OFFLINE NETWORK INTRUSION DETECTION: MINING TCPDUMP DATA TO IDENTIFY SUSPICIOUS ACTIVITY

KRISTIN R. NAUTA AND FRANK LIEBLE

Abstract

With the boom in electronic commerce and the increasing global interconnectedness of computer systems, infrastructure protection has quickly become the 'next big problem' after Year 2000 preparedness. Public and private sector organizations have a created a multi-billion dollar market demand for tools to protect themselves against cyber attacks. System administrators are scrambling to find in-line and off-line solutions for detecting network vulnerabilities, and to understand the behavior of network resource consumers.

This paper will provide an introduction to the intricacies of reconciling TCPDUMP packet flow data in an effort to construct IP conversation behavior profiles. The resulting behavior profiles are warehoused for reporting and trend analysis. The behavioral data is mined to identify potential suspicious or intrusive network activity. Finally, by applying the patterns discovered through mining to the randomly timed samples of TCP packet flow data, systems administrators can monitor behavior exception reports for infrastructure vulnerabilities.

OVERVIEW OF OFFLINE INTRUSION DETECTION

With the boom in electronic commerce and the increasing global interconnectedness of computer systems, infrastructure protection has quickly become the 'next big problem' after Year 2000 preparedness. Public and private sector organizations have a created a multi-billion dollar market demand for tools to protect themselves against cyber attacks. These organizations hope to protect the consumer information found in transaction systems; employee data stored in newly implemented ERP systems; and intellectual property stored in their internal knowledge bases, data warehouses, and decision support systems. It is estimated that a full 80% of an organizations' intellectual assets are in digital form. Table 1 displays the Gartner Group's suggested information security hierarchy.

Information Security Hierarchy
Level 6 **Validation**
Level 5 **Auditing, Monitoring, Investigating**
Level 4 **Information Security Technologies and Products**
Level 3 **Information Security Awareness and Training**
Level 2 **Information Security Architecture and Processes**
Level 1 **Information Security Policy and Standards**

Table 1: Information Assurance Hierarchy, Gartner Group

Often, network security violations originate within the network itself, coming from disgruntled or otherwise ill-motivated employees; or from external individuals who have gained access to internal accounts. Their behaviors may involve disruption of services; or the theft or destruction of important files. The fear of gaining a bad reputation through public exposure often prevents attacked organizations from revealing attacks. Thus, although it is difficult to track the losses associated with intrusive acts, system administrators are scrambling to find in-line and off-line solutions for detecting network vulnerabilities, and to understand the behavior of network resource consumers.

There are two types of software solutions available to aid in this quest:

> **"Intrusion detection systems** collect information from various vantage points within a computer system or network, and analyze that information for symptoms of system breaches."
>
> **"Vulnerability systems** check the network and hosts for configuration problems that could give rise to security vulnerabilities."

This paper will discuss implementing an off-line network intrusion detection system to periodically analyze or audit batches of TCP/IP network log data. Further we will discuss how to apply an initial analysis to data collected in future periods so that systems administrators can use exception reports to identify suspected intrusions. It is important to note that no software can monitor all network traffic because the data processing becomes prohibitive. By including network intrusion detection into the comprehensive security infrastructure, system administrators can provide the organization with a more secure computing environment.

USING SAS SOFTWARE FOR OFFLINE INTRUSION DETECTION

To solve the complex issues involved in turning TCP/IP network transactions into data suitable for warehousing, mining, and exception reporting we utilized a synergistic combination of products, namely SAS Enterprise Miner and SAS IT Service Vision to attack the problem.

The SAS Enterprise Miner package can be used to analyze incoming TCP/IP traffic data, analyze the data for unauthorized activities and notify system administrators of identified threats. SAS Enterprise Miner includes the Base SAS software package along with its 4th generation data manipulation language which can be used to re-construct TCP/IP conversations in whole, and to create data content features relevant to the behavior of non-normal network activity. Once the appropriate data structure is created, SAS Enterprise Miner provides a full suite of supervised and unsupervised data mining algorithms, which can be used to formulate intrusion detection models. In the initial stages of intrusion detection system implementation the status of current network activity is often unknown making unsupervised models required to provide the categorization of conversations as normal or non-normal. Non-normal categorization can be specific, i.e., denial of service, port scanning probes, and super user access. SAS Enterprise Miner includes tools to visualize data and model results. These models can be applied in a near real time scenario against incoming TCP/IP data to detect unauthorized activity.

In addition, SAS IT Service Vision will allow administrators to build a read-only, analytical data warehouse, which will be used as the foundation for managing large quantities of incoming TCP/IP traffic data. This data warehouse, known as the performance database (PDB), will provide the ability store data into five levels, Detail - cleansed, transformed data in a non-aggregated format, and Day, Week, Month, and Year - detail data summarized into specific descriptive statistics by different time intervals for long term historical reporting and analysis. Once the TCP/IP traffic data is loaded into a PDB it can accessed using any of the SAS Software tools.

SAS IT Service Vision can be used to identify and monitor simple relationships in the TCP/IP network logs. With SAS Enterprise Miner, complex multivariate models can be created to profile intrusive behaviors.

DATA COLLECTION AND ENHANCEMENT

TCP/IP traffic data is generated using a data collection application, which is referred to as a **collector**. A collector can be a "home grown" application or purchased from one of many network performance vendors available in the market. When network data is generated it is stored in files commonly known as **network data logs**. These data logs can be represented in a wide variety of formats, i.e. ASCII, binary, RDBMS tables which can pose as a problem in reporting and analysis. Also pre-processing of the network data within the logs is usually required so that the data can be standardized. This will ensure that the data is processed correctly and that there are no discrepancies in the representation of the data.

