Experimental Studies Using Median Polish Procedure to Reduce Alarm Rates in Data Cubes of Intrusion Data

Jorge Levera[1], Benjamin Barán[2], and Robert Grossman[1]

[1]University of Illinois at Chicago, Chicago, IL, USA 60612
jlevera@cs.uic.edu, grossman@uic.edu
http://www.uic.edu/
[2] Centro Nacional de Computación, Universidad Nacional de Asunción,
San Lorenzo, Paraguay
bbaran@cnc.una.py
http://www.cnc.una.py

This is a draft of the article: Jorge Levera, Benjamin Barán, and Robert Grossman, Experimental Studies Using Median Polish Procedures to Reduce Alarm Rates in Data Cubes of Intrusion Data, Intelligence and Security Informatics for National and Homeland Security, Hsinchun Chen, Reagan Moore, Daniel Zeng, John Jeavitt, editors, LNCS 3073, Springer Verlag, New York, 2004, pages 482-491.

Abstract. The overwhelming number of alarms generated by rule-based network intrusion detection systems makes the task of network security operators ineffective. Preliminary results on an approach called EXOLAP shows that false positives alarms can be avoided by detecting changes on the stream of alarms using a data cube and median polish procedure. A data cube aggregates alarms by hierarchical time frames, rule number, target port number and other feature attributes. The median polish procedure is used on materialized relational views of the data cube to detect changes on the stream of alarms. EXOLAP shows promising results on labeled and unlabeled test sets by focusing on exceptions on the normal stream of alarms, diverting the attention away from false positives.

1 Introduction

Given the proliferation of valuable assets on the Internet, it has become clear that network security operators are looking for robust intrusion detection systems (IDS) that are efficient, effective and easy to manage. A popular approach to network intrusion detection is rule-based intrusion detection systems (RBIDS) [1].

Roughly speaking, RBIDS inspect network packets passing through the network and compare them with a set of rules. A match triggers an alarm. RBIDS are very effective against known attacks and efficient when the set of rules is kept to a reasonable size [1, 2]. Their main drawbacks are the impossibility of detecting unknown attacks and the overwhelming amount of alarms that could be generated [2, 3, 4].

A great number of alarms generated by RBIDS are false positives [2, 3, 4] (i.e., attack did not actually take place). False positives alarms could be reduced by tuning the IDS or by eliminating the rules that causes the noise. Sometimes, it is difficult to apply any of those changes because either the IDS belongs to another organization (e.g., outsourcing, cooperative distributed IDS) or it is not safe to delete a specific rule. Then, the security operator is faced with the problem of detecting true positive alarms among a pile of false positives.

Several approaches have been proposed to reduce the number of alarms. Solutions were proposed from the realm of data mining, machine learning and visualization. The approach studied in this paper is based on Exploratory Data Analysis (EDA)[5] and On-line Analytical Processing (OLAP) techniques, and is called EXOLAP (EXploratory OLAP).

EDA and OLAP techniques do not make assumptions about the data, and they are particularly effective when it is not known a priori what is being sought within the data [1, 2]. Experimental results show that EXOLAP can detect changes on the trend of alarms by focusing on exceptional data. Generalized alarms could be generated to turn the operator's attention towards the interesting part of the data instead of a pile of false positives.

EXOLAP is based on the progressive aggregation of alarms in multiple summarized views of a time-related data cube [6]. The median polish procedure [5] is used on two-dimensional views of the data cube to detect changes on the stream of alarms. A predictable and manageable number of generalized alarms is generated to help network security operators focus on the most interesting data first.

The rest of the paper is organized as follows: Section 2 surveys related work on alarm reduction. Section 3 presents the EXOLAP approach to alarm reduction. Section 4 shows experimental results. Section 5 discusses future work and concluding remarks.

2 Related Work

Erbacher and Sobylak [2] use exploratory data visualization tools to improve the work of forensic analysts. Vert et al. [7] propose a geometric approach to help in the analysis of large amounts of audit data. Their work is very similar to EXOLAP because it aids the analysis process by exploratory means. However, their set of tools focuses more on visualization and raw audit data whereas EXOLAP focuses on real-time alarm reduction using EDA techniques.

