Alert Management Systems: A Quick Introduction

Robert Grossman
University of Illinois at Chicago and
Open Data Partners

July 11, 2003
Draft 3.1

This is a draft of a paper from the book Managing Cyber Threats: Issues, Approaches and Challenges, edited by Vipin Kumar, Jaideep Srivastava, Aleksandar Lazarevic, Kluwer Academic Publisher, to appear.

Abstract

We describe a type of data mining system designed to screen events, build profiles associated with the events, and send alerts based upon the profiles and events. These types of systems are becoming known as alert management systems (AMS). We give some examples of alert management systems and give a quick introduction to their architecture and functionality.

Keywords: data mining, alert management systems, events, profiles, alerts

1 Introduction

In this note, we give an overview of systems designed to screen events, build profiles associated with the events, and send alerts based upon the profiles and events. These types of systems are becoming known as alert management systems (AMS). In this paper, we give some examples of alert management systems and give a quick introduction to their architecture and functionality.

Section 2 contains a brief description of related work. Section 3 contains the key definitions. Sections 4 and 5 describe the functionality and architecture of alert management systems. Section 6 describes several examples.

Section 7 describes some alert management systems built by the author and Section 8 contains the conclusion. Skimming the examples in Section 6 first may make the paper easier to understand.

2 Background and Related Work

One of the best understood examples of alert management systems are systems designed to detect fraud. Descriptions of fraud systems can be found in [1], [4], [6], [7]. As far as we are aware of the idea of abstracting the concepts of events, profiles, and alerts and considering a class of systems that uses these concepts for scoring, matching, routing, and linking appears to be novel. On the other hand, as the large number of examples described in Section 4 shows, various examples of alert management systems have been around for quite a long time. Additional references can be found in the references of the work cited above.

3 Events, Profiles, and Updates

Alert management systems are based upon three primitive concepts: events, profiles, and updates, which we now describe.

1. *Profiles*, abstracting feature vectors, are modeled as an (unordered) set of vectors $\{x_i \in \mathbf{R}^N : i \in \mathcal{I}.\}$ The indices $i \in \mathcal{I}$ are called profile ids.

2. *Events*, abstracting transactional data, are modeled as an ordered set of $\{e_j : j \in \mathcal{J}\}$, with the following properties:

   (a) there is a map $\theta(e_j)$ assigning a profile id $i \in \mathcal{I}$ to each event e_j, $j \in \mathcal{J}$.

   (b) events can be concatenated $e_j \cdot e_k$ and this concatenation is associative $(e_i(e_j e_k)) = ((e_i e_j)e_k)$;

   The indices $j \in \mathcal{J}$ are called transaction ids.

3. An *update* is given by an action of events on profiles $e_k \cdot x_i$ with the following properties:

   (a) for an event e_k and a profile x_i,

   $$e_k \cdot x_i \in \mathbf{R}^N,$$

(b) $x_i = e_k \cdot x_i$, in case $\theta(e_k) \neq i$

(c) For all events e_j and e_k

$$(e_j \cdot e_k) \cdot x_i = e_j \cdot (e_k \cdot x_i).$$

In words: transactional data are modeled by events; profiles summarize state information derived and aggregated from their associated events; and events update profiles. The update action captures the aggregation and computation of derived attributes which is usually involved when one or more transactions are used to update their corresponding profile.

Given an initial collection of profiles, the effect of transactional event data is to move each profile along an orbit.

In the paper [9], a very similar set up, dual to the set up here, is used to model events, profiles and updates.

4 Functionality

Although different alert management systems can have quite different functionality, many of them share the following core functionality: scoring, matching, routing, and linking. In this section, we give brief descriptions of each of these.

4.1 Scoring

Scoring is a function mapping profiles to a continuous

$$f : \mathbf{R}^N \longrightarrow \mathbf{R}$$

or finite set of values or labels

$$f : \mathbf{R}^N \longrightarrow \text{Labels}.$$

Alert management systems are often used for real time scoring in the following way:

1. Let e_j be an event associated with a profile ID i, i.e., $\theta(e_j) = i$.
2. Let x_i be the profile associated with profile ID i and

$$x_i' = e_j \cdot x_i$$

be the result of updating the profile with the event.

