Genetic Algorithm for framing rules for Intrusion Detection

S. Selvakani[†] and R.S.Rajesh[††],

[†]Asst.Prof, MCA Dept, SCAD College of Engg & Technology, Tirunelveli, Tamil nadu, India
[††]Reader, Dept of CSE, MS University, Tirunelveli, Tamil nadu

Summary
With the rapid expansion of computer networks during the past decade, security has become a crucial issue for computer systems. The detection of attacks against computer networks is becoming a harder problem to solve in the field of Network security. Intrusion Detection is an essential mechanism to protect computer systems from many attacks. As the transmission of data over the internet increases the need to protect connected system also increases. Therefore, unwanted intrusions take place when the actual software systems are running. A brief overview of Intrusion Detection System, genetic algorithm and related detection techniques was presented. Developing rules manually through incorporation of attack signatures results in meaningful but weak as it is difficult to define thresholds. In this paper the method of learning the Intrusion Detection, rules based on genetic algorithms was presented. The experimental results are demonstrated on the KDD cup 99 intrusion detection data set. In our experiments the characters of an attack such as smurf and warezmaster were summarized through the KDD 99 data set and the effectiveness and robustness of the approach are proved.

Key words:
Attack Signatures, Intrusion Detection, Genetic Algorithm, KDD Cup Set, Rule set.

1. Introduction

With the rapid expansion of Internet in recent years, computer systems are facing increased number of security threats. Despite numerous technological innovations for information assurance, it is still very difficult to protect computer systems. Therefore unwanted intrusions take place when the actual software systems are running. While we are benefiting from the convenience that the new technology has brought us, computer systems are exposed to increasing security threats that originate externally or internally. Despite different protection mechanisms, it is nearly impossible to have a completely secured system. Therefore Intrusion detection is becoming an important technology that monitors network traffic and identifies network intrusions such as anomalous network behaviours, unauthorized network access and malicious attacks to computer systems.

When an intruder attempts to break into an information system or performs an action not legally allowed, that is called as an intrusion. Intruders can be divided into two groups, external and internal. The former refers to those who do not have authorized access to the system and who attack by using various penetration techniques. The latter refers to those with access permission who wish to perform unauthorized activities. An Intrusion detection system is a system for detecting intrusions and reporting them accurately to the proper authority [3].

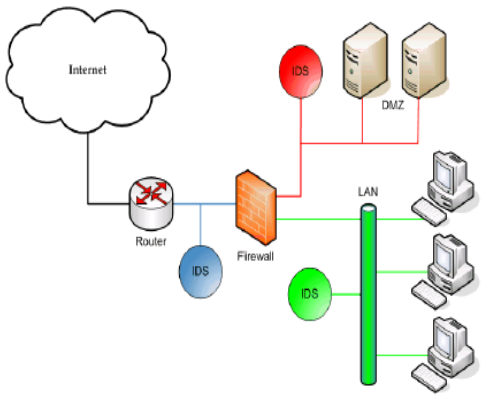

Fig 1: A Computer Network with Intrusion Detection Systems

There are two general categories of intrusion detection systems: misuse detection and anomaly detection. Misuse detection system detects intruders with known patterns, and anomaly detection 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. Fig.1 Shows the Computer Network with Intrusion Detection Systems. A host-based IDS monitors activities associated with a particular host, and a network based IDS listens to network traffic.

Manuscript received November 5, 2007
Manuscript revised November 20, 2007

A number of soft computing based approaches have been proposed for detecting network intrusions [1]. Soft computing refers to a group of techniques that exploit the tolerance for imprecision, uncertainty, partial truth, and approximation to achieve robustness and low solution cost. When used for intrusion detection, soft computing techniques are often used in conjunction with rule based expert systems acquiring expert knowledge [4] where knowledge is represented as a set of if-then rules.

In this paper, we present a GA-based approach to network misuse detection. GA is chosen because of some of its nice properties, e.g., robust to noise, no gradient information is required to find the global optimal or sub-optimal solution, self learning capabilities etc. The software is experimented using DARPA data sets on intrusions, which has become the de facto standard for testing intrusion detection systems. The experimental results show that our approach is effective and it has the flexibility to either generally detect network intrusions or precisely classify the types of misuses.

