Defending Denial of Service Attacks against Domain Name System with Machine Learning Techniques

Samaneh Rastegari, M. Iqbal Saripan* and Mohd Fadlee A. Rasid

***Abstract*—Along with the explosive growth of the Internet, the demand for efficient and secure Internet Infrastructure has been increasing. For the entire chain of Internet connectivity the Domain Name System (DNS) provides name to address mapping services. Hackers exploit this fact to damage different parts of Internet. This paper focuses on Denial of Service (DoS) attacks as the major security issue during recent years. The process of detection and classification of DoS against DNS has been presented in two phases in our model. The proposed system architecture consists of a statistical pre-processor and a machine learning engine. Three different types of neural network classifiers and support vector machines are evaluated as the engine in a simulated network. The performance of our system was measured in terms of detection rate, accuracy, and false alarm rate. The results indicated that a back propagation neural network provides a 99% accuracy and an acceptable false alarm rate of 0.28% comparing to other types of classifiers.**

***Index Terms*—Denial of service, Domain name system, Network security, Neural network, Support vector machines.**

I. Introduction

Originality DNS was designed based on an unreliable delivery protocol named User Datagram Protocol (UDP) and security of DNS was not a big issue at that point in time because the original design was sufficient to satisfy the needs of the Internet [1], [2]. Nowadays, DNS has become a vital service for the operation of the Internet and of any private network of a certain size, so this is the time to secure the DNS system from any unauthorized access.

The first objective of this paper is to evaluate different types of DoS attacks against DNS. Identifying patterns of these attacks lead us to generate the required data for different attack scenarios through simulations by varying different parameters. Two of the most common DoS attacks occur against DNS are the type of direct DoS attacks and amplification attacks. In the first one attacker tries to overwhelm the server by sending an excess traffic from single or multiple sources.

Therefore, it will cause a huge number of query packets to be received by the target name server. The name servers flooded by DoS attacks will experience packet loss and cannot always respond to every DNS request. Reference [3], points that the packet size of DNS data flow is small and this similarity to anomalous packets makes the process of detection more difficult.

On the other hand, attackers establish the most sophisticated and modern type of DoS attacks known as amplification attacks to increase the effect of normal DoS attacks. The reason that this type of attack named amplification is that the attacker makes use of the fact that small queries can generate much larger UDP packets in response [4]. Nowadays, DNS protocol (RFC 2671) is used by the attackers to magnify the amplification factor. For example a 60 bytes DNS request can be answered with responses of over 4000 bytes. This yields an amplification factor of more than 60. Several researchers have studied the effects of reflected amplification attacks. Based on their analyzes, patterns of these attacks include a huge number of nonstandard packets larger than the standard DNS packet size which was 512 bytes [5].

There were several attempts to propose a solution to defend DNS against such attacks [3], [6], but according to our knowledge, there was no specific intelligent detection system for Denial of Service (DoS) threats against DNS and this is the second objective of this work.

The rest of this paper is organized as follows. Next section describes the simulation model for generating our data set. Section III introduces the proposed model for detection and classification of DoS attacks against DNS. The results are presented in section IV and, then we draw some conclusions in section V. The initial result of this paper was presented in [7] and [8].

II. Simulation Model for Dataset Generation

When accessing to a real environment for traffic simulation is hard, we exploit the power of network simulators. According to our knowledge, there were no available generated dataset for DoS attacks against DNS. Therefore, we used simulation for generating the required data for our experiments. We simulate our model using an OTcl program in NS-2 (version 2.28). It is used to model different DoS

Manuscript received November 7, 2010.

Samaneh Rastegari is with the Department of Computer and Communication System Engineering, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia (e-mail: samaneh.rastegari@gmail.com).

Mohd Fadlee A. Rasid is with the Department of Computer and Communication System Engineering, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia (e-mail: fadlee@eng.upm.edu.my).