3. With this data, $f(x_i')$ is the result of scoring the updated profile using a scoring function $f(\cdot)$.

In other words, the event data is used to update the corresponding profile, which is then scored. The goal is to detect bad behavior as soon as possible.

Finally, the term *signature* is sometimes applied to an updating rule in which the old profile or score is averaged with the new profile or score. More precisely, using the notation above, a signature based update uses an update of the form

$$y_i' = \theta f(x_i') + (1 - \theta) y_i,$$

where y_i is the previous, y_i' is the new score, $x_i' = e_j \cdot x_i$ is the updated profile, and $f(x_i')$, the corresponding score. Here $\theta > 0$ is a constant. Signature based methods are described in [2] and [3]. Signature based methods are commonly used in alert management systems since signatures "smooth" blend new event information with historical information stored in the profile, something which in practice is quite helpful.

4.2 Matching

Sometimes associating a profile ID i in $\mathcal{I}$ with an event is straightforward and sometimes it can be quite challenging. For example, given a credit card transaction or call detail record if the profile ID is the account number or the calling number, then the profile ID is immediately and unambiguously available from the event data. On the other hand, if the profile ID must be matched against another list, such as list of customers, this can be more difficult. For example, is John Smith, 121 Oak Road, San Francisco, CA the same as J. Smithe, Oak Avenue, San Francisco, CA 94701. As the amount of data grows, this problem becomes computationally challenging. Even more difficult is the problem of associating a profile ID to an individual who is deliberately trying to make this task difficult, such as an individual engaged in fraud or other inappropriate activities. In this case, multiple variants of names, addresses and phone numbers may be used.

Alert management systems using matching to normalize names, addresses and similar information and to check names, addresses and related information against various lists. Alert management systems often contain both bad and good lists, i.e. lists containing individuals which must be check more carefully (bad lists) and individuals which are known to the system and have already been vetted (good lists).

4.3 Workflow

Often after events and profiles have been scored and checked against good and bad lists, additional work is required. Further investigation may be warranted, checks against additional lists may be formed, various alerts may be sent, etc. For this reason, alert management systems often contain a workflow component containing rules describing the various types of further processing that is required. For example, workflow rules may be used to assign further investigative work to analysts based in part on the analysts current work load and area of expertise. In many cases, the impact of an alert management system is fundamentally dependent upon the quality of the workflow component. Even if the scoring component is extremely accurate with a very low false positive rate, nothing is gained unless the alerts produced by the score get to an individual analyst who can take the appropriate action at the appropriate time after having examined the appropriate auxiliary data.

4.4 Linking

Events and profiles can often times be linked by common shared attributes or by attributes which have some suspicious relationship with each other. A few examples will make this clearer. For example fraud rings sometimes stage a number of different accidents in order to collect insurance payments. The accidents, although seemingly unrelated, may share a common cell phone number (with different addresses), may all occur within a small physical region, may all use the same body shop, or the same doctor, etc. Of course, two accidents, neither of which are fraudulent, may also share common links or attributes. The goal of linking analysis software is to identify linkages which are suspicious in some way so that further investigation may be done. Sometimes link analysis software is also known as forensic software. Some examples of link analysis can be found in [11].

5 Architecture

In this section, we describe a common architecture for an alert management system. See Figure 1. In practice, actual alert management systems are usually much more complex. The functionality for an alert management system can be divided into three general areas. First, functionality which extracts, transforms, cleans, and loads the data. Second, functionality, for the off-line (i.e. non-real time) analysis of data. This includes data analysis,

data mining, link analysis and related types of forensic activities. Third, functionality for the on-line or real time analysis, routing, and workflow.

