A NOVEL SIGNATURE SEARCHING FOR INTRUSION DETECTION SYSTEM USING DATA MINING

YA-LI DING, LEI LI, HONG-QI LUO

Pattern Recognition and Intelligence System, College of Automation, Nanjing University of Post and Telecommunications, Nanjing 210003, China
E-MAIL: yaliding@163.com,lil@njupt.edu.cn

Abstract:

Intrusion Detection System (IDS) has recently emerged as an important component for enhancing information system security. Data mining and machine learning technology has been extensively applied in network intrusion detection and prevention systems by discovering user behavior patterns from the network traffic data. In this paper, we propose a novel signature searching to detect intrusion based on data mining，which is an improved Apriori algorithm. We evaluate the capability of this new approach with the data from KDD 1999 data mining competition. Our experimental results demonstrate the potential of the proposed method.

Keywords:

Intrusion detection; Data mining; Association rule; Apriori algorithm; Frequent itemset; Scenario

1. Introduction

1.1. Intrusion detection systems (IDS)

As network-based computer systems play increasingly vital roles in modern society, they have become the targets of our enemies and criminals. It is critical to have a good protection system against intrusions. An intrusion into a computer system is any activity that violates system integrity, confidentiality, or data accessibility. In order to meet this challenge, Intrusion Detection Systems (IDS) are being designed to protect the availability, confidentiality and integrity of critical networked information systems.

There are two approaches to intrusion detection. Rule-based detection involves an attempt to define a set of rules that can be used to decide that a given behavior is that of an intruder. The advantage of this approach is that the rules can be formulated to detect specific attacks in detail. Therefore, it is guaranteed to detect known attacks and produce few false alarms. The opposite approach, anomaly detection involves the collection of data relating to the behavior of legitimate users over a period of time, and then applied statistical tests to the observed behavior to determine whether that behavior is legitimate user behavior or not. Anomaly detection has the advantage that it can detect new attacks that the system has never seen before as they deviated from normal behavior. However, current anomaly detection schemes still suffer from a high rate of false alarms. Therefore, currently IDS have many false alarms and redundancy alarms. As a result, understanding and handling IDS alarms become even more difficult. [2, 3, 6]

1.2. Apriori algorithm [5]

Let $I = \{i_1, i_2 ...... i_m\}$ be a set of items, where each element i_m is an item, and D is a set of transaction T, which is a set of item, where $T \subseteq I$,each T has only one flag called transaction mark, denoted TID .A transaction , $T \in D$ contains a set of items or an itemset , $X \subseteq I$,if $X \subseteq T$. An association rule is an expression of the form $X \Rightarrow Y$, where $X \subseteq I$, $Y \subseteq I$, and $X \cap Y = \phi$ .The rule $X \Rightarrow Y$ holds in D with support, $\sup(X \Rightarrow Y) = P(X \cap Y)$, and confidence, $conf(X \Rightarrow Y) = P(X / Y)$.

Classical Apriori algorithm:

```
(1) $C_1$ = {candidate I-itemsets};
(2) $L_1 = \{c \in C_1 \mid c.count \geq \min\sup\}$ ;
(3) FOR (k=2 $L_{k-1} \neq \phi$ ; k++) DO BEGIN
(4) $C_k$ =apriori-gen ( $L_{k-1}$ );
(5) FOR all transactions t $\in$ D DO BEGIN
(6) $C_t$ =subset ( $C_k$ , t);
(7) FOR all candidates c $\in$ $C_t$ DO
```

```
(8) c.count++;
(9) END
(10) $L_k$ = {c ∈ $C_k$ |c.count≥minsup}
(11) END
(12) Answer= ∪ $L_k$ ;
```

Apriori algorithm may need to generate a huge number of candidate generations. Each time when candidate generations are generated, the algorithm needs to judge whether these candidates are frequent item sets. The manipulation with redundancy result in high frequency in querying, so tremendous amounts of resources will be expended whether in time or in space.

Apriori algorithm is an influential algorithm for mining frequent itemsets for Boolean association rules [7]. It doesn't consider the order of items in itemsets.By the observation of the attack signatures; we find that there are some attack signatures depending on other previous attack signatures. This is due to the new attack is a derivative from the previous attack. Based on the two points, we propose an algorithm to combine the known signatures and sequence steps to find the signature of the related attack quickly which could avoid the mostly false positive effectively.

