Using Enumeration in a GA based Intrusion Detection

S.N.Pawar
Asso.Professor (E &TC),
Jawaharlal Nehru Engineering College,
Aurangabad, MS, India.

R.S.Bichkar
Professor (E &TC),
G.H.Raisoni College of Engineering &Management,
Pune, MS, India.

ABSTRACT

In the last few years there has been a tremendous increase in connectivity between systems which has brought about limitless possibilities and opportunities. Unfortunately security related problems have also increased at the same rate. Computer systems are becoming increasingly vulnerable to attacks. These attacks or intrusions based on flaws in operating system or application programs usually read or modify confidential information or render the system useless. Different soft computing techniques are used for network intrusion detection (NID).This paper presents an effective GA based approach to generate the classification rules for network intrusion detection. While applying GA an, enumeration technique is used to select the gene values in a chromosome. This enumeration technique substantially reduces the computational time required for population generation and yields more appropriate rules. These rules are then used to detect the network intrusions. Experimental results show that this technique is more effective in detecting intrusions.

Keywords

Genetic Algorithms, Intrusion detection, Enumeration.

1. INTRODUCTION

When a computer system is connected to a network it goes on a high risk. There are various threats to a computer system such as viruses, Trojan horses, worms, intrusions etc. Viruses can be greatly controlled by installing antivirus software and updating it regularly.

Any unauthorized access causing violation to the security policy of a system is called intrusion to a computer. Intrusions cannot be predicted. Hence more focus is put on intrusion detection. The sooner we are able to detect an attack, the quicker we can act. Intrusion detection can help us to collect more information about attacks, strengthening the intrusion prevention methods. Various soft computing techniques such as Genetic Algorithm, Artificial Neural Network and Fuzzy Logic are used to make an intrusion detection system (IDS) smart enough to detect the intrusions at the earliest so that future damage can be avoided.

There are number of limitations to the prevention based approach for computer network security [1]. First, it is probably impossible to build a completely secure system. Further, the prevention based security philosophy constrains the user's activity and productivity. Hence intrusion detection systems are designed based on various detection techniques, namely Anomaly intrusion detection and Misuse intrusion detection [2].

Anomaly intrusion detection:

In anomaly IDS the user's behavior is compared with a known standard behavior to detects any significant deviation from normal behavior. This approach can be more effective in protection against unknown or novel attacks since no prior knowledge about specific intrusions is required. However it may cause more false positives because abnormality can be due to a new normal behavior [3].

Misuse intrusion detection:

This is the most widely used IDS. It uses patterns of known attacks or weak spot of the system to identify known intrusions. The signatures and patterns used to identify attacks consist of various options in the packet like source address, destination address, source and destination ports and even the key words in the content area of a packet.

An IDS can also be classified in to two categories based on their location [4], as host based and network based IDS.A host based IDS monitors activities associated with a particular host; whereas a network based IDS monitors activities associated with network.

GA is found to be the most efficient technique for intrusion detection in terms of detection accuracy at the expense of time [5]. Researchers have used GA for either generation of classification rules [2, 3, 6, 7] or for the selection of appropriate features of the chromosome [8, 9].

In this paper, we put forth an effective GA based approach to generate the classification rules for network intrusion detection. GA is selected because of its robustness, simplicity of operation and power of effect [10]. Enumeration technique has been used to arrange the gene values in different arrays. These gene values are picked up while generating the population. By using the enumeration technique the search space is substantially reduced which also reduces the computational time and helps in getting the more accurate results.

2. GENETIC ALGORITHM

Genetic Algorithm (GA) is a technique which works on the mechanics of natural selection. They are based on the Darwin's theory of survival of the fittest. Simplicity of operation and power of effect are the two main attractions of the genetic algorithm approach. They combine survival of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm with some of the innovative flair of human search [10].The major application of GA is in the area of optimization.

The GA process begins with a set of potential solutions or chromosomes which are randomly generated or selected. These chromosomes are normally encoded in the binary form but other forms of encodings are also used. The entire set of these chromosomes comprises a population.

