A two-stage technique to improve intrusion detection systems based on data mining algorithms

Hachmi Fatma
Larodec, ISG, University of Tunis
Tunisia

Limam Mohamed
Dhofar University, Oman

An intrusion detection system (IDS) is the fundamental part of the security infrastructure, since it ensures the detection of any suspicious action. Although the detection of intrusions and attacks is the ultimate goal, the huge amount of generated alerts cannot be properly managed by the administrator. In order to improve the accuracy of sensors, we adopt a two-stage technique. The first one aims to generate meta-alerts through clustering and the second one aims to reduce the rate of false alarms using a binary classification of the generated meta-alerts. For the first stage we use two alternatives, self-organizing map (SOM) with k-means algorithm and neural GAS with fuzzy c-means algorithm. For the second stage we use three approaches, SOM with K-means algorithm, support vector machine and decision trees. Based on a public data set and several evaluation criteria, our proposed procedures are evaluated. Results show that our procedures outperform other competitor methods by reducing the rate of false positives.

Keywords; IDS; clustering; binary classification; alerts; meta-alerts.

I. INTRODUCTION

Computer security aims to provide a high level of protection against criminal activities such as violation of privacy, corruption of data, fraud and access to unauthorized information. In fact computers are in tremendous need for an efficient and powerful security policy to secure the information system and to prevent attackers from destroying it. Currently, we are facing an enormous growth of malicious code signature, cybercrimes and threats which can put the security administrator in very critical situations. This explains the major importance accorded to security technologies such as authentication, cryptography and intrusion detection. Given their fundamental role as layers of defense, the accuracy of those technologies should continuously be improved to ensure the detection of intrusive activities. In this work we focus on applying data mining for intrusion detection. In fact the accuracy of an IDS depends on its ability to detect real threats on the network and to avert the administrator about them. It should eliminate false attacks and generate only true ones. In order to reach this objective, data mining is applied to collect and to analyze the information stored in large databases produced by an IDS. The set of generated alerts should be explored carefully to separate intrusive activities from normal network traffic.

II. RELATED WORKS

[1] introduced four alert correlation techniques: prerequisites and consequences of individual attacks, similarity-based, pre-defined attack scenarios and statistical causal analysis. The first one aims to correlate two alerts if several necessary conditions occur which are the pre-conditions and the consequences of a specific attack. The second one is based on the similarity of some attributes to correlate alerts such as ip addresses or port number. The third one uses the concept of pre-defined attack scenario to compare low level alerts and decide whether they have to be correlated or not. The last one is based on the causal relationship to correlate low level alerts. They discussed the advantages and disadvantages of each correlation technique using several evaluation criteria. However, they gave only a theoretical view of each technique and their work does not provide any experimental study. [2] introduced a new alert correlation technique to extract attack strategies based on the causal relationship between alarms. This technique is based on two approaches: Multilayer perceptron (MLP) and support vector machine (SVM). In the experimental study they used DARPA 2000 to test their proposed technique. In fact MLP and SVM algorithm require a training set to build a model that will be used to predict the right decision for a new observation. The elaboration of training examples is not simple and requires a high level of experience in the computer security domain. [3] discussed the false alarms issue. They performed an evaluation of the IDS SNORT against DARPA 1999 to quantify the rate of false attacks generated by SNORT and to measure its performance. Results reveal that the number of false alerts is very high and actually outnumber real attacks. Also, the IDS has detected only 32 attacks which confirms its low detection performance. To overcome this problem two solutions are presented: writing specific rules for SNORT and using an alert correlation process. [4] proposed a two-stage technique in order to improve the accuracy of an IDS. The first stage aims to classify the generated alerts based on the similarity of some attributes to form partitions of alerts using SOM with k-means algorithm. The second one aims to classify the meta-alerts

