Filtering Intrusion Detection Alarms using Ant Clustering Approach

Ghodhbani Salah
ISG Sousse
Sousse University, Tunisia

Jemili Farah
ISITCOM Sousse
Sousse University, Tunisia

ABSTRACT

With the growth of cyber attacks, information safety has become an important issue all over the world. Many firms rely on security technologies such as intrusion detection systems (IDSs) to manage information technology security risks. IDSs are considered to be the last line of defense to secure a network and play a very important role in detecting large number of attacks. However the main problem with today's most popular commercial IDSs is generating high volume of alerts and huge number of false positives. This drawback has become the main motivation for many research papers in IDS area. Hence, in this paper we present a data mining technique to assist network administrators to analyze and reduce false positive alarms that are produced by an IDS and increase detection accuracy. Our data mining technique is unsupervised clustering method based on hybrid ANT algorithm. This algorithm discovers clusters of intruders' behavior without prior knowledge of a possible number of classes, then we apply K-means algorithm to improve the convergence of the ANT clustering. Experimental results on real dataset show that our proposed approach is efficient with high detection rate and low false alarm rate.

General Terms

Security, intrusion detection system, data mining.

Keywords

Intrusion detection system, alarm filtering, ANTClass, ant clustering, intruders' behaviors, false alarms.

1. INTRODUCTION

With the explosive increase in number of services accessible through the Internet, information security needs to be carefully concerned and a sufficient protection is needed against cyber attacks. Intrusion detection systems (IDSs) are an essential component in computer network security. It monitors target sources of activities, collects and inspects audit data looking for evidences of intrusive behaviors. When it detects suspicious or malicious attempts, an alarm is raised giving the network administrator the opportunity to react promptly. IDS can be classified as Host-based Intrusion Detection System (HIDS) which protects a certain host or system, and Network-based Intrusion detection system (NIDS), which protects a network of hosts and systems [2]. The NIDS uses the audit data; an audit trail is a record of actions on a system that are logged to a file in chronologically sorted order. The alarms generated by a NIDS inform network administrators that their network is possibly under attack [1]. However, NIDS might generate huge amount of alarms during the detection stage, which exhausts system administrator by rendering a large amount of a false alarms. So we need more intelligent attack analysis to deal with false alarms problems and increase detection accuracy.

Various techniques have been proposed to analyze alarms and reduce false positives. Ning and Xu [3] proposed a technique that automatically learn attack strategies as acyclic graphs and extract them from correlated intrusion alarms. Kruegel et al [4] performed real-time verification of attacks in which the NIDS reduces the priority of non real attacks and, thus, differentiates between false and non-relevant alarms. Rachman [5] positioned a data mining layer, to represent the baseline of security system deployment and to analyze new data against this baseline, as an analysis layer within NIDS. Julisch and Dacier [6] handled alarms by identifying root causes and removing them. Cuppens et al. [7] analyzed alarms by implementing functions to manage, cluster, merge and correlate alarms.

Faour et al. [8] proposed Self-Organizing Maps (SOM) coupled with Bayesian networks for alarm filtering. But, their SOM work resulted in fixed network architecture in terms of number and map units arrangement. . In [1] a data mining technique based on Growing Hierarchical Self-Organizing Map (GHSOM) which adjusts its topology during the learning process according to the inputs data (alarms) to reduce false positive alarms and to assist system administrators in analyzing alarms generated by the IDS. The proposed algorithm aims to explore the hidden structure of alarm data, and to uncover false alarms (FP & FN) hiding in normal clusters. Considerably, a data sample consisting of 1849 data patterns including 6 web attack scenarios were tested using the proposed approach. The results show that the proposed algorithm outperforms SOM algorithm in terms of both false positives and false negatives which reduced from 15% to 4.7% and from 16% to 4% respectively. A major limitation in the previous solutions of alert correlation is that the methods used led to increase false positives.

