International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622
International Conference on Industrial Automation and Computing (ICIAC) (12th&13th April 2014)

RESEARCH ARTICLE OPEN ACCESS

Intrusion Detection System Using Evolutionary Approach of Genetic Algorithm

Ms. Pooja B. Aher 1, Prof. Kapil N. Hande2
1(Department of Computer Science & Engineering, RTMNU University, INDIA Email: poojabaher1786@gmail.com)
2(Department of Computer Science & Engineering, RTMNU University, INDIA Email: kapilhande@gmail.com)

ABSTRACT
Day by day, the importance of data and information is increasing in network area. Same time, the threat to the data security is also increasing rapidly. Intrusion Detection System (IDS) is used to prevent the data from the threats. In this paper , IDS using Genetic Algorithm (GA) is proposed. With the help of evolution process of Genetic Algorithm the malicious packet on the network can be detected and IDS helps to block the respective IP addresses. Genetic algorithm is an evolutionary algorithm which is helpful for optimization purpose. An Intrusion Detection System is a system for detecting intrusions and reporting them accurately to the proper authority. GA operators are applied to the input data to find malicious IP addresses from the list given by the system. These methods have many significant advantages in case of huge data.
Keywords - Evolutionary algorithm; Genetic Algorithm(GA), IDS, Intrusion, IP address;

I. Introduction

Intrusion Detection Systems are usually specific to the operating system and used to provide security, handle intrusions, and recover from damage caused by security breaches. Existing IDS systems can be divided into two categories according to the detection approaches: anomaly detection and detection is an approach to detect intrusions, misuse detection or signature detection. Anomaly is done by first learning the characteristics of normal activity of users. Then the system uses such characteristics to judge whether the user's activity is normal or not. Misuse detection systems are the approach that tries to match user activity to stored signatures of known exploits or attacks. Such detection system uses a previously defined knowledge to check whether the new activity is in that knowledge database. If yes, the IDS considers this activity as a possible attack and then blocks it.

Genetic Algorithm (GA) has been used in different ways in IDSs. Genetic algorithm is an evolutionary algorithm used for search & optimization. Its related behavior can be translated to represent a rule to judge whether or not a real-time connection is considered an intrusion. These rules can be modeled as chromosomes inside the population. The population evolves until the evaluation criteria are met. The generated rule set can be used as knowledge inside the IDS for judging whether the network connection and related behaviors are potential intrusions.

Distributed Denial-of-Service (DDoS) [3, 5] attack is a most important threat to security. In which the victim network element(s) are bombarded with high volume of fictitious attacking packets originated from a large number of Zombies. The aim of the attack is to overload the victim and render it incapable of performing normal transactions. To protect network servers, network routers [4, 8] and client hosts from becoming the handlers, Zombies and victims of distributed denial-of-service (DDoS) attacks. Genetic algorithm approach can be adopted as a sure shot weapon to these attacks. The central theme of this paper is to explore parameters and evolution process[6] of Genetic Algorithm which helps to detect malicious packet on the network and ultimately helps to block the respective IP addresses.

II. Literature Survey

The central theme of this system is to explore parameters and evolution process [2, 3, 13] of Genetic Algorithm, which helps to detect malicious packet on the network and ultimately helps to block the respective IP addresses. There are several approaches for solving intrusion detection problems as sizes on the network.

Genetic algorithm into network intrusion detection techniques has described by Wei Li [6]. This implementation of genetic algorithm is unique as it

considers both temporal and spatial information of DARPA data set Rule Set Rule Base Network Sniffer GA network connections during the encoding of the problem; therefore, it should be more helpful for identification of network anomalous behaviours.

The Intrusion detection system for detecting DoS, R2L, U2R presented by B. Uppalalaiah, T. Bharat et al. [9], Probe from DD99CUP data set. Genetic Algorithm detects the intrusion, while correlation techniques identify the features of the network connections .The results shows that we have specified set of rules and high Dos, R2L, U2R, Probe attack detect rate. In Optimizing the parameters present in the algorithm reduces the training time.

Srinivasa K G, SaumyaChandra et al.[10] presents IGIDS, where the genetic algorithm is used for selecting best individual from the database. This makes IDS faster and intelligent.

Anup Goyal and Chetan Kumar [11] has presented a different features in network connection such as type of protocol, status of the connection & network service on destination. Each rule in rule set identifies a particular attack type. For this , GA is implemented and trained it on the KDD Cup 99 data set to generate a rule set that can be applied to the IDS to identify and classify different types of attack connections.