From this section, input the body of your manuscript according to the constitution that you had. For detailed information for authors, please refer to [1].

2. Genetic Algorithm

Genetic algorithms [2,9] employ metaphor from biology and genetics to iteratively evolve a population of initial individuals to a population of high quality individuals, where each individual represents a solution of the problem to be solved and is compose of a fixed number of genes. The number of possible values of each gene is called the cardinality of the gene.

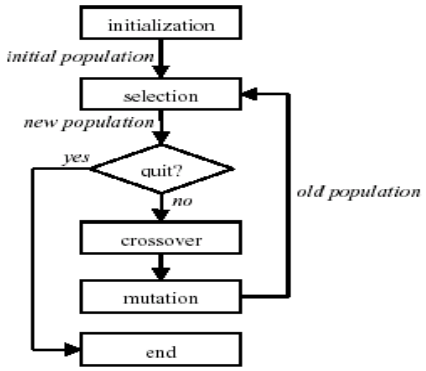

Fig.2 Structure of GA

Fig 2 illustrates the operation of a genetic algorithm. The operation starts from an initial population of randomly generated individuals. Then the population is evolved for a number of generations and the qualities of the individuals are gradually improved. During each generation, three basic genetic operators are sequentially applied to each individual with certain probabilities like selection, cross over, and mutation. 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.

When a GA is used for problem-solving, three factors will have impact on the effectiveness of the algorithm, they are: 1) the selection of fitness function; 2) the representation of individuals; and 3) the values of the GA parameters.

3. Related Work

This section briefly summarizes some of the applications of soft computing techniques for intrusion detection. However, a number of GA based IDSs are discussed in the later part of the paper in order to compare and contrast those work with our work.

GAs and GP have been used for network intrusion detection in different ways. Some approaches directly use GAs to derive the classification rules [6,7], 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 [4,8].

The early effort of using GAs for intrusion detection can be dated back to 1995, when Crosbie *et*[2] 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.

Bridges *et al.* [1] develop a method that integrates fuzzy data mining techniques and genetic algorithms to detect both network misuses and anomalies. 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. The authors propose a way to include quantitative features by introducing fuzzy numerical functions. Their preliminary experiments show that the inclusion of quantitative features and the fuzzy functions significantly improved the accuracy of the generated rules. In this approach, a GA is used to

find the optimal parameters of the fuzzy functions as well as to select the most relevant network features.

Lu *et al.* [7] 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 [6] 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.* [10] 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. This paper shall analyze selected DOS and R2L attacks. This paper studies warezmaster and smurf attacks as representative instances from the R2L and DOS category. Typical and relevant features must be observed present in the KDD data set [5] that can help with the detection of these attacks.

4. GA Applied to ID

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.

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

Smurf							
Duration	**src_bytes**	**hot**	**num-failed logins**	**count**	**srv_count**	**srv_rerror_rate**	**srv_diff_host_rate**
0	1032	0	0	508	508	0	0
0	1032	0	0	509	509	0	0
0	1032	0	0	510	510	0	0
0	1032	0	0	510	510	0	0
0	1032	0	0	511	511	0	0
0	1032	0	0	511	511	0	0
0	1032	0	0	511	511	0	0
0	1032	0	0	506	506	0	0
0	1032	0	0	509	509	0	0
0	1032	0	0	509	509	0	0
0	1032	0	0	510	510	0	0

Table 1: Smurf sample data set with reduced features

A Dos attack is an attack in which the attacker makes some computing or memory resource too busy or too full to handle legitimate requests or denies legitimate users access to a machine. 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.

```
if Number of "hot" indicators <=0.0 and
   Number of connections to the same host as the
   connection in the past  two seconds <=500.82 and
    % of connections that have "REJ" errors >0.21
    and <=0.01 and Number of connections to
     host <=41.2 and >112.3
then SMURF
```