In this work, we propose a system based on ANTClass [10] algorithm, for the first time, as Ant clustering approach for the problem of alarm filtering. Our approach, presents an unsupervised learning method that does not assume prior knowledge of the structure of the data, which supports administrators to explore alarms pattern in order to analyze intruders' behaviors and to uncover false positive and false negative alarms. The proposed method provides better detection and lower false positive rate by combining ANT and K-mean algorithms. Basically, ANT preprocessed the data to produce a number clusters with centers of intruders' behaviors. On the other hand, K-mean algorithm is applied to refine the final results of ANT module to get a more uniform partition of clusters.

We empirically show, using real world data that our proposed technique uncovers attack vectors hiding in normal clusters. That is, it reduces false negatives produced by the NIDS and it competes well with other data mining techniques like SOM and GSOM.

This paper is organized as follows. Section 2 provides a background on ant based clustering and ANTClass algorithm. Section 3 explains the architecture of ANTClass method for alarms filtering. Section 4 presents and discusses our empirical results. The last section offers the conclusions.

2. BACKGROUND ON ANT CLUSTERING

2.1 Ant based clustering

Ant colonies provide a means to formulate some powerful nature-inspired heuristics for solving the clustering problems. Several clustering methods based on ant behavior have been proposed in the literature. Ant-based clustering sorting was first introduced by Deneubourg et al. [10] He proposed the basic ant model for clustering. He focused on clustering objects by using a group of real-world robots. In his model, the ants would walk randomly on the workspace, picking or dropping one data element from it. The ants possessed only local perceptual capabilities. They could sense the surrounding objects were similar or not to the object, they were carrying. Based on this information, they would perform the pick or drop action.

The basic model (BM) of Deneubourg can be described as follows: The data items are randomly scattered into a two-dimensional grid. Initially, each data object that represents a multi-dimensional pattern is randomly distributed over the 2D space. Each ant moves randomly around this grid picking and dropping the data items. The decision to pick up or drop an item is random but is influenced by the data items in the ant's immediate neighborhood. The probability of dropping an item is increased if ants are surrounded with similar data in the neighborhood. In contrast, the probability of picking an item is increased if a data item is surrounded by dissimilar data, or when there is no data in its neighborhood. In this way, clustering of the elements on the 2D grid is obtained.

Lumer and Faieta [11] modified Deneubourg et al.'s BM [10] using a dissimilarity-based evaluation of the local density, in order to make it suitable for data clustering and it has subsequently been used in data mining [12]. This algorithm is called LF model or standard ant clustering algorithm (SACA). In this algorithm, each ant-like agent cannot communicate with each other, and they can only sense the similarity of the objects in their immediate region.

Lumer and Faieta [11] have introduced the notion of short-term memory within each agent. Each ant remembers a small number of locations where it has successfully dropped an item. And so, when picking a new item this memory is consulted in order to bias the direction in which the ant will move. Thus, the ant tends to move towards the location it last dropped a similar item. Lumer and Faieta [11] define picking up (Pp) and dropping probabilities (Pd) as follows:

- $$\mathrm{P_P}(O_i) = \left(\frac{k_1}{k_1 + f(O_i)}\right)^2 \qquad \textbf{(1)}$$
- $$\mathrm{P_d}(O_i) = \begin{cases} 2\, f(O_i) & \text{IF } f(O_i) < k_2\mathrm{z} \\ 1 & \text{IF } f(O_i) \geq k_2\mathrm{s} \end{cases} \qquad \textbf{(2)}$$
- $$f(O_i) = \begin{cases} \frac{1}{S^2} \sum_{O_j \in R_s(r(O_i))} 1 - \frac{f(O_i, O_j)}{\alpha} & \text{If } f > 0 \\ 0 & \text{Else} \end{cases} \qquad \textbf{(3)}$$

Where:

- f(oi) is a measure of the average similarity of data object oi with the other data object oj present in the neighborhood of oi
- d(oi, oj) is the dissimilarity between pair of objects (oi, oj)
- α is a factor that defines the scale for dissimilarity
- k1 and k2 are two constants that play a role similar to k1 and k2 in the BM.

