Efficient Security Alert Management System

Minoo Deljavan Anvary *Minoo.deljavan@yahoo.com*
IT Department School of e-Learning
Shiraz University
Shiraz, Fars, Iran

Majid Ghonji Feshki *Ghonji.majid@yahoo.com*
Department of Computer Science
Qzvin Branch, Islamic Azad University
Qazvin, Qazvin, Iran

Amir Azimi Alasti Ahrabi *Amir.azimi.alasti@gmail.com*
Department of Computer Science
Shabestar Branch, Islamic Azad University
Shabestar, East Azerbaijan, Iran

Abstract

Nowadays there are several security tools that used to protect computer systems, computer networks, smart devices and etc. against attackers. Intrusion detection system is one of tools used to detect attacks. Intrusion Detection Systems produces large amount of alerts, security experts could not investigate important alerts, also many of that alerts are incorrect or false positives. Alert management systems are set of approaches that used to solve this problem. In this paper a new alert management system is presented. It uses K-nearest neighbor as a core component of the system that classify generated alerts. The suggested system serves precise results against huge amount of generated alerts. Because of low classification time per each alert, the system also could be used in online systems.

Keywords: Intrusion Detection, Security Alert Management, K-nearest Neighbor, Real-time Security Alert Classification, Reduction of False Positive Alerts, Precise Classifying True Positive Alerts.

1. INTRODUCTION

An intrusion detection system (IDS) inspects all inbound and outbound network activity and identifies suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise a system [1]. There are several ways to categorize an IDS such as misuse detection vs. anomaly detection, network-based vs. host-based systems and passive system vs. reactive system. In misuse detection, the IDS analyzes the information it gathers and compares it to large databases of attack signatures. Essentially, the IDS looks for a specific attack that has already been documented. Like a virus detection system, misuse detection software is only as good as the database of attack signatures that it uses to compare packets against. In anomaly detection, the system administrator defines the baseline, or normal, state of the networks traffic load, breakdown, protocol, and typical packet size. The anomaly detector monitors network segments to compare their state to the normal baseline and look for anomalies. In a network-based system, or NIDS, the individual packets flowing through a network are analyzed. The NIDS can detect malicious packets that are designed to be overlooked by a firewalls simplistic filtering rules. In a host-based system, the IDS examines at the activity on each individual computer or host. In a passive system, the IDS detects a potential security breach, logs the information and signals an alert. In a reactive system, the IDS responds to the suspicious activity by logging off a user or by reprogramming the firewall to block network traffic from the suspected malicious source. These systems are known to generate many alerts. Analyzing these

alerts manually by security expert are time consuming, tedious and error prone. From another point of view false positive alerts have huge share of generated alerts. Identifying attack types and generating correct alerts related to attacks is another problem with IDS. In order to overcome mentioned problems 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 authors design new algorithm based on framework introduced in [2] that uses K-nearest neighbor algorithm (KNN) [3] as a core component. It classifies the generated alerts based on attack type of alerts, detects false positive alerts, high speed classification to use with alert generation in IDSs. The proposed system uses some techniques same as previous work techniques [3] such as alert filtering, alert preprocessing, and alert filtering to improve accuracy of the results.

This paper is categorized in to 5 sections. In Section 1 the alert management techniques are mentioned. Related alert management techniques are investigated in section 2, section 3 explains the suggested alert management system and describes all component of the proposed system, the experimental results are shown in section 4 and finally section 5 is a conclusion and future works.

2. RELATED WORKS

There are several techniques that used to manage security alerts one of them is clustering and classification of alerts. In [4, 5] clustering algorithms based on genetic algorithm, named Genetic Algorithm (GA) and Immune based Genetic Algorithm used to manage IDS alerts. In [6] author was introduce clustering algorithm based on root causes which finds main cause of alerts and join them together to construct clusters. He shows that by deleting these root causes consequent alerts reduced to 82%. This method is very good but it depends on undelaying network structure and the approach should be change when the structure of network is changed.

In [7] DARPA 2000 dataset [8] is evaluated by three algorithms. The proposed algorithms used alerts without any preprocessing techniques. In another work expert system used to make decision [9, 10]. Debar et al. [11] designed a system by placing them in situations aggregates alerts together. Situations are set of special alerts. Some attributes of alert are used to construct a situation.