2. The improved apriori

2.1. Principle of the improvement

The Apriori algorithm is an algorithm for mining frequent itemsets. The name of the algorithm is based on the fact that the algorithm uses the prior knowledge of frequent itemset properties. In Apriori algorithm, from C_k to L_k have two steps: (1) Pruning itemsets according to L_{k-1} .(2) Pruning itemsets according minsupport. However, Apriori algorithm has the shortage of producing a large number of candidate itemsets and scanning the database too many times. Based on reducing query frequencies, storage resources and false alarm of IDS, we design an improved Apriori algorithm that mines frequent itemsets with knowing the signature of one existing attack. Usually an attack consists of a sequence of logical scenarios. Some new attacks vary from previous attack signatures. Let us see one example about Code Red. As we know Code Red I and Code Red II are both "worms", which are attacks that propagate themselves through networks without any user intervention or interaction. They both use the flaw in Microsoft's Internet Information Services (IIS) Web server software. Code Red I cause the damage by defacing Web pages and by denying access to a specific Web site by sending it the massive amounts of data. Finally it will shut the web site down. This is known as a denial-of service (DoS) attack. Code Red II is much more powerful and more damaging. The rules for SNORT to detect Code Red I and Code Red II are the following:

Alert tcp any any -> any 80 (msg: "Code Red I Overflow"; content "|2F646566 1756C74 2E696461 3F4E4E4E|";).

Alert tcp any any -> any 80 (msg: "Code Red II Overflow"; content: "|2F646566 1756C74 2E696461 3F585858|";).

From above ,we can see that both of Code Red I and Code Red II include the same strings"2F646566 1756C74 2E696461".We can consider the same strings as a known pattern. The specific patterns are called signatures. We use the known patterns of unauthorized behavior to predict and detect the attacks. Besides that, we find the last string following the same strings decide whether the attack is Code Red I or Code Red II?

2.2. The improved algorithm in detail

Our improved algorithm includes four elements: item id, transaction id, timestamp, known-signatures. To implement the improvement, combining with an instance, the improved algorithm is described as following steps:

Step1: Scan the transaction database D to analysis the item of each transaction .If an item is met for the first time, it will be added into the candidate 1-itemset C_1 and its count will be set 1. If the item has been in C_1 already, its count will be added 1.Then scanning C_1 to delete those itemsets whose support is smaller than the min-sup. Thus, we get the frequent 1 itemset L_1.Meanwhile, we could compute the length of the largest TID M.

Table 1. The transaction database

TID	The list of item's id
T0	i_1, i_3, i_4, i_7
T1	i_2, i_3, i_5
T2	$i_2, i_4, i_7,$
T3	i_1, i_3, i_5, i_6

T4	i_2, i_5
T5	i_1, i_4, i_5
T6	i_1, i_3, i_5
T7	i_3, i_4, i_6

Scanning the above database, setting min-sup=2, we can get M=4 and L_1 as the fellows Table2:

Table 2. L_1

Itemset	support
i_1	4
i_2	3
i_3	3
i_4	2
i_5	2
i_6	3
i_7	2

Step2: Generally, suppose L_{k-1} has been generated ,then, the candidate K itemset C_k is generated by L_{k-1} joining with L_1 .But the problem for us is that each transaction is an item sequence ,the appearance of current item may depend on any item before it and it doesn't depend on items after it. For example, in above database T7 { i_3 , i_4 , i_6 } i_4 may depend on i_3 ,doesn't depend on i_6 .Also, i_6 depends on i_3 , i_4 or{ i_3 , i_4 }.So ,when L_{k-1} joins with L_1 ,we just consider i_n item where $n \geq k-1$.By this way ,we can reduce time and resource consumption .

Step3: Prune C_k to delete the itemsets whose (k-1)-subsets are incompletely included in L_{k-1} or without the known-signatures. Min-sup=2; we can get Table3.

Table 3. L_2

itemset	Support
i_1 i_3	3
i_1 i_4	2
i_1 i_5	3
i_2 i_5	2
i_3 i_4	2
i_3 i_5	3
i_3 i_6	2
i_4 i_7	2

Step4: We check all of the possible combinations of the frequent items with already known signatures, if they meet the minimum support requirement. Then, append this k-itemsets by repeating Step2 and Step 3. We can first append the backward, until the min-sup is unsatisfied or $k \succ M$ (size of largest the TD).

Suppose the known signatures are " i_1 and i_3 "; Min-sup=2; we can get Table4.

Table 4. L_3

Itemset	Support
i_1 i_3 i_5	2

The algorithm uses $L_3 \infty L_1$ to generate a candidate set of 4-itemsets, C_4 .Although the join results in { i_1 , i_3 , i_5 , i_6 },this itemset is pruned since its subset { i_1 , i_3 , i_6 } is not frequent .Thus { i_1 i_3 i_5 } is the maximum frequent itemset.

From above, we can see that T0, T3 are also include $\{ i_1\ i_3 \}$, but they are not frequent transactions. While, those transactions shouldn't be ignored. Maybe they are new attacks derivativing from an exiting attack. We should handle them manually to reduce false alarm.

3. Experiment results

The experiment was tested on a Intel Core(TM)2 Duo Processor 1.83GHZ with 1024 RAM running Window XP. To evaluate the effectiveness of our improved Apriori, we use DARPA 1999 datasets as the testing data. [1] In our experiments, the first two weeks' data are used as training data: the first week's data don't contain any attack and the second week's data contain 43 attack instances. By using known-signatures, those attacks are divided into 18 types.

In fig .1, the length of known-signatures is Len. Len=2.Size of database is 5Mbytes.

