ISSN: 2278 – 7798
International Journal of Science, Engineering and Technology Research (IJSETR)
Volume 2, Issue 10, October 2013

An effective rule generation for Intrusion Detection System using Genetics Algorithm

Miss Priya U. Kadam, Mr. P. P. Jadhav
Department of Computer Engineering, Pillai's Institute of Information Technology, New Panvel
University of Mumbai, Mumbai (INDIA)

Abstract— **In today's scenario the security is important in vast growing computer networks, so Intrusion Detection System is essential task in daily life practices. There are various approaches being utilized for intrusion detection. In this paper, we present an Intrusion Detection System using Genetic Algorithm for efficiently and effectively to identify various types of intrusions or attack. We propose effectiveness and accuracy of an approach to generate rules for different types of anomalous connection. The KDDCUP99 training and testing dataset is used to generate effective new rules by adopting reasonable detection rate.**

Index Terms— **Computer & Network Security, Intrusion Detection System, Genetic Algorithm, KDDCup99 Dataset.**

I. Introduction:

Nowadays, computer network is contaminated by various threats such as viruses, Trojan horses, worms, intrusions etc. When we connect our system to these networks may be attacked by such intrusions. For such reason the IDS is used to control the different types of attack. The various methodology and approaches has been developed and deployed for IDS using such as ANN, Data mining, Fuzzy logic etc. Intrusion detection technique is classified broadly in the two main groups: misuse intrusion detection and anomaly intrusion detection.

Manuscript received Sep, 2013.

Miss. Priya Uttam Kadam, *Department of Computer Engineering, PIIT, New Panvel, University of Mumbai Mumbai, India.*
Mr. P. P. Jadhav, *Department of Computer Science , University of Mumbai, Mumbai, India*

Misuse Detection:

Recognized intrusions are detected by looking at the computer system behavior some attribute pattern of such intrusions. This move toward some collected information about the system behavior under standard conditions and under some identified intrusions to decide the current state of the system. In this case, the intrusion detection problem is a classification problem.

Anomaly Detection:

Familiar and unidentified intrusions are detected by analyzing changes in the standard pattern of utilization or behavior of the computer system. This approach does not use information about the system behavior when an intrusion is in progress. Basically IDS is classified in to two categories: Host based and Network Based IDS.

Network Based (NIDS):

it examines network traffic to recognize threats that generate abnormal traffic flows, such as DoS attacks, scanning, and certain structures of malware.

Host-Based (HIDS):

it monitors the characteristics of a single host and the events happening within that host for doubtful activity. GA is most efficient technique for intrusion detection in terms of detection accuracy at time constraint. Researchers have used GA for either generation of classification rules or for the selection of appropriate features of the chromosome. Many of soft computing based approaches have been proposed for detecting network intrusions. Soft computing techniques are often used in conjunction with rule-based expert systems acquiring expert knowledge [1, 4, 5, 6], where the knowledge is represented as a set of *if-then* rules. Although different soft computing based approaches having been proposed, the possibilities of using the techniques for intrusion detection are still under-utilized.

In this paper, we demonstrate GA based approach to generate the effective classification rules for network intrusion detection. GA is selected because of its cost effectiveness,

All Rights Reserved © 2013 IJSETR

robustness, simplicity of operation. This technique has been used to arrange the gene values in different arrays. These gene values are picked up while generating the population. By using this technique reduces the computational time and helps in getting the more accurate results.

II. Related Work:

GAs has been used for network intrusion detection in different ways. Some of the approaches directly use GAs for to obtain the classification rules [4, 5, 6, 7], while others use different AI methods for possession of rules, where GAs are used to select appropriate features or to determine the optimal parameters of some functions [8, 9]. Li [5] represent a technique using GA to detect abnormal network intrusion. This approach includes is obtaining classification rules for quantitative and distinct features of network data. Apart from the implementation of rule generation for IDS is given but results of experiments do not exist. Bridges [3]: This method is a combines both fuzzy data mining techniques and Genetic Algorithm for detection of network anomalies and misuses. The most features are not predicted properly in various existing Genetic Algorithm based IDS's. This method uses Genetic Algorithm to recognize the optimal parameters of the fuzzy functions for selecting the features of the relevant network.