Gutowitz [13] improved this model by giving the ants the capacity to sense the complexity of their neighborhood. The ants would not try to pick or drop anything in areas with low complexity. These complexity-seeking ants were able to avoid actions that did not contribute to the clustering process, performing their task more efficiently. Monmarche [9] proposed an algorithm where several objects are allowed to be on the same cell of the workspace grid. Each cell with one or more objects together corresponds to a cluster. Each ant is also capable of carrying more than one object at a time.

Ngenkaew [14] proposed two multiple pheromone concepts in ant based clustering with ant nest algorithm and with ant memory algorithm. Artificial Trailing pheromone and Foraging pheromone help ants to decide which direction to go or where to pick up or drop the item of food.

Handl [15] presented a comparative study of the performance of ant-based clustering against traditional methods. Study realized that ant clustering is very fast algorithm for high-dimensionality data and its interesting features like no assumption on the shape of the clusters, automatic identification of the number of the clusters and robust behavior to the effects of outliers.

Brown et. al [16] proposed hierarchical clustering by making a movement zone around each cluster. Worker ants move only in this zone. Then the resulting clusters are merged by the queen ant through a connection. Li et. al [17] proposed a new for calculating the distance based on reachability paths between two objects, called distance with connection. Jiang et. al [18] redefined the behavior of ant and colony similarity. Groups of objects are activated one by one to be artificial ants. This method does not search for the sample objects as ants are the objects themselves. It can reduce the number of iterations. Feghi et. al [19] presented a new Adaptive Ant-based Clustering Algorithm for clustering data sets. The algorithm takes into account the properties of aggregation pheromone and perception of the environment.

2.2 ANTClass algorithm

Monmarche [20] [21] combined the stochastic and exploratory principles of clustering ants with the deterministic and heuristic of the popular k-means algorithm in order to improve the convergence of the ant-based clustering algorithm. The proposed hybrid method is called AntClass and is based on the work of Lumer and Faieta [11]. The AntClass algorithm allows an ant to drop more than one object in the same cell, forming heaps of objects. Another important contribution of this algorithm is that it also makes use of hierarchical clustering, implemented by allowing ants to carry an entire heap of objects.

3. ALARMS FILTERING SYSTEM BASED ON ANTCLASS ALGORITHM

This section describes the structure and the implementation of the alarms filtering system. This system try to adapt ANTClass algorithm to alarms filtering problem .it consists of 3 modules: alarms preprocessing, ANT algorithm application, K-means algorithm Application. In fig 1 we present the general architecture of the proposed system.

Before detailing our approach we suggest presenting the notion of typical behavior which constitutes the central point of this approach. We believe that the various types of alarms generated by one NIDS for every couple of machines in connection in an interval of time can be representative of the nature of this session. Besides, this behavior can be similar for several machines in connection for the different periods. Then, the classification of this similar behavior in a number of typical behaviors can create coherent clusters of the data which can be significant potential scenarios of attacks.

3.1 Alarms preprocessing module

Firstly this module receives alarms from different NIDS installed in the network and extracts important information from each brut alert received. Alerts are retrieved from log files generated by NIDS and they represent an events produced by many external machines (with IPextern) trying to connect to a number of internal machines (with IPintern). Alarms preprocessing module creates a number of data vector called VC from alerts wich reflect intruder behavior in window of time. For a certain window of time, the numbers of alarms of type *i* for each pair (IPint, IPext) is calculated. As a result, an aggregated alarm data vector called VC is represented as follows:

[DT, IPextern, IPintern] = #a1, #a2. . . *#an* where

- DT represents an interval of time,
- IPextern represents the IP of external machines,
- IPintern represents the IP of internal machines,
- #ai represents the number of alarms of type i.

