A Data Mining Analysis of RTID Alarms

Stefanos Manganaris Marvin Christensen Dan Zerkle Keith Hermiz

International Business Machines Corporation
19 Lakehurst Court, Research Triangle Park, NC 27713

{stefanos,marvinc,dzerkle,khermiz}@us.ibm.com

Abstract

IBM's Emergency Response Service provides real-time intrusion detection (RTID) services through the Internet for a variety of clients. As the number of clients increases, the volume of alerts generated by the RTID sensors becomes intractable. This problem is aggravated by the fact that some sensors may generate hundreds or even thousands of innocent alerts per day. With an eye towards managing these alerts more effectively, IBM's data mining services group analyzed a database of RTID reports. The first objective was an approach for characterizing the "normal" stream of alerts from a sensor. Using such models tuned to individual sensors, we then developed a methodology for detecting anomalies. In contrast to many popular approaches, the decision to filter an alarm out or not takes into consideration the context in which it occurred and the historical behavior of the sensor it came from. Our second objective was to identify all the different profiles of our clients. Based on their history of alerts, we discovered several different types of clients, with different alert behaviors and thus different monitoring needs. We present the issues encountered, solutions, and findings, and discuss how our results may be used in large-scale RTID operations.

1 Introduction

Organizations collect huge volumes of data from their daily operations. This wealth of data is often under-utilized. Data mining is a process of drilling through large amounts of data to discover hidden key facts that can drive decision making. Data mining helps companies reap rewards from their data warehouse investments, by transforming data into actionable knowledge, revealing relationships, trends, and answers to specific questions that are too broad in nature for traditional query and reporting tools.

Knowledge discovery and data mining is a relatively young, interdisciplinary, field that cross-fertilizes ideas from several research areas, including machine learning, statistics, databases, and data visualization. With its origins in academia about ten years ago, the field has recently captured the imagination of the business world and is making important strides by creating knowledge discovery applications in many business areas, driven by the rapid growth of on-line data volumes. Fayyad, *et al.*, [1] presents a good, though somewhat dated, overview of the field. Bigus [2] and Berry and Linoff [3], among others, have written introductory books on data mining that include good descriptions of several business applications.

Commercial intrusion-detection systems often generate mountains of data. However, most large-scale enterprise operations use fairly simple filters to screen alarms in order to cope with their sheer volume; little else is usually done with this data. Emergency Response Services and the Knowledge Discovery Consulting practice at IBM work together in a pilot project to reveal the value hidden deep inside these databases of alarms.

The first phase of our plan, described in this paper, focused on two objectives. First, we sought to improve anomaly detection. In contrast to common approaches that analyze alarms in isolation, we felt that alarm context could enhance decision making. Moreover, we wanted to develop an adaptive approach where the system would learn to detect anomalies based on its experience from the history of alarms. We expand on that work in section 2 of this paper. Our second objective was to understand the clients we serve better from the perspective of alarm histories they tend to generate. That work is described in section 3.

2 Data Mining for Anomaly Detection

IBM provides real-time intrusion detection services to clients world-wide. Commercially available sensors, such as NetRanger from Cisco Systems, are deployed on customer networks. All intrusion alarms are sent over the Internet to IBM's Network Operations Center (NOC) in Boulder, Colorado, which provides 7×24 first-level monitoring. The database of alarms in Boulder is one of the largest known collections of intrusion data and has alluring potential for insights into intrusion behaviors. Operators at the NOC deal with thousands of incoming alarms from each sensor every day, using sophisticated filtering and summarization tools to determine in real-time the extent and source of potential attacks. Even though these tools perform admirably, success currently depends critically on careful hand-crafting of the filtering and summarization rules. As the number of sensors increases, the data volume rises, and this task becomes harder to keep up with. By necessity, most manually crafted rules are fairly simple, placing a lot of weight on priority levels statically pre-assigned to different alarm types. Most of the rules do not correlate alarms and do not take into account the context in which alarms are raised or the historical behavior of the sensor raising the alarm.