The off-line analysis usually contains a data warehouse or data mart and various data analysis, data mining, and forensic analysis tools. From this off-line analysis, data mining models and profiles are often produced for the on-line system. In addition, the off-line analysis may involve extensive checking against various internal and third party databases, checking which may be too time consuming to take place on-line.

The on-line analysis usually contains one or more databases containing various watch lists which incoming events and profiles are compared to. In addition, scoring may be done using the data mining model produced from the off-line analysis. Finally, workflow and routing is usually done producing various alerts and reports.

Part of the complexity of alert management systems is that the extraction, transformation, cleaning and loading must be consistent for the both the off-line and on-line components. There is usually reporting which is part of both the off-line and on-line components of the system.

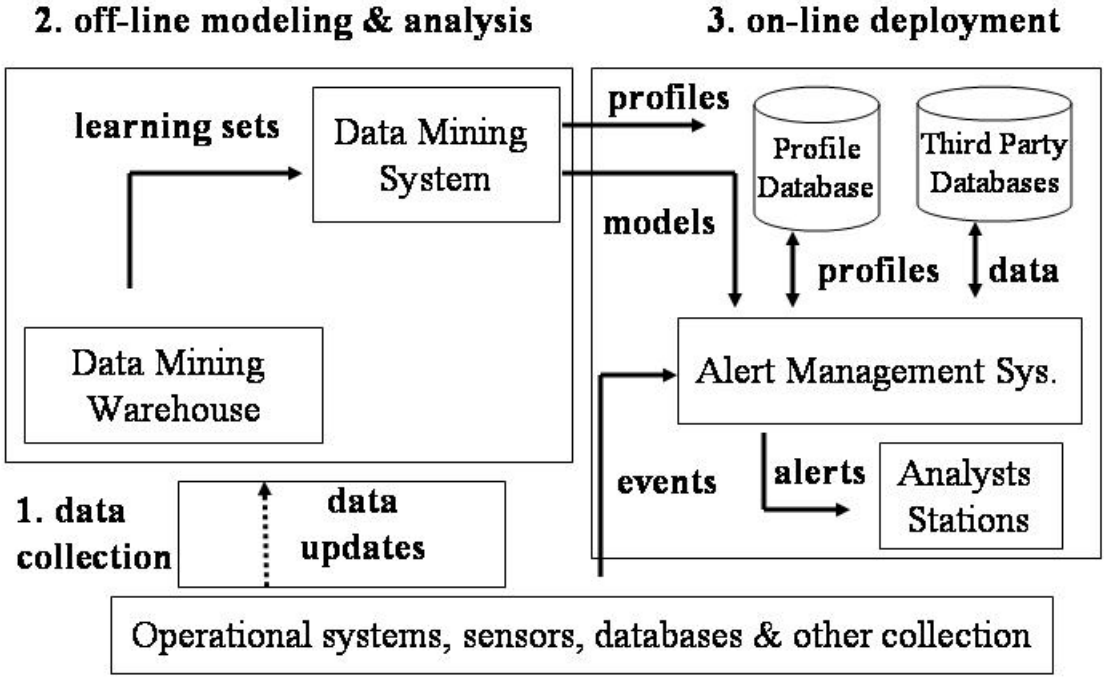

Figure 1: A typical architecture for an alert management system.

6 Examples

In this section, we give some examples of alert management systems. In most of the cases discussed below, there is a natural way to label a data set of events. For example, events may be labeled "good" or "bad"; "intrusion" or "no intrusion"; "normal" or "abnormal"; "threat" or "no threat"; "fraud" or "no fraud"; or "red" or "blue". For simplicity, we often refer to the labels simply as "bad" or "good," with the understanding that the particular meaning of these terms is dependent upon the particular example.

A labeled set of events can be used to label profiles in different ways. A common rule is to assume that profiles are initially labeled good until an event labeled bad is associated with them, after which they are labeled bad. Notice that this makes sense for credit card transactions and similar types of event data: a given credit card account can have a mixture of good and bad transactions. The goal is to detect when there are bad transactions and thereafter stop all transactions. Given a labeled set of events, we can use a variety of classification algorithms to construct a scoring function, which is simply a numerical function on state space $\mathbf{R}^N$ indicating the likelihood that a profile is bad.