PERFORMANCE DATA WAREHOUSE

Due to the sheer volume of data a methodology needs to be implemented to organize and manage the network data. The performance data warehouse (PDB) provides the ability to filter and store the TCP/IP data into a multilevel repository utilizing three major principals:

1) Data Filtering - defining what performance metrics are to be kept in the PDB. This will allow for the loading of useful data in the PDB and discard unwanted performance metrics,

2) Data Aging - defining how long performance data are to be kept in the PDB before it is deleted or archived. This will manage the size of the PDB and manage data dormancy, and

3) Data Aggregation - defining summary statistics the detail data will be reduced by, i.e., mean, maximum, minimum.

This will allow for easier reporting by reducing machine cycles and run time. With these three principals large amounts of TCP/IP traffic data can be managed and organized for reporting and analysis.

Once the traffic data has been cleansed, loaded, and reduced into a PDB it can be used for a wide variety of reporting and analysis. For example, exception reporting, once data mining determines the unauthorized activities and threats within the TCP/IP data they can be fed back into the PDB as exception rules. These exception rules can be applied against the PDB's detail and/or summarized data allowing for the ability to identify intrusions over a wider span of time.

DATA MINING

Data mining strategies fall into two broad categories: **supervised learning** and **unsupervised learning**. Supervised learning methods are deployed when there exists a field or variable (**target**) with known values and about which predictions will be made by using the values of other fields or variables (**inputs**). Unsupervised learning methods tend to be deployed on data for which there does not exist a field or variable (**target**) with known values, while fields or variables do exist for other fields or variables (**inputs**). Unsupervised learning methods while more frequently used in cases where a target field does not exist, can be deployed on data for which a target field exists. Table 2 breaks down data mining techniques by modeling objective and supervised/unsupervised distinctions.

Modeling Objective	Supervised	Unsupervised
Prediction	Regression and Logistic regression Neural Networks Decision Trees Note targets can be binary, interval, nominal, or ordinal.	Not feasible
Classification	Decision Trees Neural Networks Discriminant Analysis Note targets can be binary, nominal, or ordinal.	Clustering (K-means, etc) Neural Networks Kohonen Networks Self Organizing Maps
Exploration	Decision Trees Note targets can be binary, nominal, or ordinal.	Principal Components Clustering (K-means, etc)
Affinity		Associations Sequences Factor Analysis

Table 2: Modeling Objectives and Data Mining Techniques

Table 1 displays four modeling objectives: prediction; classification; exploration; and affinity. Prediction algorithms determine models or rules to predict continuous or discrete target values given input data. For example, a prediction problem could attempt to predict the value of the S&P 500 Index given some input data (e.g. economic crash in Asia, trade balances, etc.).

Classification algorithms determine models to predict discrete values given input data. A classification problem might involve trying to determine if transactions represents anomalous behavior based on some indicators (if the purchase was made at a pawn shop, amount of purchase, type of purchase, etc.).

Exploration uncovers dimensionality in input data. Trying to uncover groups of similar customers based on spending habits for a large, targeted mailing is an exploration problem. Affinity analysis determines which events are likely to occur in conjunction with one another. Retailers use affinity analysis to analyze product purchase combinations in grocery stores.

Both supervised and unsupervised learning methods are useful for classification purposes. In a particular business problem involving anomaly detection, the objective may be to establish a classification scheme for anomalies. Regression, decision trees, neural networks and clustering can all address this problem. Decision trees and neural networks build classification rules and other mechanisms for detecting anomalies. Clustering would indicate what types of groupings (based on a number of inputs) in a given population are more at risk for exhibiting anomalies, with the grouping membership of a transaction classifying if a transaction were at greater risk of being anomalous.

CASE STUDY: TCPDUMP HEADER DATA

In this case study we used data provided on the Internet at **http://iris.cs.uml.edu:8080/** . Raw TCPDUMP data was collected under a simulated network environment. The network simulated a baseline environment without intrusions and four types of intrusions. Each simulation was approximately 10 minutes of real time. The intrusion simulations were:

- **IP Spoofing** - With IP spoofing an intruder attempts to busy out the system by creating numerous half open connections, often within a small period of time.

- **Rlogin** - The RLOGIN attack is characterized by a high rate of connections from one node to another, often within a small period of time. In this attack, the intruder is attempting to gain access to the system.

- **Network Scanning** - Network scans are used to determine vulnerable ports. The attack is often characterized by accessing a high number of ports on limited set of addresses.

- **Network Hopping** - Network hopping is used to cover the trail of an intruder on your network, making it more difficult to pinpoint the original vulnerability that allowed access to the systems. A hopping attack appears as a chain of subsequent logins to various networked machines.

The goal of the analysis is to create descriptive information from the raw TCPDUMP header files, then to mine the data in order to determine likely intrusive TCP/IP connections.

Data Collection and Enhancement

The raw data consisted of packet level transmission data including source and destination IP address and ports; flags, acknowledgements and packet sequence numbers; and window, buffer and optional information. Table 3 displays the full TCPDUMP header data layout.

Data analysts often recognize the fact that 80% to 90% of their works is really data preparation. With the intrusion simulation data, this was certainly the case. Upon inspection of the data, you find that no single record represents a complete conversation between two IP addresses. In fact, each record is only a portion of the conversation: the source sending information to the destination, or vice versa. In order to create useful inputs for data mining, we must first determine which records are part of the same conversation. After the conversation reconciliation, we can compute meaningful variables such as the number of connections made to a one or more destination IP addresses within a two-second time window.

For data mining then, the data preparation goals is to establish the final state of a conversation; then to understand the behavior of each source IP address on the network in relation to destination IP addresses and destination ports. Steps in the data processing are discussed in the remainder of this section.