Our long-term vision for NOC is illustrated in Figure 1. Operators are assisted by an automated decision engine, which screens incoming alarms using a knowledge-base of decision rules, which is updated with the assistance of a data mining engine that analyzes historical data and feedback from incident resolutions. Taking the first steps in that direction, we wanted to investigate whether the "normal" stream of alarms, generated by sensors under conditions not associated with intrusion attacks, can be characterized. Moreover, given such models of normal behavior, we wanted to develop a method that improves incident detection while reducing the volume of false positive alarms and thus also operator workload.

The basic idea of our approach is simple: frequent behavior, over extended periods of

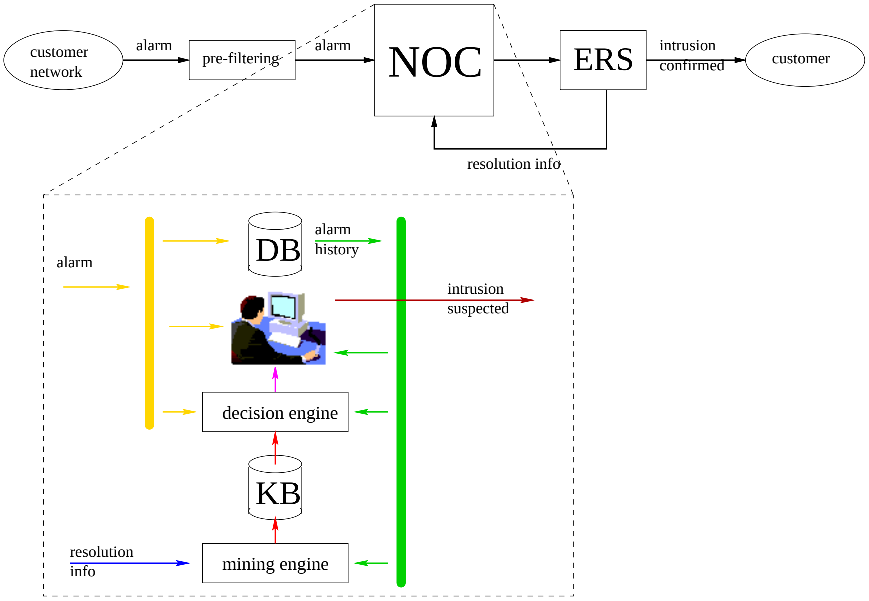

Figure 1: Vision for Network Operations Center

time, is likely to be normal. A combination of specific alarm types occurring within seconds from each other, always in the same order, every few minutes, for an extended time period, is most likely less suspicious than a sudden burst of alarms never seen before.

We used association analysis, in IBM's Intelligent Miner for Data toolkit [4], to discover all frequent sets of alarms. The problem of mining association rules was introduced by Agrawal, *et al.*, in [5]. The input consists of a set of supermarket transactions, where each transaction is a set of literals (called items). An example of an association rule is: "30% of transactions that contain beer also contain diapers; 2% of all transactions contain both of these items." Here 30% is called the *confidence* of the rule and 2% is the *support* of the rule. The problem is to find all association rules that exceed user-specified minimum support and minimum confidence thresholds. Common approaches decompose the problem into two subproblems: (i) find all combinations of items that have transaction support above minimum support; call those combinations *frequent itemsets*. A combination of items X has support s in the transaction set Y, if s% of the transactions in Y contain X. (ii) use frequent itemsets to generate the desired rules. The idea is that if, say, $ABCD$ and AB are frequent itemsets, then $AB \rightarrow CD$ is an association rule of the solution iff the ratio of ABCD support over AB support exceeds the minimum confidence threshold. The first subproblem is computationally more demanding and has been the focus of considerable work on developing fast algorithms (e.g., Agrawal, *et al.*, [6], Brin, *et al.*, [7]).

