International Journal of Electronic Commerce Studies
Vol.6, No.1, pp. 119-126, 2015
doi: 10.7903/ijecs.1392

DATA MINING BASED TECHNIQUE FOR IDS ALERT CLASSIFICATION

Hany Nashat Gabra
Ain Shams University
1229 El Sheikh Aly Gad El-Hak St., Sheraton Heliopolis, Cairo, Egypt
hanynashat@hotmail.com

Ayman M. Bahaa-Eldin
Ain Shams University
1 El Sarayat St. Abbaseya, Cairo, Egypt
ayman.bahaa@eng.asu.edu.eg

Hoda Korashy Mohammed
Ain Shams University
1 El Sarayat St. Abbaseya, Cairo, Egypt
hoda.korashy@eng.asu.edu.eg

ABSTRACT

Intrusion detection systems (IDSs) have become a widely used measure for security systems. The main problem for such systems is the irrelevant alerts. We propose a data mining based method for classification to distinguish serious and irrelevant alerts with a performance of 99.9%, which is better in comparison with the other recent data mining methods that achieved 97%. A ranked alerts list is also created according to the alert's importance to minimize human interventions.

Keywords: Intrusion Detection, Data Mining, Frequent Pattern, Frequent Itemset

1. INTRODUCTION

An IDS sensor can generate thousands of alerts in one day[1, 2]. Often, a vast majority of the alerts are false positives or of low importance[1, 3]. More than 90% of those alerts are irrelevant[4, 5, 6], so an IDS alert log's analysis techniques are often used to distinguish important IDS alerts from irrelevant events. Our results show that the performance of the proposed technique is enhanced, as we reduced the number of irrelevant alerts to 99.9%compared

with the performance of other recent techniques, which reduced the number of irrelevant alerts by only 74 to 97%[1, 2, 4, 7, 8].

2. RELATED WORK

Data mining techniques were first used for knowledge discovery from telecommunication event logs more than a decade ago[9.] Clifton and Gengo [10] investigated the detection of frequent alert sequences, and Ferenc[11], Walter A. Kosters and Wim Pijls[12] enhanced this knowledge for creating IDS alert filters. Long et al.[3] suggested a snort clustering algorithm. During the last 10 years, data mining based methods have been proposed in many research papers[3, 4, 5, 7, 8, 10].

3. MINING FREQUENT PATTERNS

Mining frequent itemsets from a database has been solved largely by algorithms that are a priori-based and by pattern-tree growth techniques. Algorithms for mining existing techniques do not include generating frequent patterns for each transaction, which is necessary for many applications.

Table 1. Example alerts/ Item data set records

Alerts	Item
Alert1	1 3 4
Alert2	2 3 5
Alert3	1 2 3 5

Assume a dataset which contains alert records generated by an IDS system in Table 1 where the set of items I = {1, 2, 3, 16, 20} and the set of Alerts = {Alert1, Alert2, Alert3}.Mining all alerts that have similar frequent itemsets at minimum support of 50% would require generating frequent itemsets with the alerts in the format [< itemset > Alert-list].

We propose the AlertFp algorithm for mining frequent patterns with the alerts where they occurred. Mining Fps with alerts on an IDS log is an important goal of this algorithm, where we link all frequent patterns to the alert transactions from where they came. Then, the number of frequent patterns found in each transactionis determined. Finally, all transactions in the dataset are re-sorted according to the number of the related frequent patterns. The AlertFp algorithm represents each frequent k-pattern as < Fk1, Alert1k1, Alert2k1, . . . , Alertmk1>, where Fk1is the first frequent k-pattern, and Alertmk1 is the mth Alert of the first frequent k-pattern. Thus, with this AlertFp technique, the data set is scanned to obtain the candidate 1-itemsets

with a list of their alerts. The alerts of each candidate pattern are implemented. Then, the number of each candidate pattern's alerts is equivalent to the support of the pattern. The frequent pattern mining algorithm is applied to past IDS alert logs (AlertFp) in order to discover patterns that describe redundant alerts. The alert weight is measured by calculating the Frequent Pattern Outlier Factor (FPOF) for each alert's transaction.

$$\text{FPOP (t)} = \frac{X \in t, X \in FPS \sum (D, minisupport) support(X)}{\| FPS(D, minisupport\ t \|}$$

The interpretation of the above formula follows[13]. If a transaction t contains more frequent patterns, its FPOF value will be large, which indicates that it is unlikely to be an outlier.

In contrast, transactions with small FPOF values are likely to be outliers or to be considered as an interesting alert to be investigated by the security analyst.

By using $X \in t,\ X \in FPS \sum (D, minisupport) support(X)$ and re-ordering the IDS alerts by the FPOF for simplicity, we have the important alerts on the top of IDS log and irrelevant alerts pushed to the end of the log file.