Lee and Stolfo [8] apply several data mining techniques to system and network features in order to learn their normal behavior. Following Lee and Stolfo's ideas, Manganaris et al. [3] use alarm features to characterize the normal stream of alarms using association rules. Ye and Li [9] use clustering and classification on alarm features. Most of the previous approaches require a carefully selected training set. Portnoy et al. [10] tackle this problem.

Julisch and Dacier [11] use a conceptual clustering technique that reduces alarms. In a later work, Julish [12] uses clustering to find the root of most false positives. Shortcomings of their approaches include periodic tuning to adjust the model to changing network conditions, and the numerous parameters that are not trivial to determine [1].

Lately, several authors have proposed techniques to correlate alarms. Cuppens and Miege [13] cluster, merge and correlate alarms in a cooperative IDS environment. Ning et al. [14] build attack scenarios. They correlate alarms by partial match of prerequisites and consequences of attacks. Correlation condenses alarms to a few groups and facilitates the distinction between false and true positives. However, prerequisites and consequences conditions are trivial to find.

3 The EXOLAP Approach to Alarm Reduction

RBIDS may generate thousands of alarms per hour. For obvious reasons, network security operators are unable to look at them individually. Instead, they need a way to quickly filter false positives and focus on real threats. Using EXOLAP, operators receive a small number of hinds pointing towards interesting segment of the alarms generated in the last few minutes.

In EXOLAP a data cube aggregates raw alarms generated by RBIDS. A data cube is a data abstraction that allows one to view aggregated data from a number of perspectives. We refer to the dimension to be aggregated as the *measure* attribute, while the remaining dimensions are known as the *feature* attributes.

EXOLAP uses the number of alarms generated as its measure attribute. The feature attributes are based on the alarms' attributes and the network packets that triggered them. A suggested set of feature attributes is: identifier of the rule that triggered the alarm, time-to-live attribute of the packet, target port number, and at least two time related attributes.

Time related attributes (e.g., T_0, T_1, …, T_K) should be hierarchical. The time attribute at the lowest level, T_0, should be small enough to detect recent attacks and big enough to avoid overloading the database storing alarms. For example, if $T_0=15$ minutes, alarms will be aggregated every fifteen minutes. Time attributes at intermediate levels, say T_i, should include the one below so that aggregated alarms at T_{i-1} could be used to fill the cells of time attribute T_i. Following the previous example, if $T_0=15$ minutes, then the time attribute above, T_1, could be sixty minutes (i.e., $T_1=T_0*4$).

EXOLAP uses data cubes because they are useful in identifying trends [15] and offer a summary of the alarms generated by the RBIDS. Considerations regarding view materialization strategies and efficient implementation of data cubes are beyond the scope of this paper. The authors used some of the ideas found in [15,16].

Figure 1 shows a three dimensional data cube with the corresponding feature attributes. Additional feature attributes can be easily included.

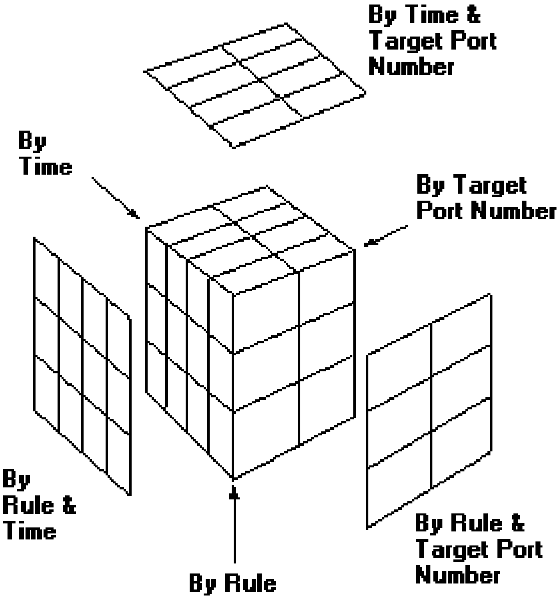

Fig. 1. A three dimensional data cube aggregates the number of alarms by feature attributes (*Rule*), (*Time*), and (*Target Port Number*). Besides, two dimensional tables show the aggregated alarms by (*Rule and Time*), by (*Rule and Target Port Number*) and by (*Time and Target Port Number*)

For simplicity, the following relational views are used in this paper:

- *By rule and time*. Every T_0 minutes, a new column $t_{0,i}$ of alarm frequencies in the database is added to a table with the trend of alarms in the last T_1 minutes at T_0 minutes intervals. This view shows the number of alarms triggered by each rule during each time interval.