*(Corresponding Author) M. Iqbal Saripan is with the Department of Computer and Communication System Engineering, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia (e-mail: iqbal@eng.upm.edu.my).

attacks against DNS. The network topology of our simulation contains a single legitimate client, an attacker, and two servers. All nodes are connected to the same router. All the links are 100Mbps and 10ms except the link between target server and router that is 10Mbps and 10ms delay. We used a queue size of 100 packets, with a drop-tail queuing strategy. There are two types of traffic generated in the network which are legitimate traffic and attack traffic. A modified version of Agent/Ping with a maximum of 3 retransmissions with 5-second timeouts is used for DNS as implemented in [9]. For doing this modification, there is a need to change behaviour of existing module (ping.cc and ping.h) in the applications folder in NS-2. The "command" method which is used for sending the requests in ping.cc, is modified using a new field defined in PingAgent class in ping.h file. This field is set to one when the transaction is successful, and it means there is no need to have retransmission. So, before sending any request by a sender, we need to check the value of this field. If it is not set to one then the "command" method will start to create a new packet and send it. This process is presented in Fig. 1.

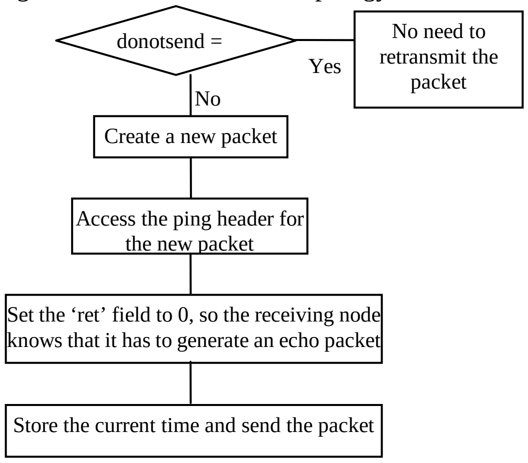
donotsend =
Yes
No need to retransmit the packet
No
Create a new packet
Access the ping header for the new packet
Set the 'ret' field to 0, so the receiving node knows that it has to generate an echo packet
Store the current time and send the packet

Fig. 1. Modified "command" method in ping.cc

In our simulation we attach the modified application to the servers. We follow the model set by [9], whereby the request interarrival period is fixed at 10s. The attacker is expected to flood the target name server with excess traffic. The DoS traffic is modeled as constant bit rate (CBR) source. CBR can be generated by the CBR traffic generator in NS-2. We chose different values of delay for applying to the attack start time in order to achieve variability. This can be done by using a RNG (Random Number Generation) object in NS-2.

III. Materials and Methods

This section presents a new attack detection system for DoS against DNS, which uses a machine learning engine to detect and classify attacks. This IDS is a network-node based IDS (NNIDS), which can be implemented on a name server for the purpose of attack detection. Fig. 2 illustrates the overall architecture of our proposed system with input-output data types.

This system starts by gathering packet stream that was received by a name sever. Next, the pre-processor starts analyzing the traffic statistically based on an administrator specified time window of 20s length, which is more than the maximum lookup latency. The parameters that are going to characterize the DNS traffic received by the name server and that constitute the input of the classifier are defined as follows:

1) Throughput of received DNS requests that is defined as the number of received bits at the server. We measured the average value of this metric for the specified time window.
2) Average size of received packets by the server during a monitoring time window.
3) Packet loss that is defined as the number of lost DNS packets that did not reach their destination due to flooding attack traffic.

The data splitting method that will be applied to our preprocessed data set should be clarified. Currently, the most commonly used validation methods for data splitting are k-fold cross validation and bootstrap method [11]. The advantage of k-fold cross validation comparing to bootstrap method is that all the cases in the data set are eventually used for both training and testing. Unlike bootstrap method k-fold cross validation does not use substitute cases to form the training data sets. Therefore, there is no overlap between the training data sets. This advantage can solve overfitting problem. Overall, 10-fold cross validation provides a strong test to obtain reliable estimates. So, during our experiments after preprocessing the data, this method should be applied to our data set. Next, the machine learning engine should be implemented. Four different machine learning engines have been evaluated for our system, which three of them are in the category of neural network classifiers and the last one is a modern algorithm based on support vectors. In the following subsections, these engines are introduced in details.

