2015 11th International Conference on Computational Intelligence and Security

A two-stage process based on data mining and optimization to identify false positives and false negatives generated by intrusion detection systems

Hachmi Fatma
ISG, university of Tunis
Tunisia
Hachmi.fatma@gmail.com

Mohamed Limam
Tunisia and Dhofar University, Oman
Tunisia

***Abstract*—To ensure the protection of computer networks, an intrusion detection system (IDS) should be integrated in the security infrastructure. However, IDSs generate a high amount of false alerts exceeding the administrator ability for analysis and omit several attacks which can threaten the network security. In this paper, a two-stage process based on data mining and optimization is proposed having as input the outcome of multiple IDSs. In the first stage, for each IDS the set of elementary alerts is clustered to create a set of meta-alerts. Then, we remove false positives from the sets of meta-alerts using a binary optimization problem. In the second stage, we discard the meta-alerts generated by all IDSs and only those missed by one, two or most of them are left. This set is called the set of potential false negatives. In fact, at this level a meta-alerts fusion is performed to avoid the redundancy between meta-alerts collected from multiple IDSs. Finally, a binary classification algorithm is proposed to classify the potential false negatives either as real attacks or not. Experimental results show that our proposed process outperforms concurrent methods by significantly reducing the rate of false positives and false negatives.**

***Keywords*-clustering; binary classification; binary optimization; false positives; intrusion detection systems; false negatives.**

I. Introduction

Computer security aims to protect networks from criminal activities such as violation of privacy, corruption of data and access to unauthorized information. In fact, to secure the information system and to prevent hackers from destroying it, computers are seeking for efficient and powerful security technologies. Hence, given their role as layers of defense, IDSs are considered as fundamental components for the protection of computer networks. Therefore, their accuracy depends on their ability to detect real threats on the network. But, IDSs generate a lot of false positives(FPs) and a high rate of false negatives (FNs). A false alerts is defined as a signal triggered mistakenly by an IDS reporting an intrusion but in reality it is just a normal network traffic. A FN is an attack missed by the IDS which can cause serious damages in the network.

So, to enhance the accuracy of IDSs, we propose a two-stage process to eliminate FPs and FNs. In the first stage, the set of alerts generated by each IDS is clustered into a set of meta-alerts based on several attributes extracted from the alert database. Then, a binary optimization problem (BOP) is formulated to identify FPs. In the second stage, the input set is created by discarding the meta-alerts generated by all IDSs and the remaining ones are grouped into one set called potential FNs (P-FNs). First, this set is clustered to group the similar meta-alerts generated from different IDSs together in order to ovoid their redundancy. Then, a binary classification algorithm (BCA) is proposed to identify FNs from the set of P-FNs.

The remainder of this paper is organized as follows. Section 2 gives an overview of the related works. Section 3 describes the proposed method for FPs and FNs reduction. Experimental results and performance comparisons are given in Section 4. Finally, conclusion and future work are given in Section 5.

II. Related Works

Many research works focused on reducing the rate of FPs generated by IDSs. [1] introduced a two-stage alert correlation and filtering technique(SOM-KM). The first stage aims to group the generated alarms based on the similarity of some extracted attributes to form meta-alerts using SOM with k-means algorithm. The second stage aims to classify the created meta-alerts into two clusters: true alerts and false ones. Unfortunately, using SOM with k-means in the second stage is not very efficient since the administrator needs to manually determine which cluster contains the true alerts by examining the two attributes: alarm frequency and time interval. [2] presented the decision support classification (DSC) alarm classification. It groups all the alerts generated in an attack-free environment. So, all alerts are considered as FPs in this environment and all the recorded patterns in this case represent the normal behavior and are called patterns of FPs. Then, DSC eliminates FPs based on these recorded patterns. [3] proposed a Beysian network model for classifying the alarms generated by IDSs as attacks or false alerts. [4] introduced a Genetic Fuzzy Systems within a pairwise learning framework for two main reasons. First, fuzzy sets enables a smooth separation between the concepts and allows a better interpretability in the rule set. Second, the learning scheme, in which all possible pair of classes are

978-1-4673-8660-9/15 $31.00 © 2015 IEEE
DOI 10.1109/CIS.2015.82

contrasted, improves the precision for the rare attacks since it provides a better separability between normal activities and attack types. [5] proposed a novel approach called the cluster center and nearest neighbor (CANN). In this approach, two distances are computed then summed. The first one measures the distance between each observation and its cluster center, and the second one measures the distance between each observation and its nearest neighbor in the same cluster.