Credit Card Transacations. One of the best examples of transactional data is provided by credit card transactions. The data in a credit card transaction is broadly based upon the ISO 8583 standard and includes the account number, the date and time of the transaction, the amount of the purchase, etc. By aggregating transactional data by account number, a profile can be built for each account number. A fraud model uses transactional data to update profiles and then scores each profile for the likelihood of fraud.

Perhaps the best known alert management system for detecting credit card fraud is the Falcon System developed by HNC [10].

Call Detail Records. A Call Detail Record (CDR) contains data about telephone calls, including the date and time of the call, the duration of the call, the calling number, the called number, the telephone number of the billed party, which may be different then the calling number (for example, with 800 numbers), and related data. By aggregating CDR data by the calling number, a profile can be created. A variety of models can be built using these profiles. As before, a fraud model can be built which updates profiles using CDR data and then scores the updated profiles for the likelihood of fraud. As another example, models can be built predicting the likelihood of customer attrition or churn, or predicting the lifetime value of

a customer. For the latter two examples, models may be built based upon a single calling number, or by aggregating all calling numbers associated with a given individual, household, or business.

Alert management systems for detecting telephone fraud have been developed by several of the large telephone companies, for example by AT&T [3].

Passenger Name Records. A third example is provided by passenger name records or PNRs. The transactional data in a PNR includes the originating city, connecting cities, if any, the destination city, flight numbers, name and address of the passenger, frequent flyer number, and related information. Giving a collection of PNRs, profiles can be built for each passenger. Using these profiles, a risk assessment can be done for each airline passenger.

An example of an alert management system for PNRs is the Computer Assisted Passenger Screening System (CAPS) used by the TSA to screen airline passengers at airports.

Network Intrusion Systems. Another example is provided by network intrusion systems employing statistical methods. Network intrusion systems monitor events derived from system logs and other sources. These are used to update various internal feature vectors, which are used as the inputs to statistical models, whose outputs trigger alerts.

Today, the most common network intrusion detection systems, such as Snort [12], look for specific patterns in the data (which are also called signatures, but different than the signatures described above) and do not employ event-profile based techniques.

Suspicious Activity Reports. The Financial Crimes Enforcement Network or FINCEN, which is part of United States Department of the Treasury, collects reports from financial institutions about various types of suspicious financial transactions. These reports are called Suspicious Activity Reports or SARs. There are a number of criteria used for deciding whether or not to file a SAR. In addition, financial institutions are precluded from doing any business with certain individuals or business which have been placed on various watch lists. Larger financial institutions use alert management systems for comparing new accounts to the watch list, as well as for scoring transactions in order to decide whether or not it is necessary to file a SAR.

Automated Manifest System. The Automated Manifest System (AMS) is a system operated by the US Customs which provides inventory control and release notification for cargo entering the US. Carriers, port authorities, service bureaus, freight forwarders, and container freight stations can

use the AMS to provide digital processing of manifest and waybill data. The AMS in turn can use manifest and waybill event data to build profiles about the users of their systems. Alert management systems associated with the AMS can score both event data (manifest and waybill data), as well profiles summarizing activities about carriers and other users of the system. Particularly important for systems like this is improving scoring by overlaying third party data over internal event and profile data.

Interagency Border Inspection System The US Customs Service and Immigration and Naturalization Service (INS) use the Interagency Border Inspection System (IBIS) to screen individuals at ports of entry to the US. IBIS data is collected from a variety of sources and profiles generated by IBIS are shared by a over 20 US federal agencies. IBIS is used at ports of entry to clear expeditiously the majority of the traveling public, while allowing attention to be focused on a relatively small number of individuals. IBIS contains data on suspect individuals, businesses, vehicles, aircraft, and vessels.