A. BP Neural Network

In this paper, we tried to find the optimized BP network that can effectively detect and classify different DoS attacks against DNS. Our BP neural network has three layers. The number of the units in the input layer is equal to the features of input vector which are three features of DNS traffic. There are also three units in the output layer representing different states of normal and DoS attacks: [0 0 0] for normal conditions, [0 0 1] for direct DoS attack and [0 1 0] for the amplification attack. Our main assumptions considered for training process of BP networks are listed in Table I. The optimal structure of our network was found by varying the number of hidden neurons from 3 to 21. This range of values was specified based on some preliminary experiments.

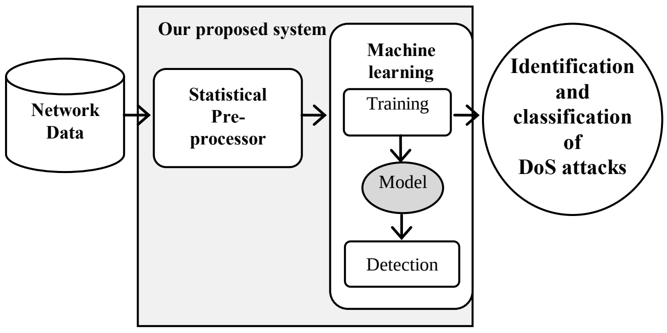
Network Data
Our proposed system
Statistical Pre-processor
Machine learning
Training
Model
Detection
Identification and classification of DoS attacks

Fig. 2. System Architecture

Table I. Assumptions in BP neural network

Parameters	Value
Number of epochs	500
MSE	0.00001
Training function	trainlm
Activation function	tan-sigmoid
Number of neurons in hidden layer	3-21
Number of neurons in output layer	3

Table II presents a comparison between different structures of BP neural network using performance metrics in section IV. The best performed structure for our BP neural network is the shaded column with 99% accuracy.

The results of BP network are illustrated in Fig. 3, Fig. 4, Fig. 5, and Fig 6. The x-axis values of the figures represent the number of neurons in the hidden layer, which are changing from 3 to 21. In Fig. 3, the y-axis values represent the mean squared errors that different structures of BP networks faced to, during the 500 epochs. The least training error and testing error were for the case of 7 neurons in the hidden layer, and as the number of neurons is increasing the MSE is also increasing.

In Fig. 4, the y-axis values represent the false alarm rate of our BP-based model while number of hidden neurons is increasing from 3 to 21. Suddenly, in the 7 neurons case we had an acceptable value near to zero for false alarm rate.

Detection rate of direct DoS attacks and amplification attacks is a criterion for classification rate of model. A good detection rate for each type of DoS attacks can show the classification power of our model. Fig. 5 presents the detection rate of direct and amplification DoS attacks against DNS while number of neurons in the hidden layer of BP network is increasing. A comparison between the different numbers of neurons shows the best detection rates (99.55% for direct DoS attacks and 97.82% for amplification attacks) in case where the number of neurons is set to 7. As this number is increasing, the detection rates of both cases are decreasing.

As IDSs require high detection rate, low false alarm rate and good accuracy; thus we compared the accuracy of different structures of BP network. Fig. 6 shows the accuracy results measured by changing the number of neurons in the hidden layer. As it can be seen, our BP network can achieve the best accuracy of 99% in the case of 7 neurons in hidden layer.

