IOSR Journal of Engineering
Mar. 2012, Vol. 2(3) pp: 468-472

Mitigating Denial-of-Service Attacks Using Genetic Approach

Mr. Anurag Andhare[1], Prof. Arvind Bhagat Patil[2]
[1](Department of Computer Technology, Yeshwantrao Chavan College of Engineering Nagpur) India
[2](Department of Computer Technology, Yeshwantrao Chavan College of Engineering Nagpur) India

ABSTRACT
Denial-of-service attacks are considered as very dangerous. These attacks have crashed many computers on internet recently. Detecting Denial-of-Service (DoS) attacks has been a difficult problem to solve. Security of computers from DoS attacks has become a crucial issue. Recognition of attacks is becoming a harder problem to crack in the field of Computer Network Security. In this paper idea for use of a Genetic Algorithm (GA) based approach, for generation of rules to identify DoS attacks on the system is used. A short general idea of Intrusion Detection System, genetic algorithm and related detection techniques is provided. The GA will be trained on the KDD Cup 99 data set to generate a rule set that can be used to identify attacks on the system. The algorithm takes into consideration different features in network connections of KDD Cup 99 dataset to generate a rule set.

Keywords: **Denial of Service attacks, Genetic Algorithm Intrusion Detection, KDD Cup 99Dataset, Rule set.**

I. INTRODUCTION

In recent years internet has grown at a remarkable rate in the terms of size and services provided. This has benefitted us remarkably but has exposed the computer systems to complex security threats. In spite of many technological innovations for computer security, it is almost impossible to have an entirely secured system. Hence it has become crucial to use an Intrusion Detection System (IDS) which monitors network traffic and identifies network intrusions such as anomalous network behaviors, unauthorized network access and malicious attacks to computer systems.
With the constant development of the intrusion technology of the network, the invasive behaviors are characterized with uncertainty, complexity, diversity and dynamic tendency etc. conventional security technologies such as static technologies like authentication systems, security routing and firewall etc played a certain role on the system to prevent the illegal intrusion, but the only defense is not enough from the perspective of safety management [15].
IDS is a tool that monitors events occurring in a computer system or network and analyzes them for signs of security threats [2]. Intruders can be divided into two groups, external and internal. The external intruders are those who do not have any authorization for accessing the system and who attack by using different attack techniques. The internal intruder refers to those who have access permissions and wish to perform unauthorized activities.
There are generally two categories of IDSs: misuse detection and anomaly detection. The misuse detection system performs the detection of intrusions through a matching with known patterns, and the anomaly detection system detects systems identify deviations from normal network behaviors and alert for potential unknown attacks [5].
Some IDS integrate both misuse and anomaly detection and form hybrid detection systems. The IDSs can also be classified into two categories depending on where they look for intrusions. A host-based IDS monitors activities associated with a particular host, and a network based IDS listens to network traffic.
In recent years, the research on intrusion detection is gradually inclined to artificial intelligence technology to improve the detection accuracy. Numbers of machine learning approaches are available for detecting network intrusions. Machine learning techniques for intrusion detection, are often used in conjunction with rule based expert systems [2] where knowledge is represented as a set of if-then rules.
In this paper, a Genetic Algorithm based approach is presented for network misuse detection. The GA is implemented and evaluated on the KDD Cup 99 dataset.
The paper is organized as follows. Section 2 focuses on the genetic algorithm and its application in intrusion detection and response systems. Section 3 describes the related work. Section 4 explains the KDD Cup 99 Dataset features. Experimental setup is explained in Section 5. Section 6 discusses results and section 7 concludes the paper.