Warezmaster exploits a system bug associated with a FTP server. Normally guest users are never allowed write permissions on an FTP server. Hence they can never upload files on the server. This attack takes place when an FTP server has, by mistake given write permissions to users on the system. Hence any user can login and upload files.

```
if duration > 265 and protocol = tcp and
    service = ftp v ftp_data and
  src_bytes > 265616 v <=283618 and dest_bytes =0 and
  hot >0 v <=2 and is_guest_login =1 and
  Number of services requested of host <=25.83
then WAREZMASTER
```

Warezmaster							
Duration	**Protocol Type**	**Service**	**Src bytes**	**Dstbytes**	**Hot**	**Loggedin**	**Isguest Login**
140	tcp	ftp_data	0	283618	0	0	1
282	tcp	ftp	158	597	0	0	2
142	tcp	ftp	142	461	0	0	1
280	tcp	ftp_data	283618	0	0	0	1
0	tcp	ftp_data	12	0	0	0	2
0	tcp	ftp_data	12	0	0	0	2
0	tcp	ftp_data	12	0	0	0	1
280	tcp	ftp_data	283618	0	0	0	2
281	tcp	ftp	162	597	0	0	1
0	tcp	ftp_data	12	0	0	0	1
282	tcp	ftp	156	593	0	0	1

Table 1: Warezmaster sample data set with reduced features

During the execution of the attack, the attacker logs on the server using the guest account. The attacker then creates a hidden directory and uploads copies of illegal software on to the server. Other users can then later download these files. If a huge amount of data is sent from source as compared to that form destination then it can be assumed that data is being uploaded.

An FTP connection is observed by verifying that the protocol is TCP and the service is FTP or FTP_DATA. This rule suggests than an FTP connection has been active for an extended period of time and a large amount of data has been transferred from the source machine with no data received from the destination machine and also the rule suggests that hidden directories (hot=1) are created when a guest has logged in. The rules are generated using logged in. The rules are generated using the C4.5 algorithm on the part of corrected KDD training data set.

Since the GA has to use such rules to detect intrusions, such rules in the rule set will be codified to the GA format. Each rule will be represented in a GA format. The first part of the GA will act as a search algorithm. It will match the rules with any anomalous connections that occur the network to detect an intrusion. Each rule will carry values for the intrusions that they have detected and a value for the intrusion. Each rule will carry values for a false alarm that the rule produces.
The second part of the GA is the fitness function. The fitness function F determines whether a rule is good or bad. F is calculated for each rule using the support confidence framework.

$$\text{Support} = |A \text{ and } B| / N$$
$$\text{Confidence} = |A \text{ and } B| / |A|$$
$$\text{Fitness} = t1 * \text{support} + t2 * \text{confidence}$$

Where
N is the total number of records
$|A|$ stands for the number of network connections matching the condition A
$|A \text{ and } B|$ is the number of records that matches the rule.
T1 and t2 are the thresholds to balance the two terms.

5. Performance evaluation of proposed rules on the KDD data set

These rules used reduced features from the KDD data set and tested on the KDD training set to observe their performance with respect to detection, false alarm and missed alarm rates. If the rules are well formed, then the detection rate is expected to be high while concurrently achieving low false alarm and missed alarm rates.
For each chromosome in the population, the number of network connections and the number of connection that matches the condition is initialized to zero. For each record in the training set, if the record matches the chromosome update the network connection by 1 and if the record only matches the condition part, then update that value A by 1. Then calculate the fitness of each rule and select the best fit rules into new population. Then apply the crossover and mutation operators to each rule in the new population. Finally, decide whether to terminate the training process or to enter the next generation to continue.

Record Type	Training Set	Testing Set
Normal	96.2%	93.8%
Smurf	90.6%	67.3%
WarezMaster	92.7%	76.6%

Table 3: Results (Detection Rates)

The experimental result, Table 3 shows that the proposed method yielded good detection rates when using the generated rules to classify the training data itself.