Table 1 shows a *Rule and Time* view of alarms generated by a Snort system [17] installed on the Public Sector Metropolitan Area Network of Asuncion, Paraguay. In this case, a fifteen minutes interval is the smallest time attribute (i.e., $T_0=15$ minutes and T_1=60 minutes).

- *By target port number and time*. This view shows the number of alarms at each time interval classified by the target port number of the network packet that triggered the alarm.

- *By time-to-live and time*. This view summarizes the number of alarms triggered at each time interval classified by the time-to-live attribute of the network packet that triggered the alarm. This view is particularly useful to detect denial of service attacks as mentioned in [18] and group several alarms triggered by the same attack.

Table 1. A two dimensional view of aggregated alarms from The Public Sector Metropolitan Area Network of Asuncion. Each column represents a fifteen minutes time interval, and each row the number of alarms triggered by the rule (*Rule ID*) during the corresponding time interval

Rule ID	$t_{0,0}$ 8:45PM	$t_{0,1}$ 9:00PM	$t_{0,2}$ 9:15PM	$t_{0,3}$ 9:30PM
1	138	72	35	21
7	0	0	0	2
10	4	2	4	4
14	1	5	1	0
16	1	0	0	0
17	60	60	60	60
18	29	28	30	28
19	440	451	434	459
22	5	3	5	6
24	0	0	0	1
27	3	0	0	0
29	0	1	0	1
38	0	1	0	0
46	0	1	0	0

Once several relational views of the cube are built, an EDA technique is used to find the most interesting subset of alarms. The following section introduces this technique.

3.1 Median Polish Procedure

EDA techniques search data for *exceptions* or abnormal data values compared to values anticipated by a statistical model [5]. The statistical model tries to approximate the whole set of values and can be considered a good approximation of it. The exceptions found with this method can be used by an analyst of the data as a starting point in the search for anomalies, guiding the analyst's work across a search space that can be otherwise very large [5, 15]. A traditional way of performing EDA is *median polish procedure* (MPP)[5].

MPP fits an additive model by operating on a data table. The algorithm works by alternately removing the row and column medians, and continues until the proportional reduction in the sum of absolute residuals is less than a specified tolerance value or until there has been a maximum of iterations specified. In principle, the process continues until all the rows in each dimension have zero median. MPP finds the *effect* that each row and column has on the model, given by the algebraic sum of the medians that have been subtracted in that row at every step. Besides, MPP provides the *residual* in each cell of the table, which tells how far apart that particular cell is from the value predicted by the model.

To illustrate the procedure, the values $Y=\{ y_{ij}\}$ given in Table 1 are used as input to MPP. The relational view *Rule and Time* can be considered a two-way table of the number of alarms generated per time frame. An *additive model* can express the relationship between time and rule. Equation 1 shows the additive model, where μ is the overall typical value for the whole table, α_i is the row effect of row i, β_j is the column effect of column j, and ι_{ij} is the deviation of y_{ij} from the model (i.e., residual)

$$r_{ij} = \mu + \alpha_i + \beta_j + \iota_{ij} \ . \qquad (1)$$

Table 2 shows the residual values for the two-way table given in Table 1. The exceptional values on Table 1 are found by locating the largest absolute values on the residual table. In Table 2, the cell on the first row and first column has the biggest absolute value.

