ISSN No. 0976-5697

Volume 2, No. 6, Nov-Dec 2011

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Genetic Algorithm based Network Intrusion Detection System

Nitin Gupta*, Nitin Pareek and Kavita Pandey
Computer Sc. Department, JIIT,
Noida, India
nitin2341989@gmail.com, nitinpareek31@gmail.com, kavita.pandey@jiit.ac.in

Abstract-The traditional prevention techniques such as user authentication, data encryption and firewalls are used as the first line of defence for computer security. If a password is weak and compromised, user authentication cannot prevent unauthorized use of system. So we developed a network intrusion detection system, it would be mainly focused on denial of service attacks. Genetic Algorithm would be used to identify the network intrusions. The most important idea that stands beyond the initial creation of GAs is the aim of developing a system as robust and as adaptable to the environment as the natural systems. For fitness criteria, probabilistic approach is used to calculate the optimality of the proposed new attack pattern.

Keywords- Genetic Algorithm, Probabilistic fitness Criteria, intrusion detection, Smurf, Neptune, Ping of Death, Back attack.

I. INTRODUCTION

With the ever-increasing growth of computer networks and emergence of electronic commerce in recent years, network security has become a priority. Society has grown to rely on Internet services, and the numbers of Internet users are increasing every day. Internet and local area networks are expanding at an amazing rate in recent years, not just in the terms of size, but also in the terms of changing the services offered and the mobility of users that make them more vulnerable to various kinds of complex attacks. While people are benefiting from the convenience that new technology has brought to us, but it also increased the number and complexity of security threats. As more and more users become connected to the network, the opportunity for malicious users to do their damage also increases. Thus there is a great need of new network security policies in order to detect and react as quickly as possible to the occurring attacks. Therefore security has become a crucial issue for networks. There are various kinds of attacks possibilities like network Trojans (spyware) which can steal important information such as your credit card password.

Intrusion Detection is an essential mechanism to protect computer systems from attacks. It is becoming an increasingly important technology that monitors network traffic and identifies network intrusions such as anomalous network behaviors, unauthorized network access, and malicious attacks to computer systems. The IDS reports corresponding alarms and may take immediate action on the intrusions. In the simplest terms, intrusion detection systems consist of three functional components:

a. An information source that provides a stream of event records
b. An analysis engine that finds signs of intrusions
c. A response component that generates reactions based on the outcome of the analysis engine.

Our aim has been to develop a network intrusion system to detect DDOS attacks. For which integration of GA is done with NIDS. The whole work is divided into no. of sections. The next section discusses the similar works done by other researches and in the following section, problem formulation part is discussed. The system implementation and results are shown in the further sections.

II. RELATED WORK

This section briefly summarizes some of the previous work done on intrusion detection systems. In [10] the authors have implemented a simple genetic algorithm which evolves weights for the features of the data set. Then k-nearest neighbor classifier was used for the fitness function of the GA as well as to evaluate the performance of the new weighted feature set. The main aim of work was to rank features according to their importance. This helped in the detection of intrusions more efficiently. GA was able to identify the important features for each class. But the application of feature selection using a simple exclusion method based on the feature ranking proved to be unsuccessful so there is need of alternative methods for evolving a feature mask to perform feature selection as well as feature extraction.

In work done by Chittur [3], he has designed a genetic algorithm that promoted a high detection rate of malicious behavior and a low false positive rate of normal behavior classified as malicious. The genetic algorithm was given "training data" from which an empirical model of malicious computer behavior was generated. This model was then tested over previously unseen data to gauge its real-world performance. The results were good and the classification accuracy was 97.8%. The results are biased towards the training dataset (KDD 10% training set), because the last rules extracted from the training phase have been tested on the rest (90%) of the KDD dataset, i.e. testing and training the dataset are from the same distribution.

Wei Li, [4] proposed a GA-based method to detect anomalous network behaviors. Both quantitative and categorical features of network data are included when deriving classification rules using GA. The inclusion of quantitative features may lead to increased detection rates. However, no experimental results are provided in the paper.