Even if reducing the rate of FPs can improve the accuracy of IDSs and help the security administrator to consider only real alerts and prevent incoming attacks, it is not enough since FNs can severely affect the reliability of IDSs. For instance, [6] introduces a new alert post processing technique based on the Majority Voting (MV) algorithm. MV consists on finding potential FPs (P-FPs) and potential false negatives (P-FNs) by comparing the generated alarms. When they supervise the same network traffic, some IDSs generate alerts and most do not, then these particular alerts are P-FPs of those IDSs. However, if some IDSs do not generate alarms but most do, then these alarms are called P-FNs of those IDSs. Thus, P-FPs and P-FNs are analyzed to verify if they are correctly classified as FPs and FNs. But, in [7], [8] and [9], authors conclude that MV usually leads to incorrect decisions producing low percentages of true P-FPs/P-FNs.

[9] developed a Creditability-based Weighted Voting (CWV) scheme to consider the various domain knowledge between multiple IDSs. CWV enhances the efficiency of alerts processing by reducing FPs and FNs. Their algorithm includes four components: Creditability Modeling (CM), Authority Selection (AS), Voter Exclusion (VE) and Weighted Voting (WV). First, CM test the IDSs detection capability for different types of network traffic and summarize the IDSs corresponding creditability by investigating their detection experience. If the creditability of some IDSs exceeds decision criteria, AS use them as authorities because they have low FP and FN rates. However, if no IDS can be selected as an authority, VE eliminates IDSs with poor performance because they give incorrect decisions.

III. The proposed method

FPs and FNs are two important metrics in the evaluation of IDS's reliability. In fact, FPs are normal network traffic classified as threats by an IDS which cause a time and effort loss for the security analyst. FNs define the failure of an IDS in detecting a real intrusion which can damage the information system and threaten the network security. Hence, we propose a two-stage process using as an input set the alerts generated by multiple IDSs. The first stage begins with a clustering step that aims to reduce the huge number of generated alerts by applying k-means algorithm. Then, the set of meta-alerts is cleaned from FPs using a BOP. Once only true attacks remain, the meta-alerts detected by all IDSs are removed and the remaining one are grouped to form the P-FN set. The second stage, begins by clustering the similar alerts coming from heterogenous IDSs together. Then, a BCA is proposed to identify FNs using a labeled training set.

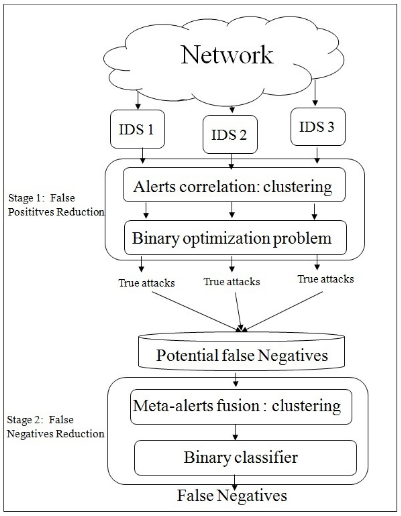
Network
IDS 1
IDS 2
IDS 3
Stage 1: False Posititves Reduction
Alerts correlation: clustering
Binary optimization problem
True attacks
True attacks
True attacks
Potential false Negatives
Stage 2: False Negatives Reduction
Meta-alerts fusion : clustering
Binary classifier
False Negatives

Figure 1. Proposed method for FP and FN reduction

A. The Training set processing

To efficiently identify FPs and FNs, a prediction model is needed to classify data vectors. In fact, deciding the nature of a given alert requires expertise in the domain which makes the detection of FPs and FNs a hard task. Thus, a labeled training set is used to create the prediction model which is processed as follows. First, the training set is divided into two clusters, the first one groups false alerts and the second one gathers real attacks. Then, all FPs reporting the same event are grouped together using k-means algorithm since one event can generate multiple signatures. The IP addresses are the identity of the hacker in the network and the timestamp defines the time frame of a given alert. Hence, the clustering of false alerts is based on the similarity of the three attributes namely, Source IP address, Destination IP address and Timestamp. This resulting set of FPs, SFP (FP_1, $FP_2 ... FP_s$) is used later for the classification. However, for the second group, the alerts are classified into five clusters: User to Root (U2R), Denial of Service (DoS), Remote to local (R2L), Data and Probe. The created set of real meta-alerts are denoted by SRA (RA_1, $RA_2 ... RA_5$).