Table 2. The residual table for the two dimensional view of aggregated alarms in Table 1. Each column represents a fifteen minute time interval, and each row the deviation of the number of alarms triggered by the rule (*Rule ID*) with respect to the model built during the corresponding time interval

Rule ID	8:45 PM	9:00 PM	9:15 PM	9:30 PM
1	84.5	18.5	-18.5	-32.5
7	0.0	0.0	0.0	2.0
10	0.0	-2.0	0.0	0.0
14	0.0	4.0	0.0	-1.0
16	1.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0
18	0.5	-0.5	1.5	-0.5
19	-5.5	5.5	-11.5	13.5
22	0.0	-2.0	0.0	1.0
24	0.0	0.0	0.0	1.0
27	3.0	0.0	0.0	0.0
29	-0.5	0.5	-0.5	0.5
38	0.0	1.0	0.0	0.0
46	0.0	1.0	0.0	0.0

In IDS terms, a big residual value means that there is a significant change in the trend of alarms. For example, the largest deviation value in Table 2 indicates an increase on alarms generated by the rule with id = 1 in the time window between 8:45 PM and 9:00 PM. Particularly, the largest positive value on the last column, 9:30 PM, indicates a recent increase on the normal values on the table.

Small residual values suggest that the stream of alarms of a particular type has been stable and can be considered "normal" noise. By examining the abnormal cells, the operator has reduced the searching space of alarms to the rule ID and time frame indicated. The reduction obtained depends on the frequencies of alarms generated during that time frame. This gives the operator enough flexibility to adapt to different scenarios. Since EXOLAP is exploratory, the final judgement is left to the expert. In addition, generalized alarms could be generated for the top *n* exceptions on the last time interval of a particular view.

The median polish procedure is used on other relational views of the data cube as well. As a result, there are as many residual tables as relational views.

In the following section, relational view construction and MPP are combined and automated to reduce the number of alarms to be inspected.

3.2 Alarm Reduction

The processes of building relational views of the data cube presented in the previous section can be automated. First, a reasonable set of relational views should be chosen. Those views are materialized and updated at predefined time intervals as more alarms are generated by the RBIDS. For example, the *Rule and Time* view given in Table 1 can be considered a moving time window. Every $T_0=15$ minutes, a new column, say $t_{0,i}$, is added with the aggregated alarms seen in the last T_0 minutes. To save space, the oldest $t_{0,j}$-minute interval (where $j<i$) could be aggregated on the next time frame level (e.g., $T_1=60$ minutes).

EXOLAP generates *trend alarms* (t-alarms) at every time interval $t_{l,i}$ of a time feature attribute T_l. Using the Rule and Time view shown in Table 1, t-alarms could be generated every fifteen minutes for each of the top *n* exceptions on that particular view. Similarly, t-alarms are generated for other views of the data cube. With t-alarms, operators can quickly locate potential hazards caused by a sudden increase or decrease on the number of alarms generated by a RBIDS.

The advantage of using t-alarms is that the operator receives a predictable number of alarms pointing towards an interesting subset of raw alarms. The number of t-alarms generated could be set ahead of time to a manageable number. In this way, the operator is not overwhelmed and can start analyzing RBIDS alarms from the most interesting part.

Equation 2 shows how to compute the number of t-alarms per hour. For every time related feature attribute T_l, we multiply the number of relational views (RV_l) involving T_l by the number of exceptional values (n) retrieved from each relational view by the number of intervals T_i included in sixty minutes.

$$\text{t-alarms per hour} = \sum_{i=0}^{k-1} RV_i * n * 60/T_i \qquad \textbf{(2)}$$

In the following section, experiments with EXOLAP show encouraging results on diverse sets of alarms, including labeled and unlabeled datasets of different sizes.

4 Experimental Results

EXOLAP was tested with four sets of alarms generated by Snort IDS version 1.9.1 with default set of rules. One of the systems was located on a Public Sector Metropolitan Area Network of Asuncion, Paraguay. Another set of alarms was generated by a Snort located on the Abilene network [19]. An additional set belongs to an IDS running at the University of Illinois at Chicago (UIC). The last set of alarms was generated using DARPA99 Intrusion Detection Set [20]. Table 3 gives an overview of the datasets used.

