Rule based Network Intrusion Detection using Genetic Algorithm

M. Sadiq Ali Khan
Department of Computer Science
University of Karachi

ABSTRACT
The rapid increase of information technology usage demands the high level of security in order to keep the data resources and equipments of the user secure. In this current era of networks, there is an eventual stipulate for development of consistent, extensible, easily manageable and have low maintenance cost solutions for Intrusion Detection. Network Intrusion Detection based on rules formulation is an efficient approach to classify various type of attack. DoS or Probing attack are relatively more common and can be detected more accurately if contributing parameters are formulated in terms of rules. Genetic Algorithm is used to devise such rule. It is found that accuracy of rule based learning increases with the number of iteration.

Keywords
Intrusion Detection; Network Intrusion Detection; Genetic Algorithm; KDD-99

1. INTRODUCTION
In this paper we described a method to apply Genetic Algorithm (GA) for Network Intrusion Detection Systems (NIDSs). In first section a concise overview of the IDS, GA and associated revealing methods are presented. In second section connection attributes and process for GA evolution are discussed in detail. The network connection information is encoded to transform into rules in IDS. Finally result of application of GA is presented.

2. BACKGROUND

2.1 Network Intrusion Detection System
Intrusion Detection System (IDS) has turn out to be an important study area in Computer based security [1]. It is a well-known skill for revealing and is used as a countermeasure to protect data reliability and system accessibility during an intrusion [2]. When a client tries to access into an information structure or carries out an action illegally, the action is referred as an intrusion that can be separated into two sets, *exterior and interior*. The exterior refers to those clients who do not have authorized access to the system and who tried to access illegally by using different saturation methods. Whereas interior refers to those which have valid access permission but desire to carry out illegal activities. Methods for the intrusion may include software bugs exploitation and misconfigurations of the system, password cracking, sniffing unsecured traffic, or utilizing the specific protocols design flaw.

2.2 Genetic Algorithm
Genetic Algorithm (GA) is an efficient investigating method used in computing to locate precise or estimated solutions to optimization and search problems [3]. Genetic Algorithms are categorized as global search heuristics. In GA heuristic search is based on concept of biological evolution In Genetic algorithms iterative mathematical modeling technique is used to find the optimal combinatorial state given a set of parameters of interest. The population evolution process is simulated through genetic programming [4]. A inhabitants of fixed-length is evolved with a GA by employing crossover and mutation operators along with a fitness function that concludes how likely individuals are to reproduce [5]. A testing solution is developed each of which is estimated (to yield fitness) and a new generation is created from the better of them. The procedure is sustained through a numerous generations with the endeavor that the population should evolve to contain a solution which is acceptable.

Following steps are involved in GA application:

1. At the start of a run of a GA, a huge population of random genes is created. Each represents a different solution to the problem at hand. Let's say in the initial population there are N chromosomes then the following steps are repeated until a solution is found.
2. Every chromosome is tested to see how better it is at solving the problem at hand and allocate a fitness score accordingly.
3. From the current population two members are selected. The chance of being selected is proportional to the chromosomes fitness. Normally Roulette wheel selection method is used.
4. Depending on the crossover rate the bits from each selected chromosome are crossover randomly at chosen point.
5. Step through the chosen chromosomes bits and flip dependent on the mutation rate.

Method is repeated until a fresh inhabitant of N elements has been formed. For terminating the conditions any one of the following condition can be used:

1. An entity is found that assures least criteria.
2. Fixed number of generations reached.
3. Financial Plan: due computation time/funds used up.
4. The peak ranking individual's fitness is reaching or has reached a plateau such that successive iterations are not producing better results anymore.

