Research of Intelligent Rule-base Based on Multilayer Intrusion Detection

Sun Zhixin
Institute of Computer, Nanjing University of Posts and Telecommunications,
State Key Laboratory of Novel Software Technology, Nanjing, P.R. China
sunzx@njupt.edu.cn, sunzx630@sina.com

Jiao Lin
Institute of Computer,
Nanjing University of Posts and Telecommunications, Nanjing, P.R. China
jiaolin@njupt.edu.cn

Abstract—**This paper presents a method to establish a rulebase based on multilayer intrusion detection. This rulebase contains two parts: the rulebase based on IP layer intrusion detection and the rulebase based on application layer intrusion detection. The former adopts a mixed quadratic network statistical model to test network traffic which has performances of dynamic principle and low False Positive Probability (FPP) and low False Negative Probability (FNP), and the rulebase is established using the twice-aggregation method. The latter is established by improved Snort. The simulation has proved that this intelligent rulebase can improve detection rate and ability to a large degree, and has low FPP and FNP.**

***Index Terms*—Misuse detection, Anomaly detection, Intelligent rule-base**

I. INTRODUCTION

At present, hackers' attack techniques and methods are becoming more and more advanced and concealed, which are harder to be described accurately only with rule characteristics. How to guard unknown intrusions efficiently is the "bottleneck" of traditional techniques of information safety. Existing safety products, such as firewall and IDS, generally work on rulebases of characteristics of attacks, which are foundations to identify intrusions and defense attacks. So far, it's still very difficult to describe intrusion characteristics roundly, but to deposit characteristics of existing attacks and/or some normal activities in rulebase.

In order to defense network attacks and various viruses, people have put forward many intrusion detection methods based on network. Reference[1]proposes a novel anomaly detection algorithm which uses feature vectors to represent the system’s attributes with slide window and to distribute normal system sates in N dimensions space. The method uses rule sets evolved by genetic algorithms to cover the abnormal space. This algorithm can improve detection rate. Reference [2]puts forward a new method for anomaly intrusion detection, based on linear prediction and Markov chain model. Linear prediction is employed to extract features from system calls sequences of the privileged processes which are used to establish the characters database of those processes, and then the Markov chain model is founded based on those features. This method is effective and efficient to monitor the computer system in real time. Reference [3] presents an adaptive marking scheme for packets with probabilities, which reduces the number of packets needed for attack path reconstruction, thus also saves time for the victim and reduces the spoofing ability for attackers. Intrusion detection techniques based on network are traditionally divided into two categories [4]:anomaly detection and misuse detection. Being independent of attack features is one of strongpoints of anomaly detection, and it can discover network abnormalities and can forecast unknown attacks according to examination targets. However, the anomaly detection has shortcomings-- the False Positive Probability and the False Negative Probability [5]. The misuse detection can detect known attack activities and has low FPP, however, the result of misuse detection lies on the maturity of the rulebase, and the rulebase must be renewed in time. In addition, the misuse detection is helpless to unknown attacks for it is established on existed intrusions.

Because anomaly detection and misuse detection have advantages and disadvantages respectively, nowadays, most IDS use these two detection methods simultaneously [6]. There are many anomaly detection methods, and using statistical method to detect attacks is an effective way to detect unkown attacks, so describing network states exactly becomes an important technique to establish anomaly detection rulebase. CIDS [7](the Correlation Intrusion Detection System) utilizes the change in cross-correlation between selected features to make sure whether there are attacks in current traffic. The CIDS constructs a cross-correlation matrix Ct, that is a k-by-k matrix representing the correlation coefficients which are different in different time. At certain degree, CIDS can describe network states in real time. However, it can't add essential statistical parameters or take out otiose parameters according to actual network, and it depends on

each parameter in a fixed degree, so this description is short of accuracies. Reference[8]puts forward five statistical models of network states, and summarizes methods across-the-aboard to describe network states. However, in these models network states also depend on each parameter in a fixed degree, so these descriptions are short of accuracies, and have FPP/FNP. Based on dynamic principle and low FPP/FNP thoughts, this paper brings forward a mixed quadratic network statistical model G(X2,DKS,DKKS,DAKS,DCC) that used in IP layer intrusion detection. This model can dynamically set and change parameters and their weights, so that can reflect states accurately and have lower FPP/FNP.

The distinguished detection based on rules is Snort [9,10], which has distinctive, simple and efficient rules. Rule options in Snort have clear definitions, and have no affiliations with each other, and these rules offer abundant options to match various features of message packets. Attacks based on single packet can be reflected clearly by combined options, and most rules need only one line codes. However, Snort is restricted in actual application for it is just a rulebase based on misuse detection. This paper makes an improvement on Snort in application layer intrusion detection, which can improve rule matching efficiency. In the mean time, this rulebase will be more popular combined with the abnomal detection rulebase.

