Neural Network Approach to Web Application Protection

Jane Jaleel Stephan, Sahab Dheyaa Mohammed, and Mohammed Khudhair Abbas

***Abstract*—The rapid growth of internet has created many services, which have become an integral part of our day in today life by using Web applications for making reservations, paying bills, and shopping on-line.**

The vulnerabilities in web application code provide an opportunity to the attack to be entre on applications level. Most network firewalls and antivirus software programs cannot stop attacks at the application level.

In this paper, we have developed a prototypic web application firewall to detect new types of attacks that do not require signature updates, using a neural network back-propagation approach for identifying attacks that were not detected at the stage of signature analysis.

The solution has been experimented on some parameters and some additional information about the user behaviors when the user accesses the web application and makes application-level control of the firewall in the framework of the scope of the WEB-application.

The system is found to have good performance in comparing and matching the test patterns with already stored patterns and from (24) test data, (95%) success rate have been correctly recognized.

***Index Terms*—Web applications firewall, signature, artificial neural network.**

I. Introduction

The increasing shift towards web applications opens new attack vectors. Traditional protection mechanisms like firewalls were not designed to protect web applications and thus do not provide adequate defense. Current attacks cannot be thwarted by just blocking ports 80 (HTTP) and 443 (HTTPS) [1].

Preventive measures (like Web Application Firewall rules) are not always possible. Reactive methods to detect what happened previously are usually easier but have the disadvantage of always being behind the actual event [2].

Protocol-enforcing network firewalls typically provide the first line of defense by arresting most basic protocol attacks at the network perimeter, including protocol based denial of service attacks. They primarily operate in the network, session, and transport layers of the Open Systems Interconnection (OSI) reference model [3].

Developers have also greatly enhanced the capability of network firewalls to police the protocol integrity of a wide range of upper-layer protocols such as DNS, FTP, HTTP, SMTP, and TFTP.

Network firewalls can also verify that traffic passed over non-standard ports, such as SMTP, running over port 25, conforms to valid SMTP traffic [3].

Standard firewalls can help restrict or permit network access to network ports authorized by the organization. Although application proxy firewalls exist, they cannot understand the specific content of all web applications being run by the organization.

There are two protection approaches:

1) Signature based: The WAF identifies attacks by checking web request against an "attack signature" file.
2) Abnormal behavior based: The WAF identifies attacks by detecting abnormal traffic patterns [4].

This paper will go through the concept of artificial neural networks, the way to apply it with signature analysis in the form of a web application firewall and to make application-level control of the firewall in the framework of the scope of the WEB-application.

II. Web Application Security

Web application security is a branch of Information Security that deals specifically with security of websites and web applications. It differs from the other branches of Information Security in that web application security is focused on vulnerabilities within the application code that is exposed during a user session on the web.

A majority of the attacks against web servers are through network firewalls and through the http (80) or https (443) ports. Some of the most commonly used hacking techniques include denial of service, leakage, cross-site scripting, SQL injection and disclosure [5].

To keep web application secure besides standard firewalls, various types of solutions are used in application layer: external tools – web application scanners and firewalls (WAS, WAF) and internal – the application itself must be self-defending [6].

A web application scanner is an automated program that examines web applications for specific security vulnerabilities [7]. A WAS uses the negative logic *(blacklist)* based filtering algorithms to detect vulnerabilities in web applications. Negative logic filtering built on signatures of known attacks and allows security systems to prevent any requests that appear to match the attacks' signatures from reaching protected servers. A WAF can work using any web traffic filtering mechanism (positive, negative, or session) and it either works as an embedded firewall with the web server or as a separated layer of security in a reverse proxy form [8].

III. Web Application Firewall

Web application firewalls (WAFs) are hardware or software devices positioned to monitor website traffic, with

Manuscript received December 2, 2013; revised March 6, 2014.

The authors are with Iraqi Commission for Computers and Informatics, Ministry of Higher Education and Scientific Research, Iraq (e-mail: janejaleel@yahoo.com, Sahab7dia@yahoo.com, mohammedalaaly@yahoo.com).