GENETIC ALGORITHM
"A Genetic Algorithm (GA) is a programming technique that mimics biological evolution as a problem-solving strategy."[2] It is based on Darwin's principle of evolution and survival of fittest to optimize a population of candidate solutions towards a predefined fitness. [11][13] GA uses a chromosome-like data structure and evolves the chromosomes using selection, crossover, and mutation operators. The process usually begins with randomly generated population of chromosomes, which represent all possible solution of a problem that are considered candidate solutions. Chromosomes are divided into genes, encoded as bits, characters or numbers. A "Fitness function" is used to calculate the fitness of each chromosome according to the desired solution. During evaluation, two basic operators, crossover and mutation, are used to simulate the natural reproduction and mutation of species. The selection of chromosomes for survival and combination is biased towards the fittest chromosomes. [6][8][10][12][13]

Figure 1 shows the structure of simple Genetic Algorithm. The process starts from an initial population of randomly generated individuals. Then the population is

evolved for a number of generations while progressively improving the qualities of the individuals by increasing the fitness value as the measure of quality. During each generation selection, cross over, and mutation are one after the other applied to each individual with certain probabilities. First, the numbers of best-fit individuals are selected based on a user-defined fitness function. The remaining individuals are selected and paired with each other. Each individual pair produces one offspring by partially exchanging their genes around one or more randomly selected crossing points. At the end, a certain number of individuals are selected and the mutation operations are applied. Selection is the phase where population individuals with better fitness are selected, otherwise it gets damaged.

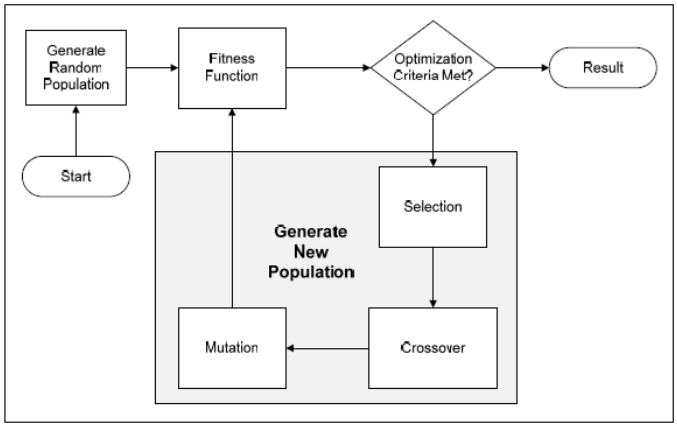

Fig. 1: Working of Genetic Algorithm [1]

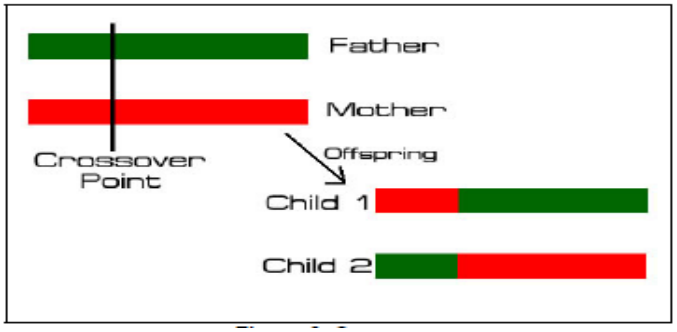

Fig. 2: Crossover

Crossover shown in Figure 2 is a process where each pair of individuals selects randomly participates in exchanging their parents with each other, until a total new population has been generated. Mutation as shown in Figure 3 flips some bits in an individual, and since all bits could be filled, there is low probability of predicting the change.