978-1-4673-5814-9/13/$31.00 ©2013 IEEE

created in the first stage into two clusters: true alarms and false ones. The binary classification is based on the collection of seven features extracted from the set of meta-alerts using SOM with k-means to cluster the input set. In the experimental study they used the DARPA 1999 to test their proposed technique. Since, the main goal is to improve the accuracy of an IDS by reducing the number of detected alerts and the rate of false positives than the idea of two-stage technique is very appropriate and efficient. Instead of performing the binary classification on the set of low level alerts, it is more efficient to cluster generated alarms into meaningful partitions. Then, the created partitions are classified whether as true or false attacks. The proposed technique allows the reduction of the huge number of produced alarms and the removal of false attacks. Unfortunately the use of SOM with k-means in the second stage is not very efficient since the administrator should manually determine which cluster contains the true alarms by examining the two attributes: alarm frequency and time interval. [11] presented an alarm clustering method to handle generated alerts more efficiently. It contrasts three dimensions: Depth of analysis, Ease of use and Bias. The first dimension implies a better clustering of real time alarm correlation. The second one implies a more simple and easy use unlike the other manual alarm correlation methods. The third one aims to select large partitions of false attacks. [10] and [12] presented new techniques based on neural networks. [7] and [9] introduced new methods based on alarm clustering.

III. ADOPTED TECHNIQUE

In this paper, we adopt the two-stage technique proposed by [4]. The first stage aims to reduce the number of generated alerts and the second one aims to reduce the rate of false positives. Our interest in the first stage is to correlate low level alerts into meaningful partitions. The correlation is based on the similarity-based technique. It is a very famous alert correlation technique based on maximizing the degree of similarity between objects in the same cluster and minimizing it between clusters as explained by [1]. Therefore, the classification is based on the similarity of some selected attributes (timestamp, source and destination ip addresses), so that alerts in the same partition are more similar to one another than they are to alerts in other partitions. Thus, the administrator will face a manageable set of alarms since all the closest ones are merged together and constitute one attack scenario. SOM and Neural GAS algorithm are applied to the data extracted from the report produced by the IDS in which lies all the information related to each detected alert. Then, as a second classifier, k-means clustering algorithm is applied to the set of neurons produced by SOM. Then, FCM algorithm is applied to the set of neurons generated by Neural GAS in order to classify the mapped data into groups of similar alerts named clusters or meta-alerts. In the second stage, our interest is to reduce the rate of false alarms. We propose three alternatives: SOM with k-means algorithm, SVM or Decision Trees (DT). As a first step, we need to define the set of data used in this stage which is the collection of some attributes extracted from each cluster created in the previous stage and judged useful for the binary classification as proposed by [3]. Seven attributes are chosen to compose each input vector: The number of input vectors is the number of clusters created during the first stage. Each meta-alert is classified as true meta alarm or false meta-alarm. Indeed, this binary classification allows the reduction of the rate of false alarms.

IV. EXPERIMENTS AND EVALUATION

Data mining techniques are efficient and suitable to be integrated in the intrusion detection domain, since they ensure the usage of large databases generated by the sensors. We perform an experimental study for the two stage alert correlation technique. To test its efficiency we used a public data set named DARPA 1999, commonly used for the evaluation of computer network sensors. Our experiments are based on the off-line evaluation sets:

- First day of the first week: it is attacks free and will be used as a training set for the SVM algorithm and DT.
- First day of the fourth week : the file outside.tcpdump is used to be our first testing data set In order to test our used procedures, we select two labeled attacks that occurred in that day as presented in Table 1.
- First day of the fifth week : the file outside.tcpdump is used to be our second testing data set. In order to test our used procedures, we select two labeled attacks that occurred in that day as presented in Table 1.

TABLE I – The selected attacks from DARPA 1999

	First testing set	Second testing set
Attack 1 ID	41.162715	51.185613
Attack 1 Name	portsweep	ls
Number of alerts	10	8
Attack 2 ID	41.182453	51.194715
Attack 2 Name	secret	dosnuke
Number of alerts	78	7

In order to obtain the set of alerts that will be used in the classification, we use the data extracted from the first day of the fourth week of the 1999 DARPA data set named outside.tcpdump as a first testing data set and the first day of the fifth week as a second testing data set. As a first step we run the IDS SNORT, characterized by its availability since it is an open source system, under WINDOWS XP against DARPA data sets. Moreover, the algorithms used in the two-stage technique: SOM, Kmeans, Neural GAS, FCM, SVM and DT are implemented using SOMtoolbox, spider toolbox, statistics toolbox and Fuzzy clustering toolbox which are running on MATLAB 7.10.0.