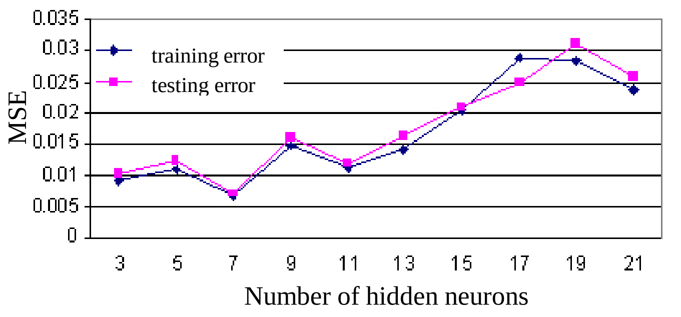

Fig. 3. Mean squared errors vs. different number of hidden neurons

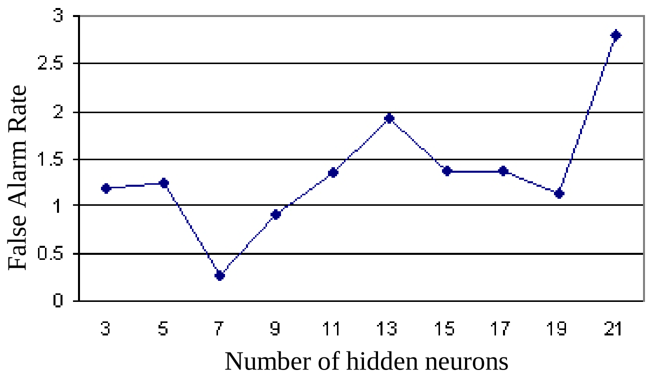

Fig. 4. False alarm rate vs. different number of hidden neurons

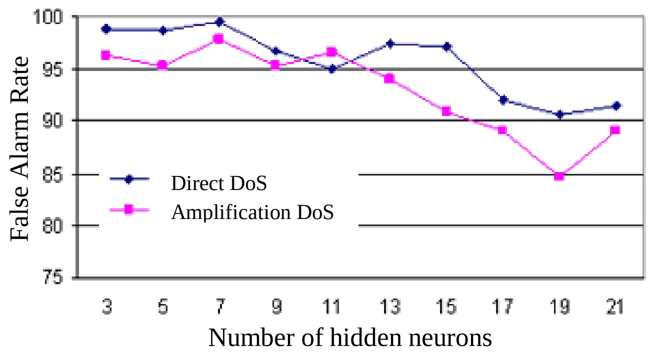

Fig. 5. Detection rate of different DoS attacks vs. different number of hidden neurons

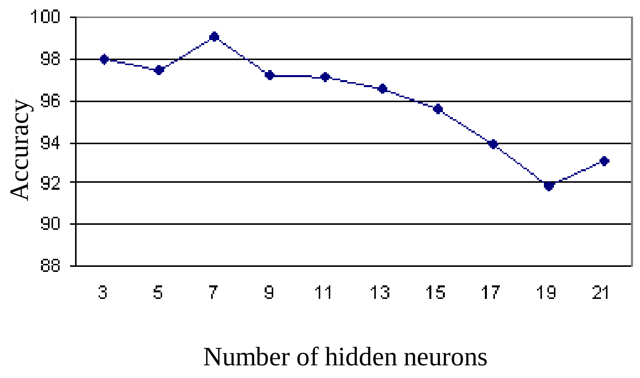

Fig. 6. Accuracy of model vs. different number of hidden neurons

As it has been presented in previous figures, we built models using various number of neurons and measured the performance parameters using 10-fold cross validation data used for training. This is a highly effective method but it is computationally expensive, because many models must be built. At first the number of neurons in the hidden layer is set to a small value and then they are increased to obtain the optimal network architecture related to convergence and generalization performance of the network. Generally, the computational complexity and convergence speed of network are affected by the number of neurons in the hidden layer since it acts poorly as model become larger and more complex. After tuning in the experiments, the optimal number of hidden neurons was found as 7. The MSE and false alarm rate of model are increasing while the number of hidden neurons is increasing. On the other hand, the detection rate and accuracy of model is decreasing while the number of hidden neurons is increasing.

