelligence Review*, Vol. 39, No. 3, Pp. 251–260.

[10] Chirag Modi, Dhiren Patel, Bhavesh Borisaniya, Hiren Patel, Avi Patel & Muttukrishnan Rajarajan (2013), "A Survey of Intrusion Detection Techniques in Cloud", *Journal of Network and Computer Applications*, Vol. 36, No. 1, Pp. 42–57.

[11] Emilio Corchado & Álvaro Herrero (2011), "Neural Visualization of Network Traffic Data for Intrusion Detection", *Applied Soft Computing*, Vol. 11, No. 2, Pp. 2042–2056.

[12] Navya Sidharth & Jigang Liu (2007), "IAPF: A Framework for Enhancing Web Services Security", *IEEE 31st Annual International Computer Software and Applications Conference (COMPSAC 2007)*, Vol. 1, Pp. 23–30.

[13] Deven Shah & Dhiren Patel (2009), "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.

[14] Deepti Sisodia, Lokesh Singh & Sheetal Sisodia (2012), "Web based Secure SOA", *International Journal of Computing Algorithm*, Vol. 01, No. 02, Pp. 63–69.

[15] N. Shahgholi, M. Mohsenzadeh, M.A. Seyyedi & S.H. Qorani (2011), "A New SOA Security Framework Defending Web Services against WSDL Attacks", *Proceeding of IEEE 3rd International Conference on Social Computing (Socialcom)*, Pp. 1259–1262

[16] Nafise Fareghzadeh (2009), "Web Service Security Method to SOA Development", *World Academy of Science, Engineering and Technology*, Pp. 815–819.

[17] Wei She, I-Ling Yen & Bhavani Thuraisingham (2008), "Enhancing Security Modeling for Web Services using Delegation and Pass-on", *IEEE International Conference on Web Services (ICWS)*, China, ISBN: 978-0-7695-3310-0, Pp. 545–552

[18] Tao Xu & Chunxiao Yi (2011), "SOAP-based Security Interaction of Web Service in Heterogeneous Platforms", *Journal of Information Security*, Vol. 2, No. 1, Pp. 1–7.

[19] 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, Vol. I.

[20] Prachi M. Kharat, Prachi A. Deshpande & Aaditya P. Bakshi (2013), "Single Sign On Certificate based Authentication for WS-Security", *International Journal of Advanced Research in Computer Science*, Vol. 4, No. 6.

[21] Nils Gruschka & Norbert Luttenberger (2006), "Protecting Web Services from DoS Attacks by SOAP Message Validation", *Security and Privacy in Dynamic Environments*, Springer US, Pp. 171–182.