We characterized normal alarm behavior in terms of frequent itemsets and association rules as follows. First, the continuous stream of alarms was partitioned into bursts of alarms. Each burst corresponds to a transaction, in the lingo of association analysis, and items refer to alarms. Alarm bursts were identified by larger than average interalarm times at their start and end points[1]. Frequent itemsets refer to combinations of alarms that tend to occur often within bursts. Association rules relate the occurrence of one set of alarms with another in the same burst. The model of normal behavior for a sensor consisted of the collection of frequent itemsets and association rules with high confidence. The minimum support and confidence thresholds were chosen empirically; more on this later.

In this work, the ordering of alarms within bursts was not modeled. The same approach, however, can be taken using sequential pattern analysis [8, 9]. Sequential patterns refer to frequent alarm sequences rather than frequent sets. We believe there are cases where ordering over time is of significance, and we are currently investigating an extension in that direction.

The algorithm for detecting deviations from normal behavior is shown in Figure 2. Given a set of frequent itemsets, high confidence rules, and an incoming burst of alarms, we first checked whether the set of alarms is a known frequent itemset. In that case, this is likely an innocent set of alarms, with expected frequency of occurrence equal to the itemset's support. Failing that, we identified the set of most specific supported itemsets M. A frequent itemset is supported by a burst of alarms, when it is a subset of the alarms in the burst. A supported frequent itemset is *most specific* when no other frequent itemset that is a superset of it is also supported. The set M shows combinations of alarms that are known to occur, independently of each other, frequently within bursts, but are not known to co-occur. This is one type of an anomaly where the context in which alarms occur is of importance. The set D contains any alarms not covered by any of the patterns in M. When D is not empty, we have a burst that contains alarms that occur with frequency lower than the minimum support threshold in any context.

Another type of anomaly involves the unexpected absence of alarms. This is where association rules come into play (cf. check_rules in Figure 2). Any high-confidence rule $A \rightarrow C$ such that A is supported in burst B while C is not, is indicating that the set of alarms C, which was known to frequently occur in the particular context, is unexpectedly missing. A measure of interestingness for this anomaly is the probability of the event, $1 - \text{confidence}(A \rightarrow C)$.

To set the minimum support threshold, take into consideration the frequency of any known innocent alarms. The higher the threshold, the fewer the frequent itemsets, and thus the fewer the alarm bursts that we can assign a precise estimate on the frequency of

[1] We are investigating a more sophisticated approach based on box counting, often used for computing fractal dimensions in the analysis of dynamic systems, which avoids the need for an arbitrary threshold on interalarm times.

```
input:    model of normal sensor behavior (frequent sets $F$ and rules $R$)
          stream of alarm bursts from sensor
output:   alarm bursts that deviate from normal behavior

LOOP
          $B$ = read_next_burst_of_alarms
          IF ($\exists s \in F : s = B$) THEN
              check_rules($R$), report_frequent($B$)
          ELSE
              $M$ = most_specific_supported_itemsets($B, F$)
              $D = B - \cup_{m_i \in M} m_i$
              check_rules($R$), report_infrequent($B, M, D$)
          END
END
```

Figure 2: Algorithm for Anomaly Detection

occurrence. More and more bursts would be flagged as anomalies, raising the rate of false positives while lowering the rate of false negatives. Regarding the minimum confidence threshold, higher values produce fewer, more confident, rules. Broken high-confidence rules are more interesting than broken low-confidence rules. The higher the confidence threshold, the fewer bursts are flagged as anomalies because of broken rules, lowering the rate of false positives and raising the rate of false negatives.

We tested these ideas off-line on two NetRanger sensors with good results, and currently plan an on-line evaluation with a more rigorous methodology. NetRanger is a misuse detection system that works by comparing network traffic against a database of signatures of known behavior that may indicate an attack. Upon spotting such behavior, NetRanger generates an alarm, showing the source and destination address of the network traffic that caused it, the type of alarm, and the time and date issued. Each alarm type is given a priority level from one to five; high level alarms are more likely to be of high concern.