The second part presents the statistical model of network states and ideas to establish this model, at the end of this part it'll give the total design of this intelligent rulebase. In the third part and the forth part, we present methods to establish rulebases on IP layer and application layer respectively. Finally, there is a simulation and conclusion.

II. TOTAL ALGORITHM DESIGN OF THE INTELLIGENCE RULEBASE

Supervise the traffic of network data stream at first, then carry out data processing. Judging the statistic features of traffic, we should make sure whether it is suspectable traffic, if it is, judge from historical records in the rulebase to see whether it is illegal traffic, if yes, then make real time response, while not, we will not release it right now, but carry out data processing, that is analysing contents of this packet. Make responses directly to those assured illegal packets or unquestionable normal packets, for example giving advance alarms, or recombining and sending out normal data packets so that they can arrive at their original destinations. It is necessary to establish a fitting statistical model of network to describe network states accurately.

A. Thoughts of establishing statistical model of network states

1. dynamic principle. Analyzing existing network states models [7,8], we can see that all these models are lack of mobility, and they can't add necessary parameters or take out otiose ones. This paper strives to establish a statistical model which can change parameters dynamically according to actual network.

2. Reinforce dependency to parameters that have special marking meaning to network states to assure low FNP, and loose dependency to parameters that can not mark varieties of network states steadily to assure low FPP.

3. Avoid selecting irrelevant parameters to enhance efficiency of the model.

B. The mixed quadratic network statistical model

The statistical model is defined as follows, according to the thoughts mentioned above in A.

Definition one: Current network states of the current node X($x_1, x_2, \cdots, x_n$) $\in[0.0,1.0]^n$, and the history network state Y（$y_1,y_2,\cdots,y_n$）$\in[0.0,1.0]^n$, $x_i(1\leqslant i\leqslant n)$ contains values of different states parameters. In order to have the same testing standard, x_i is adjusted to be between 0 and 1. Y is the latest vector of X, that is to say, Y is the vector of network traffic passed by the current node in t time span, which contains the adjusted number of different destination ip addresses, the adjusted number of different source ip addresses and the adjustede number of packets that have same destinations and so on.

Definition two: States offset vector of the current node S($S_1, S_2, \cdots, S_n$), S= Y－X , which records changed degree of network states in the latest time span.

Definition three: Weight vector W($W_1, W_2, \cdots, W_n$) of different parameters, 0<= W_i<=1, 1<= i<= n, which reflects network traffic's dependency degree towards those parameters'.

Definition four : Improved X^2

$$X^2=\sum_{i=0}^{n} Wi\,|\frac{(Xi-Yi)\hat{2}}{Xi}|=\left\{\left\lfloor\frac{S\,\hat{\otimes}\,2}{vector}\right\rfloor\right\}W'$$

In the formula $\hat{\otimes}2$ is vector operation, which does square operation to each branch of the vector; $\lfloor\overline{vector}\rfloor$ is vector operation, which does divide operation to corresponding branches of two vectors.

Definition five: improved KS

$$D_{ks}=\sum_{j=1}^{k}\{\max_{(i,j)}\}\{Wi\,|\,Xi-Yi\,|\}$$ （1<=i<=n，0<k<n）

$\{\max_{(i,j)}\}Pi$ is the number j maximum in array P_i,

$\sum_{j=1}^{k}\{\max_{(i,j)}\}Pi$ contains k numbers that are larger than the others in array P(P_i) （1<=i<=n，0<k<n）.

Definition six: improved KKS

$$D_{kks}=\sum_{j=1}^{k}\{\max_{(i,j)}\}\{Wi(Xi-Yi)\}+\sum_{j=1}^{k}\{\max_{(i,j)}\}\{Wi(Yi-Xi)\}$$

（1<=i<=n，0<k<n）

Definition seven: improved AKS

$$D_{AKS}=\{\sum_{j=1}^{k}\{\max_{(i,j)}\}\{Wi \mid Xi-Yi \mid\}\}+\{\sum_{i=1}^{n} Wi \mid Xi-Yi \mid\}$$

（0<k<n）

Definition eight: improved **CC**

$D_{cc}=\| W_i*(X'\bullet X)-W_i^{'}*(Y'\bullet Y)\|$

$X'\bullet X, Y'\bullet Y$ are n-rank self-cross matrixes of X，Y respectively. ‖ ‖ is the absolute cortege value of matrix.

Definition nine: the mixed quadratic network statistical model

If the current network state is M（t）=（X^2,D_{KS},D_{KKS},D_{AKS},D_{CC}）and the history network state is M（t0）, then

G（X^2,D_{KS},D_{KKS},D_{AKS},D_{CC}）=| M（t）•（M（t）^{T}）－M（t0）•（M（t0）^{T}）|

In normal network states G will vary in a stable area. Once there are abnormities in G, we can suppose that there is abnormity in current network traffic. At this time we should take measures to make G be normal.