In the next sections we propose to exploit results of alarms preprocessing module in order to create a classification of intruders' behaviors using ant clustering method called ANTClass which a hybridization of Ant clustering and K-means algorithm

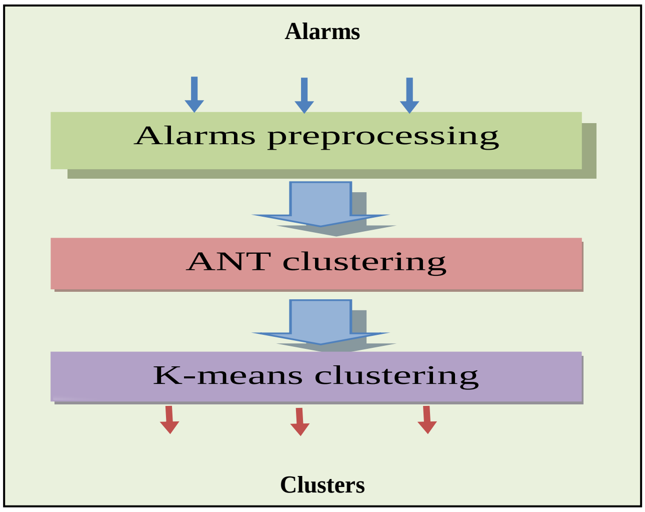

Fig 1. Alarm filtering system architecture

3.2 Application of ANT algorithm

This module receives data vectors produced by ''alarms preprocessing module'' in order to create a preliminary classification of intruders' behaviors which reflect the different actions of intruders in local machine. Hence, we use the Ant algorithm to apply clustering of data vectors; the ant algorithm is presented in fig 2. Firstly the ants are located randomly on the 2D board. Then each ant performs a move and possibly drops or pick up an object (which represent in our case a data vector). Each ant selects a random direction (among the 8 possible directions). Then each ant has a probability Pdc to further continue in this direction when moving next, else it flows randomly a new direction. We note that each ant has a speed parameter which tells of how many steps it will move in the selected direction before stopping again also the ant has a capacity which tells how many object it can transports. Once it has moved, the ant may possibly pick up or drop an object as described in formula (1) and (2). Finally we note that the stopping criterion of this algorithm is simply the number of iterations.

```
1.  Initialize randomly the ants positions
2.  Repeat
3.  For each ant anti Do
      (a) Move anti
      (b) If anti does not carry any object Then look at the
          cells in the neighborhood of anti location and
          possibly pick up an object
      (c) Else (anti is already carrying an object(O) look
          at the 8 cells around anti and possibly drop O
4.  Until Stopping criterion
```

Fig 2.Ant algorithm

3.3 Application of K-means algorithm

The previous algorithm based on ants only has the major advantage to providing a relevant partition of the data without any initial information about the future classification. However two important problems remain. The first one is due to the fact that some objects are not assigned to any heaps when the ant algorithm stops what we call ''isolated'' objects which are alone on the board. The second problem is that an object has been assigned to a wrong heap then it can take a long time until the object is transported to the right cluster. So we propose to combine ant based clustering algorithm with k-means algorithm in order deal with those problems. Hence, in this section we use k-means algorithm to remove quickly obvious errors and provide an efficient heuristic for assigning ''isolated'' objects.

The Ant algorithm provide a initial partition of clusters, then K-means algorithm computes the center of each clusters then it computes a new partition by assigning every object to the heap with center is the closest to that object. This cycle is repeated during a given number of iteration or until the assignment has not changed during one cycle. In fig 3 we present K-means algorithm. The experimentations presented in section 4 show that hybridization is efficient because it eliminate ''isolated objects'' and removes many misclassifications errors produced by Ant algorithm.

```
1.  Take as input the partition P of the data set found
    by the ants in the form of k heaps H1,…,Hk,
2.  Repeat
    (a)  Compute O center (H1), …, O center (Hk),
    (b)  Remove all objects from all heaps,
    (c)  For each object Oi ∈ E :
           i.   Let Hj, j ∈ [1,K] be the heap which
                center is the closest to Oi,
           ii.  Assign Oi to Hj,
    (d)  Compute the resulting new partition P=
         H1,…,Hk' by removing all empty clusters,
3.  Until stopping criterion.
```

