COMPUTATIONAL INTELLIGENCE FOR EVALUATION OF ATTACKS IN THE NETWORK

Visumathi. J[1] and Dr.Shunmuganathan K.L.[2]

Researh Scholar, Sathyabama University, TamilNadu, India – 600119.
Professor & Head, Department of CSE, RMK Engineering College, TamilNadu, India – 601 206

Abstract

Intrusion detection system(IDS) has recently emerged as an important component for enhancing information system security. Intrusion detection technology is an effective approach to dealing with the problems of network security. In this paper, we present a data mining-based network intrusion detection framework in real time (NIDS). This framework is a distributed architecture consisting of sensor, data preprocessor, extractors of features and detectors. To improve efficiency, our approach adopts a novel FP-tree structure and FP-growth mining method to extract features based on FP-tree without candidate generation. FP-growth is just accord with the system of real- time and updating data frequently as NIDS. We employ DARPA intrusion detection evaluation data set to train and test the feasibility of our proposed method. Experimental results show that the performance is efficient and satisfactory. Finally, the development trend of intrusion detection technology and its currently existing problems are briefly concluded.

1. Introduction

With the evolution of the technology of information, especially the prevalence of the technology of Internet/Intranet, security of more and more organization and individual's computer system establishment and information resource was threatened. Therefore, the security of information is become the one of the most important task in the domain of the technology of information. Traditional model of intrusion detection is been established inefficient and the cost of research is so much. The technology of data mining takes on particular predominance in the domain of unexpected knowledge acquiring. Thereby Data Mining-based Intrusion Detection is become prevalent [1], [2], [3], [4]. In essence, Network security is just network information security. In general, all technologies and theories about secrecy, integrality, usability, reality and controllable of network information are the research domain of network security. Intrusion is an action that tries to destroy that secrecy, integrality and usability of network information, which is unlicensed and exceed authority. Intrusion Detection is a positively technology of security defend, which gets and analyses audit data of computer system and network from some network point, and to discover whether there is the action of disobeying security strategy and whether be assaulted. Intrusion Detection System is the combination of software and hardware of Intrusion Detection System.

The rest of the paper is organized as follows. Section 2 describes how to extract features from audit data. Section 3 outlines the main components of our framework. Section 4 reports the results of our experiments. Section 5 draws the conclusion.

2. Feature Extraction for NIDS

Feature extraction adopts a FP-tree structure and FP-growth mining method [7] based on FP-tree without candidate generation, which optimized from Apriori algorithm. FP-growth is just adapt to the system of real time and updating data frequently like NIDS. Apriori [6] is a basal algorithm of generating frequent patterns. Apriori employs an iterative approach known as a level-wise search, where k-itemsets are used to explore $(k+1)-i$ temsets. Apriori is an influential algorithm for mining frequent itemsets for Boolean association rules. Many association-mining algorithms evolve from it. In some application cases the Apriori behave not as good as expect (i.e., need to repeatedly scan the itemsets, inefficient, consuming abundant resource of CPU). FP-growth is optimized algorithm from Apriori. FP-growth adopts a divide-and-conquer strategy that compresses the database representing frequent items into a frequent-pattern tree (FP-tree), and proceeds mining of the FP-tree. FP-tree is a good compact tree structure, which contains the complete information of the database in relevance to frequent pattern mining, and its size is usually highly compact and much smaller than its original database. The method is highly compressed so frequent item-sets generation is integrated and don't need to repeatedly scan the itemsets. Therefore NIDS adopts FP-growth, and the conclusion is whether resource using or efficiency is advanced.

The main steps of FP-growth method are as follows:

(i) Construct conditional pattern base for each node in the FP-tree.

(ii) Construct conditional FP-tree from each conditional pattern-base.

(iii) Recursively mine conditional FP-trees and grow frequent patterns obtained so far.

(iv) If the conditional FP-tree contains a single path, simply enumerate all the patterns.

Let's look at an example of extraction of features

Example 1

This example based on preprocessed data of Table 1. Assume the minimum support count is 2. There are four transactions in this database. Fig.1 is the FP-tree constructed from the Table 1. Table 2 is the first scan of the database candidates 1-itemsets and their support counts. Table 3 shows the result

Table 1: Preprocessed Data

TID	Items
T100	TCP-po, 192.168.0.1-sIP, 80-sPt, 192.168.0.2-dIP, 1717-dPt
T200	TCP-po, 202.198.16.220-sIP, 80-sPt, 10.60.46.58-dIP , 2209-dPt
T300	TCP-po, 192.168.0.1-sIP, 3050-sPt, 192.168.0.2-dIP, 1717-dPt
T400	TCP-po, 202.198.16.220-sIP, 80-sPt, 10.60.46.58-dIP, 1717-dPt

Table 2: Frequent Item set and support count

Itemset	Support count
TCP-po	4
80-sPt	3
1717-dPt	3
192.168.0.1-sIP	2
202.198.16.220-sIP	2
192.168.0.2-dIP	2
10.60.46.58-dIP	2
3050-sPt	1
2209-dPt	1

3. The Framework and Implement of NIDS

The overall system distributed architecture framework is designed to support a data mining- based NIDS, as shown in Fig. 2. The architecture adopts a detection mode of real time based on network. The system is consists of several divided-module namely data collection module (sensor), data preprocessor, threads control module, extractors of features, detectors and result return module between user and computer.

