Journal of Scientific Research and Development 2 (5): 316-319, 2015
Available online at www.jsrad.org
ISSN 1115-7569
© 2015 JSRAD

Evaluating Artificial Neural Network in IDS Alert Management System

Fatemeh Charlank Bakhtiari[1], Mir-Kamal Mirnia[2,*]

[1]*Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Urmia, Iran*

[2]*Department of Applied Mathematics, Tabriz University, Tabriz, Iran*

Abstract: Intrusion Detection Systems (IDS) are designed to protect systems used by organizations against threats. The main drawbacks of IDS are the number of alerts generated and failing. By using Artificial Neural Networks (ANN), a system is proposed to be able to classify IDS alerts and to reduce false positives alerts. The experimental results on DARPA KDD cup 98 show that the system can cluster and classify alerts and causes reducing false positive alerts considerably. Also the proposed system could improve the accuracy of identification rate per attack type.

Key words: IDS, Alert management, Artificial Neural Network, false positive alert reduction

1. Introduction

An Intrusion Detection System (IDS) is a hardware device or software program that analyzes computer system activities and/or network traffics to detect malicious activities and produces alerts to security experts (Debar H. et al., 1999). These systems are known to generate many alerts. Analyzing these alerts manually by security expert are time consuming, tedious and error prone (Debar H. et al., 2001). From another point of view false positive alerts have huge amount of generated alerts. Identifying attack types and generating correct alerts related to attacks are other problems with IDS. In order to overcome mentioned problem, alert management systems was introduced. Alert management systems help security experts to manage alerts and produce a high level view of alerts.

In this paper by using Artificial Neural Network (ANN) (Gurney K., 2009), an alert management system is proposed which classifies alerts in real time and detects false positive alerts. To improve accuracy of the results, the proposed system uses some techniques such as alert filtering and alert preprocessing.

The alert management system is introduced in Section 1. Section 2 reviews related works, section 3 explains the suggested alert management system. The experimental results are shown in section 4 and finally section 5 is conclusion and future works.

2. Related Works

Clustering of alerts is an example of alert management techniques. In (K. Julisch, 2003) a clustering method is introduced based on discovering root cause of false positive alerts. The results in (K. Julisch, 2003) show that a small number of root causes implies 90% of alerts. By removing these root causes, the total number of alerts comes down to 18%. One of the main problems of this technique is that it depends on under laying network structure.

In (Maheyzah, M. S. et al., 2009) some heuristic and neural network based techniques are used to cluster alerts. Another clustering technique is used in Mirador project with expert systems by Cuppens, the similarity between two alerts is calculated by expert system (F. Cuppens., 2001; E. MIRADOR, 2000). The results of two genetic clustering algorithms, named Genetic Algorithm (GA) and Immune based Genetic Algorithm (IGA) are compared in (Wang, J. et al., 2010; Wang J. et al., 2009).

Wespi and Debar (Debar H. et al., 2001) design an algorithm that places alerts in situations which a set of special alerts created with source, destination and attack class attributes. In (Krovi R., 1992) an algorithm is introduced that create hyper alert from existing alerts. A hyper alert is an aggregation of related alerts. An alert management system is introduced in (Amir Azimi Alasti Ahrabi et al., 2010) using Self-Organizing Maps (SOM) to cluster and classify IDS alerts. In (Amir Azimi Alasti Ahrabi et al., 2010) several operations and methods such as alert filtering, alert preprocessing and cluster merging are introduced.

In (Bahrbegi H. et al., 2010) an alert management system similar to (Amir Azimi Alasti Ahrabi et al., 2010) is presented in which the performance of seven genetic clustering algorithms named Genetic Algorithm (GA) (Krovi R., 1992), Genetic K-means Algorithm (GKA) (Krishna K. et al., 1999), Improved Genetic Algorithm (IGA) (Fuyan L. et al., 2005), Fast Genetic K-means Algorithm (FGKA) (Lu Y. et al., 2004), Genetic Fuzzy C-means Algorithm (GFCMA), Genetic Possibilistic C-Means Algorithm (GPCMA) (Bahrbegi H. et al., 2010) and Genetic Fuzzy Possibilistic C-Means Algorithm (GFPCMA) (Bahrbegi H. et al., 2010) are used to cluster and classify true positive and false positive alerts. After clustering alerts the system prioritize produced clusters by using a Fuzzy Inference System (Bahrbegi H. et al., 2010).

In this paper an alert management system similar to system (Amir Azimi Alasti Ahrabi et al., 2010) are designed that uses ANN to classify generated alerts. The system will be able to improve the accuracy of results, identifying attack type of alerts accurately and also reducing the number of false positive alerts considerably.