Fig 3.K-means algorithm

4. EMPIRICAL WORK

4.1 Implementation

We have implemented our proposed approach with java language, using eclipse software, and we have applied this approach to a real dataset which retrieved from NIDS: Snort [1] log files. These files included 32031 alarm event records generated over duration of 20 days, from 20/11/2004 to 10/12/2004. These alarm events were produced by 4638 external machines trying to connect to 288 internal machines. The logs generated by SNORT include 16 real attack scenarios and one non-attack/normal scenario as identified by the domain expert. Attack scenarios include: 4 scenarios brute force on POP3, 3 scenarios crawler Web, 2 scenarios Web IIS, 2 scenarios scanner of vulnerability, 1 scenario IIS attack against apache server, 3 scenarios brute force against FTP server, and 1 scenario SNMP attack[1].

In order to evaluate our system, we have divided the dataset into 4 sets as it shown in Table 1.

Table 1.Test and training data set

Data set	Number of alerts	Number of data vectors	Number of windows
Test 1	166	50	10
Test 2	2815	1693	159
Test 3	11962	4000	330
Training	32031	12000	435

4.2 Experimentation

The main criteria that we have considered in the experimentation of our system are the detection rate, false positive rate and false negative rate.

- Detection Rate (DR): is defined as the number of attack scenario correctly classified by our system divided by the total number of test examples.
- False positive rate (FPR) : is defined as the proportion that normal data is falsely detected as attack behavior
- False negative rate (FNR): is defined as the proportion that attack data is falsely detected as normal behavior

Table 2.Comparison of results

Data set	FPR	FNR	DR
Test 1	166	50	10
Test 2	2815	1693	159
Test 3	11962	4000	330
Training	32031	12000	435

Table 2 shows that the low rate of FP and FN are generated with Test3 data set, so we can note that the rate was improved by increasing the size of the dataset; what means that the addition of new normal connections moves closer to the profile establishes in the real profile of the users. Also the Table 2 shows that DR is improved by increasing the number of data set element.

Table 3.Comparison of results

Approach	FPR	FNR
ANTClass	2%	2%
GHSOM	4.7%	4%
SOM	15%	16%

Table 3 shows the results of our proposed ANTClass on the testing data and how it compares with other system which used the same datasets: GHSOM [1] and SOM [2]. Results show high performance of our system. The approach based on ANTClass gives the smallest false alerts rates. ANTClass outperforms SOM and GHSOM on false positives and false negatives. Our system showed a reduction in false positives to 2% and of false negatives to 2%.

5. CONCLUSION

We have proposed a data mining technique based on ANTClass algorithm, to deal with the problem of the huge number false alarms, which represent a new decision support layer for network administrators to analyze and sort out alarms generated by a network intrusion detection system (NIDS). The application of our system in alarms filtering in intrusion detection context helps detect attacks and reduce false alarms with very considerable rates. There are still some challenges in our system. First, we can improve the clustering method in order to approximate the right number of classes. Secondly we can improve our system by integrating a prediction module which allows the prevention of intruders' actions.

6. REFERENCES

[1] N.Mansour & M. I. Chehab & A. Faour (2009), “Filtering intrusion detection alarms,” Cluster Computing, vol. 13, no. 1, pp. 19-29.

[2] Przemyslaw kazienko & Piotr Dorosz (2003),”Intrusion Detection Systems (IDS)’’, Part 2 Classification, Methods and Techniques. IT FAQ.

