Achieving Query Efficiency in IDS

Ramya J[1]
Communication and Networking,
Trichy Engineering College,
Trichy, India

Hemalatha J[2]
Electronics and Communication Engineering,
K.Ramakrishnan College of Technology,
Trichy, India.

***Abstract* — This paper describes data mining and data warehousing techniques that can improve the performance and usability of Intrusion Detection Systems (IDS). Current IDS do not provide support for historical data analysis and data summarization. This paper presents techniques to model network traffic and alerts using a multi-dimensional data model and star schemas. This data model was used to perform network security analysis and detect denial of service attacks in distributed environment. Our data model can also be used to handle heterogeneous data sources (e.g. firewall logs, system calls, net-flow data) and enable up to two orders of magnitude faster query response times for analysts as compared to the current state of the art. Our system has helped the security analyst in detecting intrusions and in historical data analysis for generating reports on trend analysis.**

Keywords -- Data warehouse. OLAP. Data mining and analysis. Computer security. Intrusion detection

Keywords—component; formatting; style; styling; insert (key words)

I. INTRODUCTION

A. Data warehousing and data mining

Knowledge discovery encompasses algorithms and tools for bringing together data from distributed information repositories into a single repository that can be suitable for data analysis. Data warehousing and data mining has been used for data analysis applications in the area of retail, finance, network/Web services and bio-informatics. For example, a communications services provider is interested in collecting its network usage information and then identifies usage patterns, catch fraudulent activities, makes better use of resources and improve the quality of service.

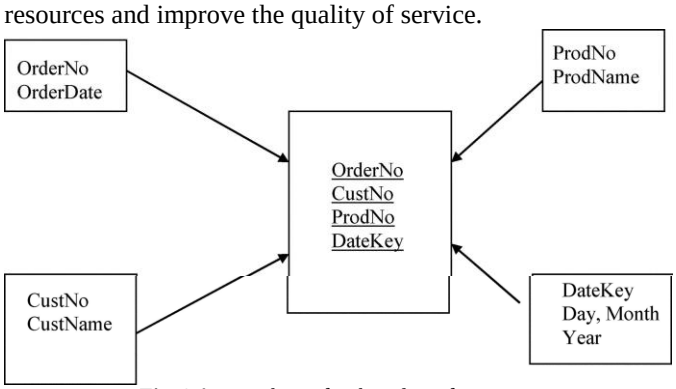


Fig. 1 A star schema for the sales information

Data warehousing and data mining techniques can analyze the data automatically, summarize it and predict future trends. Data Warehouse uses a data model that is based on a *multidimensional data model.* This model is also known as a *data cube* which allows data to be modeled and viewed in multiple dimensions. *Dimensions* are the different perspectives for an entity that an organization is interested in. For example, a store will create a *sales* data warehouse in order to keep track of the store' sales with respect to different dimensions such as *time*, *branch*, and *location.* "Sales" is an example of a central theme around which the data model is organized. This central theme is also referred as a *fact table. Facts* are numerical measures and they can be thought of as quantities by which we want to analyze relationships between dimensions. Examples of facts are *dollars sold*, *units sold* and so on. The *fact table* contains the names of the facts as well as keys to each of the related dimension tables .The entity-relationship data model is commonly used in the design of relational databases.

B. Introduction to intrusion detection systems

Since the cost of information processing and Internet accessibility is dropping, more and more organizations are becoming vulnerable to a wide variety of cyber threats. According to a recent survey by CERT, the rate of cyber-attacks has been doubling every year in recent times. Therefore, it has become increasingly important to make our information systems, especially those used for critical functions such as military and commercial purpose, resistant to and tolerant of such attacks. Intrusion Detection Systems (IDS) are an integral part of any security package of a modern networked information system. An IDS detects intrusions by