C. Analysis the statistical model

Set weight for each selected parameter to realize the dynamic principle. Weights that vary from 0 to 1 reflect the model's dependency degree towards those parameters. The model is independent of one parameter if the corresponding weight is 0, and the model has the biggest correlation to the parameter if the corresponding weight is 1. A method to set weights which are set dynamically by the model is shown as follows:

if（G is normal）{

for (I=1; I<=n;I++)

// U_i, D_i are upper limit and lower limit values of parameters that varied in normal network states

{

if ($\left|x_i-x_i^{'}\right|>U_i$)

//αand βare grow gene and minish gene respectively ,and $0<\alpha,\beta<1$

{ $W_i=W_i\times\beta$ }//loose the dependency degree

else if ($\left|x_i-x_i^{'}\right|<D_i$)

{$W_i=W_i+(1-W_i)\times\alpha$ }//strengthen the dependency degree

else { $W_i=W_i$ }}}else //G has abnormity

{for (I=1; I<=n;I++)

// $U_i^{'}, D_i^{'}$ are upper limit and lower limit values of parameters that varied in abnormal network states{

if ($\left|x_i-x_i^{'}\right|>U_i^{'}$)

{$W_i=W_i+(1-W_i)\times\alpha$}

//strengthen the dependency degree

else if ($\left|x_i-x_i^{'}\right|<D_i^{'}$)

{$W_i=W_i\times\beta$ }//loose the dependency degree

else { $W_i=W_i$ }}

}

Change weight values dynamically. At normal network states, 1) minish the weights of parameters whose weights vary in a large scope and loose the model's dependency degree to them for they can't mark network states steadily. 2) increase the weights of parameters whose weights vary in a small scope and enhance the model's dependency degree to them for they can mark network states steadily. At abnormal network states,1)increase the weights of parameters whose weights vary in a large scope and strengthen the model's dependency degree to them for they have special marking meaning to abnormal network states. 2)minish the weights of parameters whose weights vary in a small scope and loose the model's dependency degree to them for they hardly have any special marking meaning to abnormal network states.

This paper synthesizes thoughts of five statistics models--X2, DKS, DKKS, DAKS and DCC, and make after treatments by joining the dynamical principle to establish a model on these models--mixed quadratic network statistical model. This model has advantages of X2, DKS, DKKS, DAKS and DCC, however, we don't hope that it has all functions that other five models have, but expect it is better than any other one model, so it can describe network states more accurately.

Because G（X2,DKS,DKKS,DAKS,DCC）=| M（t）•（M（t）T）－M（t0）•（M（t0）T）|, so if X2, DKS, DKKS, DAKS and DCC are normal, then G will be normal. What we should do is to set logical fluctuant scopes, for there is a special situation that G will exceed its stable scope while X2, DKS, DKKS, DAKS and DCC are all normal, because it is likely that G is superposition of normal variety value of those five models. However, this is an extreme situation because X2, DKS, DKKS, DAKS and DCC have small varieties in normal network traffic and the superposition of these five values is very little comparing with the abrupt variety of abnormal network traffic.

D. Description of the algorithm

Pseudocodes of the algorithm to establish the intelligent rulebase in intrusion detection are shown as follows:

DA=IPAddress(packet) //get IP address

X=getX //X is a vector to save the current network states

Gnew=Test(X) //Gnew save tests results on X

CompG=Compare(Gnew,G) //return departure between //Gnew and G

If （CompG ≥departure scope of the threshold）{

Record and deal with these data, change corresponding weight W; //there is abnormal traffic

}

else{

analyse data packets, change corresponding weights W;

get destination port dPort；

```
if (dPort in NormalPorts ){          //dPort is common port
matching the rulebase according to the index dPort;
if matched{
confirm abnormity, give real time response according to rule's options;
}
else{
data packet is normal, give response;
get characteristics of normal data packets to optimize the rulebase;
}
}
else{      //dPort is aggregate port
dPortH1= H1(p1,p2,…,pn);          //pi is destination portdPortH2= H2(p1,p2,…,pn);      // Hi is a user-defined hash function
matching the rulebase with the index (dPortH1,dPortH2);
if matched{
confirm abnormity, give real time response;
}
else{
data packet is normal, give response;
get characteristics of normal data packets to optimize the rulebase;
}
}
}
```

Detect and analyse network traffic and IP addresses of data packets in IP layer based on network states. Give real time response directly to normal traffic or illegal traffic in IP layer, and analyse suspectable traffic in application layer. The intrusion detection rulebase in application layer is divided into three parts: index, rule head and rule option. We consider destination ports as the first index according to their characteristics. There are two types of destination ports: common ports and collective ports. The former type can be considered as index directly, while the latter type can not. So we give some user-defined hash functions H1(p1,p2,…,pn) and H2(p1,p2,…,pn), and (dPortH1,dPortH2) is considered as index. Rule heads record some header information, such as source IP/port, destination IP/port, and have pointers which point at next rule heads or at lists of rule options which are attached to certain headers. Rule options save all information of rule options, option functions and pointers that point at rule headers or at associate option nodes. When matching the rulebase of application layer, we should find out the rule header according to destination ports, and then carry out operations according to rule options corresponding to the rule head.

