A Novel Network Intrusion Detection System(NIDS) Based on Signatures Search of Data Mining

Hu Zhengbing[1,2], Li Zhitang[1],Wu Junqi[2]
1) *Huazhong University of Science and Technology, College of Computer Science and Technology, kievpastor@yahoo.com*
2) *Huazhong Normal University, Department of Information Technology, kievkpi@hotmail.com*

Abstract

Network security has been a very important issue, since the rising evolution of the Internet. There has been an increasing need for security systems against the external attacks from the hackers. One important type is the Intrusion Detection System (IDS). There are two major categories of the analysis techniques of IDS: the anomaly detection and the misuse detection.Here we forcus on misuse detection, the misuse detection collected the attack signatures in a database as the same as virus protection software to detect the relate attacks. we propose an algorithm to use the known signature to find the signature of the related attack quickly.

1. Introduction

There are two types of IDS: The one is Network-based IDS(NIDS) and the other is Host-based IDS(HIDS). The NIDS monitors the packets from the network and HIDS analyzes the audit data of the operation system. There are also two major categories of the analyzes techniques of IDS:the anomaly detection and the misuse detection. Anomaly detection uses the established normal profiles to identify any unacceptable deviation as the result of an attack. Misuse detection uses the "signatures" of knows attacks to identify a matched activity as an attack instance. Both of them can be used in the NIDS or HIDS. For the NIDS, when using the misuse detection technique, the operation is similar to the detective actions of the virus protection software where using the already known virus code to detect the known virus. The misuse detection collected the attack signatures in a database as the same as virus protection software to detect the relate attacks. Misuse detection techniques have been widely used in all commercial IDS products.

In the misuse Network-base IDS, the signature search problem is a time-consumption and error-pruning work. Therefore, some papers introduce the data mining [1,2,3] technique into IDS. However, they all concentrate mining association of attribute in the transfer protocol but not mining the traffic content. The problem was solved until the Signature Apriori algorithm [5] was proposed. but the signature Apriori waste much time for generate the unnecessary candidate itemsets and scan the database .If the size of database is large, the Signature Apriori will be not effective in the signature search . In this paper, we propose an algorithm to use the known signature to find the signature of the related attack quickly. We discuss a scenario that if we have known the signature of one existing attack, then we propose a more efficient way than the Signature Apriori algorithm to find the signature of the modified attack, the modified attack is derived from the known attack.

2. Intrusion Detection Techniques

We described the misuse detection commonly called as the signature detection. We use the known patterns of unauthorized behavior to predict and detect the attacks. These specific patterns are called signatures. For the HIDS, one example of a signature is that if the user cannot login to the system three times, then the HIDS will detect this behavior by the pattern of "three failed logins". For the NIDS, a signature can be as simple as a specific pattern that matches a portion of a network packet. The signature of packet content and header content can indicate the unauthorized actions. The occurrence of a signature may not signify an actual attempted unauthorized access, but it is a good idea to take each alert seriously. For instance, a user could post an article on a web site containing Code Red information. This site could contain an analysis of the attack that contained "|2F646566 1756C74 2E696461 3F4E4E4E|" in the text. This would then be triggered anytime anyone accessed this web site. Depending on the robustness and seriousness of a signature that is triggered, some

alarm, response, or notification should be sent to the proper authorities.

There are two advantages of misuse detection technique. The first is that it is very effective for detecting the attacks without generating an overwhelming number of false alarms. The second is that it can quickly and reliably diagnose the use of a specific attack tool. On the other hand, the disadvantage of misuse detection technique is that it can only detect those attacks that have been described in the database. Therefore, the database must be constantly updated with signatures of new attacks.

Anomaly intrusion detection tries to determine whether deviation from the established normal usage patterns can be flagged as intrusion. Anomaly detection technique assumes that all intrusive actives intercross the set of anomalous activities instead of being exactly the same; there will be two cases. The anomalous actives that not intrusive are flagged as the intrusion. This will result the false positive. The other more situation is that the intrusive actives that are not anomalous result are regarded as the false negatives.