In every generation the fitness of these chromosomes is checked. Fitness function is used to determine the fitness of the chromosomes. Based on the fitness, fittest chromosomes are selected. The chromosomes with poor fitness value are discarded. The selected fit chromosomes undergo crossover,

mutation to form a new population. This new population is used for the next generation. Normally, the algorithm terminates when either a set number of generations or a satisfactory fitness level has been achieved.

Genetic algorithm is composed of three operators. They are reproduction or selection, crossover or recombination and mutation.

Selection is the process of choosing parents for reproduction. There are many techniques to select the chromosomes, e.g. Roulette wheel selection, Boltzman selection, tournament selection, rank selection, steady state selection etc.

After two parents have been selected by the selection method, crossover takes place. Crossover is an operator that mates two parents (chromosomes) to produce two offspring. Two newborn chromosomes may be better than their parents and the evolution process may continue. The crossover is carried out according to the crossover probability. There are various crossover techniques like one point, two point, uniform etc.

Mutation is a genetic operator that alters one or more gene values in a chromosome. Mutation is carried out according to the mutation probability. It prevents the genetic population from converging to a local minimum. It adds new gene values to the gene pool. These new gene values may cause the genetic algorithm to arrive at better solution. The types of mutation are flip bit, random and min-max etc.

The chromosomes are then evaluated using a certain fitness criteria. When the GA terminates, the chromosome with best fitness is taken as the best solution of the problem.

3. RELATED WORK USING GA APPROACH

Different researchers have implemented GA in a different way for intrusion detection.

Middlemiss et al (2003) [8] have used GA for weighted feature extraction with specific application to intrusion detection data. It evolves weights for the features of data set. A k-nearest neighbor classifier was used for the fitness function of GA as well as to evaluate the performance of the new weighted feature set. These weighted features are used to scale the input variables provided to the classifier system.

This improves classification accuracy as feature space is compressed.

Gong et al. (2005) [4] have used GA for the generation of network intrusion detection rules. They have used support confidence function as a fitness function to judge the fitness of each rule. These rules are used for intrusion detection.

Zhao et al (2005) [11] have put forth a novel approach of using clustering GA to solve the computer network intrusion detection problem. The algorithm combines two stages in to the process including clustering stage and genetic optimization stage. The algorithm can not only cluster the cases automatically, but also detect the unknown intruded action.

Tao Xia et al (2005) [7] have put forth a hybrid method which uses both information theory and GA for intrusion detection. They used information theory to filter the traffic data and thus reduce the complexity. They used a linear structure rule to classify the network behavior in to normal or abnormal behaviors.

Chi Hoon Lee (2006) [9] presents the novel feature selection method that maximizes class separation between normal and attack patterns of computer network connections. The researchers have focused on selecting a robust feature subset based on the genetic optimization procedure in order to improve a true positive intrusion detection rate.

Saqib Ashfaq et al (2006) [6] have used a GA for generating efficient rules for cost sensitive misuse detection in intrusion detection systems.

Yong Wang et al (2009) [3] have put forth an efficient rule generator for denial of services of network intrusion detection. The rules generated by their algorithm are suitable to continuously changing misuse detection.

Chen Zhongmin et al (2009) [12] have designed a training algorithm model based on abnormality detection. The proposed experimental model is based on a hypothesis that if variable x appear more times than the desired value, there is possibility of occurring abnormality.

Thus, the researchers have implemented GA to generate the classification rules or to select the appropriate features. From the above discussion it is clear that Middlemiss [8] and Lee [9] have used GA for the selection of appropriate features where as Gong et al.[4], Saqib Ashfaq et al. [6], Tao Xia et al [7] and Yong Wang et al [3] have used GA for the generation of classification rules. Jiu-Ling Zhao et al (2005) [11] have used GA by integrating clustering analysis for intrusion detection.

4. THE PROPOSED GA BASED IDS USING ENUMERATION

In this work the classification rules are generated using GA approach. These rules are then used to classify or detect the infected connections.

4.1 Data set

MIT Lincoln Laboratory, under Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory (AFRL) sponsorship, has collected and distributed the first standard data for evaluation of computer network intrusion detection systems. This Data is DARPA 1998 data [13].It consist of tcpdump and BSM list files. Each line in a list file corresponds to a separate session. Each session corresponds to an individual TCP/IP connection between two computers. The first nine columns in list files provide information which identifies the TCP/IP connection.