The SAS System 1

CONTENTS PROCEDURE

Data Set Name: WORK.NET0 Observations: 49

Member Type: DATA Variables: 21
Engine: V612 Indexes: 0
Created: 11:34 Monday, August 16, 1999 Observation Length: 257
Last Modified: 11:34 Monday, August 16, 1999 Deleted Observations: 0
Protection: Compressed: NO
Data Set Type: Sorted: NO

-----Engine/Host Dependent Information-----

Data Set Page Size: 8192
Number of Data Set Pages: 2
File Format: 607
First Data Page: 1
Max Obs per Page: 31
Obs in First Data Page: 20

-----Alphabetic List of Variables and Attributes-----

#	Variable	Type	Len	Pos	Format	Informat	Label
9	ACK	Num	8	57	BEST12.	BEST32.	Return Sequence Number From Other
11	BUF	Num	8	73	BEST12.	BEST32.	Receive Buffer Space Available
18	CONV1	Char	35	175			Initiator: Conversation Signature
19	CONV2	Char	35	210			Responder: Conversation Signature
14	DATETIME	Num	8	123	DATETIME23.6		Datetime Stamp
4	DEST_ADD	Char	8	24	$8.	$8.	Destination Address
5	DEST_POR	Char	8	32	$8.	$8.	Destination Port
16	DIPADDR	Char	20	151	$20.		Destination Address and Port
21	DPTYPE	Char	4	253			Destination Port Group
17	DTYPE	Char	4	171	$4.		Simulation Data Type
6	FLAG	Char	1	40	$1.	$1.	Flag
13	OP	Char	34	89	$34.	$34.	Optional Information
20	PORTTYPE	Char	8	245			Destination Port Description
7	SEQ1	Num	8	41	BEST12.	BEST32.	Packet Sequence Number
8	SEQ2	Num	8	49	BEST12.	BEST32.	Return Sequence Number
15	SIPADDR	Char	20	131	$20.		Source Address and Port
2	SRC_ADDR	Char	8	8	$8.	$8.	Source Address
3	SRC_PORT	Char	8	16	$8.	$8.	Source Port
1	TIME	Num	8	0	TIME15.6	BEST32.	Time
12	ULEN	Num	8	81	BEST12.	BEST32.	UDP Packet Length
10	WIN	Num	8	65	BEST12.	BEST32.	Receive Buffer Space Available

Table 3: TCPDUMP Header data layout

The first step in data preparation is to group all records from a single source IP to destination IP using a unique indicator. The basic SAS data processing language was used for this step. This creates the **conversation group** field.

Next, we used the packet flags and acknowledgements to determine the current status of a connection. For example, is the connection currently being opened, is already opened, is being closed, is completely closed. Once the records are grouped from the preceding step, the data is sorted by conversation grouping and time of connection. We then retain the flag across each record, compare with the current flag, and determine the connection state. We applied simple logic such as, if a conversation has three records with a SYN flag, then assume a conversation is opened. This step could be improved by refining the logic to determine conversation status. This creates the **final state** field.

To get information on what action the user was attempting, we mapped the destination ports to specific user actions. The destination port determines the function the user is trying to access, while the source port is assigned randomly. Most network administrators use a common set of port mappings. By transforming port numbers into action types, we can determine basic information about what a user was attempting. For this case study, we created high level groupings of the port types as login, email, system status check, SNMP, date, who, chat, and other. This creates the **destination port type** filed.

In order to consider automated versus manual input stream, we needed to determine the amount time elapsed between connections to different destination IP addresses from a single IP source address, and to different destination ports on a single destination IP address from a single IP Source address. We then created indicators for elapsed time within specific ranges. For example we indicated if the elapsed time was less than 5 seconds, between 5 and 30 seconds, greater than 30 seconds, or undeterminable. This creates the **time difference to destination address** and **time difference to destination port** fields.

Exploring Conversation Data for Simple Relationships

TCP/IP traffic data that has been cleansed, loaded, and reduced in the PDB can be exploited using data warehousing performance and exception reporting tools. This provides the system administrator ability report on simple relationships between the data. For example, we generated a performance report showing the top 10 destination addresses. This allowed for the investigation of any abnormalities within the data that could be potential intrusions, Figure 1.

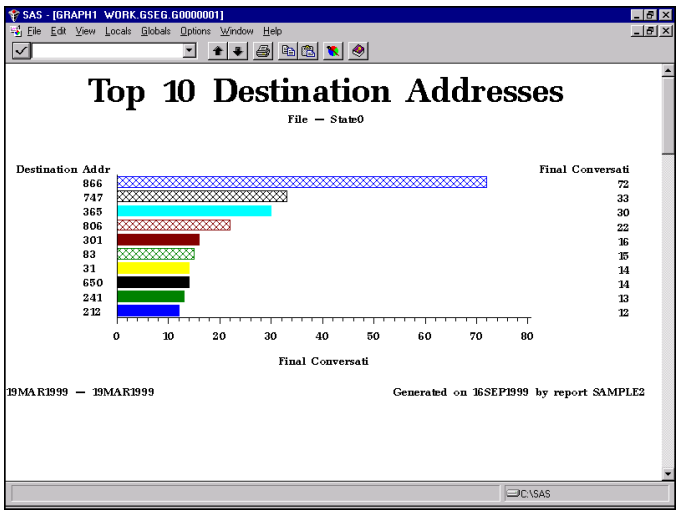

Figure 1: Performance Report

As for exception reporting we utilized exception rules derived from the data-mining tool to identify unauthorized activities and threats within the TCP/IP traffic data. These rules were defined to the data dictionary of the PDB, which were used in exception reports to discover data intrusions over a wider span

of time. For example, the destination address times for connections less then 5 seconds, less then 30 seconds, greater then 30 seconds, and unknown.

By applying the exception rules to the TCP/IP traffic data stored in the PDB an exception report was generated, Figure 2. We can clearly identify that for this specified period time 5422 connections are less than 5 seconds. This would alert the administrator to investigate this potential anomaly further.

