A NOVEL APPROACH DESIGN IN INTERFERENCE FINDING SYSTEM FOR DATA MINING USING MADAM ID

R.Venkatesan*

R. Ganesan**

A. Arul Lawrence Selvakumar***

ABSTRACT

Intrusions are the activities that violate the security policy of system. Intrusion Detection is the process used to identify intrusions. Network security is to be considered as a major issue in recent years, since the computer network keeps on extending dramatically. Information Systems and Networks are subject to electronic attacks and the possibilities of intrusion are very high. In order to protect the networking system, it is mandatory to update and install Intrusion Detection System (IDS) due to the expansion of network dramatically every day along with new threats and attack. We do know current IDSs are constructed with interception of Data Mining techniques and Intrusion Detection. We also used the Data Mining techniques to design this interference finding system. The objective of this paper is state our novel approach design in Intrusion Detection for Data Mining using MADAM ID. We have analyzed this novel approach using Network Flight Recorder (NFR).

Keywords: *Data Mining, Intrusion Detection System (IDS), Network Security, Misuse Detection, Anomaly Detection, Classification, Clustering, MADAM ID, NFR*

*Research Scholar/CS, CMJ University, Shillong, Meghalaya

**Professor, Dept of E& I Engineering, N I C E, Kumaracoil, Tamil Nadu

***Professor & Head, Dept of CSE, Rajiv Gandhi Institute of Technology, Bangalore, Karnataka

1. INTRODUCTION:

In recent years many researchers are focusing to use Data Mining concepts for Intrusion Detection. Data mining is a process to extract the implicit information and knowledge which is potentially useful and people do not know in advance, and this extraction is from the huge data. In the other hand *intrusions* in an information system are the activities that violate the security policy of the system, and *intrusion detection* is the process used to identify intrusions. The objective of this paper is state our novel approach design in Intrusion Detection for Data Mining using MADAM ID in NFR (Network Flight Recorder Inc., 1997)[1], a system includes packet capturing engine and N-Code programming support for specific packet filtering logic. This is an offline evaluation, but an effective ID should be in real-time to satisfy the security policy an organization.

2. INTRUSION DETECTION TECHNIQUES:

The intrusion detection techniques based upon data mining [2], [3] are generally falls into one of two categories: *anomaly detection* and *misuse detection*. The signatures of some attacks are known, whereas other attacks only reflect some deviation from normal patterns.

2.1 Anomaly Detection

Anomaly detection attempts to determine whether deviation from an established normal behavior profile can be flagged as an intrusion [4]. Anomaly detection consists of first establishing the normal behavior profiles for users, programs, or other resources of interest in a system, and observing the actual activities as reported in the audit data to ultimately detect any significant deviations from these profiles. Most anomaly detection approaches are statistical in nature.

2.2 Misuse Detection

Misuse detection works by searching for the traces or patterns of well-known attacks. Lee et al. [5] designed a signature-based database intrusion detection system (DIDS) which detects intrusions by matching new SQL statements against a known set of transaction fingerprints. Misuse detection is considered complementary to anomaly detection.

2.3 Pros and Cons of Anomaly and Misuse Detection

Table 1: Pros and Cons of Anomaly Detection and Misuse Detection

Technique	Pros	Cons
Anomaly Detection	Is able to detect unknown attacks based on audits	High false-alarm and limited by training data.
Misuse Detection	Accurately and generate much fewer false alarm	Cannot detect novel or unknown attacks

2.4 Drawbacks of current IDS

Intrusion Detection Systems (IDS) has become a standard component in security infrastructures as they allow network administrators to detect any violations. These security violations range from external attackers trying to gain unauthorized access to insiders abusing their access. Current IDS have a number of significant drawbacks [6]:

- *False Positives* – A common complaint is the amount of false an IDS will generate.
- *False Negatives* – In this case, IDS does not create a signature or alarm, when an intrusion is actually happened.
- *Data Overload* – In this aspect, Misuse Detection cannot be related directly, however, it is very important to analyze how much data an analyst can efficiently and effectively analyze.

2.5 Need of using data mining approaches in IDS

A team in Minnesota University (1990) recognized the need for existence of standardized dataset to train IDS tool. Minnesota Intrusion Detection System (MINDS) combines signature based tool with data mining techniques. Signature based tool (Snort - freeware) are used for misuse detection & data mining for anomaly detection. The reasons for using Data Mining approaches in IDS are:

1. It is very difficult to build IDS using programming languages, which requires more explicit data and functional knowledge.
2. The reliability, compatibility and dynamic nature of machine-learning make it a suitable solution for this situation.
3. The environment of an IDS and its classification task highly depend on user-driven preferences.