[3] Ning & Xu, (2003),”Learning attack strategies from intrusion alerts”. In: Proc. 10th ACM Conf. on Computer and Communications Security, pp. 200–209. Washington D.C

[4] Kruegel & Robertson & Vigna, (2004),"Using alert verification to identify successful intrusion attempts". Pract. Inf. Process. Commun. **27**(4), 220–228.

[5] Rachman,(2005),"Baseline analysis of security data. Securimine Software Inc", www.securimine.com

[6] Julisch & Dacier, (2002), "Mining intrusion detection alarms for actionable knowledge", Proc. International Conference on Knowledge Discovery and Data Mining, pp. 366–375. Edmonton, Canada

[7] Cuppens & Miege, (2002), "Alert correlation in a cooperative intrusion detection framework". In: Proc. 23rd IEEE Symposium on Security and Privacy, pp. 202–215. Toulouse, France.

[8] Faour & Leray, (2006), "Automated filtering of network intrusion detection alerts", In: Proc. 1st Joint Conf. on Security in Network Architectures and Security of Information Systems, pp. 277–291. Seignosse, France.

[9] Monmarche & Slimane, 1999, "On improving clustering in numerical databases with artificial ants", Advances in Artificial Life, pp. 626-635.

[10] Deneubourg & Gross, 1991, "The dynamics of collective sorting: Robot-like ants and ant-like Robots", In Proceedings of the First International Conference on Simulation of Adaptive Behavior: From Animals to Animats, Cambridge, MA, MIT Press, pp. 356-363

[11] Lumer & Faieta, 1994, "Diversity and adaptation in populations of clustering ants", Cambridge: MIT Press, In D. Cliff, P. Husbands, J.-A.Meyer, & S.W. Wilson (Eds.), From animals to animats: Proceedings of the Third International Conference on Simulation of Adaptive Behavior,pp. 501-508.

[12] Lumer & Faieta, 1995, "Exploratory database analysis via self-organization".

[13] Gutowitz, 1993, 'Complexity-seeking ants', In Proc. of the Third European Conference on Artificial Life.

[14] Ngenkaew & Satoshi Ono, 2008, "Pheromone- Based Concept in Ant Clustering", In Proc. of 3rd International conf. on Intelligent System and Knowledge Engineering, pp. 308, 312.

[15] Handl & Knowles, 2003,"Ant Based clustering: a comparative study of its relative performance with respect to k-means average link and 1-D-som", Technical Report No. TR/IRIDIA/2003-24, Universite Libre de Bruxelles, Belgium.

[16] Brown & M. Huber, 2010, "Pseudo-hierarchical ant-based clustering using Automatic Boundary Formation and a Heterogeneous Agent Hierarchy to Improve Ant-Based Clustering Performance", IEEE international conference on SMC, pp. 2016-2024, 2010.

[17] Shanfei Li & Wei Huang, 2010, "An Improved Ant-Colony Clustering Algorithm Based On the Innovational Distance Calculation Formula", Third International Conference on Knowledge Discovery and Data Mining,pp. 342-346, 2010.

[18] Hong Jiang & Qingsong Yu, 2010, "An Improved Ant Colony Clustering Algorithm", 3rd International Conference on Biomedical Engineering and Informatics, IEEE 978-1-4244-64982/10, pp. 2368-2372.

[19] I. El-Feghi & M. Errateeb, 2009, "An Adaptive Ant-Based Clustering Algorithm with Improved Environment Perception", International Conference on Systems, Man, and Cybernetics, San Antonio, TX, USA - October 2009 published in IEEE 978-1-4244-2794-9/09,pp.1431-1438.

[20] Monmarche & Slimane, 1999, "On improving clustering in numerical databases with artificial ants", Advances in Artificial Life, pp. 626-635.

[21] Monmarche, 1999, "On Data Clustering with Artificial Ants", In: Freitas AA, (ed.), Data Mining with Evolutionary Algorithms: Research Directions – Papers from the AAAI Workshop, AAAI Press, pp. 23-26.

.