3. Proposed Alert Management System Based on ANN

Fig. 1 shows the proposed system. In this paper we use binary traffics files of a network, DARPA 98 dataset (MIT Lincoln Lab.) instead of real network traffics. Snort tool (Snort) is used to produce alerts of DARPA 98 dataset network traffics. Snort is an open source signature based IDS which gets DARPA 98 online traffic and then generates alert log files (Amir Azimi Alasti Ahrabi et al., 2010). After generating alert log files, these files are entered into the proposed system as inputs.

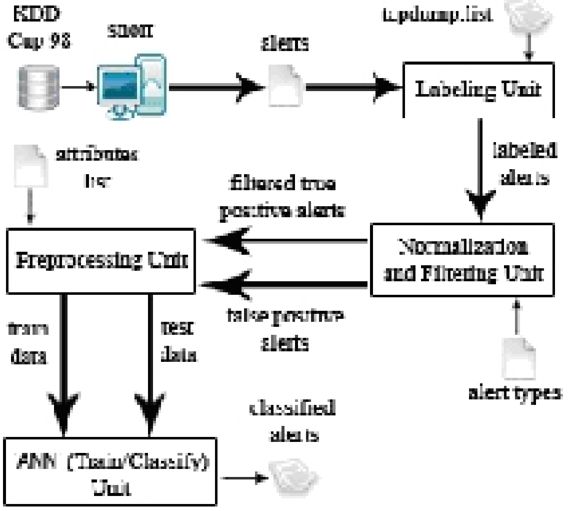

Fig. 1: Proposed alert management system

3.1. Labeling unit

Labeling unit gets the generated alert from Snort and also tcpdump. List files of DARPA 98 dataset and then generates labeled alerts. A labeled alert is an alert with its own attack type. The tcpdump.list files contain information about all packets in DARPA 98 dataset. These labels are used to train ANN and evaluate results of ANN (Amir Azimi Alasti Ahrabi et al., 2010; Bahrbegi H. et al., 2010).

3.2. Normalization and filtering unit

In this phase accepted attack types are entered to the unit and only the alerts that are in class of predefined attack types are selected (Amir Azimi Alasti Ahrabi et al., 2010; Bahrbegi H. et al., 2010; S Terry Brugger and Jedidiah Chow, 2007). This unit uses eight attributes of alert to filter, which are: Signature ID, Signature Rev, Source IP, Destination IP, Source Port, Destination Port, Datagram length and Protocol (Snort Manual).

3.3. Preprocessing unit

Preprocessing unit converts string values of attributes of alert to numerical data. It also reduces the range of attribute values and converts alerts into data vectors (1), (2) and (3).

$$IP = X_1.X_2.X_3.X_4, \qquad (1)$$
$$IP\_VAL = (((X_1 \times 255) + X_2) \times 255 + X_3) \times 255 + X_4$$

$$protocol\_val = \begin{cases} 0, protocol = None \\ 4, protocol = ICMP \\ 10, protocol = TCP \\ 17, protocol = UDP \end{cases} \qquad (2)$$

$$IUR = 0.8 \times \frac{X - X_{\min}}{X_{\max} - X_{\min}} + 0.1 \qquad (3)$$

3.4. Ann training and classification unit

In this unit we use ANN as a classifier. ANN should be trained with train dataset and then gets the test dataset to classify them.

An artificial neural network (ANN) is a mathematical model that is inspired by the structure and/or functional aspects of biological neural networks. A neural network is made up of a number of simple, highly interconnected processing elements called neurons, which mimics the biological neural network in order to process information by their dynamic state response to external inputs. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. Modern neural networks are non-linear statistical data modeling tools. They are usually used to model complex relationships between inputs and outputs or to find patterns in data (Gurney K., 2009).

ANN has three major learning algorithms named: Supervised Learning, Unsupervised Learning and Reinforcement Learning. In this paper supervised learning approach is used. In supervised learning, a set of examples are entered into system and then network updates neurons weights according to error values calculated. After an ANN being trained, the test dataset should be entered into the ANN to classify dat+a vector in dataset.

4. Results and discussion

To simulate the proposed system C#.net programming language, MATLAB software and artificial neural network toolbox is used (Neural

Network Toolbox, 2011; Matlab Software). The simulation parameters are shown below.

Suggested ANN is a backpropagation network and it has 3 layers of which first layer consists of 8 input neurons, hidden layer consists of 120 neurons and last layer has 1 output neuron. The ANN gets a data vector of train data and each data vector consists of 8 attributes. Training phase consists of 50 epochs. Transfer function for hidden layer neurons is hyperbolic tangent sigmoid transfer function and for output neurons is linear transfer function. Backpropagation network training function is Levenberg-Marquardt backpropagation. Train data contains 70% of total filtered alert data vectors or 10166 data vectors. The false positive count in the training dataset is 4113. Test dataset includes 30% of the data vectors of labeled alerts; it means 2591 data vectors of true positive, and 1764 data vectors of false positive alerts.