5. Manual scrutiny: May need start-and-stop capability permutations of the above.

2.3 Genetic Algorithm in IDS

In different ways Genetic Algorithm can be used in Intrusion Detection System, the IDS can be illustrated as rule based system and to generate knowledge for the RBS GA is used as tools [6]. In order to identify disturbing behaviors for a LAN, connections of the network must be used to describe normal and abnormal behaviors [7, 8]. Occasionally an attack can be as straightforward as inspecting for ports in-hand on a server. But characteristically they are multifaceted and are created by automated tools that are easily downloadable from the Internet. Trojan horse or a backdoor is the example that can run for a limited time period, or can be initiated from diverse places [9]. In order to identify such interruptions, both chronological and spatial information of network traffic must be incorporated in the rule set. These issues are not address by the present GA, they just broadly demonstrates how information of the network connection can be replicated as genes and how the parameters in GA can be defined in this respect [10, 11]. To show the implementations few examples are used.

3. METHODOLOGY

In this study KDD-99 data set is used with reduced set of attributes. Principal Component Analysis can be used to reduce data set. Reduced data set of parameters proposed in [12]. For application of GA, gene of each activity, connection chromosome and initial population is defined.

3.1 Gene Definition

The reduced sets of attributes are used to define the initial population. The following attributes each representing the gene are selected

service, flag, land, logged_in, root_shell, su_attempted, is_hot_login, is_guest_login

For rule definition connection attributes and range of values of each field is defined as follows

S#	Attribute	Value Range
1	service	0-63
2	flag	0-10
3	land	0, 1
4	logged_in	O,1
5	root_shell	0,1
6	su_attempted	0, 1, 2
7	is_hot_login	0,1
8	is_guest_login	0,1

Gene for Service

1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0

Gene for Flag

1	2	3	4
1	0	1	0

Gene for Land (L)

1
1

Gene for Logged in (LN)

1
1

Gene for Root Shell(R)

1
1

Gene for Su attempted(S)

1
1

Gene for Is_host-Login(H)

1
1

Gene for Is_guest _Logged in(G)

1
1

3.2 Evaluation Function

The subsequent steps are used to compute the estimation function. First of all outcome is calculated based on whether a connection field matches the pre-classified training data set, and then the weight of that field multiplied. The values *Matched* is set to either 1 or 0.

Outcome = $\Sigma_{i=1}$ Number of match X weight of field

Initial Population

11101101100011011
00101001010101101
00001101110110110
00101100001101101
00111110110100001
00001100111011011
11100101101110100
11001010110010101
01001011011010111

3.3 Rule Definition

The purpose of applying GA to intrusion detection was to develop a knowledge base and define rule for detection of intrusion. The following rule was used to define the initial population. Let

DC= don't care
service=μ1
flag=π1
land=Ω1
logged_in=β1
root_shell= μ2
su_attempted= π2
is_hot_login= Ω2
is_guest_login= β2

If (μ2 or π2 or Ω2 or β2) == 0 AND (μ1 or π1 or Ω1 or β1) == DC

Than Categories as "Attacks"

Else Categories as "Normal"

3.4 Learning through GA

Two data set each of size **10000** record of mention parameters were formed. One set was labeled as Training Set other as Test Set. By running GA algorithm more than 2000 iteration were performed. The correct and false detection in both set were recorded along with the false alarm.

4. RESULTS AND DISCUSSION

The initial population of size 10 was evolved through repetitive iteration. Genetic algorithm is found to be more efficient in training of attack as compare to normal, the difference of accuracy in training and test data is almost less than 1 % in case of detection of attack however in case of normal the difference is around 5 % which suggest that detection of new/novel attack by genetic algorithm required to be further improved The following results was obtained in 2000 iteration.

Training Data

Type	Occurrence	Correct Identification	Incorrect Identification	Reliability (%)
Normal	5000	4460	540	89.2
Attack	5139	4864	275	94.64

Test Data

Type	Occurrence	Correct	Incorrect	Reliability
Normal	5040	4710	330	93.45
Attack	4958	4670	288	94.19

The genetic algorithm is found more efficient in terms of false alarm. In normal attacks it is only around 10%. The other method such as Naïve Bayes Classifier and Junction Tree algorithm are efficient in estimation of probability but performance in giving false alarm is not good. In attacks which are not common (imap, rootkit, satan etc), frequency of false alarm is more than 30-35 % [12]. In case of genetic algorithm false alarm in attack is less than 5%

False Alarm

Type	Detected	Occurrence	False Alarm %
Normal	Attack	540	10.8
Attack	Normal	275	2.75

The convergence in initial iteration is more rapid as the basic rule to address more relevant attribute is successful although the service category of network parameter have almost 64 different values, however in actual data set few services as http, ftp are more common so in initial iteration search is narrowed down to good extent. In later alterations rate of convergence is almost become static so increasing further iteration beyond certain limit did not improve accuracy of result.