Monitoring a network or system and analyzing an audit stream collected from the network or system to look for clues of malicious behavior. Intrusion Detection Systems (IDS) can be categorized according to the kind of information they analyze. This leads to the distinction between *host-based* and *network-based* IDSs. A host based IDS analyzes host-bound audit sources such as operating system audit trails, system logs or application logs. Since host based systems directly monitor the host data files and operating system processes, they can determine exactly which host resources are targets of a particular attack. Due to the rapid development of computer networks, traditional single host intrusion detection systems have been modified to monitor a number of hosts on a network. They transfer the monitored information from

multiple monitored hosts to a central site for processing. These are termed as distributed intrusion detection systems. A network based IDS analyzes network packets that are captured on a network. This involves placing a set of traffic sensors within the network. The sensors typically can be classified into the following categories:

1) Misuse Detection: This method finds intrusions in search of direct matches to known patterns of attack (called signatures or rules). A disadvantage of this approach is that it can only detect intrusions that match a pre-defined signature. Therefore, signature based methods cannot detect unknown attacks and they cannot protect systems from such attacks. One advantage of these systems is that they have low false alarm rates. SNORT is a widely used open source signature based network IDS which is used to perform real time traffic logging and analysis over IP networks. In data mining methods for misuse detection, each instance in a data set is labeled as 'normal' or 'intrusive' and a learning algorithm is trained over the labeled data

2) Anomaly Detection: In this approach, the system defines the expected behavior of the network in advance. The profile of normal behavior is built using techniques that include statistical methods, association rules and neural networks. Any significant deviations from this expected behavior are reported as possible attacks.

3) State Transition Based Intrusion Detection: In this approach a finite state machine is used to model different IDS states and transitions characterize certain events that cause IDS states to change.

C. The following are some of the disadvantages of a data mining based IDS.

1) Data must be collected from a raw data stream and translated into a form that is suitable for training. In some cases data needs to be clearly labeled as "attack" or "normal". This process of *data preparation* is expensive and labor intensive.
2) Data mining based IDS generally do not perform well when trained in a simulated environment and then deployed in a real environment. They generate a lot of false alarms and it can be quite labor intensive to sift through this data.

In order to overcome these problems, there is a need to develop methods and tools that can be used by the system security analyst to understand the massive amount of data that is being collected by IDS, analyze and summarize the data and determine the importance of an alert.. In this paper, we present data modeling, data visualization and data warehousing techniques that can drastically improve the performance and usability of Intrusion Detection Systems. Data warehousing and On Line Analytical Processing (OLAP) techniques can help the security officer in detecting attacks, monitoring current activities on the network, historical data analysis about critical attacks in the past, and generating reports on trend analysis. We present techniques for feature extraction from network traffic data and how a multi-dimensional data model or STAR schemas can be used to represent network traffic data and relate it to the corresponding IDS alerts. This paper is organized as follows. We first give a survey of research projects that apply data mining techniques to intrusion detection.

a) Historical Data Analysis: As networks are getting large and complex, security officers that are responsible for managing these networks need tools that help in historical data analysis, generating reports and doing trend analysis on alerts that were generated in the past. Current IDS often generate too many false alerts due to their simplistic analysis. The storage management of alerts from IDS for a complex network is a challenging task.

b) Support for Real Time Alert Correlation: Intrusion correlation refers to interpretation, combination and analysis of information from several sensors

c) Heterogeneous Data Support: In a typical network environment, there are multiple audit streams from diverse cyber sensors (1) raw network traffic data (2) net flow data (3) system calls ,output alerts from an IDS and so on. Since current IDS are not perfect they produce a lot of false alarms. There is a need for efficient querying techniques for a user to verify if an alert is genuine by correlating it with the input audit data.

d) Feature extraction from Network Traffic Data and Audit Trails: For each type of data that needs to be examined (network packets, host event logs, process traces etc.) data preparation and feature extraction is currently a challenging task. Due to large amounts of data that needs to be prepared for the operation of IDS system, this becomes expensive and time consuming.

e) Data Visualization: During attack, there is a need for the system administrator to graphically visualize the alerts and respond to them. There is also a need to filter and view alerts, sorted according to priority, sub-net or time dimensions.

