Data Mining for Network Intrusion Detection: How to Get Started

Eric Bloedorn, Alan D. Christiansen, William Hill,
Clement Skorupka, Lisa M. Talbot, Jonathan Tivel
The MITRE Corporation
1820 Dolley Madison Blvd.
McLean, VA 22102
(703) 983-5274
bloedorn@mitre.org

Abstract

Recently there has been much interest in applying data mining to computer network intrusion detection. For the past two years, MITRE has been exploring how to make data mining useful in this context. This paper provides lessons learned in this task. Based upon our experiences in getting started on this type of project, we suggest data mining techniques to consider and types of expertise and infrastructure needed. This paper has two intended audiences: network security professionals with little background in data mining, and data mining experts with little background in network intrusion detection.

Key words: data mining, intrusion detection, computer network security

1. Network Intrusion Detection: What is it?

Intrusion detection starts with instrumentation of a computer network for data collection. Pattern-based software 'sensors' monitor the network traffic and raise 'alarms' when the traffic matches a saved pattern. Security analysts decide whether these alarms indicate an event serious enough to warrant a response. A response might be to shut down a part of the network, to phone the internet service provider associated with suspicious traffic, or to simply make note of unusual traffic for future reference.

If the network is small and signatures are kept up to date, the human analyst solution to intrusion detection works well. But when organizations have a large, complex network the human analysts quickly become overwhelmed by the number of alarms they need to review. The sensors on the MITRE network, for example, currently generate over one million alarms per day. And that number is increasing. This situation arises from ever increasing attacks on the network, as well as a tendency for sensor patterns to be insufficiently selective (i.e., raise too many false alarms). Commercial tools typically do not provide an enterprise level view of alarms generated by multiple sensor vendors. Commercial intrusion detection software packages tend to be signature-oriented with little or no state information maintained. These limitations led us to investigate the application of data mining to this problem.

2. Intrusion Detection before Data Mining

When we first began to do intrusion detection on our network, we didn't focus on data mining, but rather on more fundamental issues: How would the sensors perform? How much data would we get? How would we display the data? What kind of data did we want to see, and what queries would be best to highlight that data? Next, as the data came in, sensor tuning, incident investigation, and system performance commanded our attention. The analyst team grew to

handle the load, and training and team coordination were the issues of the day. But the level of reconnaissance and attack on the internet was constantly increasing, along with the amount of data we were collecting and putting in front of our analysts. We began to suspect that our system was inadequate for detecting the most dangerous attacks—those performed by adversaries using attacks that are new, stealthy, or both. So we considered data mining with two questions in mind:

- Can we develop a way to minimize what the analysts need to look at daily?
- Can data mining help us find attacks that the sensors and analysts did not find?

3. Data Mining: What is it?

Data mining is, at its core, pattern finding. Data miners are experts at using specialized software to find regularities (and irregularities) in large data sets. Here are a few specific things that data mining might contribute to an intrusion detection project:

- Remove normal activity from alarm data to allow analysts to focus on real attacks
- Identify false alarm generators and "bad" sensor signatures
- Find anomalous activity that uncovers a real attack
- Identify long, ongoing patterns (different IP address, same activity)

To accomplish these tasks, data miners use one or more of the following techniques:

- *Data summarization* with statistics, including finding outliers
- *Visualization*: presenting a graphical summary of the data
- *Clustering* of the data into natural categories [Manganaris et al., 2000]
- *Association rule discovery*: defining normal activity and enabling the discovery of anomalies [Clifton and Gengo, 2000; Barbara et al., 2001]
- *Classification*: predicting the category to which a particular record belongs [Lee and Stolfo, 1998]

4. Start by Making Your Requirements Realistic

The seductive vision of automation is that it can and will solve all your problems, making human involvement unnecessary. This is a mirage in intrusion detection. Human analysts will always be needed to monitor that the automated system is performing as desired, to identify new categories of attacks, and to analyze the more sophisticated attacks. In our case, our primary concern was relieving the analyst's day to day burden.