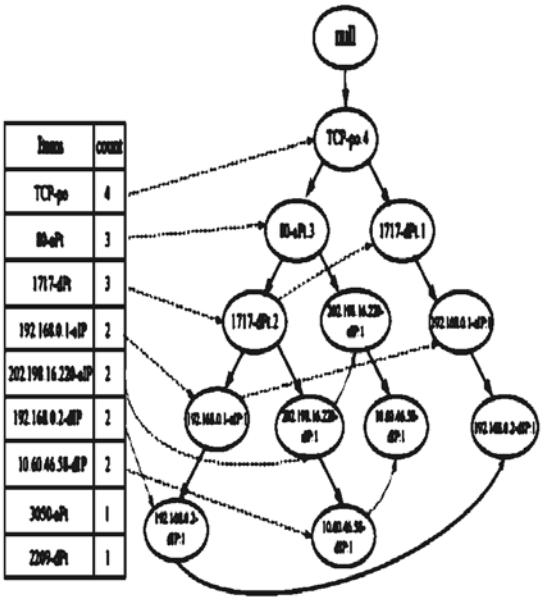


Fig 1: An FP- Tree

Data collection adopts sniffer theory using socket. After gathered, audit data must been preprocessed and cleaned (i.e., adds a sign after data items, as 'sIP' 'dIP' 'sPt' 'dPt' 'po'). Threads control module mainly dominates extractors of features to generate frequent itemsets according to the size of Time Window and the status of data collection. The dominating of extractors depends on how to glide Time Window, as shown in Fig. 3. Datasets are overlapped in two neighboring time window, and the size of those datasets overlapped just is the size of glide Time Window subtracted from the Time Window's size. How to control the size of Time Window is important. When choosing the size of Time Window, we must take into account the sort of detection, the computer and network hardware capability. Especially, how to choose the size of the overlapped data is extraordinarily important. If the size too small, the system may be miss some attacks. While if the size too large, the system will consume abundant resource of memory and CPU. The core of features generation method is just FP-growth. It is compose of two parts: FP- tree constructing, mining the FP- tree; By mining FP- tree, FP-growth method transforms the problem of finding long frequent patterns to looking for shorter ones recursively and then concatenating the suffix. It uses the least frequent items as a suffix, offering good selectivity. The method substantially reduces the search costs.

Detectors are used for comparing and identifying attacks by corresponding attack patterns, which setting master attribute and assistant attribute. It adopts classification by decision tree [8]. A decision tree is a flow-chart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and leaf nodes represents classes or class distributions.

When the system running, the system chooses the number of collecting data based on the Time Window's size and the form of data updating. When the condition accord with the requirement, all the extractors will be work. When and how extractors work lies on data preprocessor. Because the system is a real time detection system, and can not lost data during detection, so the data preprocessor is the core time line.

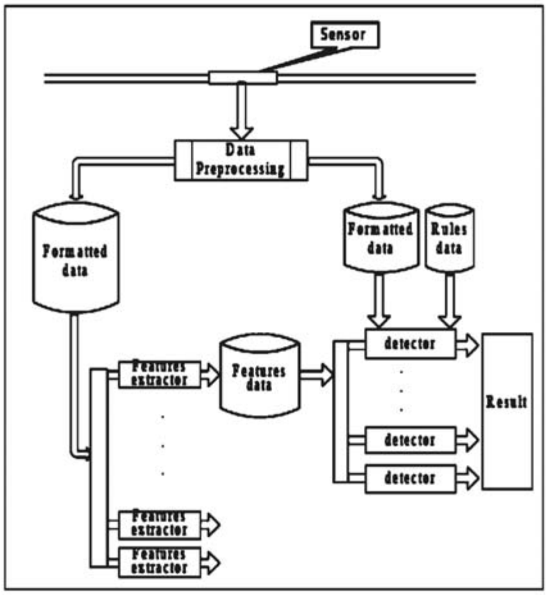


Fig 2: Architecture of our Expereiment.