A new alert management system is introduced in [2]. For evaluation of the system Azimi et. al. used generated alerts from DARPA 98 dataset [12]. The main unit of the system is cluster/classify unit that uses Self-Organizing Maps (SOM) [13] to cluster and classify IDS alerts. In another work, an alert management system is introduced [14] that similar to [2]. In that work usage of seven genetic clustering algorithms is evaluated. In [15] Learning Vector Quantization (LVQ) [16] is used as a classification engine. The accuracy of the suggested approach is acceptable [15].

In this paper an alert management system based on system proposed by authors in [2] is proposed that uses KNN as a tool to classify input alert vectors. The system will be able to improve accuracy of results and also to reduce the number of false positive alerts.

3. EFFICIENT SECURITY ALERT MANAGEMENT SYSTEM

New alert management system is described in this section. In this paper Snort [17] IDS is used to generate alerts from DARPA 98 dataset [12]. Figure 1 shows the proposed system. Snort is a free and open source network intrusion prevention system and network intrusion detection system created by Martin Roesch in 1998. Snort can reads binary dumped network traffic files named tcpdump of DARPA 98 dataset and analyze them. After producing alerts with snort; they are labeled, normalized, filtered, preprocessed and classified respectively. Each steps of the system is described below.

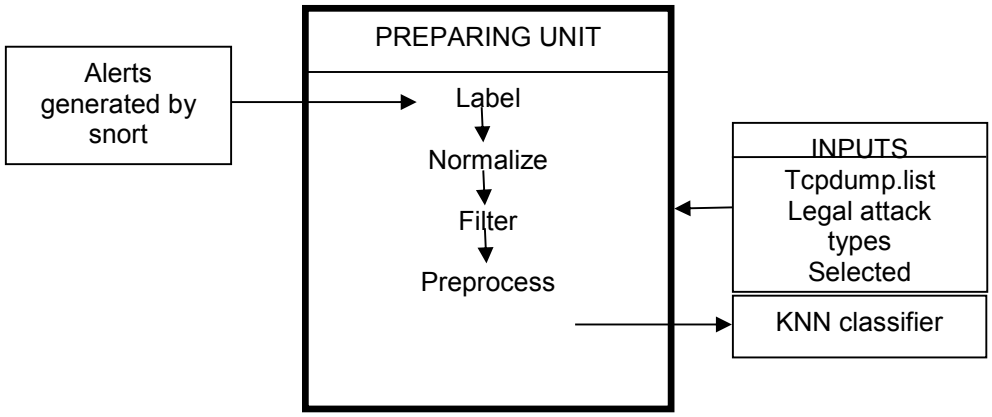

FIGURE 1: New Alert Management System.

As you see in the figure 1 preparing unit has some inputs that provide basic information about traffic and legal operations. Labeling operation according to tcpdum.list files label generated security alerts. It means that this unit appends attack type of each alert as an attribute of processed alert to proper alert. These labels are used to train and to test results of the system. Figure 2 shows the labeling algorithm.

In: TCPDUMP list files and alert log files.
Out: Labeled alerts
1. Create an empty *AttackList* and *AlertList* set.
2. For each row in TCPDUMP list files:
   2.1. If the row is a labeled attack then add the row to the *AttackList* set.
3. For each row in alert log files:
   3.1. Create key with the five attributes: *source ip, destination ip, source port, destination port, ICMP code/type.*
   3.2. If the key exists in *AttackList* set then label the selected row with the type of found attack from *AttackList* set.
   Else
   Label the selected row with the False Positive attack type.
   3.3. Add the selected row to the *AlertList* set.
4. Return the *AlertList* set.

FIGURE 2: Alert Labeling Algorithm [2, 14].

After labeling snort alerts they should be normalized. It means that alerts with unexpected attack types should be removed. According to [18] snort is unable to detect all attacks in DARPA 98 dataset then we use “alert types” file to specify accepted attack types in this investigation [2, 14]. In this paper accepted alerts attack types are: BACK, LAND, POD, PHF, ROOTKIT, NMAP, IMAP, and DICT. After normalizing, redundant alerts are removed and only one instance of them are remained in the final dataset.