Real-time automated response is very desirable in some intrusion detection contexts. But this puts a large demand on database performance. The database must be fast enough to record alarms and produce query results simultaneously. Real time scoring of anomaly, or classification models is possible, but this should not be confused with real-time model building. There is research in this area [Domingos and Hulten, 2000], but data mining is not currently capable of *learning* from large amounts real-time, dynamically changing data. It is better suited to batch processing of a number of collected records. Therefore, we adopted a daily processing regime, rather than an hourly or minute-by-minute scheme.

5. Choose a Broad and Capable Project Staff

Your staff will need skills in three areas: network security, data mining, and database application development.

- Of course the security staff need a solid grounding in networking and intrusion detection, but they also need to be able to tackle big, abstract problems.

- The data miners should have a good grounding in statistics and machine learning, but they will also need to learn detailed concepts involved in computer networking.
- The database developers will need good skills in efficient database design, performance tuning, and data warehousing.

This team will have to do a lot of cross orientation to begin working effectively. Initially, security and networking concepts must be introduced and defined. This is made more difficult by the lack of precisely predefined terminology in this field, so there will be many questions. *(What is an attack? What is normal? What is an IDS "alarm"? What constitutes an incident? What is a false alarm, and why?)*

6. Invest in Adequate Infrastructure

Significant infrastructure is required to do this sort of work. In addition to the normal processing of the data from the intrusion detection system, you will need:

- **A Database:** Because you will need to store a great deal of data, update this data regularly, and obtain rapid responses to complex queries, we recommend that you select a high-end production-quality database management system.
- **Storage Space:** In addition to the handling of normal IDS data, you will need data and working space associated with data mining. Additional data includes calculating and saving metadata, as well as sometimes copying existing data into more convenient data types. Working space will hold the various sample data sets that will be extracted for experimentation, as well as working files containing intermediate and final results. Plan for data mining to double your storage requirements.
- **Compute capability:** Data mining tools are very CPU and memory intensive. Naturally, the more memory and CPU power the better. We have found that we needed at least four times the memory and CPU power over what would be needed for an IDS database without the data mining.
- **Software:** In addition to what is required for the basic system (production quality database, Perl, database middleware, database administration and tuning aids), plan for acquisition of specialized tools. For example, we have tried the tools Clementine (http://www.spss.com/Clementine/), Gritbot (http://www.rulequest.com/gritbot-info.html), and a few others. But remember that obtaining data mining software is a necessary but not sufficient step toward a data mining capability. You still need a person who can use the software effectively. We believe that your team needs to have at least one person with some previous data mining experience.

7. Design, Compute, and Store Appropriate Attributes

Data records consist of many attributes. When doing data mining for intrusion detection one could use data at the level of TCPDUMP [Lee and Stolfo, 1998] or at the alarm level [Manganaris, et al. 2000]. In both types of data you will find fields for source IP address, destination IP address, source port number, destination port number, date/time, transfer protocol (TCP, UDP, ICMP, etc.), and traffic duration (or equivalently, both start and end times). These 'base' attributes give a good description of the individual connection or alarm, but they often are insufficient to identify anomalous or malicious activity because they do not take into account the larger context. The individual connection records in a denial of service attack are not, by themselves, malicious, but they come in such numbers that they overwhelm your network. A single connection between an outside machine and a single port on a machine inside your network is also not malicious—unless it is part of a series of connections that attempted to map

all the active ports on that machine. For this reason you will want to add additional fields containing values derived from the base fields. For example, you could distinguish traffic originating from outside your network from traffic originating inside your network.

Another type of derived data, called an *aggregation*, is a summary count of traffic matching some particular pattern. For example, we might want to know, for a particular source IP address X, and a particular IP address Y, how many unique destination IP addresses were contacted in a specific time window Z. A high value of this measure could give an indication of IP mapping, which is a pre-attack reconnaissance of the network. Aggregations are generally more expensive to compute than other kinds of derived data that are based upon only a single record.