Figure II shows Mean Square Error (MSE) for each epoch. As it can be seen in this figure, the error value is reduced when we moved forward in epoch axis; and minimum value of the error achieved in last step. In figure III, the regression graph of ANN output values are shown. In this figure we can see the output error for each attack type (number 201 – 209) is very low and regression of these data vectors is 0.9998.

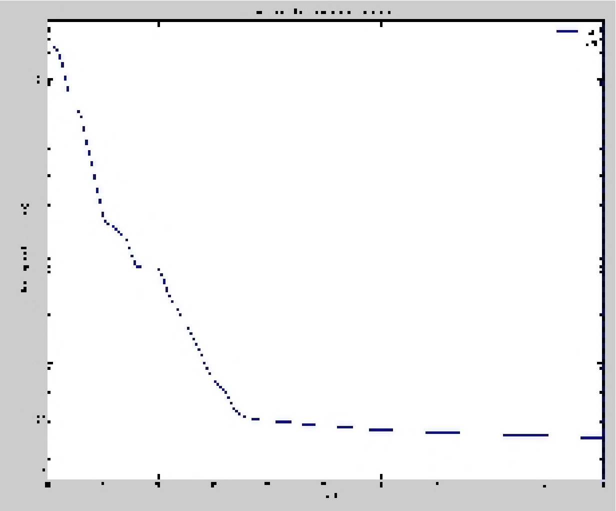

FIG. 2: MEAN SQUARE ERROR PER EPOCH

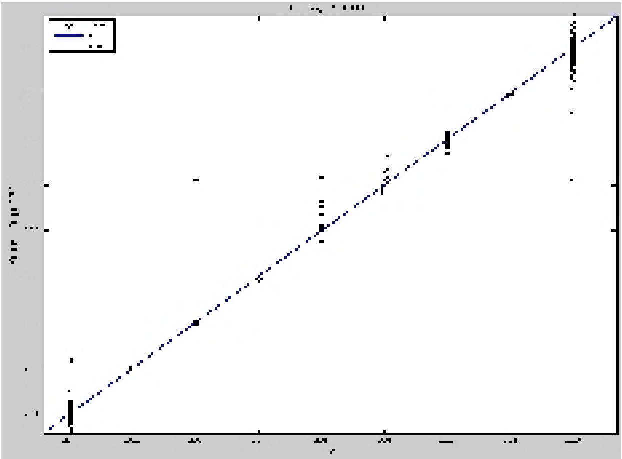

FIG. 3: ERRORS OF ANN OUTPUT VALUES PER TARGETS

Table 1: Extraxted Performance Values from This Simulation

AACT	ClaAR	ClaE	TestER	TrainER
0.000025	99.66	15	0.036	1.09×10^{-4}

To evaluate the performance of the algorithms, five measurements are introduced, as follows:

1- Train phase Error Rate (TrainER),
2- Test phase Error Rate (TestER),
3- Classification Error (ClaE),
4- Classification Accuracy percent (ClaAR),
5- Average Alert Classification Time (AACT).

In table I, the value of the metrics are shown. The value of TrainER metric shows the proposed neural network and system trained very well. The value of TestER metric is 0.036. This metric guarantee the accuracy of the results. The values of ClaE and ClaAR are 15 and 99.66 respectively that directly depend on classification error rate (Table I). It means that low rate of error in train and test phases are resulted to produce more accurate in classification of alerts. The value of AACT measurement is 0.000025 showing the proposed system can be used in real time IDS alert management systems. It means the proposed system can evaluate alerts along with production of alerts by IDS concurrently. The results of accuracy per alerts attack types are shown in Table II. The proposed system is able to identify all of the attack types of each alert with high rate of accuracy. False positive alert type identification known False Positive Reduction is an important factor of extracted values. The value of this metric is 99.72 percent.

Proposed system reaches 100 percent for LAND, DICT and NMAP attack types. Values 99.92, 97.96, 33.34, 42.86 and 33.34 extracted for attack types BACK, POD, PHF, ROOTKIT and IMAP respectively. According to the results obtained, the proposed system could identify all attack type of alerts.