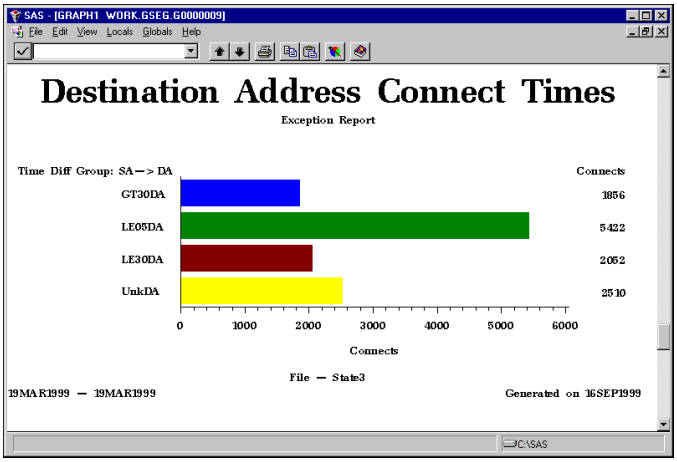

Figure 2: Exception Report

Mining Behavior Data for Complex Relationships

For data mining, we wanted to create a single record for each source IP address describing its behavior on the network. We summarized the final states, number of destination IP addresses, and time differences to destination address fields by the source IP address. This provides, for each source IP address, counts of the number of times each final state occurred, to how many IP addresses were connections made, and counts of the time difference groupings. We also summarized the destination port types and time differences to destination ports by the combined source and destination IP addresses. This provides, for each unique IP source to destination IP connection, the number of times each specific action was attempted, and the number of times port hits occurred within the specified intervals.

To create a single analysis file, we merged the two summary files by source IP address. This results in one file containing the behavior of each source IP with respect to the various destination IP addresses and ports for the given time period. The resulting file layout is displayed in Table 4.

```
                                The SAS System                                1

CONTENTS PROCEDURE

Data Set Name:  INTRUDER.MINEALL                  Observations:          1221 7
Member Type:    DATA                              Variables:             25
Engine:         V612                              Indexes:               0
Created:        11:59 Wednesday, March 24, 1999   Observation Length:    218
Last Modified:  11:59 Wednesday, March 24, 1999   Deleted Observations:  0
Protection:                                       Compressed:            NO
Data Set Type:                                    Sorted:                NO

                 -----Engine/Host Dependent Information-----
                      Data Set Page Size:        8192
                      Number of Data Set Pages:  331
                      File Format:               607
                      First Data Page:           1
                      Max Obs per Page:          37
                      Obs in First Data Page:    21

-----Alphabetic List of Variables and Attributes-----
   # Variable Type Len Pos Format Informat Label
  13 CLOSED   Num    8 114                # Final States: Closed
  25 DATAWAS  Char   8 210                Origin of Intrusion Data
   6 DATE     Num    8  49                # Date Port Requests
  12 DNS      Num    8 106                # Final States: DNS
   7 FTP      Num    8  57                # FTP Port Requests
   2 GNRL     Num    8   17               # General Port Requests
  17 GT30DA   Num    8 146                # Times Betw Dest Addr GT 30 Sec
  21 GT30DP   Num    8 178                # Times Betw Dest Port GT 30 Sec
  16 LE05DA   Num    8 138                # Times Betw Dest Addr LE 5 Sec
  20 LE05DP   Num    8 170                # Times Betw Dest Port LE 5 Sec
  18 LE30DA   Num    8 154                # Times Betw Dest Addr LE 30 Sec
  22 LE30DP   Num    8 186                # Times Betw Dest Port LE 30 Sec
   5 LOG1     Num    8  41                # Login 1 Port Requests
   3 LOG2     Num    8  25                # Login 2 Port Requests
   4 MAIL     Num    8  33                # Mail Port Requests
  10 NDADDR   Num    8  90                # Dest Addr from Src Addr
   8 NDPORT   Num    8  65                # Dest Ports for SA||DA
  11 NEW      Num    8  98                # Final States: New
  23 NSADDR   Num    8 194                # Src Addr if this was Dest Addr
  14 OPENED   Num    8 122                # Final States: Opened
   9 SRC_ADDR Char   17  73 $8.    $8.     Source Address
   1 SRC_DEST Char   17   0                Source Address||Destination Address
  15 UNKDA    Num    8 130                # Times Betw Dest Addr Unknown
  19 UNKDP    Num    8 162                # Times Betw Dest Port Unknown
  24 WHO      Num    8 202                # Who Port Requests
```

Table 4: TCPDUMP Data Prepared for Mining

Mining the Data

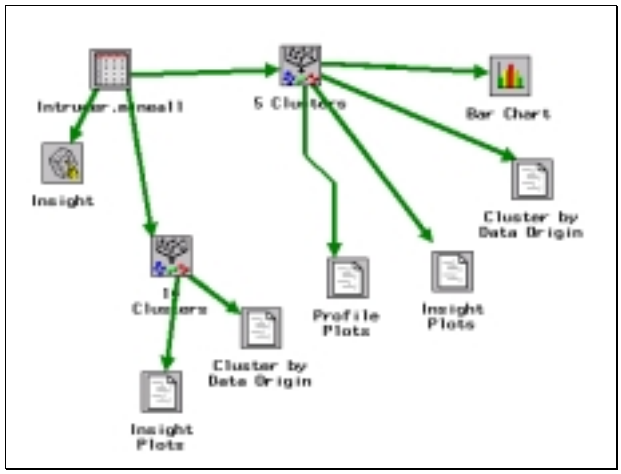

Figure 3: Intrusion Detection Analysis Flow

We used SAS Enterprise Miner to look for intrusive patterns in the data. Enterprise Miner provides a drag and drop interface for creating analysis paths through a data stream. Because we have no information on the data indicating when specific intrusions were simulated, we used the unsupervised data mining technique of clustering. To assist in understanding the clustering results, we initially request only 5 clusters. Later we increased this to 10 clusters to more separation that better correlates to specific intrusion types.

The clustering algorithm used a least squares criterion to determine cluster membership. Results from this analysis follow.