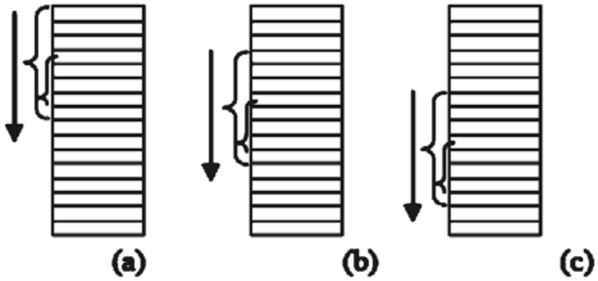


Fig 3: Gliding of Time window

The thread control module drives extractors to work in multithreading when the condition accord with the requirement. If an extractor is working and the data preprocessor's condition accord with the requirement, another extractor is constructed immediately. The system sets some buffers in the memory. Thus the system is rather adapt to real-time requirement.

4. The Experiments and Results

In the experiment, we partly use 2000 DARPA Intrusion Detection Scenario Specific Data Sets [9] to train and test our NIDS prototype. It provided a standard corpus for evaluating intrusion detection systems. It also introduced more stealthy attacks, insider attacks and attacks against the windows operating system. Attacks fall into four main categories:

DOS: denial of service

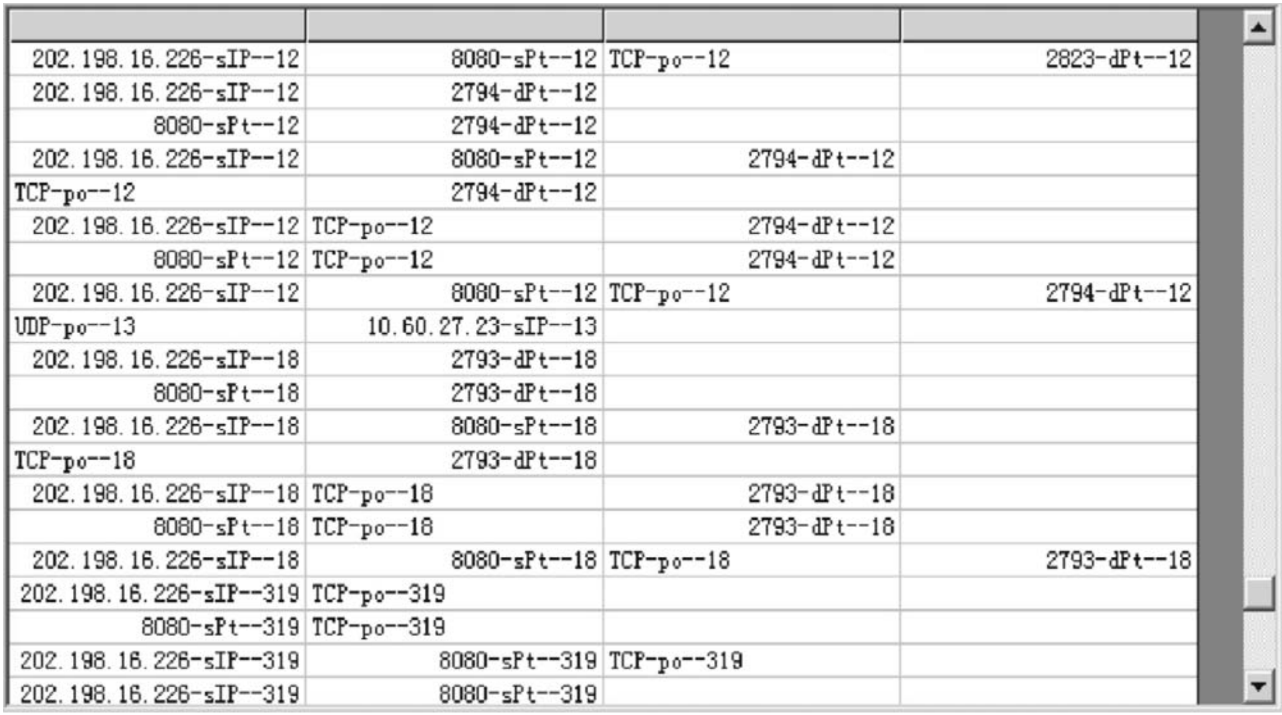
R2L: unauthorized access from a remote machine U2R: unauthorized access to local super user (root) privileges

Probing: surveillance and other probing

We built a NIDS and implemented it by Microsoft Visual C++ 6.0. Fig. 4. shows some part of label data patterns. The size of time window is 350, and the minimum support count is 10. The size of the datasets overlapped is 50 percent of the size of time window. Table 4 shows the Performance of our system.