III. RULEBASE BASED ON INTRUSION DETECTION IN IP LAYER

This rulebase which mainly records statistical characterics of abnormal traffics is used to detect attacking data in abnormal traffic. This detecting method tries to accurately describe characteristics of acceptable static modes and dynamic actions to recognize abnormal and potential intrusions. IDS that uses this method must build a set of action characteristics for every normal action. In this part we adopt the twice-aggregation method [11].

A. Once-aggregation

Under normal conditions, the traffic towards every destination is even in principle, so we carry out once aggregation using this speciality. Dividing network traffic into several different aggregations by destination IP addresses of data packets, counting the number of data packets arrived at every destination, we can get reference values of detections of abnormal traffics. This process is shown by the following algorithm:

① Whether there are data packets towards purpose network?

② If yes, judge which aggregation the IP_daddr belongs to? If no, then turn to (1);

③ If the IP_daddr belongs to a certain Aggr[i], then add IP_num in this aggregation and turn to (1); otherwise increase a new aggregation and turn to (1) too;

Thereinto, IP_daddr is the destination address of one IP packet; Aggr[i] is one aggregation; IP_num is the number of data packets in an aggregation. At the same time, we need to set an upper limit and a lower limit for IP_num based on several statistical results to reduce the FNP and FPP. In the beginning, the number of aggregations is 0, and division for aggregations is implemented in the process of counting numbers of data packets. When detect intrusions in IP layer, if the number of data packets towards one destinaton is alien seriously from the corresponding IP_num, then we can see that the current traffic is abnormal.

B. Twice-aggregation

Based on the results of once-aggregation, we can give more elaborate division, for example, division into Tcp packets, Udp packets and ICMP packets and so on; we can also proceed with more elaborate divisions, such as IP In-packet Length、IP In-packet Traffic、IP In-Byte Traffic 、IP In-packet Rate、IP In-Byte Rate and so on. Then gain statistical states from the network statistical model and write them into the rulebase, which also deposits weight vector W(W1, W2, … , Wn), 0<= Wi<=1,(1<= i<= n) , thereinto, W reflects the testing model's dependency degree towards every action parameter. An action is considered as intrusion if it is alien from its corresponding set of action characteristics, and we should change these parameters' weights. This process can be described as follows:

① Acquire current network states X;

② Get the network statistical states Gnew(X2, DKS, DKKS, DAKS, DCC) according to these test algorithms mentioned above;

③ Compare Gnew with the history G, make sure which one or some parameters has been changed according to the deviation degree of test options, and change its(their) weitht(s);

④ Update current network statistical states in the rulebase based on Gnew.