Shaik Akbar et al. [13] presents an algorithm which identifies damaging type connections called Genetic Algorithm. The algorithm considers different features by protocol type, duration, src_bytes , & is trained on the KDDCUP99 Data Set in order to generate a collection of rules which different types of attacks.

Yonghui Shi, Jun Bao, Zhongzhen Yan, and Shengping Jiang present a new detection method. Based on GA-Chaos optimization and RBF neural network is proposed. The GA-Chaos was firstly used to optimize the structure of the RBF as well as its weight values to obtain high learning and generalization ability of the RBF detected model. Then the RBF model was working to prepare and check the intrusion data sets.

III. Genetic Algorithm

3.1 Introduction

GA plays a important role to implement IDS and interface of any Rule base system and firewall system. There are three operators of genetic algorithm selection, crossover and mutation. In a genetic algorithm, chromosomes are used to encode and displace solutions. The chromosomes are represented binary or in any real numbers. In GA, these chromosomes are used to generate the fitness function. With the help of these fitness functions, the best chromosomes are selected. These chromosomes are then used in the next iteration of the algorithm. Generally, the algorithm terminates when a acceptable fitness level has been reached. Fig1. shows the basic structure of Genetic algorithm.

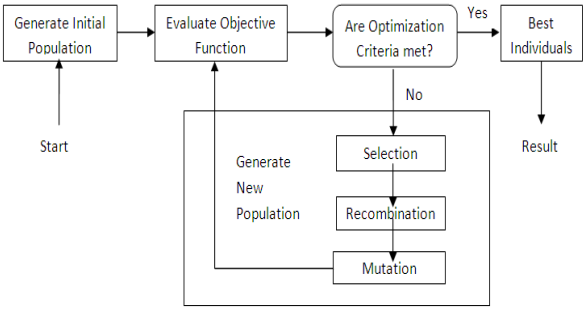


Fig 3.1 Structure of a Genetic Algorithm

3.2 Genetic Algorithm Process

GA evolves the population of chromosomes as the process of natural selection.[12] By using the operators of GA, new chromosome are processed. GA process uses a set of genetic operators such as selection operator, crossover operator and mutation operator, with the help of this it evaluate chromosome using the fitness function. GA selects those chromosomes whose fitness value are best . Chromosomes that are most fit, likely to survive. GA terminate the process by attainment of an acceptable fitness level, or if there are no improvements in the population for some fixed generations, or for any other reason. The standard GA processes is shown in fig 3.1. It contains various steps as shown in fig 3.2. The process will stop when it get the best individuals.

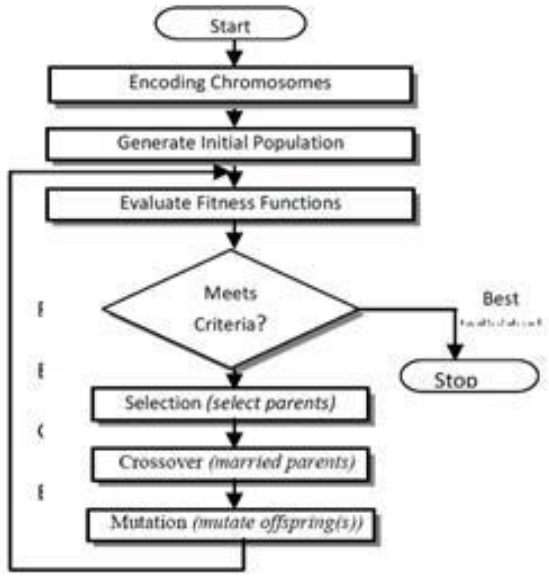


Fig 3.2. Genetic Algorithm Process

3.3 Genetic Algorithm Operator

- Encoding of the Chromosomes: In the GA process, the input is given in the form of encoded that is in binary as well as in real value form.
- Applying fitness function: It measures the performance of all chromosomes in the population by applying fitness function.
- Selection operator: This operator selects the chromosomes whose fitness value is best for recombination. The selected chromosomes are called parents. Such selection methods are: fitness- proportion selection, roulette-wheel selection, stochastic universal sampling, local selection and rank selection.
- Crossover operator: The recombination of chromosomes are done by one of the crossover methods. It produces one or more new chromosome(s) called offspring(s). Such methods are: Single Point Crossover, Multipoint Crossover, Uniform Crossover and Arithmetic Crossover.
- Mutation operator: New genetic material could be introduced into the new population through mutation process. [12] This will increase the diversity in the population. For each offspring mutation randomly alters some gene(s).Some encoding schemas: binary encoding and real- number encoding.