3. DATA MINING APPROACHES

Data mining generally refers to the process of (automatically) extracting models from large stores of data [7]. The recent rapid development in data mining has made available a wide variety of algorithms, drawn from the fields of statistics, pattern recognition, machine learning, and database. There are several types of algorithms [7] which are particularly related to intrusion detection.

- **Classification**: classifies a data item into one of several pre-defined categories. These algorithms normally output “classifiers”. An ideal application in intrusion detection would be to gather sufficient “normal” and “abnormal” audit data for a user or a program, then apply a Classification algorithm to learn a classifier that can label or predict new unseen audit data as belonging to the normal class or the abnormal class.

- **Link analysis**: determines relations between fields in the data base records. Correlations of system features in audit data, for example, the correlation between command and argument in the shell command history data of a user, can serve as the basis for constructing normal usage profiles.
- **Sequence analysis**: models sequential patterns. These algorithms can discover what time-based sequences of audit events are frequently occurring together. These frequent event patterns provide guidelines for incorporating temporal and statistical measures into intrusion detection models.

3.1. Systematic Framework

Our framework consists of data-mining programs for learning detections models, a translator for converting learned rules to real-time models, and NFR for capturing network traffic and applying the real-time N-code modules for ID. A framework has been developed, first proposed in [4], of applying data mining techniques to build intrusion detection models. This framework consists of programs for learning classifiers as well as a support environment that enables system builders to interactively and iteratively drive the process of constructing and evaluating improved detection models. The end product of this process is a set of concise and intuitive rules (that can be easily inspected and edited by security experts when needed) that can detect intrusions. The rules are then subsequently ported over to N-code as sub-routines or independent functions.

First, the network security expert needs to analyze and categorize attack scenarios and system vulnerabilities, and then we need to code the corresponding rules and patterns manually in N-code for misuse detection. In this manual development process, current IDSs including NFR have limited extensibility and adaptability. Our aim is to develop IDS which should be substantial to reduce this effort by automating:

1) The task of building intrusion detection through data mining.

2) Generating the N-code for NFR to detect intrusions via a *machine translator*.

3.2 Mining Data to Construct Attributes

In order to mine the data, first we must process and summarize packet-level network traffic data into “connection” records. We initially start out with the raw audit data (commonly *tcpdump* binary output) of the designated network we wish to monitor. This is then subsequently pre processed into individual packets/events in the ASCII format. As the packets are summarized according to their separate connections, we record their within connection features which may be deemed as “traditional attributes” of a connection record. We use the mined patterns from network connection records as guidelines to construct temporal statistical attributes for building classification models [9]. We performed pattern mining and comparisons using intrusion data of several well known

attacks e.g., port-scan, ping-sweep, etc., as well normal connection records. Each of the unique intrusion patterns are used as guidelines for adding additional features into the connection records to build better classification models.

3.3 Learning Detection Rules

We apply RIPPER [10] to the connection records to generate the classification rules for the intrusions. Like other rule learning systems, this method is used for classifications problems. “*count* “ the count of such connections rej count the count of connections that get the flag “REJ” met by a particular host *S01* count the count of connections that send a SYN packet but never get the ACK packet (S0), or receive an ACK on SYN that they never have sent (s1) diff services the count of unique (different) services diff srv rate diff services / count...

A “training” period is initially required for RIPPER to gather the necessary data on the network to compute models. The purpose is two-fold:

1) Establishing “normal” traffic patterns and variants that the network may encounter to establish anomaly detection,
2) Introducing known intrusion methods and attack scripts into the network in order to inductively learn the classification models of intrusions.

An example rule has been given here for better understanding and it is used to detect known attacks. In particular, when we illustrate how to detect and recognize an attack which is categorized as *denial-of service*.

Rule Translation

A detection rule, for example,

pod :- wrong_fragment >= 1, protocol_type = icmp. can be automatically converted into the following N-code:

```
filter pod ()
{
if (wrong_fragment() > 1 &
protocol_type () == icmp)
alarm_pod ();
}
```

As long as the features, i.e., wrong_fragment and protocol type, have been implemented as N-code filter functions.