Lu [7]: In this method classification rules are generated by Genetic Programming. Detection or Classification of intrusions in the network with the help of the fitness function is fine tuned by this method. The time required to train the system with huge data creates Genetic Programming implementation difficult. Crosbie [2]: Different agent techniques and Genetic Programming can be used to detecting network intrusions. The set of agents that determine the network behaviors can be finding out by an agent who monitors one parameter of the network audit data and Genetic Programming. Many small autonomous agents can be used in this method which is an advantage and the communication among the agents is a drawback. Selvakani [3]: This system identifies the attacks using rule set by proceeding Genetic Algorithm, then exploit rules only for R2l and DoS type of attacks. Between these two attacks, one from each is selected. The common performance of the system is less than 60%.

A Genetic Algorithm (GA) is a programming technique that reproduces biological evolution as a problem-solving strategy. GA is a technique which works on the mechanics of natural selection. It is based on the Darwin's theory of survival of the fittest. 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 find out the fitness of the chromosomes and then selection operator will choose the fittest chromosomes using tournament selection. The chromosomes with poor fitness value are discarded.

GA uses an evolution and natural selection that uses a chromosome-like data structure and evolve the chromosomes using selection, recombination (crossover), and mutation operators. The process generally begins with arbitrarily generated population of chromosomes, which represent all potential solution of a problem that are measured applicant solutions. Different positions of each chromosome are encoded as bits, characters or numbers, which is refer as genes. An evaluation function is used to compute the decency of each chromosome according to the desired solution is known as "Fitness Function". For the period of evaluation, the basic two operators, crossover and mutation, are used to imitate the natural reproduction and mutation. The selection of chromosomes for survival and combination is biased towards the best fit chromosomes [9].

The following figure shows the structure of a simple genetic algorithm. Starting by a random generation of initial population, then evaluate and evolve through selection, recombination (crossover), and mutation. Finally, the best individual (chromosome) is picked out as the final result once the optimization meets it target.

Advantages:

- Genetic algorithms are parallel, because of multiple offspring can be generated are utilized as possible solution.
- It offers multiple solutions for particular problem.
- It also optimizes the new rules or improves the rule set for IDS.

Application by GA to IDS

The GA is applied to IDS for feature selection and rule generation. In feature selection some of the parameter of connection records having value 1 is feature is present is selected. are selected in such way that they a

III. GA For IDs:

GA_Rule _Generation

Input: Encoded binary string of length n (where n is the number of features being passed), number of generations,

All Rights Reserved © 2013 IJSETR

population size (, crossover probability (Pc), mutation probability (Pm).

Output: A rule set generation for IDS.

1. Initialize the population randomly with the size of each chromosome to be 41.
2. Initialize N (total number of records in the training set), Pc=0.8 and Pm=0.088.
3. for each chromosome in the new population
4. Calculate fitness= Fx/Sum (Fx)
5. End for
6. Select 50% best fit chromosome and remove worse fit chromosome.
7. Apply Crossover to best selected chromosome.
8. Apply Mutation for each chromosome to generate new population .go to step no3.
9. Stop

GA Parameters

GA has some general elements and parameters which can be defined:

GA Operators The different GA parameter selection mutation and crossover are the most successful parts in the algorithm as they are contribute in the generation of each population.

Selection phase where population individuals with superior fitness are selected, otherwise it gets damaged.

Crossover is a method in each pair of each individuals selects arbitrarily participates in exchanging their parent's genes with each other, until an entire new population has been generated.

Mutation flips some of the bits in an individual, and since all bits could be filled, there is low probability of predicting the change.

Fitness Function The fitness function is defined as a function which scales the value individual relative to the rest of population. It generates the best possible solutions from the amount of candidates located in the population.

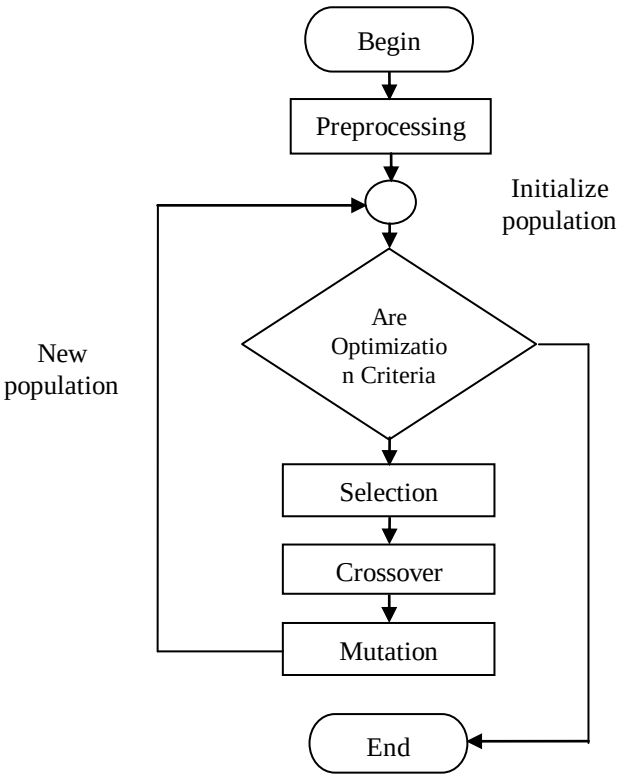
In preprocessing phase the KDDCUP99 Dataset is processed by using Weka tool which is used to remove the redundant data from existing Dataset which result in tested Dataset. The removal of redundant data or records from Dataset it improves the detection rate of desired result and improves the performance of our system.