5. Conclusion

In this paper, we outlined and implemented the architecture of the data mining-based network intrusion detection system in real-time (NIDS). We analyze a frequent patterns mining algorithm that integrate Apriori candidate generation into FP-growth method. FP-growth adopts a divide-and-conquer strategy that compresses the database representing frequent items into a frequent-pattern tree (FP-tree), and proceeds mining of the FP-tree. The method is highly compressed and frequent itemsets generation is integrated and don't need to repeatedly scan the itemsets. Therefore extractor of features adopts FP-growth, and the conclusion is that both resource consuming and efficiency are satisfied. We also expect more such attempts in the future. We are also developing unsupervised anomaly detection algorithms to reduce the reliance on labeled training data. Experiments on DARPA shows the performance of the NIDS is satisfied. Our future work includes researching some new algorithms and refining the existing system.



202.198.16.226-sIP--12	8080-sPt--12	TCP-po--12	2823-dPt--12
202.198.16.226-sIP--12	2794-dPt--12		
8080-sPt--12	2794-dPt--12		
202.198.16.226-sIP--12	8080-sPt--12	2794-dPt--12	
TCP-po--12	2794-dPt--12		
202.198.16.226-sIP--12	TCP-po--12	2794-dPt--12	
8080-sPt--12	TCP-po--12	2794-dPt--12	
202.198.16.226-sIP--12	8080-sPt--12	TCP-po--12	2794-dPt--12
UDP-po--13	10.60.27.23-sIP--13		
202.198.16.226-sIP--18	2793-dPt--18		
8080-sPt--18	2793-dPt--18		
202.198.16.226-sIP--18	8080-sPt--18	2793-dPt--18	
TCP-po--18	2793-dPt--18		
202.198.16.226-sIP--18	TCP-po--18	2793-dPt--18	
8080-sPt--18	TCP-po--18	2793-dPt--18	
202.198.16.226-sIP--18	8080-sPt--18	TCP-po--18	2793-dPt--18
202.198.16.226-sIP--319	TCP-po--319		
8080-sPt--319	TCP-po--319		
202.198.16.226-sIP--319	8080-sPt--319	TCP-po--319	
202.198.16.226-sIP--319	8080-sPt--319		

Fig 4: Data set

Acknowledgement

We take immense pleasure in thanking our Chairman Dr. Jeppiaar M.A, B.L, Ph.D, the Directors of Jeppiaar Engineering College Mr. Marie Wilson, B.Tech, MBA., (Ph.D) Mrs. Regeena Wilson, B.Tech, MBA., (Ph.D) and the Principal Dr. Sushil Lal Das M.Sc (Engg.), Ph.D for their continual support and guidance. We would like to extend our thanks to my guide, our friends and family members without whose inspiration and support our efforts would not have come to true. Above all, we would like to thank God for making all our efforts success.

References

[1] W. Lee, S. J. Stolfo, and K. Mok. Data mining in work flow environments: Experiences in intrusion detection. In Proceedings of the 1999 Conference on Knowledge Discovery and Data Mining (KDD-99), 1999.

[2] A. Ghosh and A. Schwartzbard. A study in using neural networks for anomaly and misuse detection. In Proceedings of the Eighth USENIX Security Symposium, 1999.

[3] L. Pornoy. Intrusion detection with unlabeled data using clustering. In Undergraduate Thesis, Columbia University, Department of Computer Science, 2000.

[4] W. Lee, S. J. Stolfo, P.K. Chan, E. Eskin, W. Fan, S. Hershkop M. Miller, and J. Zhang. Real time data mining-based intrusion detection. In DARPA Information Survivability Con-ference and Exposition (DISCEX II'01), Anaheim, California, June 2001.

[5] D. Zamboni. Using clustering to detect abnormal behavior in a distributed intrusion detec-tion system. Unreleased Technical Report, Purdue University. August, 2001.

[6] Jiawei Han, and Micheline Kamber. Data Mining:Concepts and Techniques. Higher Educa-tion Press, 2001.

[7] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation'', SIGMOD Conference 2000: 1-12.

[8] Johannes Gehrke, Raghu Ramakrishnan, and Venkatesh Ganti. Rainforest - a framework for fast decision tree construction of large datasets. Data Mining and Knowledge Discovery, 4(2/3):127-162, 2000.

[9] http://www.ll.mit.edu/IST/ideval/data/200 0/2000_data_inde x.html.

J.Visumathi B.E.,M.E.,(Ph.D) works as Assistant Professor in Computer Science and Engineering Department of Jeppiaar Engineering College, affiliated to Anna University, Chennai, TamilNadu, India. She is a PhD student at sathyabama University. She has more than 10 years of teaching experience and her research interests are Networks, Artificial Intelligence, Data mining.