II. SYSTEM DESIGN

A. System architecture for IDS

In this section we propose a set of techniques that will considerably improve the performance of intrusion detection systems. The improvement is focused in the area of multi-dimensional data model that can be used to represent alerts and to detect new kinds of attacks. Techniques for feature extraction from network traffic data and alert correlation are also presented.

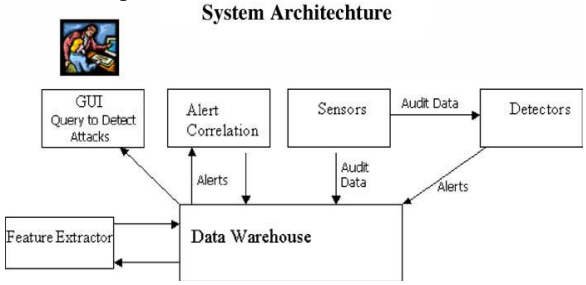


Fig. 4 Data architecture for intrusion detection system

B. A software architecture and data model for intrusion detection

Figure 4 shows an architecture diagram of our system. In a typical network environment there are many different audit streams that are useful for detecting intrusions. For example, such data includes network packets (headers, payload features), system logs on the host and system calls of processes on these machines. These types of data have different properties. Also, the detection models can vary. The most widely used detection model is a signature based system while data mining based approaches are also being explored. It is important to have an architecture that can handle any kind of data and different detection models. Our architecture supports the following components

1) Real time components that includes sensors and detectors
2) A data warehouse component to store the data efficiently
3) Feature extraction component that reads the audit data from the data warehouse, extracts some features and computes some aggregates and then stores the information back in the data warehouse. These features are useful to the analysts to detect attacks.
4) Visualization engine that presents information to the analyst.

The architecture we proposed has several advantages:

1) *Modularity*: All the data is stored in one central place and can be easily queried by the security analyst or the intrusion detection applications.
2) *Support for multiple detectors*: We have separated the sensor component from the detector component. This allows us to use a signature based detection engine and a data mining based detection engine on the same set of audit data.
3) *Correlation of audit data from multiple sensors*: Since the data from multiple sensors is stored in one central place, a detection engine can easily access the data from multiple sensors by executing a database query.
4) *Reusability*: Since the features extracted from the audit data are stored in one central place, they can be re-used by multiple applications to detect attacks.

III. SYSTEM ANALYSIS

A. Data modeling for historical data analysis using STAR schema

If an alert needs further investigation we plan to support the capability of querying and browsing a historical database. We propose to model the alert data as a multidimensional dataset and borrow the model used in On Line Analytical Processing (OLAP)In our case, the underlying relation is the alerts that are generated from IDS. The alerts can be viewed as a multidimensional data. This schema is known as the *star schema.* In it, the main table is called the *fact table.* The attributes are the dimensions of the data. Examples of dimensions are *Time Date*, *Sdinfo*, *Service*, *and Attack. Time Date* contains information of date and time when the attack was staged. *Sdinfo* describes the Source/Destination IP addresses and destination port information. This dimension encompasses a hierarchy which shows how this information can be aggregated to produce different views. Both, the source and destination IP addresses are composed of 4 bytes *Sip1Sip2Sip3Sip4* and *Dip1Dip2Dip3Dip4.* Dropping one or more of these fields produces a higher level view of the address. For example, *Sip1Sip2* corresponds to a series of domain of IP addresses each characterized by the first 2 bytes of the address. The *Service* dimension table contains the service (or protocol e.g. ftp, http) name that was attacked and the class of service (e.g. TCP, UDP). The hierarchy for these dimensions is also shown. For example, the service ftp and http belong to the TCP class. Similarly, the dimension table contains *Attack* contains both the name of the attack and its type (e.g. DOS, Probe). The dimension *Time Date* presents different views of timing information. Finally, the attribute *Duration* contains the length of the attack. This can also be viewed as *long*, *medium* or *short.* Figure 5 shows the STAR schema. Using this schema, a corresponding cube would be a five dimensional structure in which cell contains aggregates of the operations measures