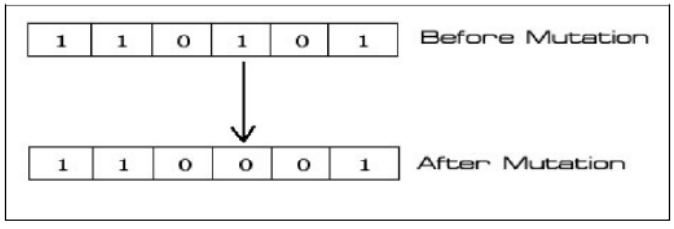

Fig.3: Mutation

II. RELATED WORK

Genetic Algorithms has been used for network intrusion detection in different ways. Some approaches directly use GAs to derive the classification rules [5], while some others use different AI methods for acquisition of rules, where GAs are used to select appropriate features or to determine the optimal parameters of some functions . The early effort of using GAs for intrusion detection can be dated back to 1995, when Crosbie et.al [3] applied the multiple agent technology and GP to detect network anomalies. Each agent monitors one parameter of the network audit data and GP is used to find the set of agents that collectively determine anomalous network behaviors. This method has the advantage of using many small autonomous agents, but the communication among them is still a problem. Also the training process can be time-consuming if the agents are not appropriately initialized. In most of the existing GA based IDSs, the quantitative features of network audit data are either ignored or simply treated, though such features are often involved in intrusion detection. This is because of the large cardinalities of quantitative features. Lu et al. [6] present an approach that uses GP to directly derive a set of classification rules from historical network data. The approach employs the support-confidence framework as the fitness function and is able to generally detect or precisely classify network intrusions. However, the use of GP makes implementation more difficult and more data or time is required to train the system. Li [5] propose 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 available yet. Xiao et al. [7] present an approach that uses information theory and GA to detect abnormal network behaviors. Based on the mutual information between network features and the types of network intrusions, a small number of network features are closely identified with network attacks. Then a linear structure rule is derived using the selected features and a GA. The use of mutual information reduces the complexity of GA, and the single resulting linear rule makes intrusion detection efficient in real-time environment. However, the approach considers only discrete features. Literature survey shows that the Intrusion detection models proposed for R2L attacks failed to demonstrate desirable performance with high detection and low false alarm rates using KDD data set. Typical and relevant features must be observed present in the KDD data set [8] that can help with the detection of these attacks.

III. TRANING AND TESTING DATASET

This idea uses the KDDCUP 99 data set to train and test the system classifier. The dataset has been provided by MIT Lincoln Labs. It contains a wide variety of intrusions simulated in a military network environment set up to acquire nine weeks of raw TCP/IP dump data for a local-area network (LAN) simulating a typical U.S. Air Force LAN. The LAN was operated as if it were a true Air Force environment, peppered with multiple attacks. Hence, this is a high confidence and high quality data set. [8]

They set up an environment to collect TCP/IP dump data from a host located on a simulated military network. Each

TCP/IP connection is described by 41 discrete and continuous features (e.g. duration, protocol type, flag, etc.) and labeled as either normal, or as an attack, with exactly one specific attack type as shown in Table 1.

Attacks fall into four main categories:

a) Denial of Service Attacks (DOS) in which an attacker overwhelms the victim host with a huge number of requests.

b) User to Root Attacks (U2R) in which an attacker or a hacker tries to get the access rights from a normal host in order, for instance, to gain the root access to the system.

c) Remote to User Attacks (R2L) in which the intruder tries to exploit the system vulnerabilities in order to control the remote machine through the network as a local user.

d) Probing in which an attacker attempts to gather useful information about machines and services available on the network in order to look for exploits.

The KDD cup 99 corrected dataset is 97.6M large and test data unlabeled dataset is 461M large. 65535 records are selected from the each dataset. For this idea, it is decided to use 10% of the training set which contains 494,021 connections. The testing set is the entire set of labeled connections consisting of around 4.9 million connections. Thus, entire data set could be used to test the system on unknown attacks. A connection is a sequence of TCP packets starting and ending at some well defined times, between which data flows to and from a source IP address to a target IP address under some well defined protocol. Each connection is labeled as either normal, or as an attack, with exactly one specific attack type.

Table 1:Attack types from KDD Cup 99 Dataset

DOS	*R2L*	*U2R*	*Probing*
pod	ftp_write	buffer overflow	ipsweep
teardrop	guess_password	load module	nmap
land	imap	perl	portsweep
smurf	multi hop	rootkit	santan
back	phf		
neptune	spy		

APPROACH USED FOR EXPERIMENT

By using GA on KDD Dataset we are intended to develop the rule set which will be integrated with network sniffer to detect the Denial of Service attacks and then such attacking connection will be terminated. The schematic of the idea used for experiment is as shown in figure 4.