IV. System Overview

The planned system overview is shown in fig 4.1. which starts from capturing firewall data entry and then initial filtering is done on the basis of rule defined by the system. This précised data is then input to the GA based algorithm which generates the best individuals.

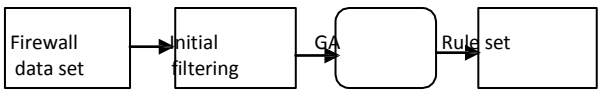


Figure 4.1: architecture of Genetic Algorithm

The detail proposed architecture is shown in fig 4. It starts from initial population from pfirewall.log file generated by the firewall system. The packets are the filtered out on the basis of rules defined by system. Then the précised data packets go through several steps that is through the operators of GA. These processes gets generate best individuals. The generated individuals are the verified by the fitness function to generate the population for next generation.

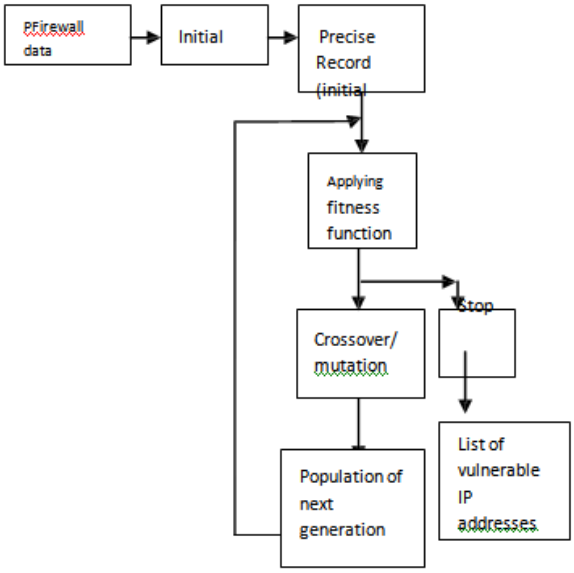


Fig 4.2 Detailed system architecture for GA.

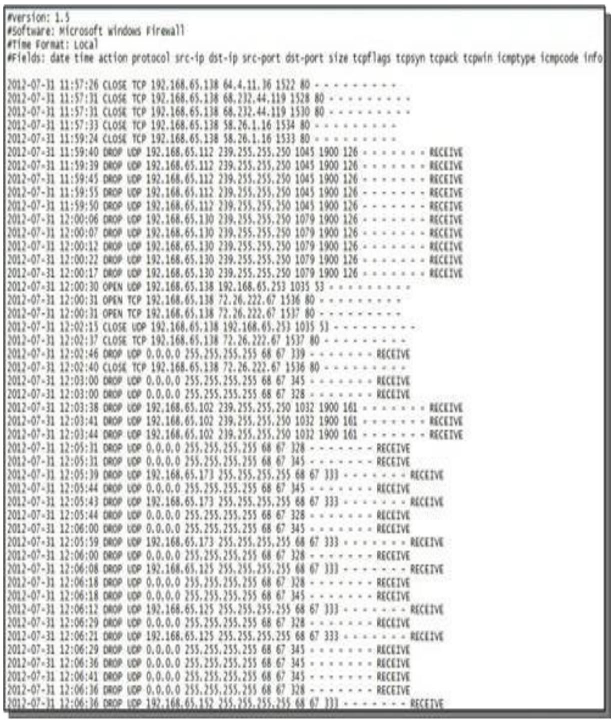
V. Experimental Setup

The scope of this paper is focused to generate list of IP addresses and there packets which are susceptible to the server or destined system. The testing is done on the entries generated by the firewall system of machine in pfirewall.log file. The pfirewall.log file contains the entries of incoming packets with various fields like date/time, action, protocol, srcip, destip, srcport, destport, size, flag, ack, type and info. But for making the connection profile we have used only 5 important fields of it. These are src-ip, dst-ip, src-port, dst-port and size.

For this experiment result GA operator along with evolution process are implemented using java. The training data is stored into the wamp server which is used as the backend to the system.