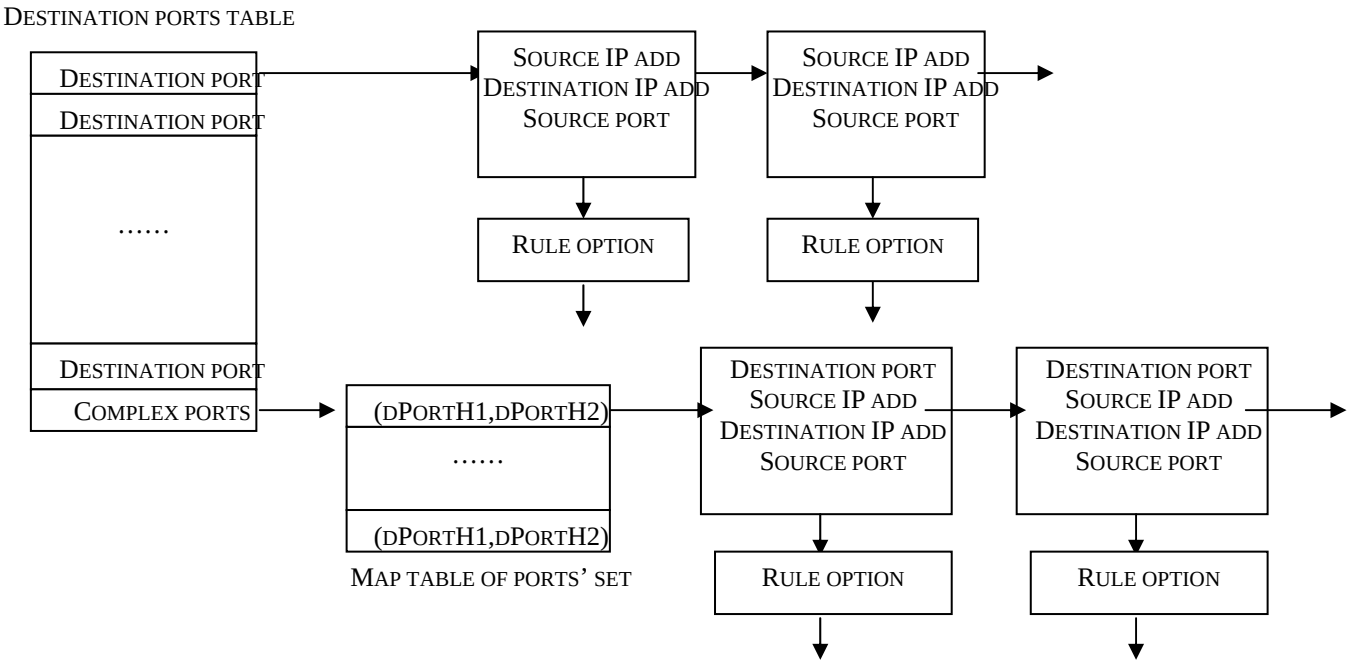

FIG.1 Construct the rule base

Intrusion detection in IP layer, that is anomaly detection, is not popular at present, because it has high FPP and FNP. This paper adopts a mixed quadratic network statistical model G(X2,DKS,DKKS,DAKS,DCC) to construct network statistical states to describe the current network accurately. In addition, using twice-aggregation to establish rulebase offers a new reference standard for keeping lower FPP and FNP.

IV. RULEBASE BASED ON INTRUSION DETECTION IN APPLICATION LAYER

Intrusion detection in application layer mainly detects contents of data packets, that is detecting intrusion by matching known intrusion modes. This rulebase is established using a method that is similar to Snort, and is divided into three logical parts: index, rule header and rule option. Analysing a large number of rules, we find that most destination ports are exclusive though they appear in sets together with source ports. We can see from all collected rules that most of them have single destination ports, and there are only about 100 kinds of rules sorted by destination ports, meanwhile, most of these rules have single destination ports. So destination ports can be considered as the first index. Moreover, there are some rules that can't be marked by single destination ports. For the sake of improving detection rate, this paper uses hash mapping to divide collective destination ports into distinguishable ones to form another index. We have two user-defined hash functions H1(p1,p2,…,pn) and H2(p1,p2,…,pn) to insure exclusion and noncrossing in mapping results, thereinto, pi is a destination port, and we consider pi as 0 if there is no pi. The rule header defines actions of rules, protocals of matching network packets, source addresses, destination addresses and so on. Rule options contain alarm information and other determinant information such as flag field in tcp and contents in data fields.

A. Establish rulebase on application

Rulebase is constructed based on following rules:① constructing the minimum and most effective rulebase; ② constructing a discrete rulebase. These rules guarantee that there is only one rule set for every data packet. In initializing stage, we construct the rulebase using independent parameters. First, because every transport protocols have different parameters that make them special; second, independent parameters conduce to establishing a smaller rulebase for detection engine; most important, it allows data packets pass through their corresponding rule subsets according to their characteristics.

Make destination ports as the first index, and arrange all rules according to the rule header which is the main list. Then insert one rule into this list according to rule options, so that there is only one rule corresponding to one option node. The rule option node deposits information of its rule options, option functions and pointers which point at one rule header node and another option node respectively. Rule header nodes deposit information of rule header, including source IP/port, destination IP/port, and pointers that point at next rule header nodes, rule option lists and rule header lists.

Destination ports can be divided into two kinds, one are ports that can mark rules uniquely, so they can be used as detecting index directly; another one are collective ports, which are mapped by two hash functions H1(p1,p2,...,pn) and H2(p1,p2,...,pn), and the mapped results are considered as detecting results. The mode constructing the rulebase is shown as Fig. 1. There are two tables in this part:

(1) destination ports table: it deposits ports that mark rules uniquely, and its last item deposits complex ports. Each item has a chain. Chains from unique ports link rule headers whose destination port is the certain port, and chains from complex ports link map tables of ports' sets.
(2) map table of ports' set: it deposits mapping results of ports' sets, that is a pair of (dPortH1,dPortH2). Each item in this table also has a chain, as the destination ports table has, which links mapping results and their corresponding rule headers.

B. Intrusion detection in application

When detecting one data packet, firstly find out whether its destination port is unique port or complex port. If it belongs to the former, find its corresponding port in the destination ports table and check parameters such as source IP address/port, destination IP address/port in rulesets along with the chain of this destination port, then if parameters of the data packet match those parameters in one ruleset then all rules in this ruleset as well as all other parameters in each rule are detected in file, finally turn to next ruleset if all rules in the previous ruleset have been detected and none of them is well matched. If the destination port is one of complex ports, compare the mapping result of this destination port with fields of the map table of ports' set and find out its corresponding (dPortH1,dPortH2), and then carry out the similar matching process mentioned above. So, intrusion detection is a process that matching data packets captured from network with rules in those rulesets. If there is one rule that matched a certain data packet very well then we can suppose there is attacking, and tackle this packet according to actions designed in this ruleset; if there is no matching rule in all rulesets then we suppose that this data packet is normal.