There are some advantages and disadvantages of anomaly detection technique. First, the IDS based on anomaly detection has the ability to detect symptoms of attacks, and the information can be used to define the signatures for misuse detectors. However, the anomaly detection usually produces a large number of false alarms due to the unpredictable behaviors of users and networks. Therefore, anomaly detection approaches often require the extensive "training sets" of system event records in order to characterize the normal behavior patterns[4].

3. Measurement Criteria for IDS

The term false positive is to describe a situation in which a IDS triggers a false alarm, but it is a wrong alarm, in fact, there is no attack .

The False positive may be caused by the following situations. First , reactionary traffic alarms is caused by another network event . An example is that an IDS triggering an ICMP flood alarm when it is really several destination unreachable packets caused by equipment failure. Secondly, equipment-related alarm is triggered by odd, unrecognized packets generated by certain network equipment. Thirdly, protocol violations is often caused by poorly or oddly written client software. The final is that true false positives is generated through some real occurrence that is not malicious in nature.

False negative is the term used to describe a network intrusion device's inability to detect the true attack. In other words, malicious activity is not detected and alerted. Fortunately, there are some actions that can be taken to reduce the chance of false negative conditions without increasing the number of false positives [5,6].

There are a number of potential reasons for causing the false negatives. The first is the network design issues, i.e., the traffic exceeds the ability of a switch . The second is the encrypted traffic design flaw, because the IDS is unable to understand encrypted traffic. Placing the NIDS behind VPN termination points is a good way to ensure the NIDS realizes all traffic. The third is the improper written signatures. Although the attack and signature are known, the signature does not properly represent the attack because it not well defined. The forth is the unpublicized attack. The attack is not publicly known, therefore vendors have no knowledge and signature about the attack .The final is the NIDS design flaw. The NIDS device simply does not detect the attack due to the poor design or signature implementation.

In this paper, we only concern with the false positive. This is due to that we can change the experimental parameters in hands to avoid the mostly false negatives.

4.The Signature Apriori Algorithm

The concept of Signature Apriori algorithm [7] is derived from Apriori algorithm [8]. 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. Apriori employs an iterative approach known as a level-wise search, where k-itemsets are used to explore (k+1)-itemsets. First, the set of frequent 1-itemsets is found. This set is denoted as L_1. L_1 is used to find L_2, the frequent 2-itemsets, which is used to find L_3， and so on, until no more frequent k-itemsets can be found. The finding of each L_K requires one full scan of the database.

A. The Proposed Algorithm

By the observation of the attack signatures, we find that there are some attack signatures dependent on other previous attack signatures. This is due to the new attack is a derivative from the previous attack . So far as we know, there are at least two kinds of attacks have this property.

For the first case, a new attack is variation of an existing attack .one example is Code Red. Generally, 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 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 CodeRed I and CodeRed II are the following:

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

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

The first rule means that if a packet incoming from any port of TCP contains a string: 2F646566 1756C74 2E696461 3F4E4E4E. Then, it is possible to incur the CodeRed I attack. The second rule is similar to the first rule. We could see that both CodeRed I and CodeRed II have the same string "|2F646566 756C74 2E696461|".

The other case is that a new attack is relative to an existing attack such as lIS illegal access attack. There are two example rules for SNORT as follows:

- alert tcp any any ->$HTTP SERVERS 80 (msg: "WEB-IIS access"; uricontent:"/iisadmpwd/aexp2.htr";).

- alert tcp any any -> $HTTP_SERVERS 80 (msg:"WEB-IIS/scripts/iisadmin/default.htm access"; uricontent:"scripts/iisadmin/default.htm";nocase;).

The first rule means that if an incoming packet from any port of TCP contains a string: /iisadmpwd/aexp2.htr, it is possible that the WEB-IIS /iisadmpwd/aexp2.htr access. The second rule is also an IIS illegal access attack and it is similar to the first rule. Both of these two rules contain "/iisasm" string.