III. PROBLEM FORMULATION

A number of soft computing based approaches have been proposed for detecting network intrusions. Soft computing

CONFERENCE PAPER
International Conference on Issues & Challenges in Networking, Intelligence & Computing Technologies
Organized by Krishna Institute of Engineering and Technology (KIET) Ghaziabad, India

refers to a group of techniques that exploit the tolerance for imprecision, uncertainty, partial truth, and approximation to achieve robustness and low solution cost. The principle constituents of soft computing are Fuzzy Logic (FL), Artificial Neural Networks (ANNs), Probabilistic Reasoning (PR), and Genetic Algorithms (GAs).

The use of genetic algorithms to detect malicious computer behavior is a novel approach to the computer network intrusion detection problem, as it is required for IDS to adapt itself with changing time. The initial population consists of the previously detected attacks. The chromosomes chosen consist of those fields of the attack patterns which are most likely to change in the new patterns.

The chromosome for the smurf attack consists of ten genes which are relevant fields that characterize this attack. The pattern of the smurf after heuristic search space reduction is as follows-

"icmpecr_i 1032 263 263 0.00 0.00 1.00 0.00 0.00"

If different pattern of smurf attack are studied then only five genes undergo changes in their values. So by the virtue of this we consider only these five fields for employing genetic algorithm. These are gene 3,4,5,7,8 (in the smurf case only).

The fitness function used in the GA is probability based. Each gene field (column) is given a weightage which is given by degree of variability of the column / tot number of distinct element and the individual weightage of each gene i.e. its frequency in the column.

New pattern are formed on the basis of selection criteria which uses fitness score of the chromosome to pass the pattern to the new generation, which then undergo mutation and crossover for further evolution.

GA is chosen because of some of its nice properties, e.g., robust to noise, no gradient information is required to find a global optimal or sub-optimal solution, self-learning capabilities, etc. As genetic algorithms work with populations of candidate solutions rather than a single solution and employ stochastic operators to guide the search process, they cope well with attribute interactions and avoid getting stuck in local maxima. In this way generalization and data fragmentation are avoided which are both inappropriate for dealing with rare classes.

A. Dataset:

Learning algorithms have a training phase where they mathematically 'learn' the patterns in the input dataset. The input dataset is also called the training set which should contain sufficient and representative instances of the patterns being discovered. A dataset instance is composed of features which describe all possible aspects of the phenomenon that is being observed. Learned patterns can be used to make predictions on a new dataset instance based on its diversity from normal patterns, its similarity to known attack patterns or a combination of both.

We have used NSL-KDD dataset for the training and testing purpose. It is modification of the KDD'99 dataset, but structure is similar to that of previous but with less redundant pattern.

NSL-KDD features can be classified into three groups:

a. Basic features: this category encapsulates all the attributes that can be extracted from a TCP/IP connection.
b. Traffic features: this category includes features that are computed with respect to a window interval and is divided into two groups:
   a) "Same host" features: examine only the connections in the past 2 seconds that have the same destination host as the current connection, and calculate statistics related to protocol behavior, service, etc.
   b) "Same service" features: examine only the connections in the past 2 seconds that have the same service as the current connection. The two aforementioned types of "traffic" features are called time-based.
c. Content features: To detect these kinds of attacks, we need some features to be able to look for suspicious behavior in the data portion, e.g., number of failed login attempts.

IV. SYSTEM IMPLEMENTATION

The core modules and functions that were used in the research work implementation are described as follows:

Pattern matching is essential for the inspection of network flows at the line speed. The intrusion detection system uses string matching to compare the payload of the network packet and/or flow against the pattern entries of intrusion detection rules. The Boyer Moore algorithm is the most well-known single pattern matching algorithm. The BM algorithm utilizes two heuristics, bad character and good suffix, to reduce the number of comparisons. Both heuristics are triggered on a mismatch. The BM algorithm takes the far most shift caused by the two heuristics.

We restricted our *Hypothesis search space* to the few most important fields that we could identify. Although, we can extend our search space to all 41 fields, that will however require the computation to go over many hours. Hence, for our experiment we confined our search space to only some important fields.We selected these fields based on a heuristic analysis of the training data to identify potential fields that seemed unique to a particular attack type.