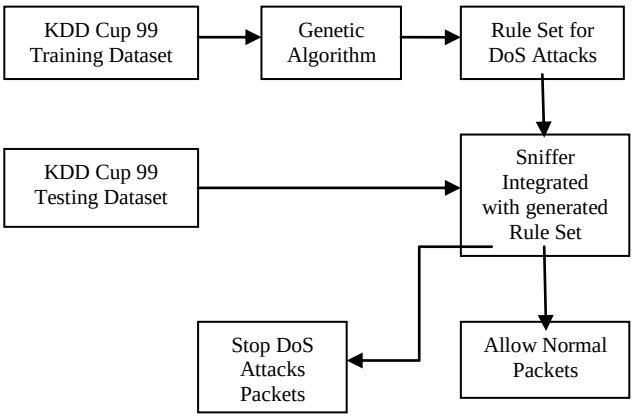

Fig. 4: Proposed System

Use and implementation of GA is discussed below.

A. Data Encoding

Every network connection in KDD dataset has 41 features. Out of these 41 only those which are having high possibilities to be involved in intrusions has to be selected. Some features have symbolic form (e.g. protocol). These features will be converted into numerical ones by assigning a unique number for each feature. The resulting map will be used to do the same for the testing data set. Chromosomes will be encoded using binary encoding.

Following features shown in Table 2 will be selected for detection of Denial of Service attacks.

Table 2: Attack features selected

Attribute	*Description*	*Data Type*	*Gen Numbers*
duration	length of connection	numeric	1
protocol	tcp, udp,icmp	symbolic	1
service	http,ftp,smtp	symbolic	1
flag	normal or error status	numeric	1
src_bytes	bytes from source	numeric	1
dst_bytes	bytes from dstination	numeric	1
attack type	pod, smurf attack	symbolic	1

B. Rules for IDS

By analyzing the dataset, rules will be generated in the rule set. These rules will be in the form of an 'if then' format as follows [1].

if {condition} then {act}

The condition using this format refers to the attributes in the rule set that forms a network connection in the dataset. The condition will result in a 'true' or 'false'. The attack name will be specified only if the condition is true.

KDD dataset has connections labeled with various attacks, as stated in previous section. For example, a Denial of Service (DoS) attack is an attack in which the attacker makes system resources so busy that they remain unavailable for legitimate requests from the users. In the *smurf* attack, attacker use "ICMP" echo request packets directed to IP

broadcast addresses from remote locations to create a DOS attack. These are three parties in these attacks; the attacker sends ICMP echo request packets of the broadcast address of many subnets with the source address spoofed to be that of the intended victim. Any machines that are listening on these subnets, will respond by sending ICMP "echo reply" packets to the victim. Following type of rule can be generated using KDD Cup 99 dataset to detect "smurf" attack.

if *(duration="0:0:1" and protocol="finger" and source_port=18982 and destination_port=79 and source_ip="9.9.9.9" and destination_ip="172.16.112.50")*
then *(attack_name="smurf")*

The above rule expresses that if a network packet is originated from IP address 9.9.9.9 and port 18982, and sent to IP address 172.16.112.50 and port 79 using the protocol finger, and the connection duration is 1second, then most likely it is a network attack of type smurf that may eventually cause the destination host out of service.

C. Fitness Function

To determine the fitness of a rule, the support-confidence framework [9] will be used. If a rule is represented as *if A then B*, then the fitness of the rule will be determined using following equations:

$$support = |A \text{ and } B| / N$$
$$confidence = |A \text{ and } B| / |A|$$
$$fitness = X1 * support + X2 * confidence$$

Here, N is the total number of network connections in the audit data, |A| stands for the number of network connections matching the condition A, and |A and B| is the number of network connections that matches the rule if A then B. The weights X1 and X2 are used as a threshold.

IV. EXPERIMENTAL SETUP

As mentioned before, the scope of our experiment was focused to generate classifiers or rules for six attack types belonging to two different classes. The training set contains maximum connections of the smurf attack type, 280,790 to be precise. The number of other connections are: 107201 neptune connections, 21 land connections, 15 satan connections, 30 ipsweep connections and 15 portsweep connections.

The implementation of genetic algorithm was done by using GLIB Library. Ubuntu operating system based computer with a Core i3 processor, 500GB of hard disk space and 4 GB of RAM was used to execute the computer program.