Preprocessing operation convert string data types such as protocols and IP values to numerical one, and also transform their values to unit range. The formula to convert IP values and protocol values are in equation (1) and (2) respectively. There are two methods to reduce value range attributes named Unit Range (UR) and Improved Unit Range (IUR). As described in [2] the accuracy of IUR is better than UR method, so in this paper IUR method is used.

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

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

$$UR = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (3)$$

$$IUR = 0.8 \times \frac{x - x_{\min}}{x_{\max} - x_{\min}} + 0.1 \quad (4)$$

In this unit KNN algorithm is used as a classifier. KNN is one of famous and traditional classification algorithms [3]. It widely used in pattern recognition. It determines class type of input vectors according to its neighbors. So some data should be entered as a training data set and then test data set is entered to classification purpose. KNN has two steps. In first step when an input vector entered to system for classification, k nearest data vectors to input vector from training data set is selected. In the second step class of input vector according to selected data vectors from previous step is calculated. In this paper we use Euclidean Distance as a distance metric [3].

4. EXPERIMENTAL RESULTS

To simulate the new system C#.net programming language, MATLAB software and SPRTool toolbox is used [19, 20]. The parameters of simulation are shown below.

In the investigation the number of nearest neighbor (K) is in range of [1 100]. The attack types used in this simulation are: BACK, POD, IMAP, NMAP, DICT, ROOTKIT, PHF and LAND. Total number of alert vectors is 14279 that splits into two sets named training and testing datasets that contains 70%, 30% of totals alert data vectors respectively. About 30% of alerts in each dataset is false positives.

The result of the proposed system is evaluated by four measurement named Classification Error (ClaE), Classification Accuracy percent (ClaAR), Average Alert Classification Time (AACT) and False Positive Reduction Rate (FPRR).

As you can see in table 1 the classification error rate reduces by reducing the value of K. When K is 1 the best result is achieved. It means that one neighbor can describe one point well.

In table 1 value of these metrics are shown. The best values of ClaE and ClaAR are 8 and 99.82% respectively. The value of AACT measurement is 0.006623 that shows the proposed system can be used in active IDS alert management systems that evaluate alerts beside alert production by IDS concurrently.

Table 2 shows the results of accuracy of proposed system in identifying attack type of each alert vector in test phase. As it can be seen in table 2, the proposed system can identify all of attack types of alerts with high rate of accuracy when k is 1. Table 2 shows that when value of K is decreased the accuracy of the metrics is increased.

An important point is accuracy percent of false positive identification. That is the proposed system can reduce false positive alerts with 99.94 percent. Which shows to be a solution of an important problem of IDSs. Proposed alert management system reaches 100 percent for BACK, LAND, POD, DICT and NMAP attack types. For attack types PHF, IMAP and ROOTKIT accuracy percent values are 66.67, 66.67 and 28.57 respectively.

Table 3 shows the results of approach [14]. It is obvious that the proposed framework gets better accuracy than genetic clustering based approach. Also the result of the system is better than its

base framework based on SOM in [2]. Table 4 shows the result of SOM based framework. One of the benefits of proposed system is capability of working in real-time mode.

K	ClaE	ClaAR	AACT
100	119	97.27	0.014054
50	83	98.09	0.007908
25	65	98.51	0.007699
20	61	98.60	0.007911
15	49	98.87	0.007100
10	46	98.94	0.006864
5	22	99.49	0.006135
1	8	99.82	0.006623

TABLE 1: Extracted performance metric values from simulation.

Back	Land	Pod	Phf	Rootkit	Imap	Dict	Nmap	False Positive(FPRR)	K
99.69	0	89.80	0	0	0	100	85.68	98.47	100
99.69	0	93.88	0	0	0	100	89.15	99.49	50
99.84	100	97.96	0	0	0	100	90.89	99.60	25
99.84	100	97.96	0	0	0	100	91.76	99.60	20
99.84	100	97.96	0	0	0	100	94.14	99.69	15
99.84	100	97.96	0	0	0	100	94.36	99.77	10
99.92	100	97.96	66.67	12.29	33.34	100	98.26	99.83	5
100	100	100	66.67	28.57	66.67	100	100	99.94	1

TABLE 2: Proposed system accuracy percent for each attack type of alerts.