Figure 4 displays the cluster statistics graphically, while Figure 5 displays the same information in tabular format. The height and color of the cluster 1 slices indicates that this cluster contains a very large portion of the data; yet the data are rather compact in the multidimensional space. Because we would expect the most of the conversations on a given network to be normal network activity, cluster 1 is probably identifying normal conversations. Other clusters pick up anomalies in the data. We need to investigate the data patterns associated with each cluster in order understand how the clusters relate to specific intrusions types.

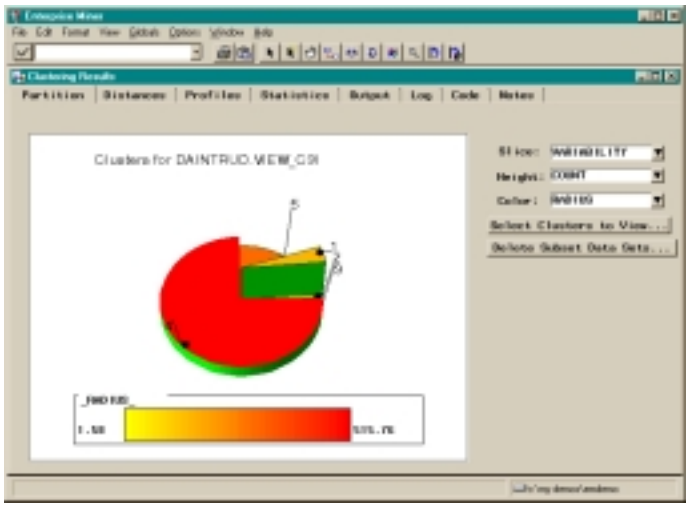

Figure 4: Graphical Cluster Statistics

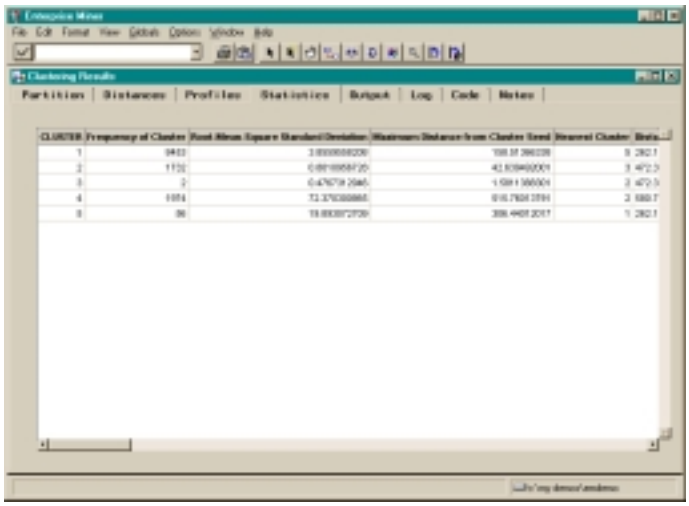

Figure 5: Tabular Cluster Statistics

Cluster 4 appears to be the second largest grouping of data points; and to have large variability in the multi-dimensional space. Cluster 5 is rather small, both in membership and radius. Clusters 2 and 3 are also relatively small. In fact cluster 3 has only 2 observations. We need to determine if these two small clusters are really identifying intrusions, or are just picking up outliers in the data.

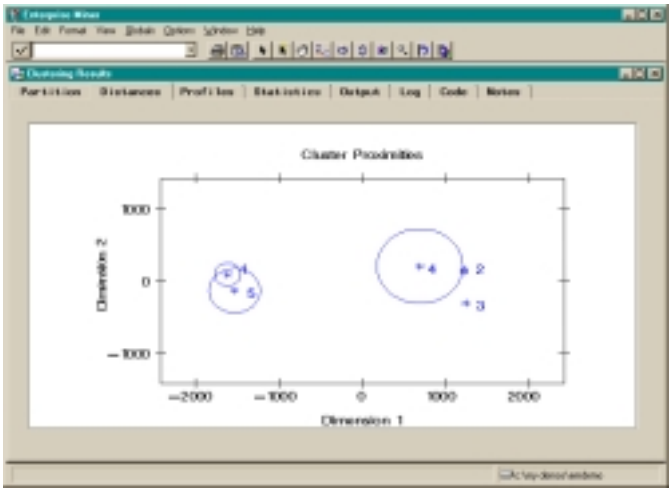

Figure 6: Cluster Proximities

Figure 6 displays the clusters on the principal component axes of the multi-dimensional space we analyzed. Each cluster mean is surrounded by a circle indicating the cluster radius. From this graphic it is obvious that the observations in clusters 1 and 5 have similar raw data (network behaviors); and the same is true for clusters 2, 3 and 4.

To associate the cluster values with specific intrusion types, we need to explore the data further. When we manipulated the data for mining, we created groups of variables that inherently make sense to evaluate together. For example, we counted the number of different connection final states for each source IP address. These final states counts were represented as unique independent variables in our analysis. To understand which clusters may represent RLOGIN attacks, we need to compare the average number of LOGIN attempts versus other port related activities, across the clusters.

Figures 7 through 11 display profile plots of the appropriate groups of variables. Figure displays the profile plot for port types by cluster. In figure 7 we see that cluster 3 as a high number of login attempts relative to other clusters, implying that this cluster could be segmenting out RLOGIN attacks. The observation that there are a large number of general ports for clusters 3 and 5 is probably no interesting. However we may need to break the general port category into other more meaningful groupings to find interesting relationships in the data.