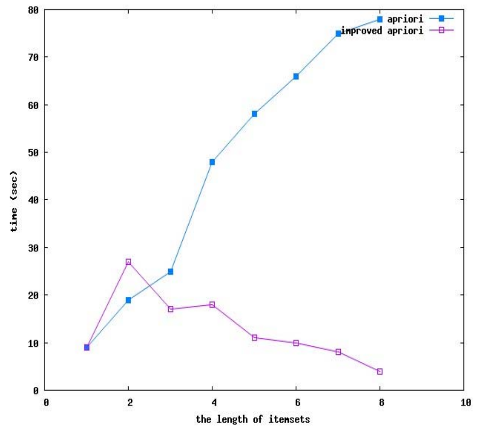

Figure 1. Different items of consumption-time comparison

Fig.1 tells us that Apriori consumes much time to generate frequent itemsets. However, many of them are not useful for finding intrusion signatures. We could see that our algorithm is faster than Apriori algorithm except the 2 –itemsets, because our algorithm should do pattern matching with the known signatures. The reason is that the number of candidate 1-itemsets is not very large, and those including the known-signatures are smaller. Therefore, in the real environment, there are not too much candidate itemsets to be generated during each pass of finding coincidental signatures.

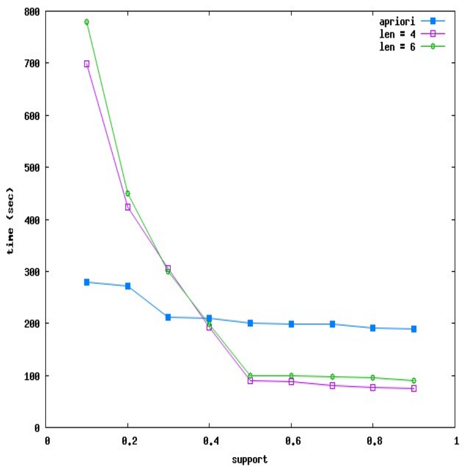

Figure 2. Different levels of support comparing consumption of time

The result in Fig.2 is based on different min-sup and different the length of known signatures. The database size is 10Mb. We set len=4 and len=6 and compare them with Apriori algorithm. We use nine different minimum supports from 0.1 to 0.9 to test the processing time of them. As shown in the above figures, the minimum support cannot be set too low or the processing time would rise sharply. We will generate very large candidate itemsets to compare with the known signatures. Besides that, we find that when the minimum support decreases, the processing times of all algorithms increase because the total number of candidate itemsets increase. If the minimum support is not too low, we have known part signatures, so that, we can reduce the times of scanning database. It is clear that the final results are correlated with the length of known signatures.

4. Conclusions

This algorithm is an improvement of Apriori, which uses the known patterns of unauthorized behavior to predict and detect the attacks. Considering each transaction is an item sequence, we should research on the dependent relations between alerts. By these measures, we can find out the new attack's signatures more efficiently. How to further improve the detection performance and optimize the algorithm is our coming work.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (Grant Nos: 10371106, 10471114) , Natural Science Foundation of Jiangsu Province (Grant Nos: 04KJB110097, 08KJB520023) and the Climbing Plan of Nanjing University of Post and Telecommunications (Grant No: NY207064).

References

[1] KDD (1999), the third international knowledge discovery and data mining tools competition data set (KDD99 Cup). http://kdd.ics.uci.edu/databases/kddcup99.html.

[2] Hu Zhengbing, Li Zhitang, Wu Junqi,“A novel network intrusion detection (NIDS) based on signatures search of data mining”, 2008 Workshop on Knowledge Discovery and Data Mining,pp.10-16， 2008.

[3] Safaa O.Al-Mamory,Zhang Hongli and Ayad R.Abbas,“IDS alarms reduction using data mining”,2008International Joint Conference on Neural Networks”,pp.3564-3570.

[4] Ding Yuxin, Wang Haisen, Liu Qingwei,“Intrusion scenarios detection based on data mining”, Proceedings of the Seventh International Conference on Machine Learning and Cybernetics, Kunming,12-15 July 2008,pp.1293-1297.

[5] Jiawei Han Micheline Kamber,“Data Mining Concepts and Techniques ”，Second Edition.

[6] Wang, Q. and V. Megalooikonomou.“ A clustering algorithm for intrusion detection. in SPIE Conference on Data Mining”，Intrusion Detection, Information Assurance, and Data Networks Security. 2005. Orlando, Florida, USA.

[7] Liu Shan, Liao Yongyi, “An improved Apriori Algorithm”, Modern Electronic Technique, Vol. 30,No. 4, pp.106-110, 2007.

[8] Jha S, Sheyner O, Wing J., “Two formal analyses of attack graphs”, Proceedings of the 15th Computer Security Foundations Workshop, Nova Scotia, pp.49-63, Jun.2002

[9] Lee, W., Stolfo, S.J. and Mok, K.W.,"A Data Mining Framework For Building Intrusion Detection Model", Proceeding of the IEEE Symposium on Security and Privacy, 1999, pp.153-157.