DOI: 10.7763/IJIET.2015.V5.493

the ability to enforce policy on browser/server transactions. WAFs are similar, though not identical to, network firewalls where policies are typically applied to IP addresses, ports, and protocols. WAFs are specifically designed to inspect HTTP(s) traffic and regulate data contained within headers, URL parameters, and web content. Another similarity is network firewalls are used to protect insecure hosts from remote exploitation. WAFs do the same for insecure websites. With a WAF in place, malicious hackers may target insecure websites, but attacks are intercepted and denied before reaching the custom web application code [9].

Web Application Firewalls (WAF) are mitigations for these vulnerabilities that do not aim at fixing the actual vulnerable application, but that try to detect and to prevent rogue requests.

To distinguish normal requests from rogue requests, WAFs use a set of filter rules in the form of white lists, black lists, or a combination of both. Commonly, the WAF will pass only those requests to the application that are classified as normal requests. Requests classified as rogue are usually blocked and thus not passed on to the application. Creating filter rule sets is challenging because on the one hand if the WAF blocks some normal requests (false positive), then the application may not function any more. On the other hand, if the WAF does not block all rogue requests (false negative), then the attacker may circumvent the WAF and exploit a vulnerability in the application [10].

Positive logic filtering allows valid requests based on a signature set *(white list)* detailing what types of communications protected server's know- how to handle; it prevents any requests not known to be valid from reaching secured servers.

Session based filtering (also called event-driven dynamic rules) utilizes positive logic based rules but allows the inclusion of variables in the rule set. The values of the variables are set dynamically during user sessions.

The disadvantages of positive logic filtering: is requirement of large vulnerabilities database based of regex rules. It causes poor bandwidth, requires more resources, hardly adaptable to large web systems.

By reducing the number of rules, in order to improve bandwidth, decreases vulnerability detection quality. In order to increase the WAF performance, bandwidth, WAF developed using artificial intelligence techniques Artificial Neural Networks, fuzzy logic [11].

Two approaches are used to detect the attacks; signature based and anomaly-based. The former is used to identify known attacks and a regular update of signatures is required whereas the later is used to identify unknown or new attacks, which will be the deviation of the model constructed in the initial attacks-free learning phase. Both of the approaches will identify the attack instead of legitimate web requests [12].

There are differences between normal firewall and web application firewall. The normal firewall deals with network layer (Layer-3 OSI) while web application firewall deals with application layer (Layer-7 OSI) [13].

IV. RTIFICIAL NEURAL NETWORK

An artificial neural network consists of a group of processing elements (neurons) that are highly interconnected and convert a set of inputs to a set of preferred outputs [14]. The first artificial neuron was formed in 1943 by the neurophysiologist Warren McCulloch and the logician Walter Pits [15].

The basic component of an ANN is the neuron. Each neuron has three important components as shown in Fig.1, a set of synaptic connections (which are represented by a set of synaptic weights and bias, (i.e. wki and bk); a propagation function (Σ) which is a linear combination between the input elements modified by the set of synaptic weights and bias; and an activation function (φ) which takes the output of the propagation function as its input and generates the output of the neuron. It is the set of synaptic weights and bias that stores the knowledge acquired during the learning phase [16].

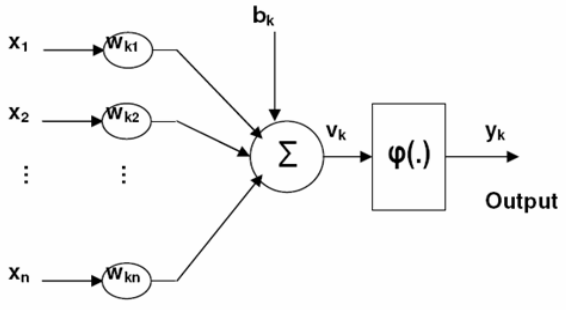


Fig. 1. A model of a neuron.

The way neurons are connected to one another will define the architecture of an ANN. In this research, Multilayer Feedforward Networks (MLN) is used. The architecture of MLNs is shown in Fig. 2.