Table 3. Datasets used on experimental results. Some characteristics are shown to indicate their diverse nature and evaluate the results on each set

Dataset	Network Type	Number of Alarms	Distinct Rules	Time Span (in days)	Distinct Source IP Addresses	Distinct Target IP Addresses
UIC Network	LAN	592,121	52	180	20,161	2,429
Pub. Sect. MAN	MAN	430,794	67	7	514	272
Abilene Network	WAN	10,357,673	49,571	90	208,527	253,790
DARPA99	Synthetic	23,050	84	10	109	187

Three relational views were used during the experiment: Rule and Time, Time-to-live and Time, and Target-port-number and Time views. The data cube used time attributes T_0=15 minutes and T_1= 60 minutes (i.e., *k=1*). MPP ran every T_0 minutes on all views and t-alarms were triggered for the biggest exception on each view (i.e., *n=1*). The reduction is computed as the number of RBIDS alarms to be examined on the exceptional cell pointed by t-alarms over the total number of RBIDS generated during the same time interval T_0. Table 4 shows the reduction obtained.

Table 4. Average reduction on four datasets divided by Rule and Time view, Time-to-live and Time view, and Target-port-number and Time view

Dataset	Avg. Reduction on Rule-Time view	Avg. Reduction on TTL-Time view	Avg. Reduction on Port-Time view
Public Sector MAN	77.98 %	84.39 %	67.72 %
Abilene Network	80.52 %	85.49 %	78.98 %
UIC Network	46.75 %	74.07 %	41.25 %
DARPA 99	48.69 %	36.88 %	54.77 %

In general, datasets DARPA99 and UIC network do not have many alarms at 15 minute intervals. In order to obtain greater reduction on these datasets, a larger T_0 value was needed.

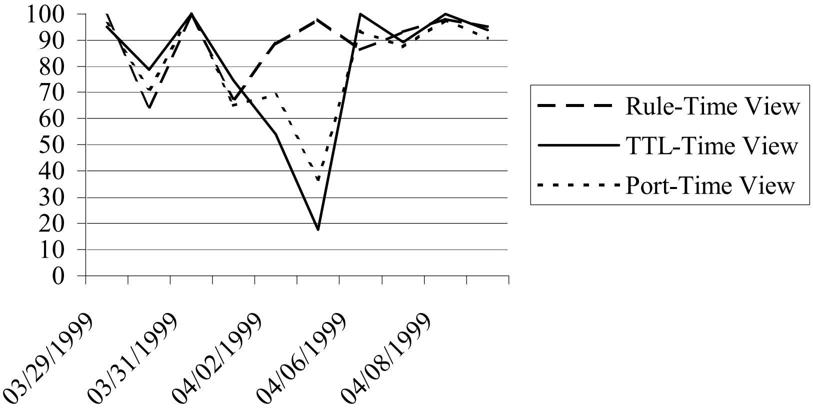

Fig. 2. Percentage of true positives alarms detected on the labeled dataset DARPA99 using EXOLAP. T-alarms were defined for the cell with the largest deviation value during the last time frame

Figure 2 shows the percentage of true positives detected on the labeled dataset DARPA99 using EXOLAP. Most true positives detected were in the subset of raw alarms indicated by t-alarms (i.e., the cell with the biggest residual value). A big drop observed on April 5, 1999 was caused by many attacks taking place on an eleven minute interval and network packets having many different time-to-live and target port number values. In this case, most true positives were among the top five exceptional values indicated by t-alarms. Rule-Time view was not affected and managed to detect most true positives with the top t-alarm.

5 Conclusion and Future Work

Experimental studies with EXOLAP aim at reducing the number of intrusion alarms to be analyzed by network security experts. A multidimensional data cube aggregates alarms by several feature attributes. MPP finds exceptional values or changes on the stream of alarms on two-dimensional views of the data cube. Tests on several datasets show promising reduction on the number of alarms to be examined. In particular, the labeled dataset DARPA99 showed that EXOLAP could improve the effectiveness of network security operators by focusing on the most interesting data first.

In the future, EXOLAP will be tested with an n-dimensional extension of MPP for data cubes proposed by Barbará and Wu [15]. This will reduce the number of t-alarms without significant computational demand, integrating several views into only one global view. Intuitively, many change detection algorithms can also be applied to the data cube. A performance comparison would be appropriate to study complexity and effectiveness of different algorithms.

Further testing needs to be done on labeled datasets like DARPA99 to verify the reduction on the number of false positives versus the percentage of true positives detected by t-alarms.