We have also used this DARPA 1998 data for the generation of GA based rule set and then for the detection of infected connections.

We have used two subsets from this data set: One is used for training the system and the other is used to test the detection rate.

4.2. Feature selection and representation

Seven most important features of the data set are selected for defining the intrusion rules. These are duration (h: m: s), service (int.), source port (int.), destination port (int.), source IP (a. b. c. d), destination IP (a. b. c. d), attack name (string).

These features or genes can be represented in different form. They can be encoded in the binary, integer or float.

The three parts of the 'Duration' such as hours, minutes, and seconds are encoded in one gene of type byte. The features 'Protocol', 'Source port', 'Destination port' and 'Attack name' are encoded using one gene of type integer [4].

The feature 'Source IP' and 'Destination IP' have four octet parts such as a, b, c and d. Each part is encoded in one gene of type byte. All the classification rules are in if-then form. All the features except attack name forms a condition 'A' where as attack name is the outcome 'B' of that condition [4].

We can use wild card values in each of the fields of the rule. We have used wild card values in the third and fourth octet of both source IP and destination IP. We have put -1 in the field chosen for the wild card.

4.3 Enumeration

Enumeration is the ordered listing of all the elements in a set. Normally the elements of a set are arranged in the natural ordered form. But if the natural ordering is not possible, then a particular ordering is imposed. Enumeration for a set is possible only if the set is countable.

Normally, while generating the genes, the range of values for each gene is defined and then randomly each gene is generated [14].We have instead used enumeration technique to determine the value of each gene for the chromosome. Each gene value from the data set is listed in an ordered fashion. Then the id of gene value is randomly chosen out of these listed set. By using the enumeration technique the search space is substantially reduced which also decreases the computational time and helps in getting more accurate results.

4.4 Fitness function

The fitness function of a rule is decided by checking the number of times it matches with the record connections [4]:

$Support = |A \text{ and } B| / N$

$Confidence = |A \text{ and } B| / |A|$

$Fitness = w_1 * support + w_2 * confidence$

Where, N is the total number of connections, $|A|$ is the number of connections where the rule matches the portion of the connection matching the first six features. $|A \text{ and } B|$ represent the number of connections that matches both condition part A and outcome part B of the rule. The weights w_1 and w_2 can be adjusted to fine tune the algorithm and have the values of $w_1 = 0.2$ and $w_2 = 0.8$.

4.5 Crossover and Mutation

Crossover is one of the important steps in GA. There are three types of crossover techniques. They are one point, two point and uniform cross over technique. We have used a two point crossover technique with a probably Pc.

Each gene in each chromosome is checked for possible mutation by generating a random number between zero and one and if this number is less than or equal to the given mutation probability i.e. Pm, then the gene value is changed. Mutations create diversity to search in domain regions that may otherwise be excluded.

5. IMPLEMENTATION AND RESULTS

The genetic algorithm for rule generation is implemented using Java language (JDK6) in NetBeans7.0.The front end development environment used is NetBeans7.0. Two subsets were developed from DARPA 1998 data.

Table 1 gives the distributions of record types in both training and testing data set. The first row gives the number of normal network records. The second row gives the distributions of Smurf attack whereas the third row gives the distribution of Neptune attack.

Table 1. The distribution of record types

Record Type	Training Set	Testing Set
Normal	45711	4513
Smurf	524	101
Neptune	15	10

The implementation is done in two phases. In the first phase the classification rules are generated using genetic algorithm. Support confidence function as fitness function. The GA parameters used were w1 = 0.2, w2 = 0.8, 2000 generations, population of 300 rules, crossover rate of 0.5, two-point midsection crossover and mutation rate of 0.01.

After generating the classification rules in the first phase, the top ten fittest rules were taken for detection purpose.

Table 2 gives the top ten rules generated for attacks. For Smurf attack, 10 fittest rules are selected where as for Neptune; only one fittest rule is generated. All these rules are different from each other.