B. False positives reduction

The first stage aims to reduce the number of FPs generated by each IDS. First, the sets of alerts generated by the multiple IDSs are clustered to group the alerts generated by the same event together. This clustering is performed using the K-means algorithm based on the similarity of the attributes used to cluster the training set. Then, FPs reduction problem is formulated as a BOP because two alternatives occur whether a meta-alert is a false alert or not. If the cluster is a FP, it will be assigned the value 1 otherwise 0. In fact, the distance is used as a measure of similarity between observations. If the distance between two data vectors is large enough than they are dissimilar otherwise they are similar and a correlation between them is feasible. This means that a false meta-alert is more similar to SFT than SOT. Therefore, the formulation of the BOP is as follows:

$$\min \sum_{i=1}^{n} D_{ik} x_i \qquad (1)$$

$\sum_{i=1}^{n} (\sum_{j=1}^{p} D_{ij}) x_i \leq \sum_{i=1}^{n} (\sum_{y=1}^{s} D_{iy}) x_i$

$$x_i \in \{0,1\}, i = (1,2,....,n), j = (1,2,....,p), y = (1,2,....,s)$$

In the objective function, the decision variables $X(x_1, x_2, x_3.....x_n)$ are the distinct meta-alerts in a given testing set. The objective is to select the clusters having the minimum distance to clusters in SFP than those in SOT. D_{ik} is distance measure between x_i and the centroid of SFP (SFP_k) which is equal to $-D_{ik}$ if $D(x_i, SFP_k) \preceq$ Max(Distances between all FP_j in SFP) otherwise it is D_{ik}. D_{ij} is the distance between x_i and F_j and D_{iy} is the distance between M_i and T_y.

C. False Negatives Reduction

This stage aims to detect the attacks missed by each IDS. So, the input set is collected from the true attacks remaining from the previous stage but not generated by all the IDSs. In fact, the meta-alerts missed by one, two or all the IDSs represent the FNs of those particular IDSs. Hence, a set of P-FN that groups all the possible FNs is created. However, a major problem arises at this level which is the redundancy of meta-alerts generated by different sensors but reporting the same threat. Thus, a clustering step is used to group all the similar meta-alerts in the P-FNS set eliminating the aforementioned redundancy problem. As done previously, the same attributes selected for the training set clustering are used. These listed attributes allows the definition of a single event even if it is detected by different IDSs. So, the meta-alerts fusion is mainly performed to cluster the meta-alerts defining the same event but generated by various sensors. Once the clustering step is finished, A BCA is used to classify each meta-alert from the testing set either as FN or not. The proposed algorithm is detailed as follows.

Algorithm 1: Proposed classification algorithm

Input: meta-alerts of the testing set, training clusters
Output: FN

```
Begin
N is the number of meta-alerts, SFP is the FP training set
SOT is the training set of true alerts
FN: number of clusters inside SFP
TN : number of clusters inside SOT, $C_1$:centroid of SOF
$C_2$ : centroid of SOT
For each meta-alert ($M_i$) from N
    If Dist($M_i$,$C_2$) $\preceq$ Dist($M_i$,$C_1$)
        For each meta-alert ($RA_j$) from TN
            If Dist($M_i$,$RA_j$) $\preceq$ Max(Dist($RA_j$ in SOT)
            insert $M_i$ in the cluster SOT
            TN=TN+1
            End IF
        $C_2$ is the new centroid of cluster SOT
        End For
    End IF
End For
End
```

First, we compute the Euclidean distance between each meta-alert (M_i) from the testing set and the centroid C_1 and C_2 of the two clusters SFP and SRA respectively. If the distance between M_i and C_2 (Dist(MA_i,C_2)) is lower than M_i and C_1 (Dist(M_i,C_1)), then the probability that M_i is a FN is high. To ensure that M_i is correctly classified as FN, we test its similarity with each meta-alert RA_j inside SRA. Therefore, if the distance between M_i and RA_j is lower than the maximum distance between true meta-alerts inside SRA, then M_i is a FN otherwise it is a FP.

IV. Experimental results

To evaluate the performances of the proposed process, a public data set named DARPA 1999 is used. which is commonly used for assessing computer network sensors. Our experiments are based on the off-line evaluation sets:

- As a training set, we use the three weeks of training data.
- A sample from the fourth week is used as our first testing data set.
- A sample of the fifth week is used as our second testing data set.

Moreover, three open source IDSs are involved in the classification namely Snort, Surcita and Ossec and all the experiments are performed using Matlab 7.10.0 under windows Server 2008.