These examples show that there exist many different attacks have the same signature indeed. Hence, we want to use some part of the known signature to find out the new attacking signature. The Signature Apriori algorithm can achieve the purpose, but it will take much time to generate candidate itemsets and scan the database. Therefore, we will propose an algorithm to find out the new signature by using the already known signatures. Finally our algorithm saves the processing time.

B. System Diagram

Fig. 1 shows our system works as follow: (1)The attacker gets an attacking tool to attack victim.(2)To simulate real-life traffic, we generate normal packets as background traffic. (3)Our algorithm will receive two kinds of data: one comes from the packet sniffer and the other one comes from the known signature. (4)Our algorithm will output the frequent itemset with the maximum length.(5)We use a famous open source NIDS - SNORT to test the signature accuracy we find out.

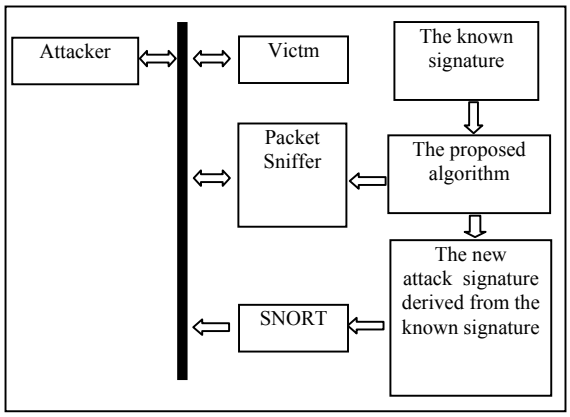

Fig.1-System Diagram

C. The Proposed Algorithm

D	A set of transactions where every transaction is a packet.
c	The candidate itemset.
S(c)	The support of the candidate c is the percentage of transactions that contain c.
min_sup	Minimum support meets the threshold.
Known _signature	The part of signature we have known.
Max_len	Maximum length of frequent itemset that contains the known signature.
N	Length of *known_signature + 1*.
k–itemsets	An itemset having k items.
Sig	Current *frequent-tiemset* in L_k
L_k	A set of frequent k-itemsets (with minimun support). The set has two fields: itemset and support count.
C_k	A set of candidate *k-itemsets* (potentially frequent itemsets). The set has two fields: itemset and support count.

Table 1-The parameters used in our algorithm

We will make the detailed description of our method. The Table 1. lists the parameters and their

descriptions that will be used in our proposed algorithm.

In the Fig. 2., the steps of how to find out a frequent k-itemsets will be as follow. At the first step, all of the frequent items will be found. And then we use a simple way toscan the database in order to find the ftequency of occurrence of each item, and decide which one meets the minimum support. Secondly, we generate the candidate n-itemsets by checking all of the possible combinations of the frequent items with already known signatures, if they meet the minimum support requirement. Then, append this n-itemsets from right. We can first append the backward, until the minimum support is unsatisfied.

Then, we append forward, and stop when the same condition occurred. Finally, the maximum length of frequent-item set can be mined by our method. A formal algorithm is described below.

Input: D, min _sup.
Output: L_k
Algorithm:
Step 1: L_1*= find_frequent_1_itemsets(D, min_sup);*
Step 2: L_n*= find_frequent_n_itemsets(D,* L_1*, min_sup,known_signature);*
Step 3: for k=n+ 1 to max_len do
Step 4: C_k*= candidates_gen (D,*L_1*,*L_{k-1}*, min_sup);*
Step 5:Answer=L_k

Fig.2-The proposed algorithm

There are three procedures used in Fig. 2. The first is find_frequent_1-itemsets, and the second is find_frequent_n-itemsets. The third is candidates_gen. We describe them as follows.

Input :D, min_sup.
Output: L_1
Algorithm:
Step 1: $L_1 = \Phi$ *;*
Step 2: while D has available transactions do begin
Step 3: read one transaction in D
Step 4: increase every item support count in the transaction.Same item only counted by one in the same transaction.
Step 5: end
Step6:$L_n=\{c \in C_1(c) | support (c) \geqslant min\_sup;$

Fig. 3 - Algorithm *find_frequent_1-itemsets*

In Fig. 3, we show a simple way to find the frequent items. We read one transaction each time from database and then count the support of each different item. If an item occurs twice in the same transaction, the support count of this item will increase once. Repeat until no transactions available in database. Finally, we will check all items in candidate 1-itemset and append the item that meet the minimum support into L_1.