S #	**Iterations**	**Accuracy (%)**	
		Training	**Test**
1	500	74	71
2	1000	81	79
3	1500	86	84
4	2000	93	91

5. CONCLUSION

The study showed that GA can be effectively used for formulation of decision rules in intrusion detection through the attacks which are more common can be detected more accurately. Rule based classification of DoS and Probe attacks can be used for effective monitoring of the network. Application of GA as compare to expert based knowledge is more fruitful as different possible combination of attribute is tested against training data and later validated through test data. It is also determined that increasing number of iteration of application of algorithm contributes in accuracy of data. However initial iteration converges to result more quickly as compare to later iteration.

The factors which need further investigation are the classification of generated rules which converge to solution more quickly. Beside this, development of knowledge base as a result of GA application can be utilized for further investigation for identification of attribute which contribute for accurate classification of attack.

6. REFERENCES

[1] McHugh, John, 2001. "Intrusion and Intrusion Detection." Technical Report. CERT Coordination Center, Software Engineering Institute, Carnegie Mellon University.

[2] D. Wagner and D. Dean, "Intrusion detection via static analysis," in Proc. IEEE Symposium on Research in Security and Privacy, Oakland, CA, 2001.

[3] Miller, Brad. L. and Michael J. Shaw. 1996. "Genetic Algorithms with Dynamic Niche Sharing for Multimodal Function Optimization." In Proceedings of IEEE International Conf. on Evolutionary Computation, pp. 786-791.Nagoya University, Japan.

[4] Wan Tang; Yang Cao; Xi-Min Yang and Won-Ho So, "Study on Adaptive Intrusion Detection Engine Based on Gene Expression Programming Rules", Computer2008, Vol 3; Pages 959-963.

[5] Santhosh Kumar, S.; Vignesh, J.; Rangarajan, L.R.; Narayanan, V.S.; Rangarajan, K.M.; Venkatkrishna, A.L., "A Fast Time Scale Genetic Algorithm based Image Segmentation using Cellular Neural Networks", Signal Processing and Communications, 2007. ICSPC 2007, IEEE, Pages 908-911.

[6] N. Ye, X. Li, Q. Chen S. M. Emran, and M. Xu, "Probabilistic techniques for intrusion detection based on computer audit data," IEEE Trans. SMC-A, vol. 31, pp. 266–274, Jul. 2001.

[7] Baoyi Wang; Feng Li; Shaomin Zhang, "Research on Intrusion Detection Based on Campus Network", Intelligent Information Technology Application, 2009 Vol 1, Pages 468-471.

[8] Akyazi, U; Uyar and A.S.E., "Distributed Intrusion Detection Using Mobile agenets against DDoS attacks", Computer and Information Sciences 2008, Pages 1-6.

[9] Fiskiran, A.M; Lee, R.B., "Runtime Execution Monitoring to detect and prevent malicious code execution", Computer Design: VLSI in computers and Processors 2004, IEEE International Conference Pages 452-457.

[10] Jiang, M.; Munawar, M.; Reidemeister, T.; Ward, P, "Efficient Fault Detection and Diagnosis in complex Software Systems with Information- Theoretic Monitoring", IEEE transactions on Dependable and Secure Computing 2011, Issue 99.

[11] Nicoletta Dessì and Barbara Pes, "An Evolutionary Method for Combining Different Feature Selection Criteria in Microarray Data Classification", Journal of Artificial Evolution
and applications, Vol 2009.

[12] Burney S. M. Aqil, Sadiq Ali Khan and Jawed Naseem, 2010, "Efficient Probabilistic Classification Methods for NIDS", (IJCSIS) International Journal of Computer Science and Information Security, Vol. 8, No. pp168-172