B. RBF Neural Network

In order to implement an optimized RBF neural network for our classification problem, we need to specify the activation function for the hidden units and the centers and widths of RBFs. The mostly used activation function for the hidden layer is a Gaussian function which has been used for the hidden units in our RBF classifier.

Table II. Results of different structures of BP network

	Architecture									
Performance Parameters	3-3-3	3-5-3	3-7-3	3-9-3	3-11-3	3-13-3	3-15-3	3-17-3	3-19-3	3-21-3
Training time	15.7448	12.1422	11.0348	10.0457	13.0594	2.5455	16.5638	36.8279	22.9827	5.6821
Training error	0.0092	0.0109	0.0068	0.0149	0.0113	0.0141	0.0205	0.0287	0.0283	0.0238
Testing time	0.0294	0.0212	0.0221	0.0215	0.0218	0.0197	0.0500	0.0420	0.0425	0.0312
Testing error	0.0103	0.0123	0.0073	0.0161	0.0120	0.0164	0.0208	0.0248	0.0311	0.0259
False alarm rate	1.19	1.24	0.28	0.92	1.35	1.91	1.38	1.38	1.13	2.79
Detection rate (Direct DoS)	98.83	98.67	99.55	96.71	95.05	97.38	97.11	92.02	90.59	91.46
Detection rate (Amplification attack)	96.33	95.34	97.82	95.30	96.55	94.09	90.86	89.11	84.70	89.05
Accuracy	98.00	97.40	99.00	97.20	97.10	96.50	95.60	93.90	91.90	93.10

The centroid locations have been chosen by K-means clustering algorithm [12], and then the width parameter was calculated using the following equation:

$$\sigma = \textit{Maximum distance between any 2 centers/number of centers} \qquad (1)$$

A total of 10 experiments were performed using the RBF algorithm. Each simulation initials clusters using the K-means algorithm. The same input features and output states as BP neural network are used for implementing the RBF neural network in our detection system. Because of high calculation power requirements, it was not possible to achieve the same MSE as BP neural networks in our preliminary examinations. Therefore, we set the value of MSE to 0.001.

The results in Table III (average results of 10 experiments on RBF neural network) show that the overall detection rate and accuracy are high enough for classification and detection.

Table III. Results of RBF neural network

Parameters	**Values**
Training time	2.892
Training error	0.012
Testing time	0.082
Testing error	0.027
False alarm rate	0.23
Detection rate (Direct DoS)	99.62
Detection rate (Amplification)	89.48
Accuracy	95.90

A. SOM Neural Network

In this experiment, the input vector of three features has been normalized due to the large variations of input values. If the raw data is applied to the network directly, the input samples with higher values may lead to suppress the influence of smaller values. So, the standard normalization given by the following equation was used:

$$nv[i] = v[i] \div \sqrt{\sum_{x} v[x]^2} \qquad (2)$$

Different number of neurons was tested to find the best performed network. We obtained sample results by looking at the output of the classifier applied to the trained data and noticed that all normal traffic was clustered between a specified range and the suspicious traffic was outside this cluster indicating a possible attack. When we were confident about the results, the trained network was evaluated by subjecting it to the test data. Therefore, the main assumptions considered for implementing the SOM neural network were as follows: number of epochs = 1000, number of neurons = 25, neighbours topology = Hextop, distance function = Linkdist, ordering phase learning rate = 0.9, ordering phase steps = 1000, tuning phase learning rate = 0.02 and tuning phase neighbour distance = 1.

Table IV shows the average results of 10 experiments applied to training data using SOM network. The results show that we cannot classify types of attacks in the case of 3 and 6 numbers of neurons, because the network did not assign one or more specific neurons to each case in order to cluster the traffic.

Finally, because the SOM networks with 3 and 6 numbers of neurons could not satisfy us by their results, we tried to evaluate the trained network with 25 and 40 neurons by subjecting them to the test data. Table V illustrates the testing performance of SOM network for different number of neurons. It is clear that even the SOM neural network could not achieve a good performance in terms of detection rate and accuracy, but the case of choosing 25 neurons for our network gives us the best structure.