For the preliminary results we report here, we used two weeks worth of data for training normal behavior models, and tested on the following week. For one of the sensors, indicatively, we had 393 thousand alarms to train on, with 107 distinct alarm types, in about nine thousand bursts of between one and 47 distinct alarm types per burst. The derived model of normal behavior consisted of about 600 frequent itemsets and 850 rules (minimum support of 1% and confidence of 98%). During testing, we processed 170 thousand alarms and flagged 314 anomalies of various types. After focusing on anomalies that (i) have low support and (ii) contain alarms not frequently occurring in the context of the flagged burst, their number is reduced significantly, to about ten per day per sensor on average. We compared this reduced number of anomalies with the incidents recorded on the call-log maintained by the operators at the NOC, and found that (i) we had detected each of the incidents recorded on the call-log, and (ii) we had flagged an additional three to eight anomalies per sensor per day that had gone undetected at the NOC, when they could have provided earlier warning and increased robustness.

Association analysis has been used previously for intrusion detection. Lee *et al.* [10] present a data mining framework for inducing concise and intuitive classification rules that can detect intrusions. At the core of their framework is a classifier that can be trained to discriminate between normal and other intrusion behaviors. The success of a classifier system in this task depends critically on (i) having sufficient data to cover the behaviors of interest and (ii) on engineering the right set of features to describe instances of behavior. In their approach, association rules and frequent episodes are computed from audit data as the basis for guiding the audit data gathering and feature selection processes.

Wespi *et al.* [11] also present a behavior-based approach for intrusion detection. The system looks for patterns in audit data, which are then used as models of normal behavior. Patterns are induced by the Teiresias algorithm [12]. After training, a pattern matching algorithm compares the observed behavior to the stored patterns. When the quality of the match deteriorates, a deviation is flagged. Differences between the two works exist in (i) what constitutes a pattern and how they are derived and (ii) how patterns are used for intrusion detection. Teiresias analyzes strings over characters of an alphabet and identifies all rigid patterns that repeat at least K times, for some value of K. Patterns are guaranteed to be maximal in both length and composition. Patterns are strings that may contain an arbitrary combination of “don’t care” characters. As applied for anomaly detection, Teiresias analyzed the sequence of system calls generated by a Unix process. Common invocations of the same process exhibit certain patterns in the sequence of calls. Intrusions are assumed to exercise abnormal paths in the executable code and the sequence of calls does not match the expected patterns.

3 Sensor Profiling

Each sensor has its own history of alarm behavior; sensors vary in terms of alarm types, alarm rates, distribution of alarms over day-of-week and time-of-day, etc. Can sensor behaviors can be assigned to general sensor profiles? The question we set out to explore was whether clients clustered into natural groupings based on the similarity of alarm histories from their sensors. Moreover, we wanted to profile the various types of alarm behavior and corresponding segments of customers.

Our motive behind this work is deeper understanding; insights in the behaviors of alarms, sensors, computer networks, customers, industries, and geographies. If indeed there are distinct segments of alarm behavior, we can much more easily put our arms around the set of sensors monitored and begin tailoring servicing strategies at the segment-level rather than the individual sensor-level. Moreover, once segments have been profiled, one can investigate factors that may affect alarm behavior. For example, is industry or geography a factor? In what ways do they affect alarm behaviors? Insights regarding behavior, often lead to insights regarding customer needs and thus opportunities for differentiation from the competition.

In marketing, the partitioning of a population of customers based on criteria that discriminate their wants and needs is called *market segmentation*. There are several examples in the knowledge discovery literature where mining techniques have been applied to this problem with good results. We are not aware, however, of related prior work in the segmentation of customers based on the alarm histories of their intrusion detection sensors.

We illustrate our work on a sample of 27 NetRanger sensors corresponding to 25 unidentified clients. We monitored these sensors for a period of about one month and collected roughly 12×10^6 alarms, corresponding to about 2GB of data. Each alarm was described in terms of its type, priority level, time stamp, source and destination IP address and port, and customer and sensor ID.