In addition, a distributed EXOLAP system is being designed to exploit the fact that views can be exchanged in a distributed IDS environment to improve the efficiency and cooperation between composing IDS.

References

1. Eleazar Eskin, Andrew Arnold, Michael Prerau, Leonid Portnoy and Salvatore Stolfo, "A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data," Data Mining for Security Applications. Kluwer 2002.
2. Robert F. Erbacher and Karl Sobylak, "Improving Intrusion Analysis Effectiveness," Workshop on Statistical and Machine Learning Techniques in Computer Intrusion Detection, George Mason University, September 24-26, 2003.
3. Manganaris, Steganos, Christensen, Marvin, Zerkle, Dan and Hermiz, Keith, " A Data Mining Analysis of RTID Alarms," Computer Networks, 34(4) , October 2000.
4. Klaus Julisch, "Mining alarm clusters to improve alarm handling efficiency," In 17th Annual Computer Security Applications Conference (ACSAC), pp 12-21, December 2001.
5. John W. Tukey, *Exploratory Data Analysis*, Addison-Wesley, 1977.
6. J. Gray, A. Bosworth, A. Layman, H. Pirahesh, "Data Cube: A Relational Aggregation Operator Generalizing Group-by, Cross-tabs and Sub-totals," In Proceedings of the 12th Int. Conf. on Data Engineering, pp 152-159, 1996.
7. Greg Vert, Deborah A. Frincke, and Jesse C. McConnell, "A visual mathematical model for intrusion detection" In Proceedings of the 21st National Information Systems Security Conference, Crystal City, Arlington, VA, USA, October 5-8 1998.
8. Wenke Lee and Sal Stolfo. "Data Mining Approaches for Intrusion Detection" In Proceedings of the Seventh USENIX Security Symposium (SECURITY '98), San Antonio, TX, January 1998.
9. Nong Ye and Xiangyang Li, "A Scalable Clustering Technique for Intrusion Signature Recognition," In Proceedings of the 2001 IEEE, Workshop on Information Assurance and Security, United States Military Academy, West Point, NY, 5-6 June, 2001.
10. Leonid Portnoy, Eleazar Eskin and Salvatore J. Stolfo. "Intrusion detection with unlabeled data using clustering" In Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001). Philadelphia, PA: November 5-8, 2001.
11. Klaus Julisch and Marc Dacier, "Mining Intrusion Alarms for Actionable Knowledge," in SIGKDD'02, Edmonton, Alberta, Canada, 2002.
12. Klaus Julisch, "Clustering Intrusion Detection Alarms to Support Root Cause Analysis", in ACM Transactions on Information and System Security 6(4), November 2003.
13. F. Cuppens and A. Miege, "Alert correlation in a cooperative intrusion detection framework," In Proceedings of the 2002 IEEE Symposium on Security and Privacy, May 2002.

14. Peng Ning, Yun Cui and Douglas S. Reeves, “Constructing attack scenarios through correlation of intrusion alerts,” In Proceedings of the 9th ACM conference on Computer and communications security, Washington, DC, USA, 2002.
15. Venky Harinarayan, Anand Rajaraman, and Jeffrey D. Ullman, “Implementing data cubes efficiently,” In Proceedings of the ACM SIGMOD '96, pp 205-216, Montreal, June 1996.
16. Daniel Barbara and Xintao Wu, “Using approximations to scale exploratory data analysis in datacubes," Proceedings of the ACM SIGKDD International Conference, August 1999
17. M. Roesch, "Snort - lightweight intrusion detection for networks," in Proceedings of Thirteenth Systems Administration Conference (LISA '99), pp. 229--238, The USENIX Association, Berkeley, California, 1999.
18. Alefiya Hussain, John Heidemann and Christos Papadopoulos, “A framework for classifying denial of service attacks,” In Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications, pp 99-110, Karlsruhe, Germany, August 25-29, 2003.
19. Advanced Network Management Lab, *The Abilene Project*, University of Indiana, Bloomington, Indiana, USA.
20. Lincoln Laboratory, Massachussets Institute of Technology, DARPA 99 Intrusion Detection Data Set Attack Documentation. [Online] Available: http://www.ll.mit.edu/IST/ideval/docs/1999/attackDB.html.