Algorithm 1. (Alert:Computing Frequent Patterns with Alerts)
Algorithm AlertFp()
Input: A list of k-items, Alert Set of k-Alerts, mini-support s.
Output: A list of frequent patterns Fps and the relative Alert.
Begin
1. Scan the Data Set once to compute
2. Compute frequent pattern F1 from candidate k-itemsets
C1 as F1 = {list of k itemset with Alertslist count ≥ minsupport , Alert1counter}.
3. For Fi < k i=1 m=0 Counter=0 do
Begin
3.1. If Fi € Alertmi then counter(m)++
3.2. i = i+1, m=m+1
3.3. Compute the next candidate set Ci+1 as F1
End

4. CASE STUDY

Snort[14] used an IDS sensor package that applies attack signatures for detecting suspicious network traffic and can emit alerts as syslog. Consider the below Snort sample (figure 1). This sample will be used to clarify the idea.

```
7 1 508 WEB-MISC/doc/access 25 2 6/11/2010 8:57 AM 1136881320
2148203530 6 46,865 80
7 2 508 WEB-MISC/robots.txt/access 25 2 6/11/2010 8:57 AM
3632363311 2148203629 6 34,074 80
7 3 508 WEB-MISC/robots.txt/access 25 2 8/11/2010 8:59 AM
3632363313 2148203229 6 34,075 80
```

Figure 1. Snort sample

The frequent patterns discovered from the sample IDS log are shown in Figure 2.

```
*****,t1,t2,t3                                                    Support:3
****25,t1,t2,t3                                                   Support:3
7***(25),t1,t2,t3                                                 Support:3
(7)***(25)2,t1,t2,t3                                              Support:3
(7)*508*(25)(2),t1,t2,t3                                          Support:3
(7)*(508)*(25)(2)****6,t1,t2,t3                                   Support:3
(7)*(508)*(25)(2)****(6)*80,t1,t2,t3                              Support:3
(7)*(508)WEB-MISC/robots.txt/access(25)(2)****(6)*80,t1,t2,t3     Support:3
(7)*(508)*(25)(2)*8:57AM**(6)*(80),t1,t2                          Support:3
7)*(508)*(25)(2)6/11/2010(8:57AM)**(6)*(80),t1,t2                 Support:3
```

Figure 2. Sample alert patterns

Finally, the alerts are sorted in ascending order according to their simple FPOF. The alerts are sorted in ascending order according to their weight (FPOF) as shown in Figure 3.

```
Simple FPOFt(3)=7----7 3 508 WEB-MISC/robots.txt/access 25 2 8/11/2010
8:59AM 3632363313 214203229 6 34,075 80
Simple FPOFt(1)=8----7 1 508 WEB-MISC/robots.txt/access 25 2 6/11/2010
8:57AM 1136991320 2148203530 6 46,865 80
Simple FPOFt(2)=9----7 2 508 WEB-MISC/robots.txt/access 25 2 6/11/2010
8:57AM 3632363311 2148203629 6 34,074 80
```

Figure 3. Output sample

5. IMPLEMENTATION AND PERFORMANCE

In this section, we describe our classifier implementation and experiments. In our setup, alerts are sorted in a new separate log file for further review. Classifiers are rebuilt every day at midnight using the IDS sensor log data. Once the frequent pattern has been detected, it is used for further alert classification. This allows for the classifier to adapt to new routine alert patterns with a reasonable learning time. The outlier factor is calculated for each transaction, and then the transactions are re-sorted accordingly. In our experiments, we apply five artificial hacks from a specific source IP to be monitored. Table 2 presents our experimental results on the June 22, 2010 sample (with 28,670 records).

Table 2. Experimental results

mini-support	frequent itemsets	attempted 5 attacks place	reduction
2	101101522	first 7 records	99.975 %
4	32589268	first 24 records	99.916 %
6	23664252	first 34 records	99.882 %

During the experiments, we measured the system reliability and accuracy (figure 4) for different support values comparable with the original attempted attacks and their place in the output file.

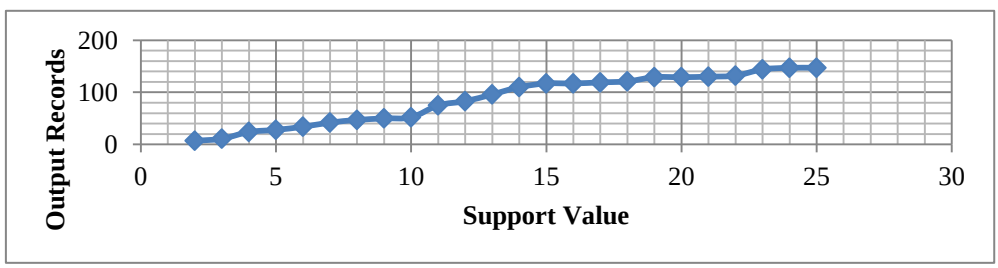

Figure 4. Mini support value vs. the 5 attacks in output

6. OPEN ISSUES AND FUTURE WORK

In this paper, we present a novel data mining based IDS alert classification method sorted for security analysts according to alert importance. Although our preliminary results are promising, one issue remains open: major changes in the arrival rate of routine alerts might be symptoms of large scale attacks but are hard to detect. However, this is an inherent weakness of alert classification and sorting systems

(e.g., see J. Viinikka' [15, 16] for a related discussion). For future works, we plan to research our classification method further and study various statistical algorithms (e.g., time series analysis) for detecting unexpected fluctuations in the arrival rates of routine alerts.