The rulebase will become increasingly huge along with the continuous joining of new rules, which will not only trouble ordinary maintenance but also affect performances of local hosts and the whole network. In addition, some existing detection rules may be invalid along with the variety of the environment, which can result in redundancy, so the rulebase must have intelligence to renew itself. Optimizing the detection rulebase is to describe the lifecycle of rules, including self-study of rulebase and resolving rules redundancy. We adopt filtering algorithm to achieve self-study and define "lifecycle of detection rules" and "quiet time of detection rules" as follows to resolve the problem of rules redundancy:

Life-cycle of detection rules T: T of one rule is the time span between its creation and deletion, which is set statically by intrusion detection system according to the current network. Different rules have different life-cycle. Generally we get the value of T using statistic methods. T must be moderate, because there is also rule redundancy if T is too large, while some useful rules are likely to be deleted if T is too small, which will heavy unnecessary burden.
Quiet time of detection rules quietT: quietT of one detection rule is the time span between its twice enabled time. The initialized value of quietT is 0, and it increases with time increasing. Whenever one rule is enabled, we need to set quietT as 0.

If the sleep time of one rule is larger than or equal to T, we suppose that this rule is invalid for there are no intrusions matching this rule to arrive and we should delete it from the rulebase; whereas, we suppose that this rule is still valid for current detection.

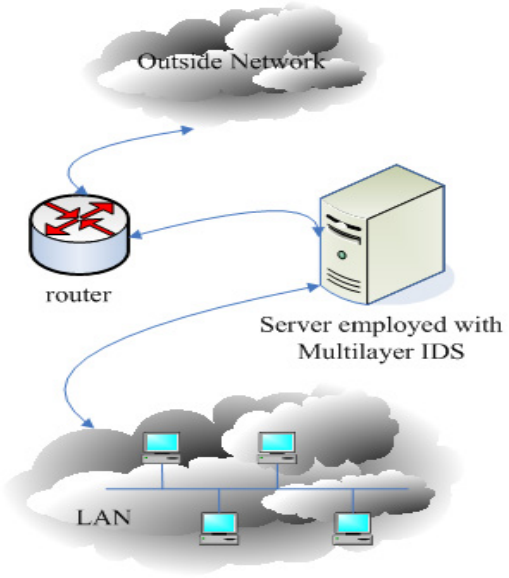

Fig. 2 The simulate topology of multiplayer IDS

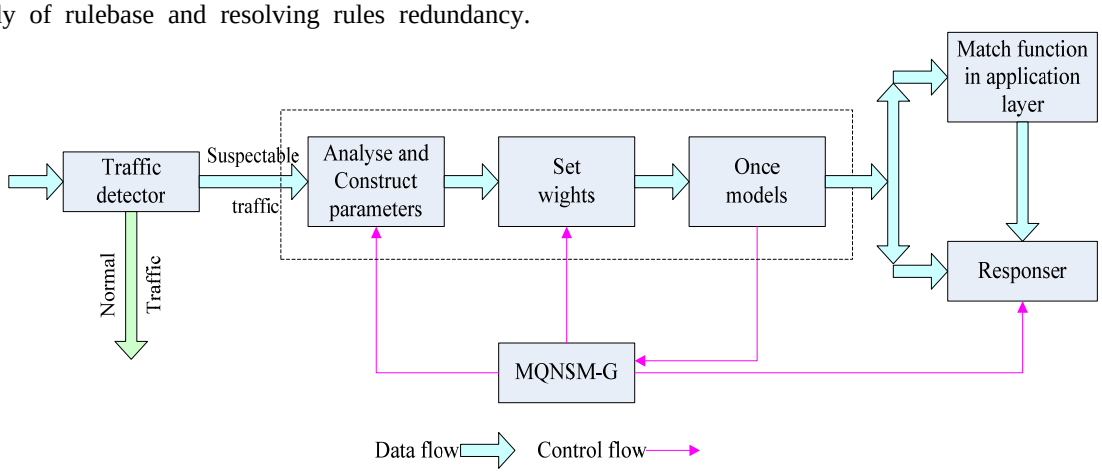

Fig. 3 System flow chart of IDS

V. THE SIMULATION

We apply this intelligent rulebase based on multi layer intrusion detection into an IDS to validate its performances and effection. The IDS is disposed on a server that has high performances between the router and LAN. The topology structure of the IDS is shown in Fig. 2 and its software flow chart is shown in Fig. 3. Data used in this experiment is from the MIT Lincoln Laboratory(American Lincoln's laboratory of M.I.T.) DARPA'98 database.