Bahrbegi et. al. in [14] proposed a framework that uses genetic algorithm families to clustering and classification propose. As two works are similar we have to compare our results with their work. These results are shown in table 3. For all metrics the proposed system has high value in contrast of all GA based techniques. As shown in table 3, these algorithms could not be able to work actively because of the execution times are high. Although the proposed method reaches high accuracy results per alert attack types.

Algorithm	ClaE	ClaAR	FPRR	AACT
GA	1218	72.03	52.15	Offline
FGKA	314	92.79	97.51	Offline
GKA	1011	75.2	62.11	Offline
IGA	306	92.97	95.24	Offline
GFCMA	148	96.60	97.51	Offline
GPCMA	91	97.91	96.03	Offline
GFPCMA	148	96.60	97.51	Offline

TABLE 3: Results of GA-Based Algorithms [14].

ClaE	ClaAR	FPRR	AACT
33	99.36	99.71	0.003

TABLE 4: Results SOM based Algorithms [2].

5. CONCLUSION AND FUTURE WORKS

In this paper a fast and accurate algorithm is proposed that is used KNN algorithm as its classification engine manage IDS alerts. The results show that accuracy and false positive reduction rate of the system is high. Also the system is able to identify the attack types of the

alerts more accurate in real-time manner. Using other artificial intelligence techniques such as evolutionary algorithms and decision trees to improve the accuracy of the system are future works of this paper.

6. REFERENCES

[1] Debar, H., M. Dacier, and A. Wespi, *Towards a taxonomy of intrusion-detection systems.* Computer Networks, 1999. **31**(8): p. 805-822.

[2] Ahrabi, A.A.A., et al., *A New System for Clustering and Classification of Intrusion Detection System Alerts Using Self-Organizing Maps.* International Journal of Computer Science and Security (IJCSS), 2011. **4**(6): p. 589.

[3] Cover, T. and P. Hart, *Nearest neighbor pattern classification.* Information Theory, IEEE Transactions on, 1967. **13**(1): p. 21-27.

[4] Wang, J., H. Wang, and G. Zhao. *A GA-based Solution to an NP-hard Problem of Clustering Security Events*. 2006. IEEE.

[5] Wang, J. and B. Cui. *Clustering IDS Alarms with an IGA-based Approach*. 2009. IEEE.

[6] Julisch, K., *Clustering intrusion detection alarms to support root cause analysis.* ACM Transactions on Information and System Security (TISSEC), 2003. **6**(4): p. 443-471.

[7] Maheyzah, S.Z., *Intelligent alert clustering model for network intrusion analysis.* Journal in Advances Soft Computing and Its Applications (IJSCA), 2009. **1**(1): p. 33-48.

[8] *DARPA 2000 Intrusion Detection Evaluation Datasets*, M.L. Lab., Editor. 2000.

[9] Cuppens, F. *Managing alerts in a multi-intrusion detection environment*. 2001.

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

[11] Debar, H. and A. Wespi. *Aggregation and Correlation of Intrusion-Detection Alerts*. in *Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection*. 2001.

[12] *DARPA 1998 Intrusion Detection Evaluation Datasets*, M.L. Lab., Editor. 1998.

[13] Kohonen, T., *Self-Organized Maps*. 1997, Science Berlin Heidelberg: Springer series in information.

[14] Bahrbegi, H., et al. *A new system to evaluate GA-based clustering algorithms in Intrusion Detection alert management system*. 2010. IEEE.

[15] Ahrabi, A.A.A., et al., *Using Learning Vector Quantization in IDS Alert Management System.* International Journal of Computer Science and Security (IJCSS), 2012. **6**(2): p. 1-7.

[16] Kohonen, T., *Learning vector quantization, in M.A. Arbib (ed.), The Handbook of Brain Theory and Beural Networks*. 1995: MIT Press.

[17] Snort, *The open source network intrusion detection system*. 2012.

[18] Brugger, S.T. and J. Chow, *An Assessment of the DARPA IDS Evaluation Dataset Using Snort*, D. UC Davis Technical Report CSE-2007-1, CA, Editor. 2007.

[19] Franc, V. and V. Hlavác. *Statistical pattern recognition toolbox for Matlab*. Center for Machine Perception, Czech Technical University 2004; Available.

[20] Mathworks, *MATLAB*. 2014, http://www.mathworks.com.