V. EXPERIMENTAL RESULTS

From the above experiment, rules for detection on DoS attacks were created that could successfully classify all of the 280,790 smurf type of attack connections. It also classified 410 normal connections as attack. The false positive rate is thus around 0.08%. In the entire testing data set, the smurf rule set correctly classified 2,807,880 connections, and with a false positive of 0.17%. The rule set that classifies Probe attacks was able to correctly classify 52 Probe attack connections in the training data set, out of a total of 60 probe type of attacks. The rule set for Probes on the entire test data showed results as follows: total number of probe attacks = 38,786, total classified correctly = 35,829. The percentage accuracy = 92.3%.

VI. CONCLUSION

The results of experiment are very encouraging. All types of smurf attack labels in the training data set were classified using generated rules. False positive rate is also quite low at 0.2% and accuracy rate is as high as 100%. This approach will be very useful for the attack detection in today's changing attack methodologies. If the rules are updated dynamically with the firewall's log data, then this method will be very effective against new attacks.

REFERENCES

[1] Gong RH, Zulkernine M, Abolmaesumi P. "A Software Implementation of a Genetic Algorithm based approach to Network Intrusion Detection". In: Proceedings of the sixth international conference on software engineering, artificial intelligence, networking and parallel/distributed computing and first ACIS international workshop on self-assembling wireless networks (SNPD/SAWN'05), 2005.

[2] Chittur A. "Model Generation for an Intrusion Detection System Using Genetic Algorithms, publications/gaids-thesis01.pdf, accessed in 2006.

[3] Crosbie, M., and Spafford, G. Applying genetic programming to intrusion detection. In Proc.1995 AAAI Symposium on Genetic Programming, pp. 1-8.

[4] D. Dasgupta and F. A. Gonzalez, "An Intelligent Decision Support System for Intrusion Detection and Response", MMM-ACNS, Lecture Notes in Computer Science, vol. 2052, pp. 1-14, 2001.

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

[6] Lu W and Traore I, "Detecting New Forms of Network Intrusion using Genetic Programming", comp utational Intelligence, Vol.20, pp.3, Blackwell Publishing, 2004.

[7] T. Xiao, G. Qu, S. Hariri, and M. Yousif, "An Efficient Network Intrusion Detection Method Based on Information Theory and Genetic Algorithm", Proceedings of the 24th IEEE International Performance Computing and Communications Conference (IPCCC '05), Phoenix, AZ, USA. 2005.

[8] KDD-CUP 1999 Data, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[9] Middlemiss M and Dick G, "Feature Selection of Intrusion detection data using a hybrid genetic of hybrid Intelligent systems, IOS Press Amsterdam, PP.519-527, 2003.

[10] Helmer, G., J. Wong, V. Honavar, and L. Miller, "Automated Discovery of Concise Predictive Rules for Intrusion Detection.", Recursions Software Inc. Ames, IA: Department of Computer Science Iowa State University Ames, IA,2002.

[11] Sinclair,C.,L.Pierce, S. Matzner,"An Application of Machine Learning to Network Intrusion Detection", Proceedings of the 15th Annual Computer Security Applications Conference, December 1999, page 371, Phoenix, AZ.

[12] Grefenstetle J, "Optimization of control parameters for genetic algorithm," IEEE Trans on Syst, Man and Cybern, vol. 16, no. 1, pp. 122-128, 1986.

[13] Juan M E T, Pedro G T, Jesus E D V, et al, "Anomaly detection methods in wired networks", A survey and taxonomy, Computer Communications, vol. 27, no.16, pp. 1569-1584, 2004.

[14] Theuns V, Ray H, "Intrusion detection techniques and approaches", Computer Communications, vol. 25, no. 15, pp. 1356-1584, 2002.

[15] Wang Guojun, Yue Zhiqiang, "Application Research of Support Vector Machine in the Intrusion Detection", GUANGXI JOURNAL OF LIGHT INDUSTRY, no. 7, pp. 51-52, 2008.