A third type of derived data is a flag indicating whether a particular alarm satisfies a heuristic rule. Because data mining methods handle many attributes well, and because we don't know for sure which one will be useful, our approach is to compute a large number of attributes (over one hundred) and store them in the database with the base alarm fields.

8. Install Data Filters

In our sensor log table, upwards of 95% of the traffic fit the profile of an IP mapping activity. That is, a single source IP was attempting a connection to hundreds or even thousands of destination IPs. Before security specialists can start providing input to the data mining effort, this traffic must be filtered. It is a straightforward task to create a filter that can find these patterns within a data table of traffic.

At MITRE, this preliminary filter is called HOMER (Heuristic for Obvious Mapping Episode Recognition). The heuristic operates on aggregations by source IP, destination port, and protocol and then check to see if a certain threshold of destination IPs were hit within a time window. If the threshold is crossed, an incident is generated and logged to the database. The reduction obtained by HOMER is significant. For example, for the period of Sep. 18 to Sep. 23, 2000, MITRE network sensors generated 4,707,323 alarms (71,094 of priority 1). After HOMER there were 2,824,559 (3,690 of priority 1) – a reduction of 40% (94% of priority 1).

IP mapping activity does not pose much of a security threat in itself, but it can be a prelude to more serious activity. Thus, HOMER provides one other important function. Even though the bulk traffic due to the mapping activity is not shown to the analyst, the source host itself is placed on the radar screen of our system. Please note that some normal activity (e.g., name servers, proxies) within an organization's intranet can match the profile of an IP mapping. HOMER handles this situation by means of an exclusion list of source IPs.

A second heuristic under development, called GHOST (Gathering Heuristic for Obvious Scanning Techniques), plays a slightly different role than HOMER. Port scanning is a more targeted form of information gathering that attempts to profile the services that are run on a potential intrusion target. The GHOST heuristic uses a different set of fields, and has its own configurable time window and port threshold, which if exceeded, triggers a security incident.

9. Refine the Overall Architecture for Intrusion Detection

Our current architecture for intrusion detection is shown in Figure 1. Network traffic is analyzed by a variety of available sensors. This sensor data is pulled periodically to a central server for conditioning and input to a relational database. HOMER filters events from the sensor data before they are passed on to the classifier and clustering analyses. Data mining tools filter false alarms and identify anomalous behavior in the large amounts of remaining data. A web server is available as a front end to the database if needed, and analysts can launch a number of predefined queries as well as free form SQL queries from this interface. The goal of this operational model is to have all alarms reviewed by human analysts.

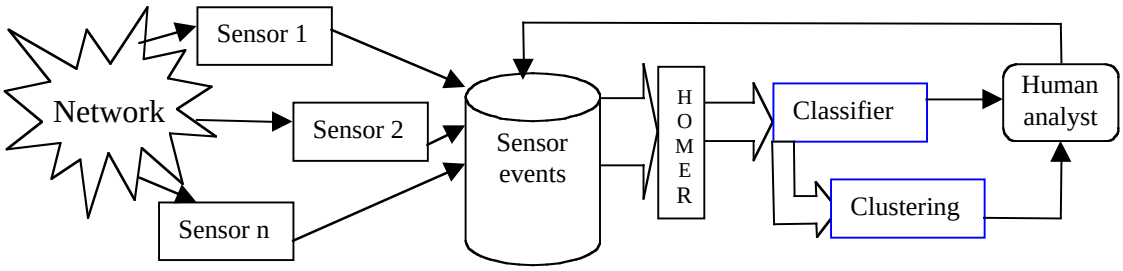

Figure 1. How sensors feed into overall intrusion detection system

Without automated support, this task is increasingly difficult due to the volume of alarms. In one recent day at MITRE for example, sensors generated about 3.4 million alarms, of which about 48,000 are labeled priority 1. Attacks and probes can be frequent and noisy, generating thousands of alarms in a day. This can create a burden on the network security analyst, who must perform a triage on the enormous flood of alarms.