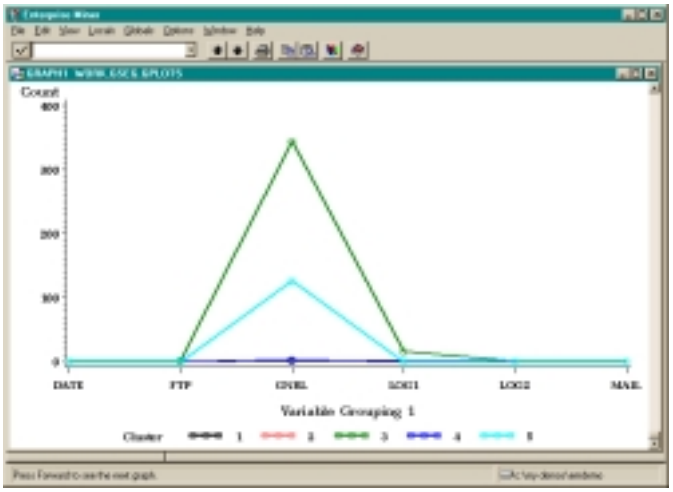

Figure 7: Port Types Profile Plot

Figure 8 displays the profile plot for numbers of destination addresses, destination ports, and source ports. In this plot, the high number of addresses associated with clusters 2, 3, and 4 is not necessarily interesting; and cluster 1 again seems to be picking up normal, light network activity. However the activity of cluster 5 is very interesting. Cluster 5 shows very few addresses connecting to a relatively high number of ports. This behavior is indicative of network hopping.

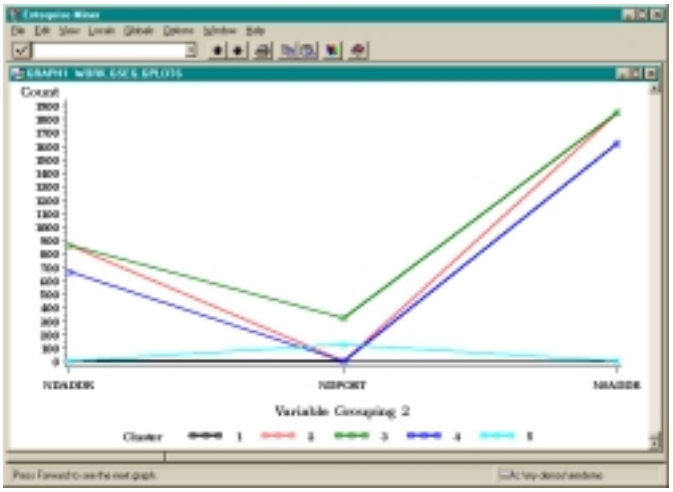

Figure 8: Address and Ports Profile Plot

The profile plot for the final conversation states is displayed in figure 9. This plots reveals that clusters 3 and 4 each have high number of connections left open, and large numbers of connections not fully established (NEW) which could indicate IP Spoofing Activity.

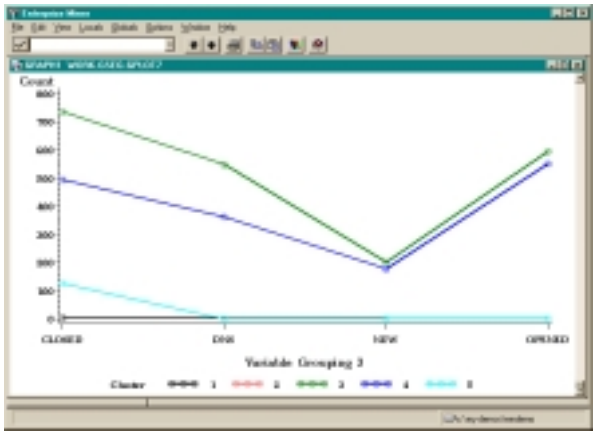

Figure 9: Final State Profile Plot

Finally, figure 10 and 11 display the profile plots for the number of destinations addresses and ports accessed within specific time duration. Again clusters 3 and 4 have a large number of addresses and ports being accessed in less that 5 seconds, as does cluster 5.

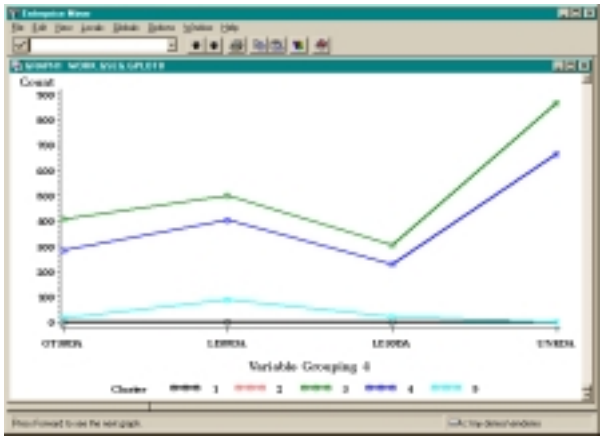

Figure 10: Destination Address Profile Plot

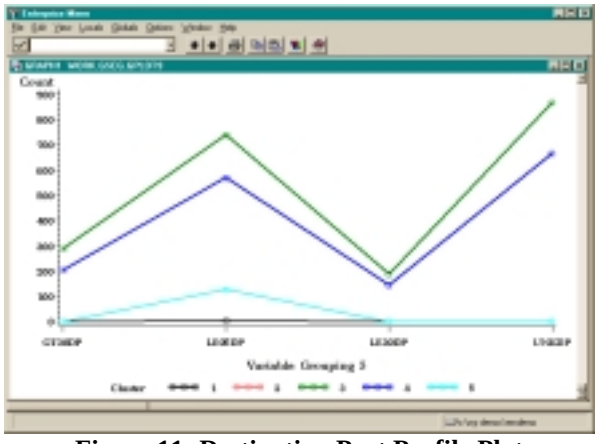

Figure 11: Destination Port Profile Plot

Table 5 pulls together the interpretations from all the profile plots and indicates the cluster size. We used this chart to help relate clusters to intrusion types.