A. Evaluation of False positives reduction methods

A false alarm is an event triggered by the sensor as an attack but in reality it is just a normal network traffic. Therefore, two rates are considered in the evaluation of FP methods: True negatives rate (TNR) which are true attacks successfully labeled as true attacks and true positives rate

(TPR) which are false meta-alerts successfully classified as FPs. TPR and TNR are given by:

$$TPR = \frac{TP}{TP + FN} * 100\%. \quad (2)$$

$$TNR = \frac{TN}{TN + FP} * 100\% \quad (3)$$

Table I
TPR AND TNR FOR TESTING SETS 1 AND 2

Methods	metrics	Testing set 1	Testing set 2
BOP	TPR	95	97
	TNR	94	96
CWV	TPR	68	71
	TNR	65.3	62
SOM-KM	TPR	78	75
	TNR	75	69

Table 1 shows that the proposed BOP detects up to 97% of FPs for Testing set 1. The other methods are less performing than BOP with a detection rate of 68% for CWV and 78% for SOM-KM. For Testing set 2, TPR generated by BOP still better than other methods.
BOP outperforms CWV and SOM-KM by detecting up to 96% of true attacks.

B. Evaluation of false Negatives reduction methods

TPs and TNs define the success of an IDS in detecting normal and malicious activities, respectively. Therefore, to evaluate the performances of BCA , TPR and TNR are used. Table 2 illustrates the TPRs and the TNRs for the two testing sets respectively. As shown, BCA generates better results than MV in reducing FNs. Hence, TPRs generated by BCA reach 97.1% and TNRs generated by BCA reach 96.5%. Therefore, BCA ouperforms MV and CWV.

Table II
TPR AND TNR FOR TESTING SETS 1 AND 2

Methods	Metrics	Testing set 1	Testing set 2
BCA	TPR	93.5	97.1
	TNR	94.5	96.5
MV	TPR	54.8	42.1
	TNR	58	51
CWV	TPR	72	70
	TNR	79	61

C. Time Performance Evaluation

To evaluate the time performance of our proposed process, we compare its running times with CWV using the aforementioned testing sets. Table 3 shows the good performance of our proposed process compared to CWV.

Table III
EXPERIMENTAL RESULTS OF RUNNING TIMES

Methods	Testing set 1	Testing set 2
Proposed process	0.20	0.31
CWV	0.41	0.65

V. CONCLUSION

An IDS is a fundamental part of the security package since it prevents incoming intrusions. Despite its role in the security architecture, an IDS generates high rates of FPs and FNs. Hence, we propose a two-stage alarm correlation and optimization technique to enhance the accuracy of an IDS. The first stage aims to reduce the large volumes of detected alerts. The second one aims to detect missed attacks or FNs. As our proposed process is tested using off-line data sets, it will be of interest to extend this work to be applied on real network traffic.

REFERENCES

[1] Tjhai C., Furnell M., PapadakI M., Clarck L.: A preliminary twostage alarm correlation and filtering system using som neural network and k-means algorithm. Computers and Security, vol. 29, pp. 712-723. (2010)

[2] Zhang Y., Huang S., WangY.: IDS alert classification model construction using decision support techniques. International conference on computer science and electronics engineering, pp. 301-305. (2012)

[3] Benferhat S., Boudjelida A., Tabia K., Drias H.: An intrusion detection and alert correlation approach based on revising probabilistic classifiers using expert knowledge. International Journal Appl. Intell, vol. 38(4), pp. 520-540. (2013)

[4] S. Elhag, A. Fernndez, A. Bawakid, S. Alshomrani, F. Herrera. On the combination of genetic fuzzy systems and pairwise learning for improving detection rates on Intrusion Detection Systems. Expert Systems with Applications, 2015; 42: 193202.

[5] W-C. Lin, S-W. Ke, C-F. Tsai. CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. Knowledge-Based Systems, 78: 1321, 2015.

[6] Chen I-W., Lin P-C., Luo C-C., Cheng T-H., Lin Y-D., Lai Y-C.: Extracting attack sessions from real traffic with intrusion prevention systems. In: Proceeding of IEEE international conference on communications (ICC). (2009)

[7] Latif-shabgahi G., Bass JM., Bennett S.: A taxonomy for software voting algorithm used in safety-critical systems. IEEE Transactions on Reliability, vol. 53(3), pp. 319-28. (2004)

[8] Parham B.: Voting algorithms. IEEE Transactions on Reliability, vol. 43(4), pp. 617-629. (2002)

[9] Ying-Dar L., Yuan-Cheng L., Cheng-Yuan H., Wei-Hsuan T.: Creditability-based weighted voting for reducing false positives and negatives in intrusion detection. Computers & security, vol. 39, pp. 460-474. (2013)