10. Build Classification Rules

Classification is used to assign examples to pre-defined categories. Machine learning software performs this task by extracting or learning discrimination rules from examples of correctly classified data. Classification models can be built using a wide variety of algorithms. Henery [1994] classifies classification algorithms into three types:

- extensions to linear discrimination (e.g., multi-layer perceptron, logistic discrimination),
- decision tree and rule-based methods (e.g. C4.5, AQ, CART), and
- density estimators (Naïve Bayes, k-nearest neighbor, LVQ).

In this work we have, so far, used only decision tree and rule-based methods because of their familiarity to us and because of their ability to give human understandable results.

Good Examples. The 'quality' of the training data is one of the most important factors in achieving good classifier performance. Training data quality is a function of the number of examples, how representative the examples are, and the attributes used to describe them.

Labeled Data. Supervised classification uses labeled training examples to build a model. The labels usually come from a human expert (or experts) who manually review cases. In our application of classification to intrusion detection we obtained labeled examples by building a web-based interface that required a label to be assigned to a new incident each time it was constructed by an analyst. Using this feedback we were able to collect 12,900 labeled examples of seven different classes of incidents from August 2000 and 16,885 for September 2000.

Classes. Another factor in getting good examples is to have a well-defined set of classes. It is important to maintain consistency in assigned labels over time, both for a single person and across multiple people. Label inconsistency can make classification very difficult especially if identical examples are labeled ambiguously.

11. Perform Anomaly Detection

Both intruder techniques and local network configurations will change. In spite of efforts to update defenses, new attacks may slip through defenses and be labeled as either normal network traffic, or else filtered as a known but benign probe. Anomaly detection techniques can help humans prioritize potentially anomalous records for review. Catching new attacks can not depend on the current set of classification rules. Since classification assumes that incoming data

will match that seen in the past, classification may be an inappropriate approach to finding new attacks. Much of the work in outlier detection has been approached from a statistical point of view and is primarily concerned with one or very few attributes. However, because the network data has many dimensions, we have investigated use of clustering for anomaly detection.

Clustering is an unsupervised machine learning technique for finding patterns in unlabeled data with many dimensions (number of attributes). We use k-means clustering to find natural groupings of similar alarm records. Records that are far from any of these clusters indicate unusual activity that may be part of a new attack.

The network data available for intrusion detection is primarily categorical (i.e., Attributes have a small number of unordered values). Clustering approaches for categorical data, such as in [Guha et al., 1999] are not generally available commercially. Unsupervised approaches for detecting outliers in large data sets for the purposes of fraud or intrusion detection are starting to appear in the literature, but these approaches are primarily based on ordered data. Knorr and Ng [1998] recently developed a distance-based clustering approach for outlier detection in large data sets. Ramaswarny, et al. [2000] define a new outlier criterion based on the distance of a point to its k^{th} nearest neighbor. Breunig et al. [2000] define a new local outlier factor, which is the degree to which a data point is an outlier.

12. Make Your System Efficient

There are a number of practical considerations in building an effective intrusion detection system. Some of these derive from the use of data mining, but many of them would be present in any intrusion detection system:

- **A central repository must be designed and enabled.** The repository must allow for inputs from a potentially large number of diverse network sensors, preferably within a single data table. Any derived data, such as data mining attributes, should also be stored in this central location. It must also support the creation and tracking of security incidents.
- **Efficient querying is essential to feed the daily operations of security analysts.** A bottleneck in querying the data will affect everything else in the system. Some steps that can be taken to improve query efficiency include the inclusion of a database performance guru on the project team, statistical/ trend analysis of query performance over time, elimination of time-consuming queries, or the retirement of old data from the database.
- **Efficiency can also be improved by selecting appropriate aggregations of attributes and statistics.** A manual analysis of network activity will reveal that a large volume of atomic network activity breaks down into a much smaller set of meaningful aggregates. At MITRE, two of the more useful aggregates were (source IP, destination port), used for catching some IP mapping activity, and (source IP, destination IP), used for catching port scanning activity. But, any combination of fields or attributes could also be used, resulting in a wealth of choices. Regardless of the fields used, aggregates reduce the downstream volume of data.
- **While most attributes and aggregates are used to feed an automated process, don't forget the analysts.** Analysts must have efficient tools to spot check the automatically generated security incidents, and to manually comb through the raw sensor data for new or complex patterns of malicious activity. The MITRE interface is centered on a set of predefined queries of the sensor database, and a browser of the incident database. With this tool, an analyst can create new security incidents or update existing incidents with new status information.
- **Due to the high volume and frequency of data inputs, and the variety of both automated and human data sources, there will invariably be some process failures.** When a failure does occur, the condition must be caught and the security team notified. Scripts that verify

the integrity of the data tables, and repair inconsistencies, are useful. If possible, the process should be halted until the error is corrected. But, in some situations, the ability to operate normally regardless of errors, and then rollback and correct statistics and attributes at the team's convenience, may be a more practical recovery strategy.

- **Scheduling is an important aspect of the operational environment.** Each organization must decide for itself how much of its intrusion detection system truly needs to be "real-time". The calculation of real time statistics must be completed in a matter of seconds, and the amount of data available in this manner will always be limited. But daily batch processing of data may be adequate in many cases.

13. Summary

We have described our experiences with integrating data mining into a network intrusion detection capability. We believe that when starting such a project you should:

- Choose your requirements carefully and be realistic.
- Assemble a team with broad, relevant capabilities.
- Invest in adequate infrastructure to support data collection and data mining.
- Design, compute, and store appropriate attributes with your data.
- Reduce data volume with filtering rules.
- Refine the overall architecture for your system, taking into account both automated processing and human analysis.
- Use data mining techniques such as classification, clustering, and anomaly detection, to suggest new filter rules.
- Make sure that automated data processing can be done efficiently.

Additional information on our specific approach to data mining for intrusion detection can be found in [Skorupka et al., 2001] and [Bloedorn et al., 2001].

References

Barbara, D., N. Wu, and S. Jajodia [2001]. "Detecting Novel Network Intrusions Using Bayes Estimators", Proceedings Of the *First SIAM Int. Conference on Data Mining*, (SDM 2001), Chicago, IL.

Bloedorn, E., L. Talbot, C. Skorupka, A. Christiansen, W. Hill, and J. Tivel [2001]. "Data Mining applied to Intrusion Detection: MITRE Experiences," submitted to *the 2001 IEEE International Conference on Data Mining.*

Breunig, M. M., H. P. Kriegel, R. T. Ng, and J. Sander [2001]. "LOF: Identifying Density-Based Local Outliers", Proceedings of the *ACM Sigmod 2000 Intl. Conference On Management of Data*, Dallas, TX.

Clifton, C., and G. Gengo [2000]. "Developing Custom Intrusion Detection Filters Using Data Mining", *2000 Military Communications International*, Los Angeles, California, October 22-25.

Domingos, P., and G. Hulten [2000]. "Mining High Speed Data Streams", in Proceedings of the *Sixth ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, p. 71-80.

Guha, S., Rastogi, R., and Shim, K. [1999]. "ROCK: A Robust Clustering Algorithm for Categorical Attributes", Proceedings of the *15th Int. Conference On Data Eng*., Sydney, Australia.

Henery, R. J. [1994]. "Classification," *Machine Learning, Neural and Statistical Classification*, Michie, D., Spiegelhalter, D. J., and Taylor, C. C. (Eds.), Ellis Horwood, New York.