	○ High Number		◎ Medium Number	○ Low Number	
Cluster Number	1	2	3	4	5
Approximate Size	**8500**	**1700**	**2**	**1900**	**56**
General Ports	○	○	◎	○	◎
Login 1 Ports	○	○	○	◎	◎
DNS Ports	○	○	◎	◎	○
Dest Ports	○	○	◎	○	◎
Dest Addrs	○	◎	○	◎	○
Source Addrs	○	◎	◎	◎	○
Opened Connections	○	○	◎	◎	○
Closed Connections	○	○	◎	◎	◎
Addrs <5 sec	○	○	◎	◎	◎
Ports < 5 sec	○	○	◎	◎	◎
Suspected Intrusion Type	**Normal Activity**	**?**	**?**	**IP Spoofing**	**Network Scanning**

Table 5: Relating Clusters to Intrusion Types

Cluster 3 has only two IP conversations in it, and for each the source and destination IP addresses are the same. Based on the activity displayed we suspect this cluster is finding the server for each simulation file. However, we would then expect to see 5 observations in the cluster - one for each simulation. To isolate this activity, we added a new variable to the data to indicate same source and destination addresses and re-ran the analysis. These results will be discussed later.

In an effort to understand the remaining segment, cluster 2, we used visualization software included with Enterprise Miner. Figures 12 and 13 display scatter plots of the variable combinations used for the profile plots earlier. The points in the plots are colored according to cluster membership. Notice that these plots reveal the same multi-dimensional spacing as figure x did, with clusters 1 and 5 being similar and clusters 2,3,and 4 being similar.

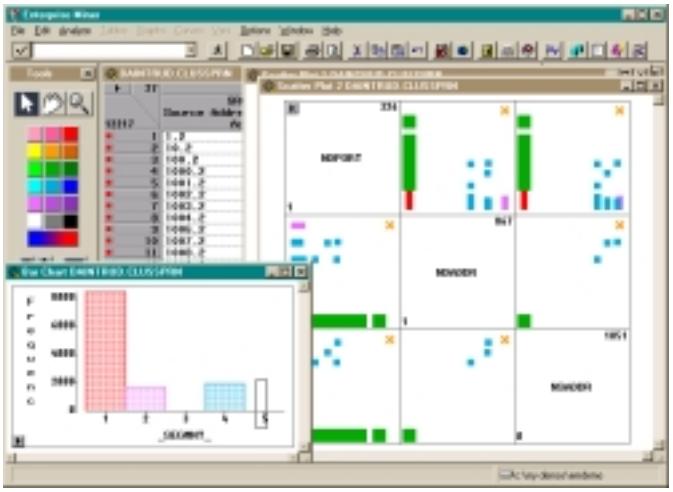

Figure 12: Exploring the Clusters Interactively, Part I

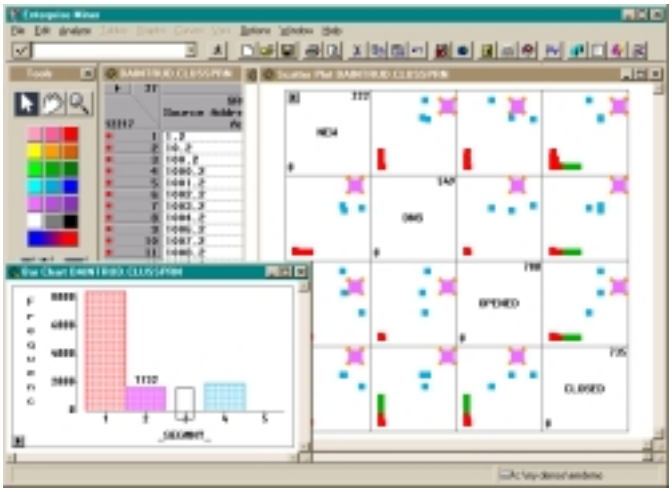

Figure 13: Exploring the Clusters Interactively, Part II

Using the dynamic linking capabilities of these charts, we can highlight all the observations in cluster 2. We can the extract them into their own data table for further investigation. Figure 14 displays the data for cluster 2 along with a bar chart for the source addresses present in the table. Cluster 2 is comprised of various conversations originating from source address 2 (note the IP addresses were cleansed in the raw data). Further, all the records in this cluster originated from the Hopping and IP Spoofing simulations. This cluster has found some behavior that is similar among activities in these two files.

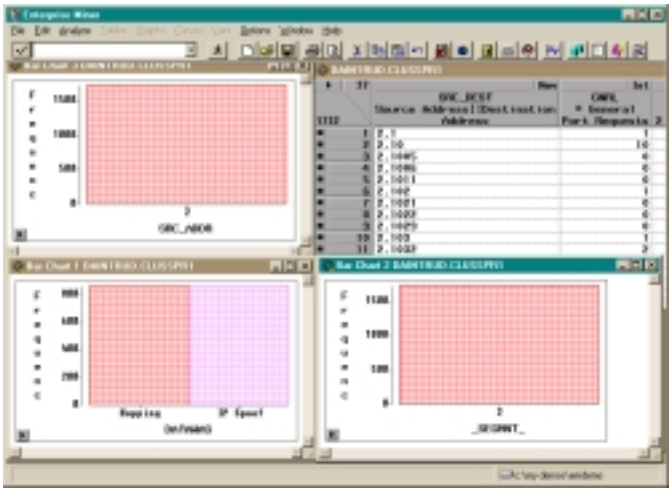

Figure 14: Isolating Cluster 2 Records

As a final check on the appropriateness of our analysis, we investigated how our clusters mapped to the original data sources. If the clusters were effectively mapping to different intrusion types, we would expect to see high counts in only 1 DATWAS category, with the exception on the normal activity cluster. From figure 15 it appears that we are isolating some specific intrusions, however increasing the number of clusters could improve the results.