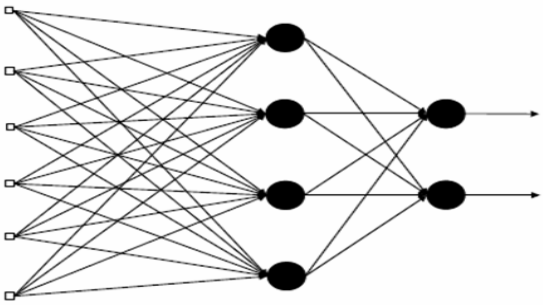


Fig. 2. Multilayer feed forward Network (MLN).

With the learning ability, ANN can be trained to perform different engineering tasks. Some of the tasks that can be identified are pattern recognition, pattern association, function approximation, control systems, filtering, and beam forming [16].

Artificial neural networks are alternatives. The first advantage in the use of a neural network in the attack detection would be the flexibility that the network would provide. A neural network would be capable of analyzing the data from the network, even if the data is incomplete or unclear. Similarly, the network would possess the ability to conduct an analysis with data in a non-linear fashion. Further, because some attacks may be conducted against the network in a coordinated attack by multiple attackers, the ability to process data from a number of sources in a non-linear fashion is especially important.

The problem of frequently updation of traditional attack

detector is also minimized by ANN. It has generalization property and hence able to detect unknown and even variation of known attacks. Another reason to employ ANN in probing attack detection is that, ANN can cluster patterns which share similar features, thus the classification problem in attack detection can be solved by ANN. The natural speed of neural networks is another advantage [17].

There are some algorithms that can be used to train an ANN for a pattern recognition task, such as: Back Propagation, Radial-basis Function, and Support Vector Learning, etc. Among them, *Back Propagation* is the algorithm that is specifically devised to train a multilayer perceptron.

V. Back-Propagation Learning

Back- propagation learning has emerged as the most significant result in the field of artificial neural networks. The back-propagation learning involves propagation of the error backwards from the output layer to the hidden layers in order to determine the update for the weights leading to the units in a hidden layer. The error at the output layer itself is computed using the difference between the desired output and the actual output at each of the output units. The actual output for a given input training pattern is determined by computing the outputs of units for each hidden layer in the forward pass of the input data. The error in the output is propagated backwards only to determine the weight updates [18].

VI. The Proposed System

The rapid development of the Internet and increase in the growth of attack and cyber-crime against web applications does not allow way to update the database of signature. It should be noted that even if there is a centralized operating system to update the database of signature, it is not efficient solution in the long term and the updated run database of signature may come too late, then the attack takes place.

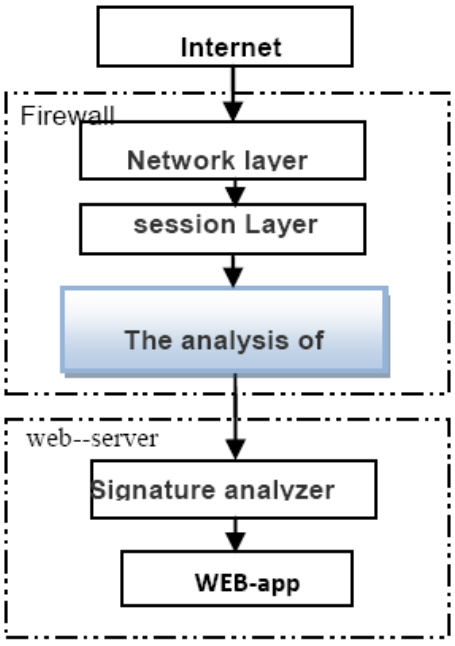


Fig. 3. Classical diagram of protection WEB-application.

Most of the attacks on the WEB-application can be recognized only at the application level. Fig.3, shows the classical diagram of protection of WEB-application, which involves an analysis of the data at the application level in the firewall.

However, the firewall often cannot distinguish between malicious user actions and the actions of a legitimate user. From this situation, there are two possible options: First option is the firewall trains all the features of user behavior for each protected WEB-application. The second option which is the most logical and convenient is to make application-level control of the firewall in the framework of the scope of the WEB-application see Fig. 4 When the analysis is at the application layer is transferred into the framework of the WEB-application, it takes on a different type, offers a number of options to analyze and gather information about potential attacks.