In detection phase the Genetic Algorithm is applied on chosen features data set and locate fitness for every rule with the following fitness function.

Fitness = Fx / sum (Fx)

Where Fx is the fitness of individual x and sum (Fx) is the entire fitness of all individuals.

Diagram:



IV. Data Set:

KDDCUP99 is based on DARPA data from MIT Lincoln Laboratory is broadly used to evaluate IDSs. In this study, we used the KDDCUP99 training and testing datasets.

Attack Type	Trained Data	Test Data
Normal	972781	60593
DoS	3883370	223298
Probe	38786	2293
R2l	12616	6075
R2r	46	39
Total	4907599	292298

Table : 1

All Rights Reserved © 2013 IJSETR

Each record of the datasets consists of 41 network features and 1 manually assigned record type. Nine network features were used in the GA which is *source IP address, destination IP address, protocol, source port, destination port, connection duration and dst_hst_service_count.* etc. The record type indicates whether a record is a normal network connection or abnormal network connection.

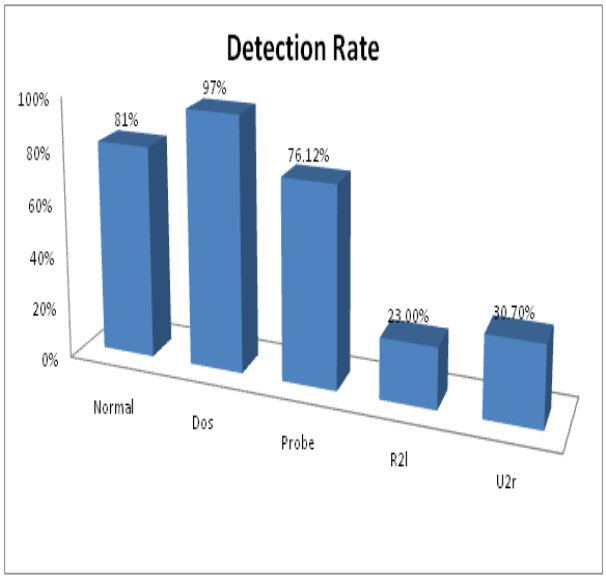


Table shows the distributions of record types in testing datasets. The records are categorized as: DoS, Smurf, R2l and U2r. These are the network attacks.

The KDD 99 intrusion detection benchmark consist different components: kddcup.data; kddcup.data_10_percent; kddcup.newtestdata_10_percent_unlabeled; kddcup.testdata.unlabeled; kddcup.testdata.unlabeled_10_percent; corrected.

V.Result:

From our implementation, we have effectively produced some rules those classify the declared attack connections. Applying Genetic Algorithm on chosen features set and finds the fitness value for every creation

	Trained Data	Test Data	Detection Rate
Normal	972781	60593	81.25%
DoS	3883370	223298	97.80%
Probe	38786	2293	76.12%
R2l	12616	6075	23.00%
R2r	46	39	30.70%
Total	4907599	292298	

Table 2 : Different attack types with detection rate

VI. How to apply GA:

In the preprocessing phase, we are just removing the redundant records from dataset because it affects the detection rate. For t we use Weka tools to tested dataset and we got non-redundant dataset. The Detection phase, an initial population is available from tested data. The population is selected randomly with size of each chromosome 41 from dataset and values are encoded in binary strings as 0s and 1s. Crossover probability is set to 0.8 and mutation probability to 0.08. From this new population calculates the fitness function as strength of one record divided by upon sum of strength of all records. Among these records select the 50% best fit chromosome and removed the other worse fit chromosome. After that we apply the Crossover to generate new population again and flip the bit for mutation. If our criteria are matches with existing pattern then stop.