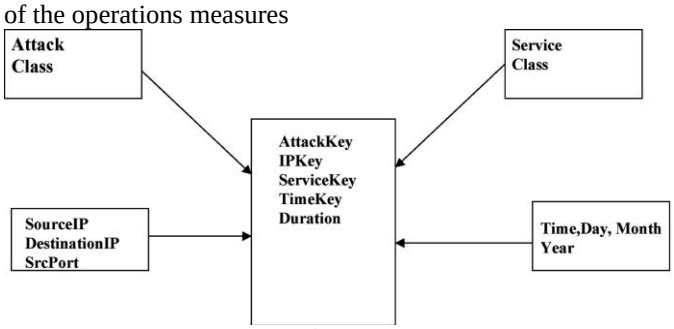


Fig. 5 A star schema for the IDS data warehouse

B. Support for high speed drill down queries and detection of attacks/virus/worms

When an alert is generated by an IDS the analyst is interested to "drill down" and check the corresponding "raw network traffic" data to verify the alert.

Scanning Activity: Process one hour of data and look for all flows where the SYN flag was set and ACK/FIN flags are not set.

- Recently, Sasser worm was released that scans port 445. To detect this worm we wrote queries to search for flows that scan for port 445. If the analyst is interested in internal machines that have been infected he can narrow the search to only those machines with destination port 445.Aquerywaswritten that would generate the top ten source-destination IP pairs on destination port 445 for net flow data during a certain period of time.
- Another security concern is denial of service attacks. One of the common network based denial of service attacks is SYN flooding. We have written queries which are similar to those for worm detection to detect if a SYN flood has occurred. In this case we detected all source-destination IP pairs that have seen an excessive number of SYN packets.
- Worm Detection: Recently, the MyDoomworm spread via an email attachment that created a backdoor on ports 3127–3198. After the release of this worm, scanning for this backdoor increased significantly. We have written SQL queries to generate reports about the number of flows caused by this scanning in 10 minute intervals. The report shows that there is a sudden jump in the number of

bytes transferred, even though the number of flows stayed constant.

C. Feature extraction from network traffic data

A number of data mining based IDS applications need to pre-process the network traffic data before they can do their analysis. For example, the preprocessing module of ADAM generates a record for each connection from the header information of its packets based on the following schema:

R (TS, Src.IP, Src.Port, Dest.IP, Dest.Port, FLAG)

In this schema, TS represents the beginning time of a connection, Src.IP and Src.Port refer to source IP and port number respectively, while Dest.IP and Dest.Port represent the destination IP and port number. The attribute FLAG describes the status of a connection. This relation R is used for association mining. We store the connection records in the data warehouse so that they are available in one central place by several applications to do the analysis.

D. Help the security officer for forensic analysis

One of the important kind of analysis is forensic analysis. Currently forensic analysis of data is done manually. Computer experts have to search through large amounts of data, sometimes millions of records, individually and look for suspicious behavior. This is an extremely inefficient and expensive process. Since we can store all the historical data (net-flow data, system calls, fire-wall logs) in a data warehouse we can help the security officer in accessing all the records which are suspicious and possibly have some intrusions. The suspicious activity can then be labeled as either anomalous or normal using SQL statements to mark the appropriate data. Since all the data is stored in a data warehouse we can update the record and store it back in the database. Our database platform can be used to design Digital Forensics tools tailored to Information Warfare that can provide real time performance.

III SYSTEM IMPLEMENTATION

Figure 6 shows an architecture diagram of a prototype data warehouse system for Intrusion Detection.

A. First SNORT was installed and configured to read tcpdump files and store the alerts into the database. A data warehouse based on ORACLE 9i is the center piece of our architecture. We store the following kinds of data in the data warehouse.

1) Alert data

We created tables to store the alerts from SNORT. Some of these tables are event, sensor, signature and detail.