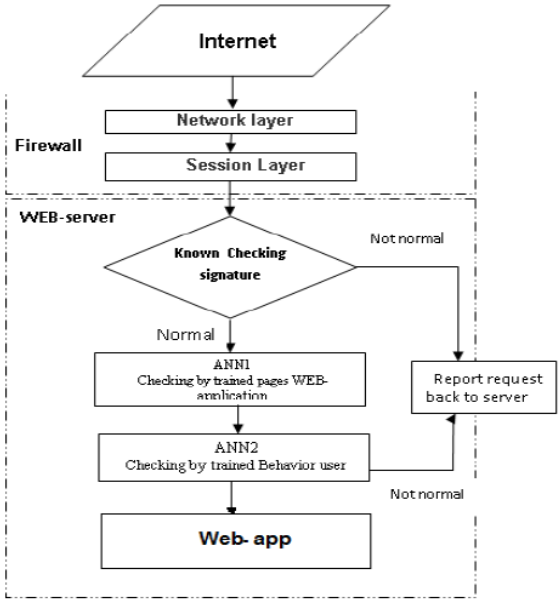


Fig. 4. The proposed diagram.

The HTTP protocol allows the user's browser to transfer information inside the URI itself (i.e., GET parameters), in the HTTP headers (e.g., in the Cookie field), or inside the request body (i.e., POST parameters). The adopted technique depends on the application and on the type and amount of data that has to be transferred [19]. proposed diagram of the analysis a system that consists of a neural network stages and a set of collected information in several sections, POST-parameters, GET-parameters, COOKIE-parameters, database operations, file system operations, errors and warnings in the user experience, which come from the information and some additional information about the user (IP address, country, name and version of browser name and version of operating system, language system, screen resolution, color depth, browser home page) are the input to the two neural networks stages .

The main tasks assigned to the neural networks, is identifying attacks far away of the stage of signature analysis. It is necessary to emphasize that there is no need to train the neural network on all currently known types of attacks. Known existing attack must be processed from signature analyzer and neural network must be able to distinguish

between legitimate user behavior and malicious user behavior.

The proposed solution summarized in Fig. 4. Before WEB-application is open to access from the Internet, the administrator provides training two neural networks to protect the page.

The administrator activates the learning mode and starts the application, trying to initiate the value of a parameter in the first network (for example, the size of uploaded files, the number of GET parameters in a singled query, etc.) .Then the set of value parameters gathered and made into a binary vector and enter to the first neural network, it is shown in Table I.

TABLE I: INPUT VALUES OF PARAMETERS

	Classes of the parameters	Scope of the authorize	Otherwise
1	the number of GET-parameters	1	0
2	the number POST-parameters	1	0
3	number COOKIE-parameters	1	0
4	The number MIME type of uploaded files	1	0
5	The number of response header HTTP	1	0
6	The names of the affected database tables.	1	0
7	The actions carried out with the tables in the database.	1	0
8	The number of errors that occurred when the scripts	1 if none occurred	0

One can distinguish four basic categories of users WEB-application from a security perspective they represent the targeted classes of two networks in binary manner are shown in Table II:

1) **Authenticated** users (are the one system administrator or more operators that control the system) not checked by neural network.
2) **Normal users** (users cannot pose a threat).
3) **Suspicious users** (users who have shown some suspicious activity, but the data collected is not enough to identify the user as an attacker)
4) **Attacker user (**users who can pose a threat).

TABLE II: OUTPUT PATTERNS NN1 AND NN2

	Category of attack	Output patterns
1	Normal users	11
2	Suspicious" users	10 01
3	Attacker	00

The output patterns of the first network represent one of user category entered as the input vector to the second network with some parameters of user behavior for increase in the coefficient of verification. In the learning process of the second neural network, the vectors is compared with the parameters laid in memory of a neural network, and concludes a normal user behavior, which led to the emergence of such a normal vector. The parameters values of the second network are represented in five bits of the binary vector plus two output bits of the first network, which are shown in Table III.

