IJCAT - International Journal of Computing and Technology, Volume 3, Issue 8, August 2016
ISSN : 2348 - 6090
www.IJCAT.org

A Novel Algorithm for Real Time Intrusion Detection Technique Using Genetic Algorithm

[1] Sheetal Verma, [2] Pravin Malviya

[1,2] SBITM,
Baitul, India

Abstract - Now a day's growth of local area networks as well as internet gives a more convenient and better business oriented technology for the users. Even though the emerging internet technology is more valuable for the users of the computer systems the critical data security threads are also increasing at a very high rate. Various firms are utilizing different protection technologies to protect their system from the intruder attacks by using antivirus application, firewall, and password protection. There are various techniques and areas which plays really important role in building more secured applications. In this paper we provide one of the most powerful technique i.e evolutionary algorithms (Genetic Algorithm) for Intrusion Detection System. It also gives a brief idea regarding proposed parameters and evolution process genetic algorithm and how to implement it in real time system.

Keywords **- DDOS, Evolutionary Algorithm, GA-RIDS, Genetic Algorthm, Intrusion, IDS, Threats.**

1. Introduction

Generally Intrusion Detection Systems are specific to the operating system in which they operate in as well as now a day it is an important tool for every organization's information as security policy in any organization handle intrusions, and recover from damage caused by malicious attack. Existing IDS systems is generally divided into two major categories according to the intrusion detection approaches first one is anomaly detection and second one is misuse detection or some time called as signature detection. Anomaly detection approach or category is also called as behavior detection. Anomaly detection is very important detection approach to detect intrusions by first identifying or learning the characteristics of normal activity of users. Then the implemented system uses such characteristics to evaluate whether the user's activity is normal or not. Misuse detection systems are the design approach that tries to match user activity to already stored signatures of known attacks. Specifically, such detection system uses a previously defined knowledge to identify whether the user new activity is in that define knowledge database. If no, the IDS considers this activity is normal otherwise it may be a possible attack and thus there is a need to blocks it. Genetic Algorithm approach has been widely used in IDSs. One network connection and its related behavior can be translated to represent a rule to find whether or not a real-time connection is considered an intrusion. These specific rules can be modeled as chromosomes inside the population. The population evolves until some evaluation criteria are met. The generated rule set can be used as previous knowledge inside the IDS for finding or identifying whether the network connection and related behaviors are potential intrusions or not. With this approach, the IDS can be viewed as a evolutionary rule-based system and GA can be viewed as a tool to aid for generating knowledge for the designed rule based system.

With this approach in mind we have designed GA based system and implemented fitness function on the processes of GA. The main goal is to get very high intrusion prediction rate and minimum false positive rates on incoming network traffic which is captured by the system. The intrusion detection training is carried out using predefined rule set and apply on real time data sets generated by the windows firewall system. The results generated after successful execution of the algorithm are thus mitigating the choice i.e. performance and applicability of genetic algorithm into the Intrusion Detection System.

2. Literature Survey

The intrusion detection mechanism currently used has many changes which uses new evolved techniques to generate better results. There are numerous approaches for resolving intrusion detection problems on the network.

Wei Li [6] proposed genetic based IDS in which he uses genetic algorithm approach to detect intrusion on the network. This approach of genetic algorithm is very unique as it focuses on both spatial and temporal information of DARPA data set. This proposed approach is really more helpful for identification of network anomalous behaviors.

B. Uppalaiah, T. Bharat et al. [7] proposed the Genetic Algorithm approach for DD99CUP data set to detect intrusion. They describe basic architecture of the proposed system along with implementation of the given approach software. There system is flexible for usage in different application areas with suitable attack taxonomy. Genetic Algorithm actually detects the intrusion and correlation techniques is used to identify the features of the network connections .Their experimental results shows that they have specified set of rules and high Dos, R2L, U2R, Probe attack detect rate. They also optimize the required parameters present in the algorithm to reduces the training time.

Srinivasa K G, SaumyaChandra et al.[8] implemented IGIDS, where the genetic algorithm is used for pruning almost best individuals in the rule set database. The describe process makes the decision much faster as the search space of the resulting rule set is compact compared to the original data set. This approach of GA makes IDS faster and intelligent.

Anup Goyal and Chetan Kumar [9] has main focus on a machine learning approach commonly known as Genetic Algorithm , to identify or detect many harmful/attack on the network. Their algorithm uses various features in network connections such as network service on the destination, current status of the connection to generate a classification rule set and type of protocol . Each rule defined in the rule set identifies a particular attack type. For experiment purpose, they implemented a GA and trained it on the KDD Cup 99 data set so that they generate a rule set that can be applied to create IDS to identify and classify different types of attack connections.

Brian E. Lavender [10] has proposed the integration of genetic algorithms (GA) into SNORT to improve SNORT at performing Network Intrusion Detection (NID).

Shaik Akbar et al. [11] has presented an algorithm which identifies damaging/attack type connections using Genetic Algorithm. The algorithm uses different parameters like protocol type, duration, src_bytes etc to generating a rule set. This Genetic Algorithm approach is trained on KDDCUP99 data set in order to generate a rule set which applied on IDS to identify different types of damaging/attack.

Major headings are to be column centered in a bold font without underline. They need be numbered. "2. Headings and Footnotes" at the top of this paragraph is a major heading.

3. Implementation Details

This approach is to build a systematic framework which can build good model by selecting appropriate features of audit data and other useful information. To build a better IDS system Genetic Algorithm is preferred.