In this intrusion detection system, statistical results of χ^2、KS、KKS、AKS and CC are input values of the mixed quadratic network statistical model, and output results of this quadratic model will be feedback values to the analysis and constructing parameters mode and the setting weights mode respectively. The mixed quadratic network statistical model G(X2,DKS,DKKS,DAKS,DCC) together with the constructing parameters mode, the setting weights mode and once statistical models constitute this IDS, as the Fig. 3 shows.

Table1 features of network traffic

Index	Name	Weight
1	Ip-in-packet-length	0.353
2	Ip-in-packet-rate	0.462
3	Ip-in-byte-rate	0.481
4	Ip-in-frag-rate	0.152
……	……	……
27	Udp-diff-dst	0.0642
28	Icmp-anomal-echo-re	0.501

Table 2 experiment traffic

Index	Background traffic(bps)	Attack traffic(bps)	Ratio(R=A/B)
1	256738	25674	10%
2	256738	16688	6.5%
3	256738	10270	4%
4	256738	5135	2%
5	256738	2567	1%
6	256738	1284	0.5%

This IDS picks up 28 features of network traffic, some of which as well as their weights are shown in table 1. We suppose 6 different kinds of network traffics to detect the robust of this model. Different network traffics have different proportions, that is attacking traffic in background traffic, change from 10% to 0.5%, as the table 2 shows.

In the process of detecting 6 different kinds of traffics, we can see that if the proportion R is large enough then performances of every test are as excellent as others, and both FPP and FNP are low enough to be considered as 0. However, different test methods have different performances when R is diminishing, and there is no one test method that is superior than others at all times, for example, in the forth traffic AKS has the lowest MR which is 1.32%, and in the fifth traffic KS has the lowest MR which is 2.592, while here the MR of AKS is 11.52% (MR is the sum of FPP and FNP). This paper uses the mixed quadratic network statistical model, and can receive best test results in different traffics. Fig. 4,5,6,7

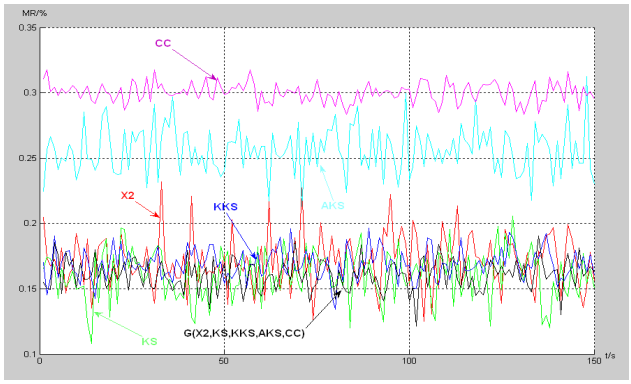

Fig. 4 test results of R=4%

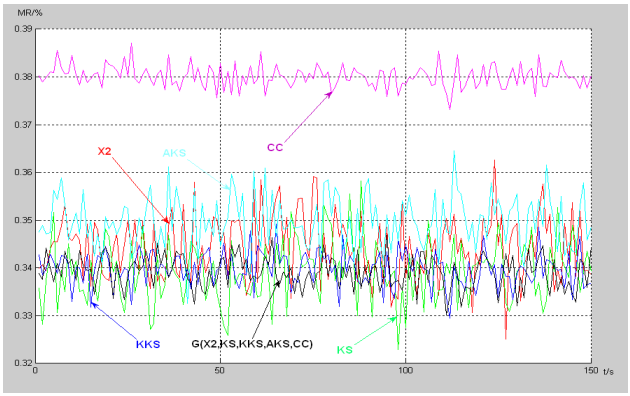

Fig. 5 test results of R=2%

respectively give different compared results between those once models and the mixed quadratic network statistical model in the latter 4 different network traffics.

When R=4% and R=2%, we test 100s respectively, and at R=1% and R=0.5% we test 150s respectively. Seen from those Fig.s, the mixed quadratic network statistical model G（X2,DKS,DKKS,DAKS,DCC） has the best and more stable performances in any network traffic.

VI. CONCLUSION

Realizing intrusion detection in multi layers is the developing trend of current security systems, which need rulebases based on multilayer detection. For the sake of improving the performances of IDS, the rulebase must be intelligent. This paper presents the research of intelligent rulebase based on multilayer intrusion detection and the rulebase is divided into two parts: the rulebase based on IP layer and the rulebase based on application layer, which are established in different methods respectively to adapt different detection needs in different layers. The simulation in the last part has proved that this rulebase has perfect performances.