After all frequent items have been mined, we will stop generating all possible candidate 2-itemsets just like the Signature Apriori algorithm. We generate the candidate itemsets only related the known signatures. Then, all of the frequent items will be concatenated to the known signature and put them into candidate n-itemsets. After that, check all item sets in the candidate n-itemset. Then, add the itemsets that meet the minimum support into L_1. For instance, assume that A, B, C and D are frequent items. {B C} are the part of known signature. We will check whether {B C A}, {B C B}, {B C C} and {B C D} meet the minimum support and determine the frequent itemsets.

Input: D, min_sup, L_1*, known _signature.*
Output: L_n
Algorithm:
Step 1: $L_n = \Phi$ *;*
Step 2: for each item in L_1 *do*
Step 3: add known_signature + item into Cn
Step 4: $L_n = \{c \in C_n \mid support\ (c) \geq min\_sup;$

Fig. 4-Algorithm *find_frequent_n-itemsets*

In Fig. 5, first, append all frequent items to sig until no any *s(sig + item)* meet the minimum support. Then append sig to all frequent items until no any *s(item + sig)* meets the minimum support.

Input: D, min_sup, L_1
Output: L_k
Algorithm:
Step 1: $L_k = \Phi$ *;*
Step 2: for each item in L_1 *do begin*
Step 3: add sig+ item into C_k
Step 4: $L_k=\{c \in C_k | support\ (c) \geq min\_sup\};$
Step5: if there is no any s(sig+ item) $\geq$ *min_sup then do step 6,7 until find maximum length of frequent itemset.*
Step 6: add item+ sig into C_k
Step 7:$L_k = \{c \in C_k | support\ (c) \geq min\_sup\};$
Step 8: if there is no any s(item+sig)>= min_sup.Then stop this procedure.

Fig. 5-Algorithm *candidates_gen*

Next, we take an example below to show how the proposed algorithm works. We assume that the transactions in the database are {{A B C D E F G

Q},{M N A B C D E F G},{ J A B C D E F G},{P Q I}}. The attack signature we have already known is {C D E}. Let the minimum support be 0.7. Applying the proposed algorithm, we can firstly get the frequent items L1={A B C D E F G}. In order to find out the derived attack signature we expanded the known signature by each frequent item, and we then we have C_n={ {C D E A},{C D E B},{C D E C},{C D E D},{C DE E},{C D E F}, {C D E G}} at the first stage. After we have C_n candidate itemsets, we scan the database to find out the L_n={C D E F}. Then we let the Ln be the new attack signature that we have already known. Repeating the step until the minimal support is no longer satisfied. We win get the L_n ={ C D E F G} in this example. Next, we expand the Ln in the inversed direction. Finally, we will get the possibility attack signature L_n ={A B C D E F G}.

From above, the proposed method will process the scan of the database in each Cn. Because the scan of the database may cost a lot of system time, therefore, a modification shown in next section win be made to enhance the efficiency.

D. The Improvement of Proposed Algorithm

We know that C_i+1 is generated from L_i* L_i. Clearly, a C_i'generated from C_i * C_i, instead of from L_i* L_i, will have a greater size than | C_i | where C_i is generated from L_i* L_i. However, if | C_i'| is not much larger than | C_i |, we may save one round of database scan. This technique is called scan-reduction [9]. Note that, when the minimum support is relatively small, the candidate itemsets become large. It may cost too much CPU time to process those candidate itemsets.

In this paper we apply the Scan-Reduction method to our algorithm. The support of itemsets is counted among every three passes. The reason why counted the amount of support count in each three passes is derived from the experiment results. The counting frequency is to be related to the hardware computational capability and others concern, such as the operational environment and parameter configuration. Therefore, the counting frequency can be an adjustable parameter that used for improving the system's perfomance according to their different environment and platform capability. How we apply the Scan-Reduction method in our algorithm described as follows.