For the smurf attack the decisive feature are protocol type, service, source bytes, Count, srv count, error rate, srv error rate, dst host same src port rate, dst host error rate, dst host srv error rate.For the Neptune attack the decisive feature are flag, error rate, srv error rate, same srv rate, diff srv rate, dst host srv Count, dst host same srv rate, dst host diff srv Rate, dst host error Rate, dst host srv error rate.

The ping of death attack and back attacks uses rules based on the heuristic analysis of the data to detect the attack from the testing dataset.

The rule for pod detection is: IF *(duration =0 and protocol=icmp and service=ecr_i || tim_iand flag=SF and (src_bytes=564 or src_bytes=1480) dst_bytes=0) Then (attack_name=" pod").*The rule for back attack is: IF *(protocol_type = tcp, service = http, logged_in = 1, dst_host_cnt = 255,dst_host_same = 0.00) Then attack name= "back".*

The overall design of the application consists of two modules – Scanning module and GA module. Scanning module is to detect the attacks, whereas GA module is to evolve new attack patterns.

A. Module 1:

The scanning module first takes the initial dataset, i.e. the network data and then takes the previously detected and the evolved patterns as input and then scans the network data for any intrusion. It uses pattern matching algorithm to detect the pattern. This scanning continues in an infinite loop which could only be broken when the network data if empty or the system administration shuts the application down.

B. Module 2:

Simple Genetic Algorithm (SGA) replaces all old rules with new produced rule preventing old good rules from participating in the next rule generation. Steady State GA (SSGA) is used to give a chance for previous rules from previous generation to participate in detecting intrusions in next generations.

As per GA principles, at the start of the experiment, we create an initial population that has individuals whose genes are selected at random from the range that they can vary in. Every population comprises of 100 individuals. We can vary the size of initial population as per our requirement. The initial population contains only the relevant field of particular attack type. It is generated from the training dataset. We have created separate datasets for each attack type. The initial population is a sequential selection of samples from these datasets. It contains only those genes of the patterns which have a tendency to change.

GA used in intrusion detection needs a fitness function, which decide whether a chromosome is right or not in a population Fitness are used to select a set of best-fit individuals from the population. The fitness criterion that is implemented is the probabilistic approach. The frequency of occurrence of each gene in the training dataset is studied.

The obtained frequency is used to calculate the probability of occurrence of this gene.

Mathematically: support= $F_i/\Sigma F_i$

Where "i" ranges from 1 to count of distinct element of that particular gene type and F_i represent the frequency of particular gene.

Now to calculate the weight of a particular column of the patterns, we use degree of variability in that column.

Mathematically: Confidence=degree of variability of the column/tot number of distinct element Fitness = support*confidence

Higher the value of fitness of a particular pattern, higher is the probability of pattern being probable attack.

The *selection* of individual that will proceed for creating the next generation population will be based on the fitness score of each individual. Top 100/200/50 (whatever the size of initial population) based on the fitness score would proceed to create next generation.

Crossover happens at any randomly selected position of the two chromosomes. This has been done to ensure that the resulting offspring's have values that maintain the range of allowable values for every field. The fit individuals selected based on their fitness score are made to undergo crossover to generate new rules or hypothesis to correctly classify attack connections that might not be there in the training data set. We have implemented only single point crossover.