ACKNOWLEDGMENT

This research work acknowledges the supports by the National Foundation of China under Grant No.60572131,

the Key Technologies R&D Program ofJiangsu Province of China under Grant No.BE2007058, the Graduate Cultivation InnovationProject of Jiangsu Province of China under Grant No.CX07S_027z, the Scientific Research Foundation of Huawei Corporation of China, and the Science Research Foundation of ZTE.

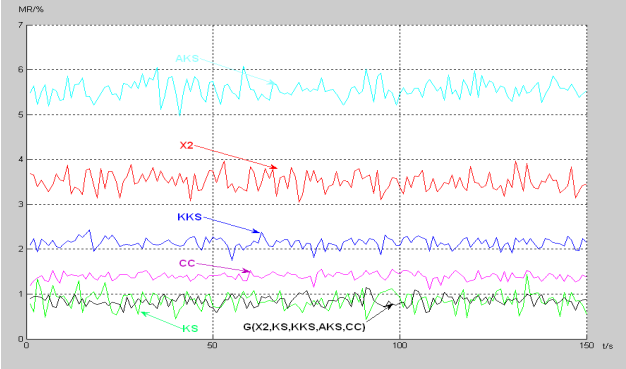

Fig. 6 test results of R=1%

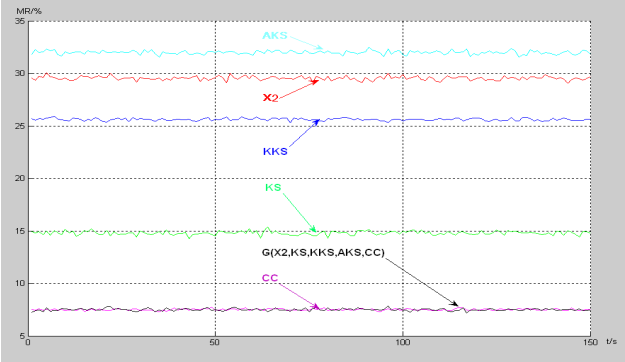

Fig. 7 test results of R=0.5%

REFERENCES

[1] ZHANG Fengbin ,YANG Yongtian ,J IANG Ziyang. “Genetic Algorithms in Intrusion Detection Based on Network Anomaly”. ACTA ELECTRONICA SINICA. Vol . 32 .No. 5 May 2004

[2] Qingbo Yin,Liran Shen,Rubo Zhang,Xueyao Li,“A New Intrusion Detection Method Based on Behavioral Model”IEEE , Intelligent Control and Automation, 2004. WCICA 2004. Fifth World Congress on , Volume: 5 , 15-19 June 2004 ,Pages:4370 - 4374 Vol.5

[3] LI Dequan1,XU Yiding , SU Purui , FENGDengguo. “Adaptive Packet Marking for IP Traceback”. ACTA ELECTRONICA SINICA. Vol . 32 No. 8 Aug 2004

[4] QING Si-han, JIANG Jian-chun, MA Heng-tai, WEN Wei-ping, LIU Xue-fei. “Research on intrusion detection techniques: a survey”. JOURNAL OF CHINA INSTITUTE OF COMMUNICATIONS. Vol.25 No.7 July 2004

[5] Bai,Y.;Kobayashi,H.;“Intrusion Detection Systems: technology and development”Advanced Information Networking and Applications, 2003. AINA 2003. 17th International Conference on , 27-29 March 2003 ,Pages:710 – 715

[6] Dwen-Ren Tsai; Wen-Pin Tai; Chi-Fang Chang;“A hybrid intelligent intrusion detection system to recognize novel attacks”, Security Technology, 2003. Proceedings. IEEE 37th Annual 2003 International Carnahan Conference on , 14-16 Oct. 2003 , Pages:428 - 434

[7] Zheng Zhang; Manikopoulos, C.N.;“Detecting denial-of-service attacks through feature cross-correlation” , Advances in Wired and Wireless Communication, 2004 IEEE/Sarnoff Symposium on , 26-27 Apr 2004 , Pages:67 - 70

[8] Jun Li; Manikopoulos, C.;“Early statistical anomaly intrusion detection of DOS attacks using MIB traffic parameters” , Information Assurance Workshop, 2003. IEEE Systems, Man and Cybernetics Society , 18-20 June 2003, Pages:53 – 59

[9] Snort Users Manual http://www.snort.org/docs/writing_rules/

[10] Rong-Tai Liu; Nen-Fu Huang; Chia-Nan Kao; Chih-Hao Chen; Chi-Chieh Chou; “A fast pattern-match engine for network processor-based network intrusion detection system”,Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on , Volume: 1 , 5-7 April 2004, Pages 97 - 101 Vol.1

[11] Ratul Mahajan, Steven M. Bellovin, Sally Floyd,John Ioannidis, Vern Paxson, and Scott Shenker. Controlling High Bandwidth Aggregates in the Network(Extended Version). AT&T Center for Internet Research at ICSI (ACIRI)and AT&T Labs Research_July 13, 2001- DRAFT