It is observed that to expand the all combinations of Cn first before scanning database in each Cn will reduce the system loading on scanning database. From above example, after we get C_n ={ {C D E A},{C D E B},{C D E C},{C D E D},{C D E E},{C D E F}, {C D E G}} at the first stage. We will expand Cn two more times in our modified method, that is, the itemsets in C_n will become 343 units. After that, we make the first scan of database to find out the Ln. The method will keep processing until the minimum support unsatisfied.

5. Experiments

We show the experimental results of our proposed algorithm and compare with the Signature Apriori algorithm. We use nine different-sized databases, From 10Mbytes to 90Mbytes.The experiment was run on a 2.4 GHz Pentium 4 microprocessor with 1 Gbyte RAM. The operating system was windows 2003 server edition.

Our experiment is divided into two parts:

1) We use the IIS illegal access problem to examine signature mining speed of Signature Apriori and the algorithm we proposed. The attack signature that we used is "iisadmpwd/aexp2.htr". It contains 19 characters. Assume that the length of the known signature we known is four and eight separately. We use the same audit databases for the Signature Apriori algorithm and our algorithm to compare the processing time of signature mining. Then, we apply the scan-reduction method to our algorithm and compare processing time again.

2) We used the well-known open source IDS - SNORT to test the accuracy of signature we mined. Here, we use Receiver Operating Characteristic (ROC), ROC curve to show the false positive rate and true positive rate (detection rate). We set the minimum support to 0.7 and find signatures of IIS illegal access attack and IIS Unicode attack by the known signature. The reason why we use the high minimum support is to raise the accuracy of signature that we mined. The output of our algorithm is "/iisadmpwd/aexp2.htr" and "/..%c0%af../", respectively. Then use the signatures as the rules in SNORT and get the experimental results.

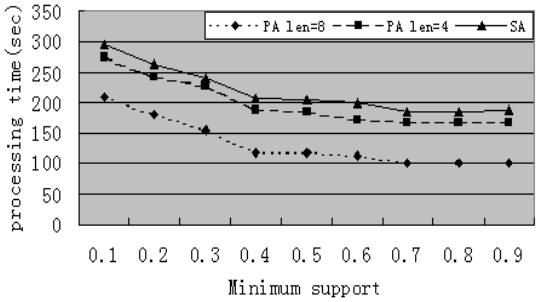

Fig. 6-The processing time for database size=10Mb

In the following Figures, SA is the abbreviation of the Signature Apriori algorithm and PA is the Proposed Algorithm. PA len=4 and PA len=8 mean using the proposed algorithm and the length of known signature is four and eight, respectively.

Fig. 6 tells us that when the minimum support decreases, the processing times of both algorithms increase because of the total number of candidate itemsets increases. We could see that our algorithm is faster than the Signature Apriori no mater what the minimum support is. The reason is that the number of candidate 1-itemsets is not very large. Therefore, in the real environment, there are not too much candidate itemsets to be generated during each pass of finding signatures.

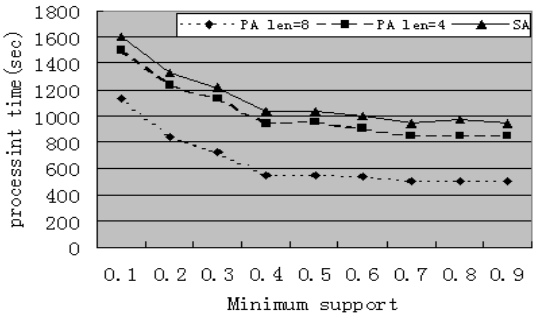

Fig. 7-The processing time for database size=40Mb

The result in Fig. 7 is similar to Fig. 6. We could see that the performance of signature discovery is mainly to be decided by the times of scanning database, especially in large database. Since we have the known signature, so that, we can reduce the times of scanning database. It is clearly that the final results are highly dependent on the length of the known signatures.