A. First stage: first alternative SOM with k-means

The main problem that an IDS must overcome is the generation of large databases which exceed the administrators

ability for analysis. In fact, one event or activity can generate a set of redundant and low priority alerts which explain their huge number. So, to deal with this issue all produced alerts in a certain time by the same event will be gathered together. As a result the set of alerts will be substantially reduced and become manageable. Actually, to group connected alerts we should rely on appropriate attributes to provide reliable results. So, from the snort database we extract three attributes: ip source, ip destination and timestamp, judged to be appropriate for the classification and able to define one attack scenario as suggested by [6] and [4]. Therefore, the similarity between two source ip addresses indicates the possibility that two alerts can be triggered by the same attacker since the ip source represents the identity of the attacker in the network. In fact, there is a very low probability to correlate two alerts having different destination ip addresses. The timestamp attribute has a high effect on the classification, since it allows the determination of the exact time in which a certain event occurs. Once the input data set is ready, SOM algorithm is applied to build the associated map. The input data fed to the k-means algorithm are the set of neurons created by the first classifier SOM. Therefore, each cluster groups 4 similar neurons together which implies the grouping of alerts triggered from the same event in a particular time based on the degree of similarity of the three chosen attributes. So, to ensure having the best clustering solution, we should test different values of k until we get the optimal portioning of the data. The latter is based on several validity measures to test the quality of the partitions. The first validity measure, Separation index (SI), determines the average number of data and the square of the minimum distances of the cluster centers. Indeed, a small value of SI indicates an optimal portioning. The second validity measure, Dunns index (DI), is used to identify whether clusters are well separated and compact or not. A big value of DI implies a good clustering. The third validity measure, Xie and Beni's index (XB), quantifies the ratio of total variation within cluster and the separation of clusters. To have an optimal number of partitions, the value of XB index should be minimized.

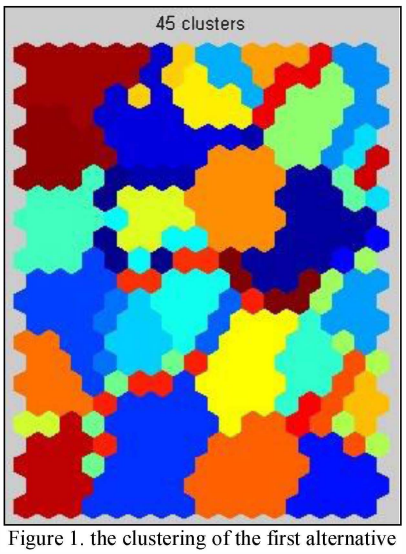

Figure 1. the clustering of the first alternative

B. First stage: second alternative neural gas with FCM

The relation between the alerts is defined by the degree of similarity of the three chosen attributes. Therefore, in this second alternative, as a first classifier we use the Neural GAS. The set of neurons created by the GAS will be fed into FCM clustering algorithm. The evaluation of FCM clustering solution is conducted through five criteria. Three of them were already introduced in the previous section and the remaining two are used specially in the case of fuzzy clustering algorithms since they use only the information of fuzzy membership to evaluate the quality of the clustering. Partition coefficient (PC) allows the measurement of the degree of overlap between clusters, a big coefficient implies an optimal clustering solution. Classification entropy (CE) is a measure of the fuzziness of a cluster partition, a small value implies a good partitioning solution

C. Summary of results

TABLE II – *Results summary of the first stage for the first testing set*

	Number of clusters	SI	DI	XB
SOM and k-means	45	3.614^{e}-004	0.0285	3.1397
Gas and FCM	45	2.866^{e}-004	0.0488	2.9285

Table 2 displays the results of the first testing data set, which consist on the recorded values of the three validity indices related to each alternative: SI, DI and XB. In fact, the second alternative looks better since it has the best values of the evaluation criteria.

TABLE III –*Results summary of the first stage for the second testing set*

	Number of clusters	SI	DI	XB
SOM and k-means	27	0.0023	0.0081	1.5889
Gas and FCM	29	9.7^{e}-004	0.1539	2.5593

Table 3 displays the values of the evaluation indices recorded from the second testing data set. We notice that both SI and DI validity indices, of the second alternative, are better than those of the first one. So, globally Neural GAS with FCM produces a better solution than SOM with k-means.