Knorr, E. M., and R. T. Ng [1998]. "Algorithms for Mining Distance-Based Outliers in Large Datasets", VLDB'98, Proceedings of the *24th Int. Conference on Very Large Databases*, Aug 24-27, 1998, New York City, NY, pp. 392-403.

Lee, W., and S. Stolfo [1998]. "Data Mining Approaches for Intrusion Detection", in Proceedings of the *7th USENIX Security Symposium*, San Antonio, TX.

Manganaris, S., M. Christensen, D. Zerkle, and K. Hermiz [2000]. "A data mining analysis of RTID alarms", *Computer Networks*, 34, p. 571-577.

Ramaswarny, S., R. Rastogi, and K. Shim, [2000]. "Efficient Algorithms for Mining Outliers from Large Data Sets", Proceedings of the *ACM Sigmod 2000 Int. Conference on Management of Data*, Dallas, TX.

Skorupka, C., J. Tivel, L. Talbot, D. Debarr, W. Hill, E. Bloedorn, and A. Christiansen [2001]. "Surf the Flood: Reducing High-Volume Intrusion Detection Data by Automated Record Aggregation," Proceedings of the *SANS 2001 Technical Conference*, Baltimore, MD.

Acknowledgment

This work was sponsored by the MITRE Technology Program as a MITRE Sponsored Research (MSR) Project.

Author Biographies

Eric Bloedorn is a lead staff member of the Artificial Intelligence Technical Center at the MITRE Corporation. His interests include machine learning and its applications, especially text. He has worked at George Mason University, Argonne National Lab, and Computer Sciences Corporation. Dr. Bloedorn received his B.A. degree in Physics from Lawrence University in 1989, and his M.Sc. and Ph.D. from George Mason University in 1992 and 1996 respectively.

Alan D. Christiansen is a Lead Artificial Intelligence Engineer with MITRE's Information Systems and Technology Division. Before joining the company in May 2000, he had been employed by Sandia National Labs, LIFIA (Grenoble, France), Tulane University, Microsoft Research, and SAIC. He received his Ph.D. in 1992 from the School of Computer Science at Carnegie Mellon University. Machine learning is one of his technical interests.

William Hill is a Senior Principal Security Engineer in Security and Information Operations Division at the MITRE Corporation. Mr. Hill has been involved in computer networking and security since 1990, working in network programming, design, operations and security, and most recently in vulnerability analysis, intrusion detection and response, and incident investigation. Prior to joining MITRE, Mr. Hill worked for Bell Atlantic, managing network operations for their Internet Center. Mr. Hill holds a B.S. from Florida State University and a Master of Science in Computer Science from George Mason University.

Clem Skorupka joined MITRE in the summer of 2000, where he works as a Lead Infosec Scientist in MITRE's Enterprise Security Solutions Division. Before joining MITRE, Clem worked for AT&T's Government Markets, where he acted as lead for firewall and network management operations for an intelligence community customer. Clem has over ten years experience in network and UNIX system administration supporting a variety of government and commercial organizations. He holds a Ph.D. in Physics from Clemson University.

Lisa M. Talbot is a consultant to MITRE through Simplex, LLC, of Leesburg, VA. She received a Ph.D. degree in Electrical Engineering from Brigham Young University in 1996. She has formerly consulted for TASC, Inc, Chantilly, VA, and has been an employee of Hughes Aircraft Company, Aurora, CO. Her research interests include pattern recognition, categorical data clustering, fuzzy and statistical signal processing, image processing, remote sensing, meteorological and atmospheric applications, and intelligent software agents.

Jonathan Tivel received a B.S. in Computer Science from the University of Virginia in 1991 and an M.E. in Systems Engineering in 1995. He has been deeply involved in MITRE research projects for the past 10 years, where he has applied his diverse knowledge of database, networking, and software technologies towards the design and production of integrated systems.