Table IV. Performance of SOM network for different number of neurons in training phase

Number of neurons / Parameters	3	6	25	40
Training time	3066.39	3070.04	3083.13	3065.97
False alarm rate	13.95	5.51	4.46	3.89
Detection rate (Direct DoS)	0	0	56.77	59.09
Detection rate (Amplification)	0	0	56.49	40.13
Accuracy	-	-	71.99	67.03

Table V. Performance of SOM network in testing phase

Number of neurons / Parameters	25	40
Testing time	0.03	0.03
False alarm rate	6.83	4.50
Detection rate (Direct DoS)	54.24	63.83
Detection rate (Amplification)	65.28	42.80
Accuracy	74.40	68.01

A. *Support Vector Machines*

SVM is another learning and soft computing technique that recently applied to IDSs. The two unique features of SVM are good generalization ability and capability to handle high dimensional data [13]. The basic SVM algorithm was designed for classification of objects into two classes [9], but many real world problems deal with more than two classes. For example, in our case, we want to classify normal traffic from direct DoS and amplification attacks. Therefore, we face to a three-class problem. In our experiments the one-against-all scheme is implemented to overcome this problem due to its lower complexity. It constructs three binary SVM classifiers, each of which separates one class from all the rest. The *i*th SVM is trained using a training set of positive labels (+1) for *i*th class and negative labels (-1) for all the others. Finally, a sample in our testing data is classified in class, *i*, which has the maximum value between all three classifiers.

During the training phase, a proper function with the corresponding parameters should be provided. This will be a time consuming process because the machine is trained with different kernel parameters and only the one which is the best performed will be selected for the testing process.

Support vector machines with three radial kernels with gamma = 1.5, 10, and 5, and the optimal regularization parameter C = 100, 1, and 1000000, were used for implementing three classifiers. The Radial basis kernel equation is as follows:

$$K(x, x') = \exp(-gamma \| x - x' \|^2) \qquad (3)$$

IV. Results and Discussions

In this section, the performance metrics used to evaluate our proposed system are introduced with their definitions:

- Accuracy, which refers to the proportion of data classified as accurate type in the total data. Accurate situations are True Positive (TP) and True Negative (TN), while false detected situations are False Positive (FP) and False Negative (FN). Accuracy of the system is calculated by the following formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \qquad (4)$$

- Detection rate (of direct DoS attacks), which refers to the proportion of direct DoS attacks detected among all direct DoS attacks.
- Detection rate (of amplification attacks), similarly refers to the proportion of amplification attacks detected among all amplification attacks. These two metrics are calculated by the following formula:

$$DetectionRate = \frac{TP}{TP + FN} \times 100\% \qquad (5)$$

- False Alarm Rate (FAR), which is defined as the percentage of the network traffic that is misclassified by the classifier. It can be calculated using the following formula:

$$FAR = \frac{FP}{FP + TN} \times 100\% \qquad (6)$$

Table VI presents the performance comparison of three neural network classifiers and SVM as well. The results show that a BP neural network outperforms other types of classifiers that have been implemented in this article. It gives us good detection rates for different types of DoS against DNS with an acceptable false alarm rate.

V. Conclusions

This paper has introduced two different types of DoS attacks against DNS which are direct DoS and amplification attacks. The investigation of the impact of DoS attacks against DNS traffic led us to find the suspicious behaviours.

Based on these patterns the required traffic data for analytical measurements was simulated using the most flexible network simulator, NS-2. Finally, a machine learning based model is proposed for detecting and classifying DoS attacks against DNS using several traffic statistics. Two different machine learning algorithms were evaluated for the detector engine which are neural network classifiers and support vector machines. For finding the optimal classifier, three of the best performed neural networks for detection and classification in intrusion detection systems were investigated. The performances of these classifiers were compared to another modern machine learning method which is SVM in terms of detection rate, accuracy, and false alarm rate. The comparison results show that a back propagation neural network outperforms other classifiers with 99.55% detection rate for direct DoS attacks, 97.82% detection rate for amplification attacks, 99% accuracy, and 0.28% false alarm rate.