TABLE III: INPUT VECTOR REPRESENTATIONS AS BITS IN NN2

a) FIRST AND SECOND VECTORS(FROM OUTPUT OF NN1) .TWO BITS

	Category of attack	Output patterns
1	Normal users	11
2	Suspicious" users	10 01
3	Attacker	00

b) THIRD AND FOURTH .TWO BITS

Client Browser	Vector Representation
Internet Explorer	00
FireFox	01
Google Chrome	10
Other	11

c) FIFTH BIT

Set Flash Player	Vector Representation
No	0
Yes	1

d) SIXTH BITS

System Language	Vector Representation
Arabic	0
Other	1

e) LAST BIT

Operating System	Vector Representation
Unix	0
Windows	1

After training the networks administrator transmitted into operating mode and provides access to the WEB-application from the Internet.

If the second network cannot say exactly what is going about the attack, the degree of deviation from normal behavior of the model is sufficiently large, the system indicates as a potential attacker to remember a network, responsible for the assignment of users to a particular category. The next time the user returns to the site and produced some of the suspected cause of action, the neural network classifies it as "suspicious" of the user, and tightens the analysis of deviations from normal behavior by the correction coefficient similarity. If suspicions are confirmed, then the user will be redirected to the category of attackers, and then all of his subsequent actions will be blocked.

The second network stores the data about the behavior of user (IP address, country, name and version of browser name and version of operating system, language system, if there is

any support for Flash, if there is support for Java, screen resolution, color depth, browser home page). In analyzing the output of the first neural network, we need to further define the identity of the user by using the second neural network. In analyzing the output of the second neural network, it is decided if the user behavior is normal, or whether it deviates from the normal scheme.

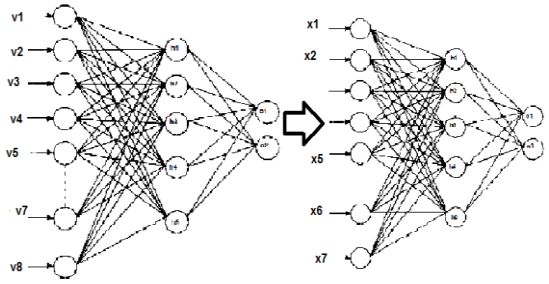
VII. Experiment

This section, explains training of ANN. Training is one of the affective process in the system. The training and recognition processes are given below.

A. Input Data

At first neural network has one input layer, one hidden layer and one output layer, the input layer size is (8) neurons, the hidden layer has (5) neurons, the output layer size is (2) neurons, Each input pattern represents a vector in a protected page as shown in Fig. 5 a).

The second neural network also has one input layer containing (7) neurons (5) input vectors representing one user behavior and the remaining (2) vectors represent the results for the first network), one hidden layer which contains (5) neurons, one output layer containing (2) neurons, as shown in Fig. 5 b).



a). First of ANN b). Second of ANN

Fig. 5. Structure of the ANN.

B. Training Neural Network

For training the neural networks, back propagation algorithm uses some parameters, which are experimentally set, that need to be addressed upon training the network, these parameters allow the algorithm to converge more easily.

In training data set is divided into two sets. The first set of data is training data sets with 240 patterns that were presented to the neural network during training. Twenty percent of the training set has been truncated for testing performance of the neural network on 48 patterns after it has been trained. The first network contains a set of the training data of 137 patterns and the second network contains 103 patterns as shown in Table IV. To increase the networks effectiveness and to make them more suitable, the best number of parameters hidden neurons , learning rate (*l*), error rate (e), which are experimentally set, These parameters allow the algorithm to converge more easily. The training stops when the generalization stops improving or when the 500 epochs is reached.

Different network architectures have been attempted by varying the number of hidden nodes such as 3, 4, 5 nodes in network1 also 3, 4, 5 nodes in network2. several experiments were carried out to select the best learning values which were chosen in the range (0-1) .in order to find their effects on the amount each weight is changed for a given error rate through network learning**.** The initial connection weights are in the range of [-1, 1].

The results and analysis of the performance of two neural networks is presented in the next step.