D. Second stage: First alternative SOM with k-means

A false alarm is an event triggered by the sensor as an attack but in reality it is not. So in order to reduce the rate of detected false attacks, a binary classification of the input data set into two clusters of true alerts and false ones, is needed. Therefore,

in the second stage the main aim is the identification of false attacks and the reduction of their rate. Then, the administrator considers only true threats and does not waste his time or effort analyzing false alerts. In the second stage, we use the set of clusters already created by the second alternative: Neural GAS with FCM since it is judged more appropriate for the second stage alarm classification. This binary affectation is based on the collection of seven extracted attributes judged efficient for the classification. In fact, the selection of the seven features are based on the association rule method of [3].

- Number of alerts in each cluster
- Signature type
- Protocol number
- Port number
- Alert priority
- Time interval
- Number of events

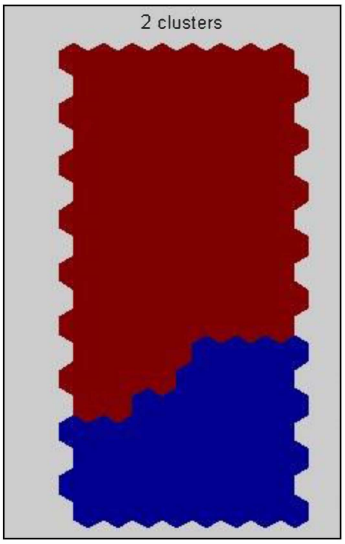

Figure 2. The binary classification of the first alternative

The first alternative consists on using SOM as a first classifier to build a map based on the given input vectors, and the k-means algorithm as a second classifier to clarify the boundaries of the partitions and to ensure a clear separation of the input data items.

TABLE IV *–Results of the first alternative for testing set 1*

	SOM and K-means	Real result
False alarms	205	213
True alarms	96	88

Table 4 displays the number of false alarms generated by the SOM and K-means for the first testing set.

Table 5 displays the number of false alarms generated by the first alternative for the second testing set.

TABLE V *–Results of the first alternative for testing set 2*

	SOM and K-means	Real result
False alarms	259	279
True alarms	35	15

E. Second stage: second alternative SVM

SVM algorithm is well known by its efficiency in binary classification. In this experimental study we used 3 different kernel functions : linear, polynomial and RBF. We start by creating the training set that will be used as a model. Since each example of the training set is marked by -1 if it is a member of the class of true alerts or 1 if it belongs to the class of false alerts, then SVM is able to predict, based on the learned model, in which class a new example will be placed.

TABLE VI –Results s*ummary of the 3 kernel functions for testing set 1*

	False alerts	True alerts
Linear SVM	193(64%)	108(36%)
Polynomial SVM	284(94%)	14(6%)
RBF SVM	212(70%)	89(30%)
Real result	213(71%)	88(29%)

Table 6 displays the rate of false and true alerts related to each kernel function. As illustrated the most appropriate function is the RBF SVM since it generates the closest value to the real result. For the first testing set the RBF SVM reveals that 70 percent of detected alerts are false attacks and only 30% of them are true threats which is the best result comparing with the linear SVM and polynomial SVM.

TABLE VII *–Results summary of the 3 kernel functions for testing set2*

	False alerts	True alerts
Linear SVM	188(64%)	106(36%)
Polynomial SVM	188(64%)	106(36%)
RBF SVM	274(93%)	20(7%)
Real result	279(95%)	15(5%)

Table 7 contains the results associated to the second testing set. The RBF SVM reveals that 93% of the detected alerts are false attacks and only 7% are true alerts. It has the closest rate compared with the real result while the others succeeded to detect just 64 percent of the false attacks and failed to detect the remaining false alerts.

F. Second stage: Third alternative DT

TABLE VIII –*Results for testing set 1*

	DT	Real result
False alarms	209	213
True alarms	92	88
Rate false alarms	69%	71%

Table 8 presents the results of the first testing data set based on the learned model created by the training data set. We notice that 69% of the alerts are false attacks and only 31% of them are real threats on the network.