7. REFERENCES

[1] R. Vaarandi, Real-time classification of IDS alerts with data mining techniques. *Paper presented at the IEEE Military Communications Conference (MILCOM* 2009), Boston, MA, October 18-21, 2009. http://dx.doi.org/10.1109/MILCOM.2009.5379762.

[2] J. Viinikka, H. Debar, L. Mé, A. Lehikoinen, and M. Tarvainen, Processing intrusion detection alert aggregates with time series modeling. *Information Fusion Journal*, 10(4), p312-324. 2009. http://dx.doi.org/10.1016/j.inffus.2009.01.003.

[3] J. Long, D. Schwartz, and S. Stoecklin, Distinguishing false from true alerts in snort by data mining patterns of alerts. *Paper presented at the International Society for Optical Engineering*, Kissimmee, Florida, USA, April 17-18, 2006. http://dx.doi.org/10.1117/12.665211.

[4] K. Julisch,and M. Dacier, Mining intrusion detection alarms for actionable knowledge. *Paper presented at the ACM SIGKDD Knowledge Discovery and Data Mining Conference*, Seattle, Washington, USA, August 22-25, 2004. http://dx.doi.org/10.1145/775047.775101.

[5] K. Julisch, Clustering intrusion detection alarms to support root cause analysis. *ACM Transactions on Information and System Security*, 6(4), p443-471, 2003. http://dx.doi.org/10.1145/950191.950192.

[6] J. Viinikka, H. Debar, L. Mé, and R. Séguier, Time series modeling for IDS alert management. *Paper presented at the ACM Symposium on Information, Computer and Communications Securit*y, Taipei, Taiwan, March 21-24, 2006. http://dx.doi.org/10.1145/1128817.1128835.

[7] S.O. Al-Mamory, H. Zhang, and A.R. Abbas, IDS alarms reduction by data mining. *Paper presented at the IEEE World Congress on Computational Intelligence*, Hong Kong, China, July 1-6, 2008. http://dx.doi.org/10.1109/IJCNN.2008.4634307.

[8] S.O. Al-Mamory,and H. Zhang, Intrusion detection alarms reduction by root cause analysis and clustering. *Computer Communications*, 32(2), p419-430, 2009. http://dx.doi.org/10.1016/j.comcom.2008.11.012.

[9] K. Hätönen, M. Klemettinen, H. Mannila, P.Ronkainen, and H. Toivonen. Mining Databases: Towards Algorithms for Knowledge Discovery. *Paper presented at the International Conference on Data Engineering*, New Orleans, LA, February 26-March 1, 1996.

[10] C. Clifton, and G. Gengo, Developing custom intrusion detection filters using data mining. *Paper presented at the 21st Century Military Communications Conference*, Los Angeles, California, October 22-25, 2000. http://dx.doi.org/10.1109/MILCOM.2000.904991.

[11] F. Bodon, A fast APRIORI implementation. *Paper presented at the IEEE ICDM Workshop on Frequent Itemset Mining Implementations*, Canada, August 10, 2000.

[12] W.A. Kosters, and W. Pijls, Depth-first implementation for APRIORI algorithm. *Paper presented at the CEUR Workshop*, Malaga, Spain, October 9, 2001.

[13] Z. He, X. Xu, J.Z. Huang, and S. Deng, FP-outlier frequent pattern based outlier detection. *Computer Science and Information System*, 2(1), p103-118, 2005. http://dx.doi.org/10.2298/CSIS0501103H.

[14] Snort, Retrieved on June 24, 2008, from http://www.snort.org/.

[15] J. Viinikka, H. Debar, L. Mé, A. Lehikoinen, and M. Tarvainen, Processing intrusion detection alert aggregates with time series modeling. *Information Fusion Journal*, 10(4), p312-324, 2009. http://dx.doi.org/10.1016/j.inffus.2009.01.003.

[16] J. Viinikka, and H. Debar, Monitoring IDS background noise using EWMA control charts and alert information. *Recent Advances in Intrusion Detection, Lecture Notes in Computer Science*, 3224, p 166-187, 2004. http://dx.doi.org/10.1007/978-3-540-30143-1_9.