3.4. Network Flight Recorder (NFR)

The Network Flight Recorder (NFR) [1] is one such extensible system that combines data collection, analysis, and storage within a single platform. IDS would normally be located between a firewall and an Internet connection, an area aptly named the *DMZ*(*De-Militarized Zone*) . This offline analysis in NFR is accomplished by scripts based on a language called *N-code*, NFR's flexible language for traffic analysis. Information is displayed in NFR will be transferred to a Web-based interface with the Java support. NFR also has a real time alerting capability and a storage subsystem that allows data to be stored and transferred to other external devices [1]. We use the frequent episodes algorithms [8] for this analysis.

3.5 Generation of N-Code Filters

The attributes of connection records are implemented as subroutines that may be called upon to check the rules that were generated by machine learning. A RIPPER rule simply consists of a sequence of attribute value tests, with each attribute implemented as an N-code filter, a rule can be automatically translated into an N-code filter that consists of a sequence of unction calls to the N-code filters.

4. EFFICIENT EXECUTIONS OF LEARNED RULES

Our plan is to implement all the required features used in the RIPPER rule set as N-code "feature filters", and implement a translator that can automatically translate each RIPPER rule into an N-code "rule filter". Although often ignored in off-line analysis, efficiency is a very important consideration in real-time intrusion detection. In our first experimental implementation of N-code "rule filters", we essentially tried to follow the off-line analysis steps in a real-time environment. A connection is not inspected (i.e., classified using the rules) until its connection record is completely formulated, that is, all packets of the connection have arrived and summarized, and all the temporal and statistical features are computed. This scheme failed miserably. When there is a large volume of network traffic, the amount of time taken to process the connection records within the past 2 seconds and calculate the statistics is also very large. During, the execution many connections may have terminated (and thus completed with attack actions) when the current connection is finally inspected by the RIPPER rules. That is, the detection of intrusions is severely delayed. Ironically, DOS attacks, which typically generate a large amount a traffic in a very short period time, are often used by intruders tofirst overload an IDS, and use the detection delay as a window of opportunity to quickly perform their malicious intent. For example, they can seize control of the operating system and “kill” the IDS. MADAM ID can be used to learn the site-specific intrusion detection rules using the locally gathered audit data, and be automatically converted in to N-code “rule filters”.

5. CONCLUSION

In this research paper, we studied the problem of how to automatically construct features from the mined patterns. We have applied relevant algorithms to construct the required model and also open the scope for further research in the areas of Misuse detection and IDS management in networking environment (large scale). The main challenge is how to efficiently execute the rules in a real-time & large-scale networking environment.

REFERENCES

[1] Inc. Network Flight Recorder. Network flight recorder. http://www.nfr.net, 1997.

[2] Daniel Barbara, Ningning Wu and Sushil Jajodia Detecting novel network intrusion using bayes estimators. In Proceedings of First SIAM Conference on data mining Chicago, 2001.

[3] Eric Bloedorn, Alan D. Christiansen, William Hill, Clement Skorupka, Lisa M. Talbot, and Jonathan Tivel. Data mining for network intrusion detection: How to get started.

[4] W. Lee and S. J. Stolfo. Data mining approaches for intrusion detection. In Proceedings of the 7th USENIX Security Symposium, San Antonio, TX, January 1998.

[5] S.Y. Lee, W. L. Low and P. Y. Wong, “Learning Fingerprints for a Database Intrusion Detection System”, In Proceedings of the 7th European Symposium on Research in Computer Security, Pages 264-280, 2002.

[6] (SANS: FAQ: Data Mining in Intrusion Detection) http://www.sans.org/security-resources/idfaq/data_mining.php

[7] W. Lee, S.J. Stolfo, K.W. Mok, Algorithms for Mining System Audit Data, in Proc. KDD, 1999.

[8] H. Mannila, H. Toivonen, and A. I. Verkamo. Discovering frequent episodes in sequences. In *Proceedings of the 1st International Conference on Knowledge Discovery in Databases and Data Mining,*

[9] W. Lee and S. J. Stolfo. A data mining framework for building intrusion detection models. In *1999 IEEE Symposium on Security and Privacy.*, Oakland, CA, May 1999.

[10] W. W. Cohen. Fast effective rule induction. *In Machine Learning: the 12th International Conference*, Lake Taho, CA, 1995. Morgan Kaufmann.