VI. Result

From the above experiment, it is able to create a rule base that could successfully categories harmful and harmless connection types. We have shown the resultant figures below by applying 100 connection entries respectively to the proposed system. After that we were able to get around 96% of accuracy to classify the connections types.



```
#Version: 1.5
#Software: Microsoft Windows Firewall
#Time Format: Local
#Fields: date time action protocol src-ip dst-ip src-port dst-port size tcpflags tcpsyn tcpack tcpwin icmptype icmpcode info

2012-07-31 11:57:26 CLOSE TCP 192.168.65.138 64.4.11.36 1522 80 - - - - - - - - - -
2012-07-31 11:57:31 CLOSE TCP 192.168.65.138 68.232.44.119 1528 80 - - - - - - - - - -
2012-07-31 11:57:31 CLOSE TCP 192.168.65.138 68.232.44.119 1530 80 - - - - - - - - - -
2012-07-31 11:57:33 CLOSE TCP 192.168.65.138 58.26.1.16 1534 80 - - - - - - - - - -
2012-07-31 11:59:24 CLOSE TCP 192.168.65.138 58.26.1.16 1533 80 - - - - - - - - - -
2012-07-31 11:59:40 DROP UDP 192.168.65.112 239.255.255.250 1045 1900 126 - - - - - - - RECEIVE
2012-07-31 11:59:39 DROP UDP 192.168.65.112 239.255.255.250 1045 1900 126 - - - - - - - RECEIVE
2012-07-31 11:59:45 DROP UDP 192.168.65.112 239.255.255.250 1045 1900 126 - - - - - - - RECEIVE
2012-07-31 11:59:55 DROP UDP 192.168.65.112 239.255.255.250 1045 1900 126 - - - - - - - RECEIVE
2012-07-31 11:59:50 DROP UDP 192.168.65.112 239.255.255.250 1045 1900 126 - - - - - - - RECEIVE
2012-07-31 12:00:06 DROP UDP 192.168.65.130 239.255.255.250 1079 1900 126 - - - - - - - RECEIVE
2012-07-31 12:00:07 DROP UDP 192.168.65.130 239.255.255.250 1079 1900 126 - - - - - - - RECEIVE
2012-07-31 12:00:12 DROP UDP 192.168.65.130 239.255.255.250 1079 1900 126 - - - - - - - RECEIVE
2012-07-31 12:00:22 DROP UDP 192.168.65.130 239.255.255.250 1079 1900 126 - - - - - - - RECEIVE
2012-07-31 12:00:17 DROP UDP 192.168.65.130 239.255.255.250 1079 1900 126 - - - - - - - RECEIVE
2012-07-31 12:00:30 OPEN UDP 192.168.65.138 192.168.65.253 1035 53 - - - - - - - - - -
2012-07-31 12:00:31 OPEN TCP 192.168.65.138 72.26.222.67 1536 80 - - - - - - - - - -
2012-07-31 12:00:31 OPEN TCP 192.168.65.138 72.26.222.67 1537 80 - - - - - - - - - -
2012-07-31 12:02:15 CLOSE UDP 192.168.65.138 192.168.65.253 1035 53 - - - - - - - - - -
2012-07-31 12:02:37 CLOSE TCP 192.168.65.138 72.26.222.67 1537 80 - - - - - - - - - -
2012-07-31 12:02:46 DROP UDP 0.0.0.0 255.255.255.255 68 67 339 - - - - - - - RECEIVE
2012-07-31 12:02:40 CLOSE TCP 192.168.65.138 72.26.222.67 1536 80 - - - - - - - - - -
2012-07-31 12:03:00 DROP UDP 0.0.0.0 255.255.255.255 68 67 345 - - - - - - - RECEIVE
2012-07-31 12:03:00 DROP UDP 0.0.0.0 255.255.255.255 68 67 328 - - - - - - - RECEIVE
2012-07-31 12:03:38 DROP UDP 192.168.65.102 239.255.255.250 1032 1900 161 - - - - - - - RECEIVE
2012-07-31 12:03:41 DROP UDP 192.168.65.102 239.255.255.250 1032 1900 161 - - - - - - - RECEIVE
2012-07-31 12:03:44 DROP UDP 192.168.65.102 239.255.255.250 1032 1900 161 - - - - - - - RECEIVE
2012-07-31 12:05:31 DROP UDP 0.0.0.0 255.255.255.255 68 67 328 - - - - - - - RECEIVE
2012-07-31 12:05:31 DROP UDP 0.0.0.0 255.255.255.255 68 67 345 - - - - - - - RECEIVE
2012-07-31 12:05:39 DROP UDP 192.168.65.173 255.255.255.255 68 67 333 - - - - - - - RECEIVE
2012-07-31 12:05:44 DROP UDP 0.0.0.0 255.255.255.255 68 67 345 - - - - - - - RECEIVE
2012-07-31 12:05:43 DROP UDP 192.168.65.173 255.255.255.255 68 67 333 - - - - - - - RECEIVE
2012-07-31 12:05:44 DROP UDP 0.0.0.0 255.255.255.255 68 67 328 - - - - - - - RECEIVE
2012-07-31 12:06:00 DROP UDP 0.0.0.0 255.255.255.255 68 67 345 - - - - - - - RECEIVE
2012-07-31 12:05:59 DROP UDP 192.168.65.173 255.255.255.255 68 67 333 - - - - - - - RECEIVE
2012-07-31 12:06:00 DROP UDP 0.0.0.0 255.255.255.255 68 67 328 - - - - - - - RECEIVE
2012-07-31 12:06:08 DROP UDP 192.168.65.125 255.255.255.255 68 67 333 - - - - - - - RECEIVE
2012-07-31 12:06:18 DROP UDP 0.0.0.0 255.255.255.255 68 67 328 - - - - - - - RECEIVE
2012-07-31 12:06:18 DROP UDP 0.0.0.0 255.255.255.255 68 67 345 - - - - - - - RECEIVE
2012-07-31 12:06:12 DROP UDP 192.168.65.125 255.255.255.255 68 67 333 - - - - - - - RECEIVE
2012-07-31 12:06:29 DROP UDP 0.0.0.0 255.255.255.255 68 67 328 - - - - - - - RECEIVE
2012-07-31 12:06:21 DROP UDP 192.168.65.125 255.255.255.255 68 67 333 - - - - - - - RECEIVE
2012-07-31 12:06:29 DROP UDP 0.0.0.0 255.255.255.255 68 67 345 - - - - - - - RECEIVE
2012-07-31 12:06:36 DROP UDP 0.0.0.0 255.255.255.255 68 67 345 - - - - - - - RECEIVE
2012-07-31 12:06:41 DROP UDP 0.0.0.0 255.255.255.255 68 67 345 - - - - - - - RECEIVE
2012-07-31 12:06:36 DROP UDP 0.0.0.0 255.255.255.255 68 67 328 - - - - - - - RECEIVE
2012-07-31 12:06:36 DROP UDP 192.168.65.152 255.255.255.255 68 67 333 - - - - - - - RECEIVE
```