TABLE IV: Training and Test Data Sets

Training Networks	Training Data	Test Data
Network1	137	24
Network 2	103	24
Total	240	48

C. The Experimental Results

To produce a good result, network1 uses (24) test data and learns the back-propagation network producing the results presented in Table V. After many epochs, i.e. training sessions, the network will give an output that is close enough to the desired output. Learning rate (*LR*=0.1) used with (5) hidden nodes produces a good result, has a smaller error than the other numbers of the hidden nodes and the lowest number of iterations is used when the training is 110 with (5) neurons implemented.

(*LR*=0.5) requires long time and gives the largest number of iterations and the time required to adjust the weights through learning the network is shorter as compared with (*LR*=0.1). Out of (24) test data, (92%) success rate have been correctly recognized. The remaining digits refer to (8%) false recognition that the system gave.

TABLE V: Results of the Networks1 with Different Learning Rates and Numbers of Hidden Nodes

Learning Rate	Hidden nodes	No. of iterations	Time minute	Rec. rate %	Reject rat %
0.1	3	150	1.12	85.73	14.27
0.1	4	128	0.66	88.84	11.16
0.1	5	110	0.45	92.17	7.83
0.5	3	210	2.15	76.28	13.72
0.5	4	192	1.75	78.27	21.73
0.5	5	184	1.46	84	16.00

TABLE VI: Results of the Networks2 with Different Learning Rates and Numbers of Hidden Nodes

Learning Rate	Hidden nodes	No. of iterations	Time minute	Rec. rate %	Reject rat %
0.25	3	80	0.56	90.54	9.46
0.25	4	65	0.42	93.45	6.55
0.25	5	50	0.35	95.39	4.61
0.75	3	112	1.10	75.36	24.64
0.75	4	103	0.89	78.33	21.67
0.75	5	95	0.66	81.45	18.55

The performance of neural network 2 using (24) test data and learning for the back-propagation network produces the results presented in Table VI. Network 2 is better when using learning rate (*LR*=0.25) with (5) hidden nodes and produces a good result with a smaller error, and gives the lowest number of iterations and time than the other numbers of the hidden nodes.

(*LR*=0.75) gives the largest number of iteration and time and requires adjusting the weights through learning the network (*LR*=0.25).

Out of (24) test data, (95%) success rate have been

correctly recognized. The remaining digits refer to (4%) false recognition that the system gave.

The two Neural networks that are used in this paper to protect individual WEB pages, work on the same principle, but the results of their work are different between them , perhaps there is a match in the output vector of these networks, or not. The result of the second netwowrk2, which represent a final result, sends to desired application when the result is normal otherwise it sends alarm to server about this situation and the user will be redirected to the category of attackers.

VIII. Conclusions

The proposed system has the ability to detect new types of attacks that do not require signature updates, as based on abnormalities of behavior, making it possible to track the user's actions, repeatedly committing attempted burglary, fully adapted to the features of the protected WEB-application. A neural network back-propagation approach is identifying attacks that were not detected at the stage of signature analysis, this is more efficient to make application-level control of the firewall in the framework of the scope of the WEB-application. The system is found to have good performance in comparing and matching the test patterns with already stored patterns. In future, the proposed system is extended for analysis of every parameters in web pages and more coverage of the user's behavior.

References

[1] R. Tiwari1 and R. Kumar, "Mobile Agent Based Distributed Intrusion Detection System: A Survey," *International Journal of Computer Applications in Engineering Sciences*, vol. II, issue III, September 2012.

[2] R. Meyer, "Detecting Attacks on Web Applications from Log Files," SANS Institute, InfoSec Reading Room, January 2008.

[3] A. J. Hacker, "Importance of Web Application Firewall technology for Protecting Web-based Resources," *ICSA Labs an Independent Verizon Business*, January 2008.

[4] Web Application Security, Top Ten Project Theme Page of the OWASP Website, OWASP_ Top _Ten Project. (February 2008). [Online]. Available: http://www.infosec.gov.hk/english/technical/files/web_app.pdf

[5] J. Panella, "Web Application Security," *the OWASP Top 10, Sapient Corporation, SapientNitro*, March 22, 2011.