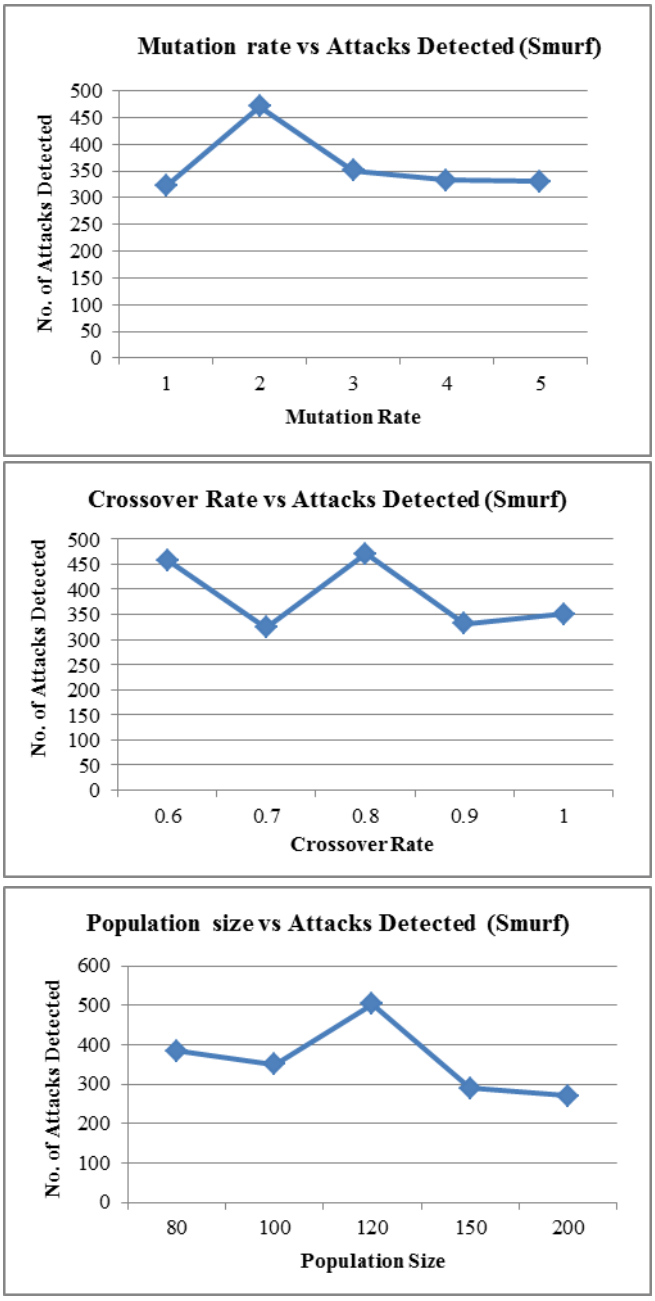

Figure 2.Smurf Attacks Detected v/s (a) Mutation Rate (b) Crossover Rate (c) Population Size

We also *mutate* the offspring's to generate new rules or hypothesis. We have used a single field mutation strategy wherein any one field in the offspring is mutated. The mutation rate has been set 1% i.e. out of 100 individuals in a population; only 1 individual will undergo mutation. It is kept in mind that it receives a different gene after mutation.

V. RESULTS

Using Genetic algorithm for implementation of Intrusion detection system we found that it successfully detects new attack pattern. The output gets changed by varying the crossover rate.

CONFERENCE PAPER
International Conference on Issues & Challenges in Networking, Intelligence & Computing Technologies
Organized by Krishna Institute of Engineering and Technology (KIET) Ghaziabad, India