Future work will study other types of security threats against DNS. Providing all patterns of traffic flows will help us to investigate the necessary features for detecting all forms of attacks against DNS. Another area of future work will be implementing the proposed model in a real environment that can investigate necessary improvement in our model such as the monitoring time.

Table VI. Performance comparison of different classifiers

Parameter / **Classifier**	**DR (direct DoS)**	**DR (amplification attack)**	**Accuracy**	**FAR**
BP	99.55	97.82	99	0.28
RBF	99.62	89.48	95.9	0.23
SOM	54.24	65.28	74.40	6.83
SVM	98.26	97	97.6	1.07

REFERENCES

[1] D. Davidowicz. (1999). Domain Name System (DNS) Security. Available: http://compsec101.antibozo.net/papers/dnssec/dnssec.html. Accessed, Feb. 2009.

[2] N. Chatzis, "Motivation for behaviour-based DNS security: A taxonomy of DNS-related internet threats," in The International Conference on Emerging Security Information, Systems, and Technologies (SecureWare 2007), Valencia, Spain, Oct., 2007, IEEE Computer Society Press, Washington DC, USA, pp. 36-41.

[3] Y. Wang, M. Hu, B. Li and B. Yan, "Tracking anomalous behaviors of name servers by mining DNS traffic," Lecture Notes In Computer Science, vol. 4331, pp. 351-357, 2006.

[4] R. Vaughn and G. Evron. (Mac. 2006). DNS Amplification Attacks (Preliminary release). Available: http://www.isotf.org/news/DNS-Amplification-Attacks.pdf. Accessed, Nov. 2008.

[5] ICANN. Factsheet root server attack on 6 February 2007. I. C. for Assigned Names and Numbers, pp. 1-6, Mac. 2007.

[6] G. Kambourakis, T. Moschos, D. Geneiatakis and S. Gritzalis, "A fair solution to DNS amplification attacks," in Second International Workshop on Digital Forensics and Incident Analysis (WDFIA 2007), Samos, Greece, IEEE Computer Society, Washington, DC, USA, Aug. 2007, pp. 38-47.

[7] Samaneh Rastegari, M Iqbal Saripan and Mohd Fadlee A. Rasid, Detection of Denial of Service Attacks against Domain Name System Using Machine Learning Classifiers, Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering 2010, WCE 2010, 30 June - 2 July, 2010, London, U.K., pp 444-447.

[8] Samaneh Rastegari, M Iqbal Saripan and Mohd Fadlee A. Rasid, Detection of Denial of Service Attacks against Domain Name System Using Neural Networks, International Journal of Computer Science Issues, Volume 6, pp. 23-27, November 2009.

[9] J. Mirkovic, A. Hussain, B. Wilson, S. Fahmy, P. Reiher, R. Thomas, W. M. Yao and S. Schwab, "Towards user-centric metrics for denial-of-service measurement," inWorkshop on Experimental Computer Science (Part of ACM FCRC), San Diego, June, 2007, ACM, pp. 1-14.

[10] K. Lan, A. Hussain and D. Dutta, "Effect of malicious traffic on the network," in Passive and Active Measurement Workshop (PAM), San Diego, CA, Apr., 2003.

[11] J. Han and M. Kamber, Data mining: concepts and techniques. Morgan Kaufmann, San Fransisco, CA, USA, 2006.

[12] S. Haykin, Neural Networks: A Comprehensive Foundation Second Edition (M). Prentice Hall, 2001.

[13] J.T. Yao, S.L. Zhao, and L.V. Saxton, "A study on fuzzy intrusion detection," in Proc. SPIE, San Jose, CA, USA, Feb., 2005, SPIE, 23-30.