Table 2. Fittest top ten rules for detection of attacks

Duration				Source Port	Destination Port	Source IP	Destination IP	Status	Attack
Hours	Minutes	Seconds	Protocol						
0	0	34	ecr/i	-	-	017.139.-1.-1	172.016.-1.-1	1	Smurf
0	0	35	ecr/i	-	-	017.139.-1.-1	172.016.-1.-1	1	Smurf
0	0	34	ecr/i	-	-	090.207.-1-1	172.016.-1.-1	1	Smurf
0	0	35	ecr/i	-	-	090.207.-1-1	172.016.-1.-1	1	Smurf
0	0	34	ecr/i	-	-	121.014.-1.-1	172.016.-1.-1	1	Smurf
0	0	35	ecr/i	-	-	121.014.-1.-1	172.016.-1.-1	1	Smurf
0	0	34	ecr/i	-	-	017.139.-1.-1	172.016.-1.-1	1	Smurf
0	0	35	ecr/i	-	-	017.139.-1.-1	172.016.-1.-1	1	Smurf
0	0	34	ecr/i	-	-	057.111.-1.-1	172.016.-1.-1	1	Smurf
0	0	35	ecr/i	-	-	057.111.-1.-1	172.016.-1.-1	1	Smurf
0	0	01	telnet	-1	23	001.002.003.004	172.016.-1.-1	1	Neptune

In the second phase, these rules are used to classify both training as well as testing data set. As shown in the table2, the attacks are of denial of service type.

Table 3 shows the percentage detection for different number of generations of GA. As the number of generations is increased, the detection rate is improved at the cost of increased time required for the generation of rules. The best results are obtained after 2000 generations.

Table 3.The Detection rates for different generations

Sr. No	Generations	Record Type	% Detection	
			Training	Testing
01	500	Normal	76.0	74.0
		Smurf	79.0	76.6
		Neptune	82.0	81.0
02	1000	Normal	82.3	81.0
		Smurf	83.5	82.0
		Neptune	85.0	83.0
03	1500	Normal	92.0	91.0
		Smurf	93.0	92.2
		Neptune	96.6	96.0
04	2000	Normal	98.3	98.2
		Smurf	98.5	96.0
		Neptune	100.0	100.0

In Table 4, the comparison between Gong et al.'s approach and our approach is given. From this comparison it is evident that the detection rate of our approach is better. As the population used and the numbers of iterations used are less in number, the computational time is also substantially reduced.

Hence the presented method is more effective in detecting the attacks.

Table 4. Comparison between Gong et al.'s approach and proposed approach

Sr. No	Parameter	Gong et al System.	Our system
1.	Average Detection Rate of attacks	Training 94.6% Testing 79.86%	Training 98.93% Testing 98.06%
2.	GA Population	500	300
3.	No of Generations	5000	2000

Table 5 gives the percentage detection rate of DOS attacks for different approaches. It is observed that Jiu-Ling Zhao et al.'s [11] approach which employs clustering genetic algorithm gives 95% detection rate. Jing Xiaopei [15] et al.'s approach used improved genetic algorithm in intrusion detection model based on artificial immune theory. Its detection rate is 99.29%.

Hua Zhou et al [16] used support vector machine and genetic algorithm for network intrusion detection. Shingo Mabu et al.'s [17] approach is based on fuzzy class-association-rule mining using genetic network programming and Yu-Ping Zhou et al [18] employed fuzzy genetics-based rule classifier in intrusion detection system.

Table 5. Detection rate comparison with other approaches

Sr. No	System by	% Detection Rate
1	Jing Xiaopei et al [15]	99.3
2	Jiu-Ling Zhao et al [5]	95.0
3	Hua Zhou et al [16]	98.9
4	Shingo Mabu et al[17]	98.7
5	Yu-Ping Zhou et al[18]	97.3
6	Our system	98.1

From the above table it is evident that the results obtained by our approach are comparable with others.

6. CONCLUSION

In this paper, a GA based intrusion detection technique is presented. The system is implemented in two steps. In the first step, GA is used to generate classification rules where as in the second step these rules are used for intrusion detection.

Enumeration technique is used for generating the population in a genetic algorithm. This reduces the search space and yields more accurate results while using smaller population and lesser number of generations compared to Gong et al.'s approach. This has reduced the time required for the generation of fittest rules.