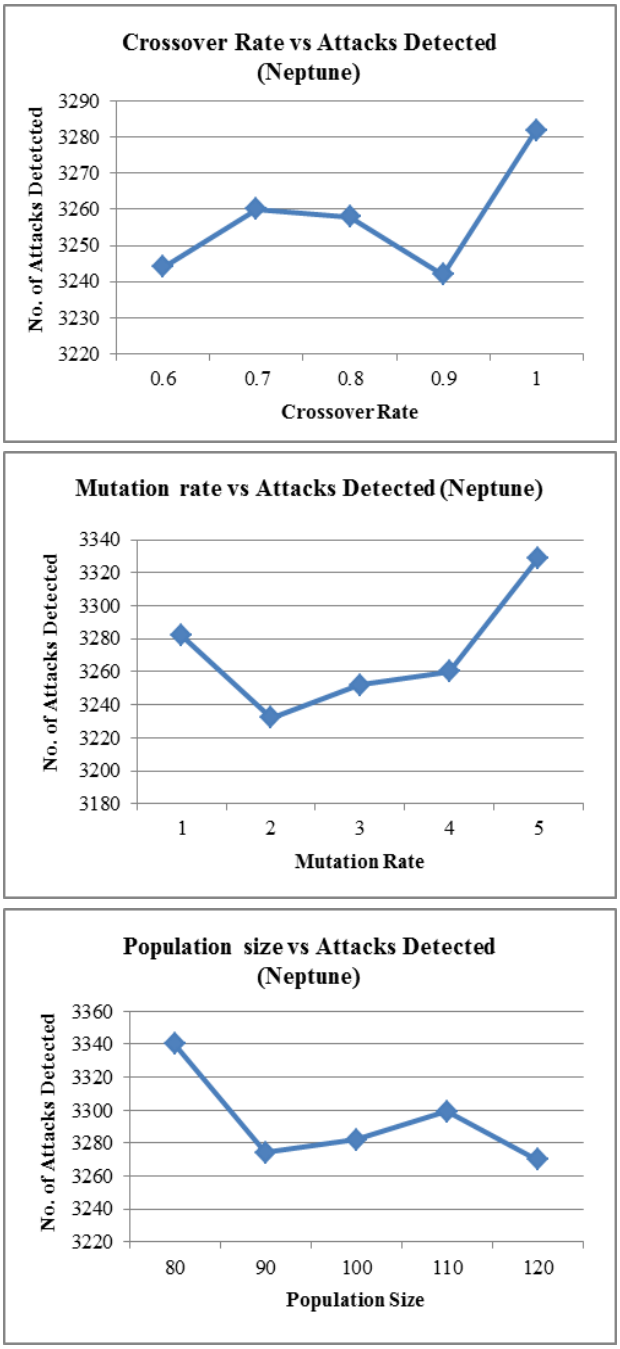

Fig 3.Neptune Attacks Detected v/s (a) Crossover Rate (b) Mutation Rate (c) Population Size

The attack detection rate was almost unchanged when we varied mutation rate. The best results that we obtained were smurf detection ~80%, Neptune detection ~73%, POD attack 100% and 100% detection rate for the back attack. The various graphs below describe the variation of attacks detected with change in the various genetic parameters. The Graphs shown in Fig 2 depicts the variation in the detection of smurf attacks on varying different parameters. Fig 3 shows the same for Neptune attack.

VI. CONCLUSION

The results of this experiment are satisfactory. This project can be very helpful in detecting Smurf, Neptune, Ping of Death and Back attacks on the network which are probably attacked by hackers. Increase in system (processor) capabilities reduces the execution time for the evolution process. Genetic algorithm gives satisfactory result but it can be improved further by using more efficient fitness methods. This research project can be extended for covering other network attacks.

VII. REFERENCES

[1] Z. Banković, D. Stepanović, S. Bojanić, O. NietoTaladriz, "Improving Network Security Using Genetic Algorithm Approach", Computers & Electrical Engineering, Vol.33, Issue 5-6, pp. 438-451.

[2] R. H. Gong, M. Zulkernine, P. Abolmaesumi, "A Software Implementation of a Genetic Algorithm Based Approach to Network Intrusion Detection", Proceedings of SNPD/SAWN`05, 2005.

[3] A. Chittur, "Model Generation for an Intrusion Detection System Using Genetic Algorithms"

[4] W. Li, "A Genetic Algorithm Approach to Network Intrusion Detection", SANS Institute, USA, 2004.

[5] Yong Wang, DawuGu, XiuxiaTian, Jing Li, "Genetic Algorithm Rule Definition for Denial of Services Network Intrusion Detection", 2009 International Conference on Computational Intelligence and Natural Computing.

[6] Wafa' S.Al-Sharafat and ReyadhNaoum, Development of Genetic-based Machine Learning for Network Intrusion Detection (GBML-NID), World academy of science, Engineering and technology 55 2009.

[7] ZoranaBanković, José M. Moya, Álvaro Araujo, Slobodan Bojanić and Octavio Nieto-Taladriz,"A Genetic Algorithm-based Solution for Intrusion Detection", Journal of Information Assurance and Security 4 (2009) 192-199.

[8] AnupGoyal, Chetan Kumar, "GA-NIDS: A Genetic Algorithm based Network Intrusion Detection System".

[9] H. GüneşKayacık, A. NurZincir-Heywood, Malcolm I. Heywood, "Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets".

[10] Middlemiss M.J. and Dick G.; "Weighted Feature Extraction Using a Genetic Algorithm for Intrusion Detection", Evolutionarycomputation,Vol. 3 pp. 1699 - 1675, (2003).

CONFERENCE PAPER
International Conference on Issues & Challenges in Networking, Intelligence & Computing Technologies
Organized by Krishna Institute of Engineering and Technology (KIET) Ghaziabad, India