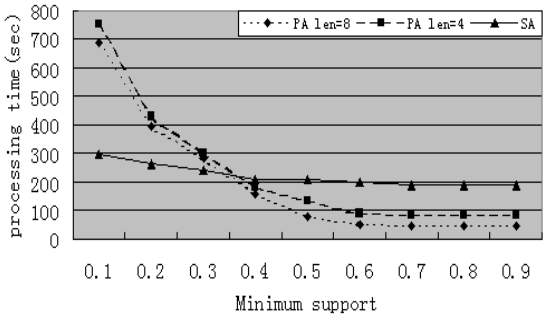

Fig.8-The processing time for database size=l0 Mb,with Scan-Reduction method.

In Fig. 8 and 9, all parameters setting are the same as Fig. 6 and 7 Here,we apply scan-reduction method to our algorithm. The scan reduction method can improve the performance of our algorithm dramatically. In our algorithm, the support is counted among every three passes. Due to this configuration, the times of database scanning is narrowed down to about 1/3 than Signature Apriori approach. Moreover, the number of candidate itemsets are small enough to feet the main memory to analysis.Thus, we can save many rounds of database scanning. As shown in the above Figures, the minimum support cannot be too low or the processing time will rise sharply, because we will generate very large candidate itemsets after each three passes.

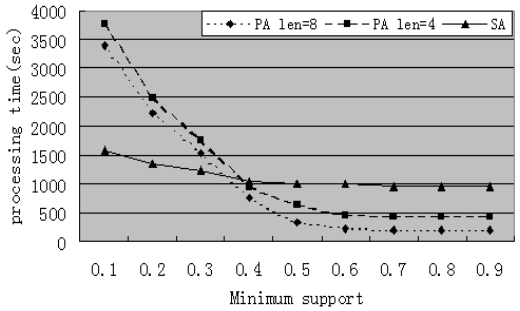

Fig.9-The processing time for database size=40Mb,with Scan-Reduction method

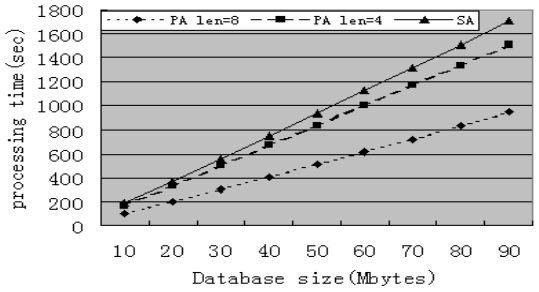

Fig.10-Processing time, database size range=10Mb - 90Mb.

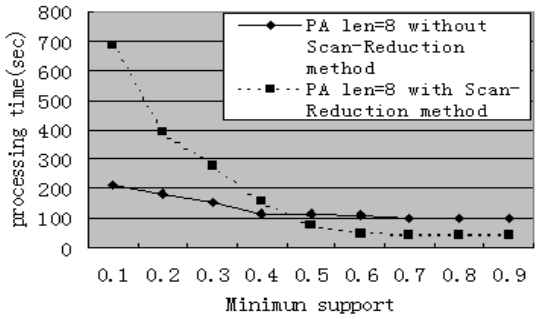

Fig.11-Compare processing time of PA with and without Scan-Reduction method, database size = 10Mb

In Fig. 10, we use nine different-sized databases from 10Mb to 90Mb to test the processing between the Signature Apriori and our algorithm. The processing time of our algorithm can be saved in proportion to the database size. We have known that the processing time of signature discovery is almost decided by scanning database time, i.e.,the bigger database, the longer scanning database time. Therefore, our algorithm can save more time when the database size increases.

In Fig. 11, we compare the processing time of the proposed algorithm with or without the scan-reduction. It is clear that the processing time of the proposed algorithm with scan-reduction method rise sharply when minimum support is smaller than 0.4. The reason is that smaller minimum support causes very large candidates be generated after three passes. Therefore, we can know that the proposed algorithm with scan-reduction method is good for the high minimum support.