The given system is run for different generations. As the number of generations is increased, more accurate intrusion detection rates are obtained.

From experimental results it is evident that the given technique is effective in network intrusion detection.

7. REFERENCES

[1] A.Vesely, D.Brechlerova, "Neural Networks in Intrusion Detection Systems", AGRIC.ECON.CZECH, 50, 2004 (1):35-39.

[2] S.Owais, V.Snasel and P.Kromer, A. Abraham, "Survey Using Genetic Algorithm Approach in Intrusion Detection Systems Techniques", 7th Computer Information Systems and Industrial Management Applications, 2008, IEEE press, June2008, pp.300-307, DOI 10.1109/CISIM. 2008.49.

[3] Yong Wang, Dawu Gu, Xiuxia Tian, Jing Li, "Genetic Algorithm Rule Definition for Denial of Services Network Intrusion Detection", International Conference on Computational Intelligence and Natural Computing, IEEE, 2009, pp.99-102.

[4] Ren Hui Gong, Mohammad Zulkernine, Purang Abolmaesumi, "A Software Implementation of a Genetic Algorithm Based Approach to Network Intrusion Detection", SNPD/SAWN'05, IEEE, 2005.

[5] Srinivas Mukkamala and Andrew H. Sung, "A Comparative Study of Techniques for Intrusion

Detection", Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'03), IEEE, 2003.

[6] Saqib Ashfaq, M.Umar Farooq, Asim Karim, "Efficient Rule Generation for Cost-Sensitive Misuse Detection Using Genetic Algorithms," IEEE, 2006

[7] Tao Xia, Guangzhi Qu, Salim Hariri, Mazin Yousif, "An efficient Network Intrusion Detection Method Based on Information Theory and Genetic Algorithm", IEEE, 2005.

[8] Melanie Middlemiss, Grant Dick, "Weighted Feature Extraction Using a Genetic Algorithm for Intrusion Detection", 2003 Congress on Evolutionary Computation (cec-03) 2003, pp.1669-1675.

[9] Chi Hoon Lee, Sung Woo Shin and Jin Wook Chung, "Network Intrusion Detection Through Genetic Feature Selection", SNPD, IEEE, 2006.

[10] Biswanath Mukherjee, L.Todd Herberlein and Karl N. Levitt, "Network Intrusion Detection", IEEE Network, 8(3):26-41, May/June 1994.

[11] Jiu-Ling Zhao, Jiu–Fen Zhao, Jian –Jun Li, "Intrusion Detection Based on Clustering Genetic Algorithm", International Conference Based on Machine Learning and, Cybernetics, IEEE, Guangzhou, 2005, pp.3911-3914.

[12] Chen Zhongmin, Feng Jianyuan, Xu Sheng, Xu Renzuo, "The Research of Intrusion Detection Technology Based on Genetic Algorithms," International Conference on Networks Security, Wireless Communications and Trusted Computing, IEEE, 2009.

[13] MIT Lincoln Laboratory, DARPA datasets, MIT, USA, http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/1998data.html

[14] Wei Li, "Using Genetic Algorithm for Network Intrusion Detection," Proceedings of the United States Department of Energy Cyber Security Group, 2004.

[15] Jing Xiaopei, Wang Houxiang, Han Ruofei, Li Juan, "Improved Genetic Algorithm in Intrusion Detection Model Based on Artificial Immune Theory," IEEE, 2009

[16] Hua Zhou, Xiangru Meng, Li Zhang, "Application of Support Vector Machine and Genetic Algorithm to Network Intrusion Detection," IEEE, 2007

[17] Shingo Mabu, Ci Chen, Nannan Lu, Kaoru Shimada, and Kotaro Hirasawa, "An Intrusion-Detection Model Based on Fuzzy Class-Association-Rule Mining Using Genetic Network Programming," IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews. IEEE, 2010

[18] Yu-Ping Zhou, Jian-An Fang, Dong-Mei Yu, "Research on Fuzzy Genetics-Based Rule Classifier in Intrusion Detection System," 2008 International Conference on Intelligent Computation Technology and Automation. DOI 10.1109/ICICTA.2008. 241. IEEE, 2008