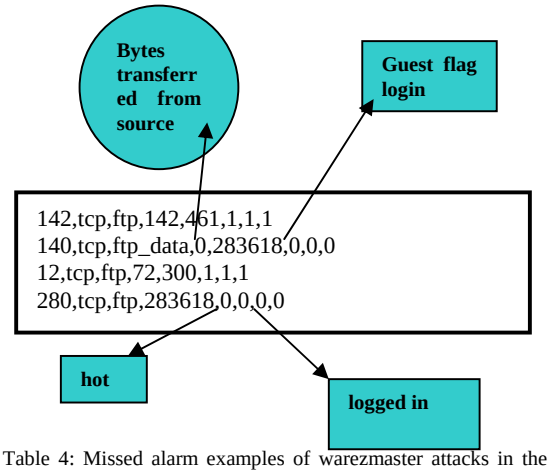

Table 4: Missed alarm examples of warezmaster attacks in the KDD data set

6. Conclusion

This paper utilized a technique for creating rules for R2L and DOS attacks. Probability of detection and false alarm rates are computed on the KDD data set. The overall performance is reasonably good with an average of 59% detection rate and achieved with only 0.1% false rate.

Although we are satisfied with the results of experiment, the number of records limited up to 2000. Most of the network security systems require capability to handle over one million concurrent sessions.

However, some limitations of the method are also observed. First the generated rules were biased to the training data set. This paper is mainly an attempt towards overcoming various short comings in the context of network intrusion detection. The data set used serves as the basis for the intrusion detection process to detect intrusions. It is represented in the form of a table or a record set. In future work, a generalized form in which the data set should be represented from various representations of data sets, still needs to be researched to be compatible with any kind of IDS.

Second while the support – confidence framework is simple to implement and provides improved accuracy, it requires the whole training data to be loaded into memory before any computation. For large data sets, it is neither efficient nor feasible. Future work will attempt to use the neural network technologies to automate the generation of rules.

Acknowledgment

I would like to thank our parents, husband, son and friends for helping us start and continue this exhausting work on a boring research topic. I would also like to thank the management of SCAD CET especially chairman Dr. Cletus Babu and VC Mrs Amali Cletus Babu for their moral and financial support.

References

[1] Bridges S.M and Vaughn R. B, "Fuzzy Data Mining and Genetic Algorithms Applied to Intrusion Detection", Proceedings of 12th Annual Canadian Information Technology Security Symposium, pp.109-122, 2000.

[2] Crosbie M and Spafford E, "Applying genetic Programming to Intrusion Detection", Proceedings of the AAAI Fall Symposium, 1995.

[3]Graham R FAQ: Network Intrusion Detection Systems, http://www.robertgraham.com, 2000.

[4]Gomez J and Dasgupta D, "Evolving Fuzzy Classifiers for Intrusion Detection", Proceedings of the IEEE, 2002.

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

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

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

[8] Middlemiss M and Dick G, "Feature Selection of Intrusion detection data using a hybrid genetic algorithm/KNN Approach", Design and Application of hybrid Intelligent systems, IOS Press Amsterdam, PP.519-527, 2003.

[9] Pohlheim H, "Genetic and Evolutionary Algorithms: Principles Methods and Algorithms", http://www.gearbx.com/docu/index.html, January 2005.

[10] Xiao. T, QU. G, Hariri. S and Yousif. M, „An efficient Network Detection Method Based on Information Theroy and Genetic Algorithm", Proceedings of the 24th IEEE International Performance Computing and Communications Conferecne, Phonix, AZ, USA, 2005.

S. Selvakani received the MCA degree from Manonmanium Sundaranar University. Her research interest includes Network Security and Soft computing. She has presented 4 papers in National Conference and 1 paper in international conference. She has published 1 paper in National journal and 1 paper in International Journal. She is currently pursuing her Ph.D degree in Network Security under the Guidance of Dr. R.S.Rajesh. Presently she is working as an Asst.Prof, MCA Dept in SCAD College of Engineering and Technology, Tirunelveli

Dr. R. S Rajesh received his B.E and M.E degrees in Electronics and Communication Engineering from Madurai Kamaraj University, Madurai, India in the year 1988 and 1989 respectively, and completed his Ph.D in Computer Science and Engineering from Manonmaniam Sundaranar University in the year 2004. In September 1992 he joined in Manonmaniam Sundaranar University where he is currently working as Assistant Professor in the Computer Science and Engineering Department. He got more than 17 years of PG teaching and Research experience. His current research interests include, Wireless networks, Pervasive computing, Digital image processing and Parallel Computing.