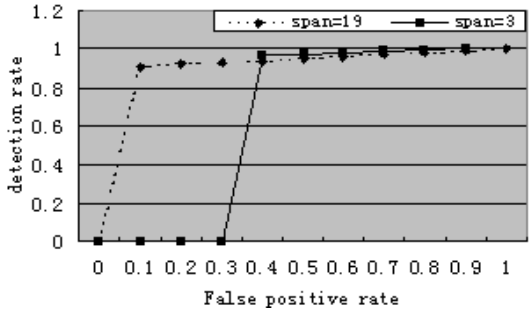

Fig.12-ROC curve for the IIS illegal access problem with the minimum support 0.7

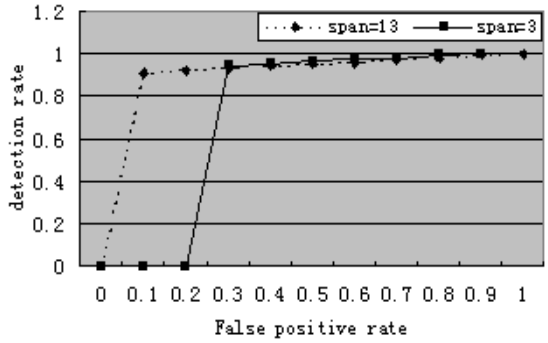

Fig.13-ROC curve for the IIS Unicode problem with the minimum support 0.7

Here, we use two different spans as for test. The result shows that by using longer string as signature for SNORT will get more accuracy than shorter string. Fig. 12 and Fig.13 show that the detection rate of the shorter span curve is higher than longer span curve. The reason is that the probability of a shorter signature in normal traffic (not attack) is high, and this will cause the high false positive rate.

6. Conclusions

In Network-base IDS using misuse detection technique, signature discovery is one of the most important technique. The Signature Apriori algorithm can be used to find out the attack signature. But consider the scenario, a new attack is relative to an existing attack signature. The Signature Apriori algorithm is a waste to make this. Therefore, we proposed a method to find out the new attacking signature based on the known signature. Since we have known that the processing time of signature discovery is mainly dedicated on database scanning, especially in large database. Hence, we also apply the Scan-Reduction method for reducing scanning times of database. By these approaches, we can find out the new attacking signature more efficiently than the Signature Apriori algorithm.

7. References

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

[2] Yang,X.R.,Song,Q.B.and Shen, J.Y.,"Implementation Of Sequence Patterns Patterns Mining In Network Intrusion Detection System", in Proceeding of ICII,2001. pp.323-326.

[3] Hu Zhengbing, Ma Ping. Data Mining Approaches to Signatures Search in Network Intrusion Detection. Control Systems and Computers (USiM). №.1, 2005. pp:83-91. ISBN:0130-5395.

[4] Hu Zhengbing,Shirochin V.P., Su Jun, An Intelligent Lightweight Intrusion Detection System(IDS), Proceedings of IEEE Tencon'2005, Melbourne, Australia, 21-24 November , 2005.pp:2211-2217.Swinburne Press, ISBN 855908149.

[5] Zhao,J.Z. and Huang, H.K., "An intrusion detection system based on data mining and immune principles",in proceeding of Machine Learning and Cybernetics International Conference,2002. pp.453-501.

[6] Yurcik,W., "Contrlling intrusion detection systems by generating false positives:squealing proof-of-concept",in Proceeding of the IEEE local Computer Network Conference,2002. pp.93-101.

[7] Han, H., Lu, Lu, X.L., and Ren, L.Y.,"Using Data Mining to Discover Signatures in Network-Based intrusion detection", in Proceeding of IEEE Computer Graphics and Applications,2002. pp.212-217.

[8] Rakesh, A., and Srikant, R.,"Fast Algorithm For Mining Association Rules",in Proceeding of the 20th international Conference on VLDB,1994.

[9] Park, J.S. , Chen, M.S., and Yu, P.S.,"Using a Hash – Based Method With Transaction Trimming For Mining Association Rules", Knowledge and Data Engineering,IEEE Transaction, 1997.