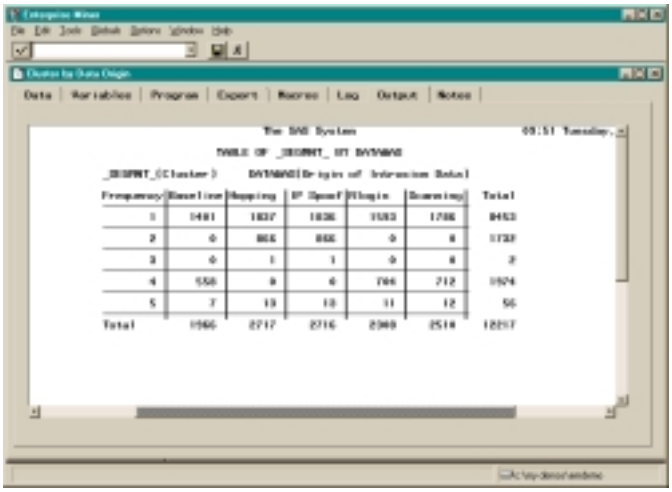

Figure 15: Cluster by Data Origin, 5 Clusters

Figure 16 displays the cluster by data origin table after increasing the number of clusters to 10 and including a variable indicating the same source and destination addresses. Cluster 5 has become the normal activity cluster. Cluster 3 is probably detecting network scanning. Cluster 6 is probably detecting RLOGIN activity. Cluster 10 should be investigated for IP spoofing. And many of the smaller clusters should be investigated for anomalous activity.

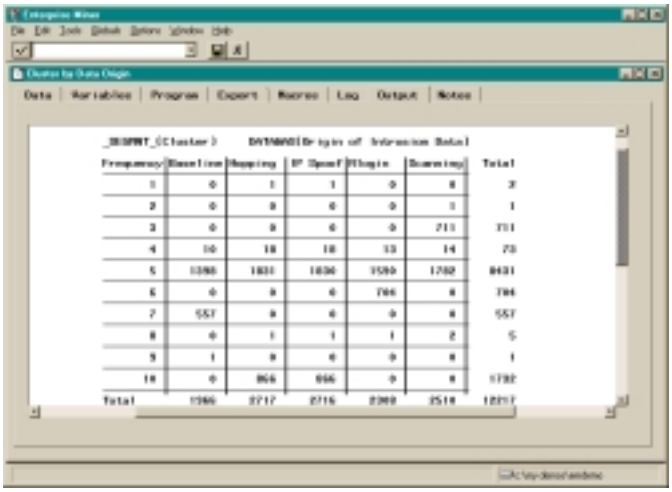

Figure 16: Cluster by Data Origin, 10 Clusters

CLOSED LOOP IMPLEMENTATION

Once the data mining analyst and the intrusion domain experts agree that the mining analysis is producing meaningful results, a strategy to automate the scoring of new log files should be developed. This strategy involves taking the data logged in the PDB, applying the data mining data transformations, then applying the cluster scoring algorithm to new data. Simple exception reports can then alert system administrators to new IP addresses and ports involved in potentially intrusive activity.

Building from Unsupervised to Supervised Learning

To further validity of the data mining results, investigators should also build a knowledge base of investigated records. This knowledge base should include not only confirmed intrusion, but also suspected intrusions that were not proved. The knowledge base can be used to validate the cluster model by feeding the known cases into a predictive model. Should investigation show the predictive model's judgment to be erroneous, the cluster analysis would need to be revisited.

The *validated* cluster model will continue to be applied to new data, producing cases to be investigated. In turn the knowledge base will accumulate known intrusive activity.

CONCLUSIONS

A combination of unsupervised data mining, data warehousing, and exception reporting allows system administrators to identify suspicious network activity, track intrusion occurrences, and automate the off-line intrusion detection process. If actual intrusive events are tracked in a warehouse, the intrusion team can build predictive models that validate unsupervised results.

As we move forward in this research we hope to address the following issues:

- Improve the logic used for determining the final conversation states.
- Explore methods of applying data mining models created for summarized data, to the raw transaction data.
- Explore how the mining results are affected by the choice of time period for data summarization and the effects of conversation censoring at beginning and end of the selected time periods.
- Add more descriptive variables to the mining data in an effort to achieve a fuller range of behavioral descriptors.
- Experiment with other clustering algorithms.

REFERENCES

An Introduction to Intrusion Detection and Assessment, Rebecca Bace, **www.iss.net/prod/whitepapers/intrusion.pdf**

Intrusion Detection: Network Security Beyond the Firewall, Terry Escamilla, ISBN: 0-471-29000-9

Information Insecurity, Government Executive, April 1999, **www.govexec.com/features/0499/0499s1.htm**

12 Mistakes to Avoid for Managing Web Security, CIO Institute, **info@cio.org**

A Perspective on New and Different Threats to Information Security, Intelligent Enterprise, Feb 1999

National Infrastructure Protection Center, **www.fbi.gov/nipc/Impdd-63.htm**

DARPA Intrusion Detection Evaluation, **www.ll.mit.edu/IST/ideval/index.html**

ABOUT THE AUTHORS

Kristin Nauta
Manager, Government Technology Center
Program Manager, Data Mining
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
Kristin.Nauta@sas.com
Phone: 919-677-8000, x4346

As manager of the Government Technology Center at SAS Institute Inc., Kristin Nauta is the federal program manager for data mining. Formerly SAS Institute's data mining program manager for Canada and the analytical products marketing manager for the US, Kristin has a BS in mathematics from Clemson University and a Masters of Statistics from North Carolina State University. Kristin has consulted in a variety of fields including pharmaceutical drug research and design, pharmaceutical NDA submissions, database marketing and customer relationship management.

Frank Lieble
Program Manager, IT Service Vision
Government Technology Center
SAS Institute Inc.
1900 Summit Tower Blvd
Suite 850
Orlando, FL 32810
Frank.Lieble@sas.com
Phone: 407-661-1711, x237

Frank Lieble is the program manager for the IT Service Vision solution at SAS Institute's Government Technology Center. His previous position was product manager for IT Service Vision at SAS Institute's Business Solutions Division. Frank has several years of experience in IT performance data warehousing which includes the management, reporting, and analysis of system and network performance data. He has a Bachelor of Science degree in Computer Science and a Master of Science degree in Statistical Computing from the University of Central Florida.