TABLE IX –*Results for testing set 2*

	DT	Real result
False alarms	261	279
True alarms	33	15
Rate false alarms	88.77%	95%

Table 9 presents the results of the second testing data set based on the developed model during the training task.We notice that 88.77% of alerts are false and only 21.23% of them are real threats on the network.

G. Summary of results

TABLE X –*Results summary for the 3 alternatives (testing set 1)*

	False alerts	True alerts
SOM with k-means	205	96
RBF SVM	212	89
DT	209	92
Real result	213	88
Detection rate (SOM)	96%	
Detection rate (RBF)	99%	
Detection rate (DT)	99%	

Table 10 reveals that SVM algorithm has the most optimal detection rate. The detection rate is the success of a certain algorithm in detecting the set of false alarms. SVM detected 99% of the total number of false attacks, in the meanwhile SOM with k-means detected only 96% of them. DT detected 98%.

TABLE XI –*Results summary for the 3 alternatives (testing set 2)*

	False alerts	True alerts
SOM with k-means	259	35
RBF SVM	274	20
DT	261	33
Real result	279	15
Detection rate (SOM)	92.8%	
Detection rate (RBF)	98.2%	
Detection rate (DT)	93.5%	

Table 11 reveals that SVM algorithm has the most optimal detection rate. As shown SVM detects 98.2% of the total number of false attacks.

V. CONCLUSION

An IDS is an essential part of any security package since it ensures the detection of intrusive activities and the notification of the network administrator if the information system is being or has been hacked. Despite its fundamental role in the security architecture, an IDS tends to generate large databases where the majority of detected alerts are false alarms. Data mining is the proper tool to improve the accuracy of IDSs since it allows the collection and analysis of data.

In this work we adopted a two-stage alarm correlation technique to improve the accuracy of an IDS. The aim of the first stage is the reduction of the large volumes of detected alerts. We used two alternatives: SOM with k-means and Neural gas with FCM, in order to cluster low level alerts into meaningful partitions or meta-alerts that contain all alarms triggered by the same event in a certain time. Experimental results show that neural gas with FCM provide a better clustering results.

The aim of the second stage is to reduce the rate of false attacks. We used three alternatives: SOM with k-means, SVM and DT, in order to perform a binary classification of the meta-alerts already generated in the previous stage. Experimental results show that SVM algorithm provides the best performance since it has the best detection rate. As our procedures were tested on only one IDS and evaluated using only four attacks, it will be of interest to extend this work to reach the complete attacks database and its application on real network traffic. Another interesting area to be explored is to study alarm correlation for multiple sensors.

REFERENCES

[1] Yusof R, Sahib S, Selamat SR. 2008. Intrusion alert correlation technique analysis for heterogeneous log. International journal of computer science and network.

[2] Zhu B, Ghorbani A. 2006. Alert correlation for extracting attack. International journal of network security.

[3] Piatesky-Shapiro G. 1991. Discovery, analysis and presentation of strong rules . Knowledge discovery in databases.

[4] Tjhai C, Furnell M, PapadakI M, Clarck L. 2010. A preliminary two-stage alarm correlation and filtering system using som neural network and k-means algorithm. Computers and Security.

[5] Tjhai C, Furnell M, PapadakI M, Clarck L. 2008. The problem of false alarms: Evaluation with snort and darpa 1999 data set. Trust, privacy and Security in digital business.

[6] Spathoulas P, Katsikas K. 2010. Reducing false positives in intrusion detection systems. Computers and

[7] Perdisci R, Giacint G, Roli F. 2006. Alarm clustering for intrusion detection systems in computer networks. Engineering Applications of Artificial Intelligence.

[8] Federico M, Matteucci M, Zanero S. 2009. Reducing false positives in anomaly detectors through fuzzy alert aggregation. Information fusion.

[9] Al-Mamory S, Zhang H. 2009. Intrusion detection alarms reduction using root cause analysis and clustering. computer Communications.

[10] Bievens A, Palagiri C, Szymanski B, Embrechts, M. 2002. Network-based intrusion detection using neural networks. Intelligent engineering system through artificial neural networks.

[11] Julisch K. 2003. Clustering Intrusion Detection Alarms to Support Root Cause Analysis. ACM Transactions on Information and System Security.

[12] Labib K, Vemuri R. 2001 .NSOM : A real time network-based intrusion detection system using self-organizing map. Networks security.