Fig 6.1. pfirewall.log file captured by firewall system

SRC-IP	DST-IP	SRC-PORT	DST-PORT	SIZE
192.168.65.138	64.4.11.36	1520	80	136
192.168.65.138	68.232.44.119	1520	80	153
192.168.65.138	68.232.44.119	1035	53	135
192.168.65.138	58.26.1.16	1537	80	136
192.168.65.138	58.26.1.16	1035	53	126
192.168.65.138	239.255.255.250	68	67	53
192.168.65.138	239.255.255.250	1032	1900	55
192.168.65.138	239.255.255.250	68	67	128
192.168.65.138	192.168.65.253	1079	1900	54

Fig 6.2. Highlighted entries are eliminated by the rule base

```
IP :- 127.0.0.1
127.0.0.1
127000000001
4
Intrusion detected from IP:239.255.255.250 from port:1900
size * weight --- 126.0
Intrusion detected from IP:239.255.255.250 from port:1900
size * weight --- 126.0
Intrusion detected from IP:239.255.255.250 from port:1900
size * weight --- 126.0
Intrusion detected from IP:239.255.255.250 from port:1900
size * weight --- 126.0
Intrusion detected from IP:239.255.255.250 from port:1900
size * weight --- 126.0
Intrusion detected from IP:239.255.255.250 from port:1900
size * weight --- 126.0
Intrusion detected from IP:239.255.255.250 from port:1900
size * weight --- 126.0
Intrusion detected from IP:239.255.255.250 from port:1900
size * weight --- 126.0
Intrusion detected from IP:239.255.255.250 from port:1900
size * weight --- 126.0
Intrusion detected from IP:239.255.255.250 from port:1900
size * weight --- 126.0
Intrusion detected from IP:255.255.255.255 from port:67
size * weight --- 339.0
Intrusion detected from IP:255.255.255.255 from port:67
size * weight --- 345.0
Intrusion detected from IP:255.255.255.255 from port:67
size * weight --- 328.0
Intrusion detected from IP:239.255.255.250 from port:1900
size * weight --- 161.0
Intrusion detected from IP:239.255.255.250 from port:1900
size * weight --- 161.0
Intrusion detected from IP:239.255.255.250 from port:1900
size * weight --- 161.0
Intrusion detected from IP:255.255.255.255 from port:67
size * weight --- 328.0
Intrusion detected from IP:255.255.255.255 from port:67
size * weight --- 345.0
Intrusion detected from IP:255.255.255.255 from port:67
size * weight --- 333.0
Intrusion detected from IP:255.255.255.255 from port:67
size * weight --- 345.0
Intrusion detected from IP:255.255.255.255 from port:67
size * weight --- 333.0
Intrusion detected from IP:255.255.255.255 from port:67
size * weight --- 328.0
```