The stream of alarms from each sensor was processed to generate a summary vector of its behavior over time that included the following attributes: (i) alarm volumes and rates by priority level and across levels, (ii) ratios of alarm volumes by level, (iii) aggregate properties of the interalarm times (such as min, max, median), (iv) number of distinct alarm types encountered by level and across levels, (v) ten top most frequent alarm types by level and across levels and corresponding relative frequencies, (vi) volume ratios by day of week, (vii) volume ratios by time of day, (viii) number of distinct IP ports and addresses and ten top most frequent IP ports and addresses, encountered as source, destination, and source/destination pairs, (ix) entropies of the probability distributions for time-of-day, day-of-week, source and destination port, address, and source/destination pair. Each entropy attribute provides an indication of how random the corresponding distribution is. There are no deep principles to justify this choice of attributes; we felt, however, they provided good summaries of alarm histories at the sensor level.

The population of summary vectors was segmented based on various attributes using *demographic clustering* in IBM's Intelligent Miner for Data toolkit [4]. This is an iterative algorithm that makes multiple passes over the set of vectors before converging to a locally optimal clustering. The quality of a partitioning is assessed by a global measure, called the *Condorset criterion*, which favors clusterings with high intraclass vector similarity and low interclass vector similarity. In other words, it favors clusters that contain similar vectors when, at the same time, vectors assigned to different clusters are dissimilar. At each step of the process, the algorithm uses the Condorset criterion to decide whether to assign a vector into an existing cluster, or whether to create a new one. The process ends when an iteration results in no changes to the clustering. In contrast to many popular techniques, demographic clustering does not require an *a priori* specification of the number of clusters to produce. It can also easily handle mixed numeric and categorical data.

Our results indicated that our sensors fell indeed into a small number of categories with distinct characteristics. A segmentation based on sensor location and alarm volumes, rates, and variety showed one cluster encompassing about 78% of the sensors, corresponding to the "average" well-tuned sensor on a typical corporate network looking at Internet or DMZ traffic. More interestingly, it also showed four other smaller segments. One corresponds to sensors looking at intranet traffic, one grouped together the sensors of a particular client, suggesting a very idiosyncratic network, and another seems to have included sensors with behavior that could improve by tuning, as indicated by the unusually high rate of low level alarms that discriminated that segment from the rest.

The demographic clustering algorithm assigns sensors to clusters with a certain degree of fitness. Low degrees of fitness indicate unusual behavior compared to peer sensors. Closer examination of sensors that stood out from the rest in our example revealed more opportunities for tuning.

Profiling segments in terms of alarm volume by time-of-day and day-of-week shed light on the profiles of demand placed on our central monitoring center by various types of sensors. This is an example where the results of segmentation can lead into a refinement of servicing strategy.

Another benefit we are currently investigating is the incorporation of a frame of reference in our reports to customers. A segment's profile can serve as frame of reference, against which we can contrast the behavior of a customer's sensors. Rather than present alarm statistics and trends for a customer in isolation, we anticipate presenting them in the context of corresponding information from the unidentified customer's peers. Such contrasts can be used to justify recommendations for changes in security-related areas.

4 Conclusions

We have presented a new approach for dealing with some of the problems of intrusion detection. Some systems use pattern matching for misuse detection while others use anomaly detection [13]. Both approaches have their advantages and disadvantages. In this paper, we have attempted to combine these approaches, by performing anomaly detection on the voluminous results produced by a misuse detection system. We have found our preliminary results encouraging: normal alarm behavior models allowed us to understand complex security-related activity in a computer network and, more importantly, allowed an automated system to ignore the enormous volume of uninteresting alarms.

Our analysis of the profiles of different IBM customers' alarm traffic also produced interesting results. It showed that the behaviors of sensors on different networks varied, but still did exhibit commonalities. The sensor profiles we induced will allow us to present our customers with useful comparative insights. The demonstration of the differences in behavior provides proof that each customer must have his monitoring service custom-tuned in order to provide the best results.