2) Network traffic data and extracted features

We created tables that correspond to network packet headers and designed a database schema to store tcpdump data. Five tables were created which correspond to Ethernet header, IP header, TCP header, UDP header and ICMP header.

a) The data we used for our project was collected by DARPA 98' . We used tcpdump data for roughly one month of network traffic as collected by a tcpdump packet sniffer. This data contains the contents of every packet transmitted between computers inside and outside a simulated military base.

b) We created loaders using Java/JDBC to load network traffic data into the ORACLE tables. We also designed programs to extract features and do data cleaning before loading it into the data warehouse.

c) We have also created a schema for storing *net flow* data and we have designing programs using database queries that do security analysis (e.g. detection of slow port scans) using net flow data. The slow port scans cannot be detected by SNORT because of time window limitations.

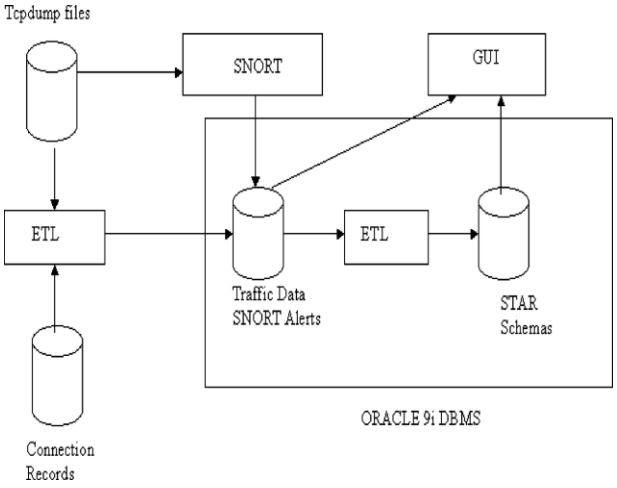


Fig. 6 Architecture of the data warehouse system for IDS

B. Aggregated data, STAR schemas and data summarization reports

d) We created STAR schemas to store the SNORT alerts and designed a extract/ transform/load program that can read data from the SNORT database and load it into the STAR schemas. Since STAR schemas use data aggregation they reduced the amount of storage usage for past history

Table 1 Detection time for DOS attacks

Intrusion type	Intrusion name	Detection time (s)
Denial of service	Smurf	95
Denial of service	SYN flood	64

e) We also created STAR schemas to store the net-flow data. Since STAR schemas use data aggregation they reduced the amount of storage usage for past history. We wrote several queries to determine how the STAR schemas helped in improving the performance of detecting attacks such as Smurf and SYN flood. This schema also helped in creating Data Summarization Reports such as Top N Lists and Black List IP. One of the reports is to sort the results by the number of flow records and then only return the first N, from the list to

present the results with the most traffic. This feature can be used by the user specifying either source or destination ip address. It can also be used on ports rather than IP address. If the user wishes, he can also look at combination of items such as top source/destination ip address pairs or source/destination port pairs.

Detection of DOS attacks

Since raw network traffic is stored in the database, we wrote queries that can detect certain kind of attacks such as Smurf and SYN flood.

a. Smurf attack is a scenario when there are a large number of replies to a particular machine from many different machines, but no "echo request" originated from that victim machine.
b. SYN flood is a scenario where there are a number of SYN packets coming to a particular machine from an unreachable host.

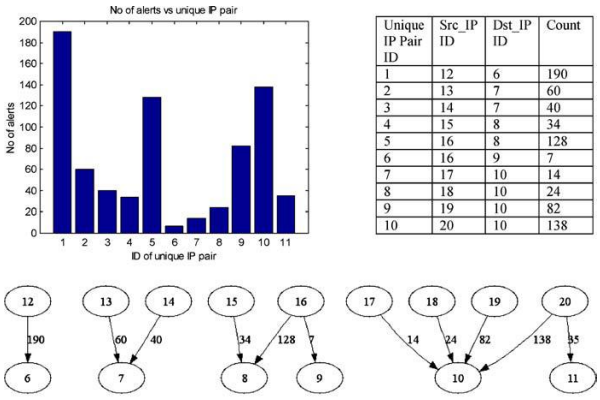
Table 1 also gives the time to detect these attacks for a ORACLE 9i database of 10 million records running on a SUN Sparc Station.