7 Status

During the period 1996-2002, Magnify developed an alert management system based upon its PATTERN data mining system [7]. PATTERN was a data mining system which was designed for mining very large data sets which did not fit into memory and was based upon the following ideas:

- PATTERN employed ensemble based modeling. Typically, ensembles were used to partition data into chunks which could fit into memory.
- PATTERN also employed boosting to improve the accuracy of the ensembles produced.
- PATTERN employed a column oriented data warehouse so that numerically intensive operations could be performed efficiently on large amounts of disk resident data.
- PATTERN was designed to run on both single workstations and clusters of workstations. MPI was used for message passing when employed on clusters.
- PATTERN used an XML representation for statistical and data mining models to provide a simple interface between the off-line data mining

component and the on-line scoring or deployment of component of the system.

- PATTERN contained specialized libraries for data transformations and data aggregations so that large numbers of events could be aggregated into profiles efficiently.

This functionality was added over a period of time. During the period, 1995-1996, the alert management system consisting of a off-line data mining system which was used for scoring. An on-line scoring component was added during 1997-1998 following the architecture described in Figure 1. A component for transforming and aggregating data was added during the period 1999-2000. A workflow and routing component was added during the period 2000-2002 [8]. Simple matching and linking was done in an ad hoc fashion, dependent upon the particular requirements of of the application.

The alert management systems built over PATTERN were used for a variety of applications including: detecting credit card fraud, detecting insurance fraud, analyzing TCP packet data for network intrusions, and uncovering suspicous events and profiles in passenger name record data.

8 Conclusion

In this note, we have provided a quick introduction to alert management systems. We have introduced the primitive concepts of events, profiles, and updates. We have also given six examples of these types of systems; many more could be given. There are four key functions usually present in an alert management system: scoring, matching, linking, and workflow, which we have briefly described. Finally, we have given a brief description of a common architecture used by alert management systems. With the increased focus on homeland defense, alert management systems will no doubt grow in importance.

References

[1] Dean W. Abbott, I. Phillip Matkovsky, and John F. Elder IV. An evaluation of highend data mining tools for fraud detection. In IEEE International Conference on Systems, Man and Cybernetics, 1998.

[2] C. Cortes, K. Fisher, D. Pregibon, and A. Rogers. Hancock: A Language for Extracting Signatures from Data Streams. In Proceedings of

the Association for Computing Machinery Sixth International Conference on Knowledge Discovery and Data Mining, pages 9–17, 2000.

[3] C.Cortes and D. Pregibon, Signature-based methods for data streams, Data Mining and Knowledge Discovery, 2001.

[4] T. Fawcett and F. Provost, Adaptive Fraud Detection, Data Mining and Knowledge Discovery, Volume 1, Number 3, 1997, pages 291-316.

[5] T. Fawcett, and F. Provost, Activity monitoring: Noticing interesting changes in behavior, Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining, 1999, pages 53-62.

[6] R. L. Grossman, H. Bodek, D. Northcutt, and H. V. Poor, Data Mining and Tree-based Optimization, Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, E. Simoudis, J. Han and U. Fayyad, editors, AAAI Press, Menlo Park, California, 1996, pp 323-326.

[7] PATTERN Data Mining System, Version 1.2, Magnify, Inc., 1997.

[8] PATTERN Data Mining System, Version 3.1, Magnify, Inc. 2000.

[9] R. L. Grossman and R. G. Larson, An Algebraic Approach to Data Mining: Some Examples, Proceedings of the 2002 IEEE International Conference on Data Mining, IEEE Computer Society, Los Alamitos, California, 2002, pages 613-616.

[10] HNC Software, a division of Fair Isaac Corporation, retrieved from http://www.fairisaac.com/fairisaac on August 20, 2003.

[11] Daryl Pregibon, Graph Mining: Discovery in Large Networks, CCR/DIMACS Workshop on Mining Massive Data Sets and Streams: Mathematical Methods and Algorithms for Homeland Defense, to appear.

[12] Snort(tm), The Open Source Network Intrusion Detection System, retrieved from http://www.snort.org on August 20, 2003.