Our next step is to integrate our anomaly detection technique into a real-time system so that results can be made available immediately to the Network Operations Center and IBM's customers. This will require adapting an infrastructure to direct the alarms and analysis results to appropriate components. It will also require the development of a dynamic retraining system, to deal with the changes in behavior that will occur over time as the monitored networks change.

References

[1] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery: An overview. In Fayyad et al. [14], chapter 1.

[2] Joseph P. Bigus. *Data Mining with Neural Networks.* McGraw-Hill, 1996.

[3] Michael J. A. Berry and Gordon Linoff. *Data Mining Techniques: For Marketing, Sales, and Customer Support.* John Wiley & Sons, 1997.

[4] IBM. *Intelligent Miner for Data: User's Guide*, 1996.

[5] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proc. of the ACM SIGMOD Conf. on Management of Data*, pages 207–216, 1993.

[6] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast discovery of association rules. In Fayyad et al. [14], chapter 12, pages 307–328.

[7] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. In *Proc. of the ACM SIGMOD Conf. on Management of Data*, 1997.

[8] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proc. of the Intl. Conf. on Data Engineering (ICDE)*, Taipei, Taiwan, March 1995.

[9] H. Mannila, H. Toivonen, and A. I. Verkamo. Discovering frequent episodes in sequences. In *Proc. of the First Intl. Conf. on Knowledge Discovery & Data Mining*, Monrteal, Canada, August 1995.

[10] Wenke Lee, Salvatore J. Stolfo, and Kui W. Mok. Mining audit data to build intrusion detection models. In *Proc. of the Fourth Intl. Conf. on Knowledge Discovery & Data*

Mining, pages 66–72, New York, NY, August 1998. American Association for Artificial Intelligence.

[11] Andreas Wespi, Marc Dacier, Hervé Debar, and Mehdi M. Nassehi. Audit trail pattern analysis for detecting suspicious process behavior. In *Proc. of the First Intl. Workshop on Recent Advances in Intrusion Detection*, Louvain-la-Neuve, Belgium, September 1998.

[12] Isidore Rigoutsos and Aris Floratos. Motif discovery without alignment or enumeration. In *Proc. of the Second Annual ACM Intl. Conf. on Computational Molecular Biology (RECOMB '98)*, New York, NY, March 1998.

[13] B. Mukherjee, L. T. Heberlein, and K. N. Levitt. Network intrusion detection. *IEEE Network*, 8(3):26–41, May 1994.

[14] Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy, editors. *Advances in Knowledge Discovery and Data Mining.* American Association for Artificial Intelligence, 1996.

Stefanos Manganaris

Stefanos Manganaris is a senior consultant in the Knowledge Discovery Consulting group at IBM. He is responsible for data mining solutions in consulting engagements, education and technology transfer on knowledge discovery issues, and research in machine learning and statistical pattern recognition with applications to business problems across industries. Dr. Manganaris earned a Ph.D. in Computer Science at Vanderbilt University.

Marvin Christensen

Marvin J Christensen is manager of Corporate Emergency Response Services (ERS) at IBM. He is responsible for vulnerability testing, compliance testing, and providing the infrastructure support for ERS's Real-Time Intrusion Detection offering. Mr. Christensen is responsible for the development and implementation of an open architecture RTID monitoring system that supports multiple commercially available RTID systems. He is also responsible for incident handling of Internet computer security incidents for IBM. Mr. Christensen earned his Masters Degree in Computer Science at the University California in Davis.

Dan Zerkle

Dan Zerkle is a senior Internet security analyst from IBM's Emergency Response Service, specializing in vulnerability analysis and in intrusion detection infrastructure support. He earned his M.S. in Computer Science from the University of California at Davis.

Keith Hermiz

Keith Hermiz serves as prinicpal of the Knowledge Discovery Consulting group within IBM's Global Business Intelligence Solutions division. He received a Ph.D. in Applied Mathematics from the University of Maryland, College Park.