Table 2: Proposed System Accuracy Percentage for Each Attack Type of Alerts

FALSE POSITIVE	NMAP	DICT	IMAP	ROOTKIT	PHF	POD	LAND	BACK
99.72	100	100	33.34	42.86	33.34	97.96	100	99.92

5. Conclusion and Future works

In this paper, we present a new approach for classifying IDS alerts and to reduce false positives alerts based on ANN. The system is able to cluster and classify alerts and causes reducing false positive alerts considerably. Results show that the output error for each attack is very low and regression of these data vectors is 0.9998. Also, experimental results show the proposed system can cluster and classify alerts and causes reducing false positive alerts considerably and also the proposed system could improve the accuracy of identification rate per attack type. Proposed system reaches 100 percent for LAND, DICT and NMAP attack types. According to the results obtained, the proposed system could identify all attack type of alerts. Putting other modules in this system that they are able to priorities the classified alerts left as a future work.

References

Amir Azimi Alasti Ahrabi, Ahmad Habibizad Navin, Hadi Bahrbegi, Mir Kamal Mirnia, Mehdi Bahrbegi, Elnaz Safarzadeh, Ali Ebrahimi, "A New System for Clustering and Classification of Intrusion Detection System Alerts Using Self-Organizing Maps", International Journal of Computer Science and Security (IJCSS), Vol. 4, Iss. 6, pp. 589 – 597, 2010.

Bahrbegi H., Navin A.H., Ahrabi A.A.A., Mirnia M. K., Mollanejad A., "A new system to evaluate GA-based clustering algorithms in Intrusion Detection alert management system", Nature and Biologically Inspired Computing (NaBIC), Second World Congress on, pp. 115–120, 2010.

Debar H., and A. Wespi, "Aggregation and correlation of intrusion detection alerts", Proc. of the 4th Int. Symp. on Recent Advances in Intrusion Detection, pp. 87–105, 2001.

Debar H., M. Dacier, and A. Wespi, "Towards a taxonomy of intrusion-detection systems", COMPUT. NETWORKS, Vol. 31, Iss. 8, pp. 805-822, 1999.

Debar H., Wespi A., "Aggregation and Correlation of Intrusion-Detection Alerts", Proceeding RAID ’00 Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection, pp. 87-105, 2001.

E. MIRADOR. "Mirador: a cooperative approach of IDS", European Symposium on Research in Computer Security (ESORICS). Toulouse, France, 2000.

F. Cuppens., "Managing alerts in a multi-intrusion detection environment", Proceedings of the 17th Annual Computer Security Applications Conference on, pp. 22-31, 2001.

Fuyan L., Chouyong C., Shaoyi L., "An Improved Genetic Approach", International Conference on Neural Networks and Brain, pp. 641-644, 2005.

Gurney K., "An introduction to neural networks", CRC Press, 2009.

K. Julisch, "Clustering intrusion detection alarms to support root cause analysis", ACM Trans. on Information and System Security, Vol. 6, Iss. 4, pp. 443 – 471, 2003.

Krishna K., Murty M., "Genetic K-means algorithm", IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics, pp. 433-439, 1999.

Krovi R., "Genetic Algorithm for Clustering: A preliminary investigation", Proceeding on 25th Hawaii International Conference on Systems Sciences (HICSS), pp. 540–544, 1992.

Lu Y., Lu S., Fotouhi F., Deng Y., Brown J. S., "FGKA: a Fast Genetic K-means Clustering Algorithm", Proceeding of the ACM Symposium on Applied computing (SAC), Nicosia, Cyprus, pp. 622-623, 2004.

Maheyzah, M. S., Mohd Aizaini, M., and Siti Zaiton, M. H., "Intelligent Alert Clustering Model for Network Intrusion Analysis", Int. Jurnal in Advances Soft Computing and Its Applications (IJASCA), Vol. 1, Iss. 1, pp. 33–48, 2009.

Matlab Software, http://www.mathworks.com.

MIT Lincoln Lab., DARPA 1998, "Intrusion Detection Evaluation Datasets. Available: http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/1998data.html", 1998.

Neural Network Toolbox, "ANN Toolbox for MATLAB", http://www.mathworks.com/products/neural-network, 2011.

Nuovo A. D. G., Catania V., Palesi M., "The Hybrid Genetic Fuzzy C-means: a Reasoned Implementation", Proceedings of the 7th WSEAS International Conference on Fuzzy Systems, ACM, pp. 33-38, 2006.

S Terry Brugger and Jedidiah Chow, "An Assessment of the DARPA IDS Evaluation Dataset Using Snort", UC Davis Technical Report CSE-2007-1, Davis, CA, 2007.

Snort Manual, http://www.snort.org/assets/82/snort_manual.pdf.

Snort: The open source network intrusion detection system. Available: http://www.snort.org.

Wang J., Baojiang Cui, "Clustering IDS Alarms with an IGA-based Approach", ICCCAS, pp. 586-591, 2009.

Wang, J., Wang, H., Zhao, G. "A GA-based Solution to an NP-hard Problem of Clustering" Security Events. IEEE, pp. 2093- 2097, 2010.