Security analysis

Attackers often perform scan on a network before they do the real attack. This type of activity will appear as a SYN scan where there might be some further communication with internal systems that respond to the scanner with a SYN-ACK. We have written queries that process one hour of data looking for all flows where the SYN flag was set and the ACK and FIN flags were not set.

Table 2 Performance comparison of query execution time

Execution time	SNORT database	Star schema	Speed up
Query a	1.56	0.22	7
Query b	40.28	0.45	90
Query c	58.01	1.08	55



Unique IP Pair ID	Src_IP ID	Dst_IP ID	Count
1	12	6	190
2	13	7	60
3	14	7	40
4	15	8	34
5	16	8	128
6	16	9	7
7	17	10	14
8	18	10	24
9	19	10	82
10	20	10	138

Fig.7 Visualization of alerts

We have developed GUI (using JAVA and JDBC) to display the alerts and sort them according to time, SourceIP, DestinationIP and so on. We can also display the alerts as Graphs where nodes are the IP address using the Graph. We can also display bar charts where *x* axis is the IP address and *y* axis is the number of alerts during a certain period of time. The following query was used to visualize alerts.

a. Give a count of all alerts on July 20th 2002 between 9 AM and 10 AM. Group the results by Source IP and Destination IP.

Allowing visualization of discovered patterns will help users to identify patterns of interest and to interact or guide the system in further knowledge discovery. A user should be able to specify the forms of presentation to be used for displaying the discovered patterns. We are also investigating techniques of concept hierarchies to implement *drill-down* and *roll-up* so that the security officer can inspect discovered patterns at multiple levels of abstraction.

7. CONCLUSION

This paper described data modeling and data warehousing techniques that drastically improve the performance and usability of Intrusion Detection Systems (IDS). Current IDS do not provide support for historical data analysis and data summarization. This paper presented techniques to model network traffic and alerts using a multi-dimensional data model and *star schemas*. This data model was used to perform network security analysis and detect denial of service attacks. Our data model can also be used to handle heterogeneous data sources (e.g. firewall logs, system calls, net-flow data) and enable up to two orders of magnitude faster query response times for analysts as compared to the current state of the art. Our system has helped the security analyst in detecting intrusions and in historical data analysis for generating reports on trend analysis. Future work will include expanding the correlation capabilities of our system. We can combine data from multiple sources or sensors to obtain a better picture of the activity in a network. This will allow us to study the relationship between data gathered from multiple sensors about the same attack. We are also looking into main memory database techniques for real time alert correlation and visualizing attack scenarios.

REFERENCES

1. W.H. Inmon, Building the Data Warehouse, 2nd edn. John Wiley, 1996. R. Kimball, The Data Warehouse, John Wiley, Toolkit, 1996.
2. S. Chaudhuri and U. Dayal, "An overview of data warehousing and OLAP technology," SIGMODRecord, March 1997.
3. J. Han and M. Kamber, Data Mining Concepts and Techniques, Morgan Kaufmann, August 2000.
4. W. Lee, S.J. Stolfo, and K.W. Kwok, "Mining audit data to build intrusion detection models," in Proc. Fourth International Conference on Knowledge Discovery and Data Mining, NewYork, 1998.
5. D. Barbara, J. Couto, S. Jajodia, and N. Wu, "Adam: Detecting intrusions by data mining," in Proc. 2nd Annual IEEE Information Assurance Workshop, West Point, NY, June 2001.
6. T. Abraham, "IDDM: Intrusion detection using data mining techniques," Technical Report DSTO-GD-0286, DSTO Electronics and Surveillance Research Laboratory, 2001.