Genetic Algorithm is a Supervised Anomaly detection system which uses evolutionary approaches to build the profile of the network connections. Within a computer simulation, a population of many individuals is created, each individual representing a possible mathematical model. Each individual has one or more chromosomes that function as basic instructions to the individual in a cause (e.g., input data) and effect (e.g., user classification) manner. An individual is measured by the aggregate performance of its chromosomes. An initial population is created by complete randomization of the chromosomes, and individuals of subsequent generations go through mutations, which are also randomized. As in Darwinism, a population that goes through many generations eliminates poor performing individuals and allows better performing individuals to replicate and mutate themselves during each generation. The implemented genetic algorithm is designed so that each individual can represented a possible behavioral model. Genetic Algorithm is a sequentionally design steps which includes encoding the chromosomes i.e. also called initial population, then applying crossover and mutation and parallel approach to check against fitness function.

The ***key idea*** of implemented Genetic Algorithm is the fitness function which is nothing but the most important factor for GA success. In this approach the fitness function is:

Fitness function= (size of packet * weight)

Where the size is the actual packet data size prescribed by the incoming packet data stream and weight is the vector which applied to each chromosome.

The implemented system starts from capturing input from the firewall data sets 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. 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. This fields input are useful to make connection profile on the basis of that we distinguishes the harmful and harmless connections. And finally we get the best individuals.

4. Experimental Setup

This experiment is set up to generate list of IP addresses and their respective packets which are vulnerable to the server or destined system. We performed testing on the entries generated by the default firewall system of the windows machine using pfirewall.log file(System generated File). The training is done on the predefined data rule set. The pfirewall.log file contains the incoming entries of packets with various parameters like date/time, protocol, srcip, destip, srcport, destport, size, action ,flag, ack, type and info. Out of these features we have used only five important parameters lik, src-port, dst-port , src-ip, dst-ip and size of data packet.

To performed experiment we have used well known java software as the frontend and Mysql 5.1 as a backend tool The overall GA operator and there evolution process code .is written using java and the training data is stored into the mysql database. This experiment is performed on windows based platform with recent computer system.

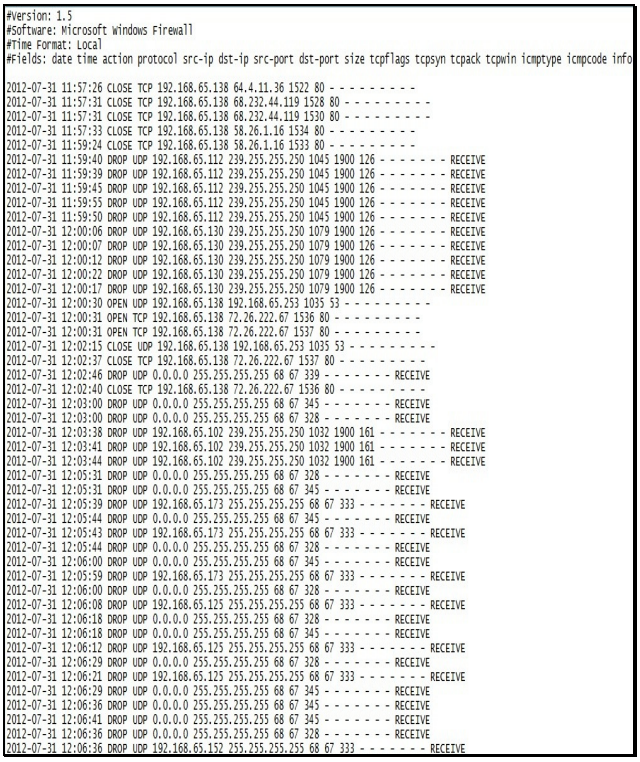
4.1 Implemented Genetic Algorithm Parameters

Table 1: parameter setting for GA algorithm

SETTING TYPE	VALUES
Encoding Scheme	Binary Encoding
Population size	100
Evolution generation	60
Selection	Fitness-proportion
Crossover	One point
Mutation	Real number
Generations	60
Fitness function	Weight * pkt_size

5. Result Discussion

From the above experiment setup, we have now able to create a rule set that could effectively categories harmful and harmless IP addresses. We have shown the resultant figures below by applying 100 connection entries respectively to the implemented system. After that we were able to get around 95% 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 5: 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: specified entries taken by the proposed system for filteration

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 7: Highlighted malicious entries are eliminated by the rule base

```
IP :- 127.0.0.1
127.0.0.1
12700000001
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 8: final list of IP addresses generated by GA-RID

6. Conclusion

In this paper we have implemented the rule set for real time system which can detect existing and new intrusions. This system can be very fruitful to integrate with any of the IDS or firewall system to improve the efficiency and the performance. In this paper, the key idea of implemented work is a fitness function of GA which is nothing but the most important factor for system success. In this approach the fitness function is: Fitness = (size * weight) Where the size is the actual packet data size prescribed by the incoming packet data stream and weight is the vector which applied to each chromosome. The above discussed approach of GA processes and evolution operators approach is really helpful to identify DDoS attack and thus provide security to organization data.

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] 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

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

[4] 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.

[5] Dietrich, S., Long, N., and Dittrich, D. 2000. Analyzing distributed Denial of service attack tools: The shaft case. In Proceedings of 14th Systems Administration Conference. New Orleans, Louisiana, USA, 329-339.

[6] Wei Li “Using Genetic Algorithm for network intrusion detection”

[7] 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.

[8] 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.

[9] Anup Goyal, Chetan Kumar, “GA-NIDS : A genetic algorithm based network intrusion detection system”,

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

[11] 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

[12] 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