[6] Web Application Security Risks, OWASP. (2010). The Ten Most Critical. [Online]. Available: https://www.owasp.org/index.php/Top_10.

[7] E. Fong and V. Okun, "Web Application Scanners: Definitions and Functions," in *Proc. the 40th Hawaii International Conference on System Sciences, IEEE*, 2007, pp. 280b-280b.

[8] I. Desmet, F. Piessens, W. Joosen, and P. Verbaeten, "Bridging the gap between web application firewalls and web applications," in *Proc the fourth ACM workshop on Formal methods in security*, Alexandria, Virginia, USA, 2006, pp. 67–77.

[9] N. Khochare, S. Chalurkar, and B. B. Meshram, "Web Application Vulnerabilities Detection Techniques Survey," *IJCSNS International Journal of Computer Science and Network Security*, vol. 13, no. 6, June 2013.

[10] I. Schmitt and S. Schinzel, *WAFFle: Fingerprinting Filter Rules of Web Application Firewalls*, University of Engineering and Technology, Taxila, Jul 2012.

[11] E. Kazanavicius, V. Kazanavicius, and A. Venckauskas, "Securing Web Application By Embedded Firewall, Electronics And Electrical Engineering," ISSN 1392-1215, no. 3, p. 119, 2012.

[12] A. H. Yaacob, "Moving Towards Positive Security Model For Web Application Firewall," *World Academy of Science, Engineering and Technology*, 2012.

[13] M. I. B. Rahimi, "Web Application Firewall," University Technology, Mara, May 2006.

[14] U. Aickelin, J. Greensmith, and J. Twycross, "Immune System Approaches to Intrusion Detection – A Review," *Natural Computing, Springer Netherlands*, vol. 6, no. 4, December, 2007, pp. 413-466.

[15] L. Fausett, "Fundamentals of Neural Networks Architecture," Algorithm, and Applications, Pearson Education, Inc., 2008, pp. 21-24.

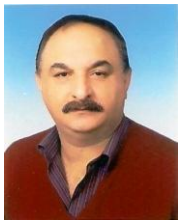
[16] S. Haykin, *Neural Networks, A Comprehensive Foundation*, 2nd Ed., New Jersey, USA: Prentice-Hall Inc., 1999.

[17] I. Ahmad, M. A. Ansari, and S. Mohsin. "Performance Comparison between Backpropagation Algorithms Applied to Intrusion Detection in Computer Network Systems," *Recent advances in systems, communications &computers*, 2008, pp.47-52.

[18] V. Blanz and T. Vetter, "Face Recognition Based on Fitting 3D Morphable Model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, September 2003, pp. 1-5.

[19] M. Balduzzi, C. T. Gimenez, D. Balzarotti, and E. Kirda, "Automated Discovery of Parameter Pollution Vulnerabilities in Web Applications," *Internet society*, 8 Feb. 2011.

Jane Jaleel Stephan was born in Baghdad in 1954. She is a Ph.D. in computer science and an assistant dean for Scientific affairs in informatics institute for postgraduate studies. She is an assist. prof. Her research interests include artificial intelligent, image processing, network security.



Sahab Dheyaa was born in Baghdad in 1960. He is M.Sc. in computer science in College of Science AL-Mustansiriyah University from 2003- 2005. From 1997-2000 he received his B.Sc. in Computer Science-Computer Dept. College of Education, AL-Mustansiriyah University. From 1980 to 1984, he received also his B.Sc in administration science in economic and administration College, AL-Mustansiriyah University. He is an assist. lauctuelar in Informatics Institute for Postgraduate Studies. Current research interests: network security, Image processing.

Mohammed Khudhair Abbas was born in Baghdad in 1982. He received his M.Sc. degree in information engineering, AL-Nahrain University from 2005 to 2008. He is an assist. lauctuelar in Informatics Institute for Postgraduate Studies. Now, he is Ph.D student in computer engineering, Istanbul Technique University, Turkey. His current research interests includes Internet security, voice over Internet protocol, design and implementation of web applications.