The group of the chromosome is close to relative of only existing chromosome of test data is return as the predicted type. Among the extracted features of the datasets, we have taken only the numerical features, both continuous and distinct, under deliberation for the sake of the generalization of the implementation.

Gets rules

Rule 1:

If (duration < 10 seconds) of an FTP connection/session, there are many Hot indicators (hot > 20) being set by a logged user then it is highly likely that warezclient attack is being executed

Rule 2

if {the connection has following information: source IP address 124.12.5.18; destination IP address:130.18.206.55; destination port number: 21; connection time: 10.1 seconds } Then {stop the connection}

Rule 3:

If (source_bytes > 265616) and(source_bytes <= 283618) Then Warezmaster Attack

Rule 4:

If (Duration <3)and (protocol_type=icmp) and (dst_byte=125016) Then Buffer_overflow

Rule 5:

If (Duration 0 to 25) and (protocol_ type = tcp and UDP) and (service=ftp OR private OR other domain) Then guesspassword

Rule 6:

If (is_host_login=0)and(service=daytime) and (flag=135) and (srcbyte > 16998.134) Then perl

Rule 7:

(duration < =6)and(protocol = tcp) and(service = ftp and icmp) and
(logged_in = 1 and is_guest_login = 1) (hot > 20)Then ftp_write

All Rights Reserved © 2013 IJSETR

VII. Conclusion:

In this paper, a method of applying genetic algorithms for network intrusion detection is presented. A number of experiments have been carried out using a benchmark data set in order to show the efficiency of our system. The major advantages of this proposed detection system can be generating the new rules to the systems as the new intrusions become known. Therefore, it is cost effective and efficient approach to IDS. A GA is used to obtain a set of classification rules network audit data. The nine features including both categorical and quantitative data fields were used when encoding and obtaining the rules. A simple but effective and flexible fitness function is used to select the appropriate rules. Depending on the selection of fitness, the generated rules can be used also to detect network intrusions or categorize the types of intrusions.

The Genetic Algorithm based Intrusion Detection System's detecting several types of attacks is possible with a high rate of rule set provided. An average detection rate of 91.025% and decrease the percentage of false alarms in the results of experiments are good. The IDS should be capable in detecting complex intrusions. The results of the paper specify the set of rules and high R2l and U2r attack detection rate.

References:

[1] Mohammad Sazzadul Hoque, Md. Abdul Mukit and Md. Abu Naser Bikas "An Implementation of Intrusion Detection System using Genetic Algorithm", International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.2, March 2012.

[2] Crosbie, Mark, and Gene Spafford. 1995. "Applying Genetic Programming to Intrusion Detection". In Proceeding of 1995 AAAI Fall Symposium on Genetic Programming, pp. 1-8. Cambridge, Massachusetts.

[3] Bridges, Susan and Rayford B. Vaughn. 2000. "Intrusion Detection via Fuzzy Data Mining", In Proceedings of 12th Annual Canadian Information Technology Security Symposium, pp. 109-122. Ottawa, Canada.

[4] Gong R.H, Zulkemine.M, Anolmaesumi.P, "A Software Implementation of a Genetic Algorithm Based approach to Network Intrusion Detection", Proceedings of the SNPD/SAWN'05, PP.19-27, Aug 2005.

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

[6]Chittur.A, "Model Generation for an Intrusion Detection System using Genetic Algorithms", High school Hornors Thesis, http://www1.cs.columbia.edu/ ids/ publications/ gaidsthesis01.pdf, accessed in 2006.

[7] W. Lu and I. Traore, "Detecting new forms of network intrusion using genetic programming", Computational Intelligence Vol.20, Issue 3, August 2004, pages 475-494

[8]J. Gomez, D. Dasgupta, "Evolving Fuzzy Classifiers for Intrusion Detection", Proceedings of the IEEE, 2002.

[8] Bridges S.M and Vaughn R.B,"Fuzzy Data Mining and Genetic Algorithms Applied to Intrusion Detection", Proceedings of 12th Annual Canadian Information Technology Security Symposium, pp. 109-122, 2000.

[9] B. Addullah, I. Abd-alghafar, Gouda I. Salama and A. Adbalhafez, "Performance Evaluation of a Genetic Algorithm Based Approach to Network Intrusion Detection System", ASAT-13-CE-14, May 26-28, 2009.

[10] KDDcup 1999 data,

http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

All Rights Reserved © 2013 IJSETR