Fig 6.3.: Final list of IP addresses generated by GA

Result

From the above experiment, it is able to create a rule base that could successfully categories harmful and harmless connection types. It evaluates the 100 connection entries of pfirewall.log file & these are given as an input to the respective to the proposed system.

Conclusion

As per the result generated; the system can be integrated with any of the IDS system to improve the efficiency and the performance. The system can also be able to integrate to the input to the firewall system. In this paper, the GA based processes as well as evolution operators along with the overall implementation of GA into proposed system are discussed.

REFERENCES

[1] T. Lunt, A. Tamaru, F. Gilham, R. Jagannathan, P. Neumann, H. Javitz, A. Valdes, and T. Garvey. "A real-time intrusion detection expert system (IDES)" - final technical report. Technical report, Computer Science Laboratory, SRI International, Menlo Park, California, February 1992.

[2] A. Chittur, "Model Generation for an Intrusion Detection System Using Genetic Algorithms", Ossining High School, Ossining NY, 2001.

[3] Anup Goyal, Chetan Kumar, "GA-NIDS : A genetic algorithm based network intrusion detection system", IJIRCCE, Vol 1, issue 7, ISSN:2320-9801, Sept 2013 .

[4] K. Ilgun, R. A. Kemmerer, and P. A. Porras. "State transition analysis: A rulebased intrusion detection approach". IEEE Transactions on Software Engineering, 21(3):181–199, March 1995

[5] John E. Dickerson, and Julie A. Dickerson "Fuzzy Network Profiling for Intrusion Detection" Electrical and Computer Engineering Department Iowa State University Ames, Iowa, 50011.

[6] Rui Zhong, and Guangxue Yue "DDoS Detection System Based on Data Mining" ISBN 978-952-5726-09-1 (Print) Proceedings of the Second International Symposium on Networking and Network Security (ISNNS '10)Jinggangshan, P. R. China, 2-4,April.2010,pp.062-065.

[7] Wei Li "Using Genetic Algorithm for network intrusion detection"

[8] B.Upalhaiah, K. Anand, B. Narsimha, S. Swaraj, T. Bharat, "Genetic Algorithm Approach to Intrusion Detection System" ISSN: 0976-8491 (online) | ISSN : 2229-4333 (print), IJCST VOL3, ISSUE 1, JAN-MARCH 2012.

[9] Shrinivasa K G, Saumya chandra, Sidharth Kajaria, Shilpita mukharjee, "IGIDS: Intelligent intrusion detection system using Genetic Algorithm", 978-1-4673-0126-8/11/2011 IEEE.

[10] Anup Goyal, Chetan Kumar, "GA-NIDS : A genetic algorithm based network intrusion detection system",

[11] Atul Kamble, "Incremental Clustering in data mining using genetic algorithm", IJCTE, Vol 2, No. 3, June, 2010

[12] Shaik Akbar, Dr. J. A. chandulal, Dr. K. Nageswara Rao, G. Sudheer Kumar, "troubleshooting technique for intrusion detection sytem using genetic algorithm", IJWBC, vol 1(3), december 2011

[13] Suhail Owais, Vaclav Snasel, Pavel Kromer, Ajith abrahim,"Survey: Using genetic algorithm approach in intrusion detection system techniques", 7th computer information system and industrial management applications,2008 IEEE

[14] Levitt, K., "GrIDS-A Graph-Based Intrusion Detection System for Large Networks", Technical Report. University of California Davis, 4 March 1996.