Intrusion Detection System based on Learning Fuzzy Rules and Membership Functions using Genetic Algorithms

Ezat Soleiman
Department of ICT Engineering
Malek Ashtar University
Islamic Republic of Iran

Abdelhamid Fetanat
Department of ICT Engineering
Malek Ashtar University
Islamic Republic of Iran

ABSTRACT
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. Different soft computing based approaches have been proposed to detect computer network attacks. Hybrid methods proved more effective and accurate, this paper tries to introduce how to use dynamic fuzzy rules and genetic algorithm in intrusion detection systems.

Keywords
Genetic algorithm, intrusion detection system, IDS, fuzzy systems.

1. INTRODUCTION
For pages other than the first page, start at the top of the page, and continue in double-column format. The two columns on the last page should be as close to equal length as possible.

Nowadays computers and the Internet are used almost in every part of our lives. Since the personal computer was invented it has been growing faster and faster and it is now impossible to imagine companies, universities or even a little shop that does not keep all the data of their customers, purchases and inventory in an electronic database or computer.

With the possibility of connecting several computers and networks was born the necessity of protecting all this data and machines from attackers (hackers) that would like to get some confidential information to use for their own benefit or just destroy or modify valuable information.

There are several security measures available to protect the computer resources of a company or a home user, but even if all expert recommendations are followed, our systems will never be safe against possible successful attacks. It is very difficult to get an invulnerable system, probably impossible and one may need to spend a lot of money designing and developing it. In companies, a very isolated system could drastically reduce productivity and for a not very experienced home user it may become a "hating technology" disease. For all these reasons the user or the security department should know what their values are, if they need to be protected and how much it costs, doing Risk Analysis [1].

According to the Edward Amoroso [2] the intrusion is "sequence of related actions by malicious adversary that results in the occurrence of unauthorized security threats to a target computing or networking domain".

An intrusion is considered as a sequence because it propagates under a current period. Actions causing the intrusion must be related, since unrelated ones are not of interest. As an intruder always has an attention to make an intrusion, he must be considered as a malicious adversary. Assuming there is one defined security policy, unauthorized security threat, is its violation.

A good security policy and a good risk analysis with well-educated users will make the system more secure to intrusions. An intrusion in the system will try to compromise one of the three main aspects in computer security.

• Confidentiality: the intruder has access to confidential information.

• Integrity: information can be modified or altered by the attacker.

• Availability: the system gets blocked so it cannot be used normally.

2. WHY USE AN IDS?

Intrusion detection allows protecting organization systems against threats that appear with increasing network connectivity and the interdependency of information systems.

IDSs have gained acceptance as a main part of the security infrastructure within an organization. There are several reasons for acquiring and using an IDS [3]:

- Avoid problems by dissuading hostile individuals
- Detect attacks and other security violations not prevented by other protection measures.
- Detect attack preambles
- Record the organization risk
- Provide useful information about the intrusions currently taking place

3. TYPES OF COMPUTER ATTACKS
The standard attack classification that is widely used is by Kendall [3]. The attacks are grouped into five major categories.

3.1 Denial of service (DOS)

The example attacks include: Apache2arppoison, Back, Crashiis, dosnuke, Land, Mail-bomb, SYN Flood (Neptune), Ping of Death (POD), Process Table, selfping, Smurf, sshprocesstable, Syslogd, tcpreset, Teardrop, Udpstorm.

3.2 User to root (U2R)

U2R refers to a class of exploit in which the attacker breaks into the system as the normal user then eventually completely control the machine as the root user. The example attacks include anypw, casesen, Eject, Ffbconfig Fdformat, Loadmodule, ntfsdos, Perl, Ps, sechole, Xterm, yaga.

3.3 Remote to local (R2L)

R2L refers to the exploits that start from remote network-based access which intend to break into the machine and obtain the user account. Example attacks include: Dictionary Ftpwrite Guest, Httptunnel, Imap, Named, ncftp, netbus, netcat, Phf, ppmacro, Sendmail, sshtrojan, Xlock, Xsnoop

3.4 Probes

The example attacks include: insidesniffier, Ipsweep, ls domain,Mscan, NTinfoscan Nmap, queso, resetscan, Saint, Satan.

3.5 Data

The example attacks include: Secret.

3.6 Trojan horses / worms – attacks:

That are aggressively replicating on other hosts

4. GENETIC ALGORITHMS

4.1 Introduction

GAs take their inspiration from biological evolution as proposed by Darwin. In biological evolution, individuals from species that adapt to their environment have a chance to survive and reproduce through natural selection. Species that survive usually develop new capabilities and capacities that can be inherited by offspring, if those prove to be worthwhile, and can be maintained through generations [4].

Table 1 shows a brief description of the correspondence between natural and artificial terminology.

Table1. GA and Natural Terminology Comparison

Natural	Genetic Algorithm
chromosome	String
Gene	feature, character or detector
Allele	feature value
Locus	string position
genotype	structure, or population
phenotype	parameter set, alternative solution, a decoded structure

4.2 Initial Population

GA starts with a population of strings to be able to generate successive populations of strings afterwards. The initialization is usually done randomly [12]. Once a population is generated, all individual in that population has to be evaluated to distinguish between good and bad individuals.

4.3 Selection

The individuals that are chosen for mating (recombination) and how many offspring each individual produces are determined by the selection method.

4.4 Recombination (crossover)

The function of the crossover operator is to allow the advantageous qualities to be spread throughout the population in order that the population as a whole may benefit from this chance discovery.

4.5 Mutation

After recombination, every offspring undergoes mutation. Offspring variables are mutated by small perturbations (size of the mutation step), with low probability.

4.6 Reinsertion

After producing offspring, they must be inserted into the population. This is especially important, if less offspring are produced than the size of the original population[5].

5. RELATED WORKS

The early effort of using GAs for intrusion detection can be dated back to 1995, when Crosbie et al. [6] 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.

Bridges et al. [7] 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.

Chittur et al. [8] 41 unique attributes were compiled from nine weeks of raw TCP dump data from a network. Five million separate connection records were created.

Li et al. [9] Applied a GA to network IDS. It considered both temporal and spatial information of network connections in encoding the network connection information into rules in the IDS. The final goal of the GA was to generate rules that matched only the anomalous connections. In this implementation, the network traffic used for GA is pre-classified data set that differentiates normal network connections from anomalous ones.

Lu et al. [10] used GP for detecting novel attacks on networks. The use of GP to detect unknown attacks was based on the hypothesis that the new rules would have better performance than the initial rules that were based on known attacks.

Xiao et al. [11] presented an approach that used 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.

6. PROPOSED METHOD

Previous works on intrusion detection systems use evolutionary fuzzy systems that use fixed rule length or fixed subset of feature space and fixed membership function for all of intrusions. One question may appear in mind is that are these assumptions effect on the result of intrusion detection system? In this work, we try to find a reasonable answer to the question using genetic algorithm. The remaining parts of proposed method are organized as follow: Section I explains dynamic length of rules versus fixed rules length. Section II explains dynamic range for membership function. Based on these introductions, we will introduce a GA based algorithm for intrusion detection systems in section III. Limitations and the applicable scope of proposed method will be introduced in section IV [12].

6.1 Dynamic rule length

In previous works on IDSs that use genetic algorithm, if-then rules had fix length. For example, if part of rules had 8 conditions.

If x1 is A1 and x2 is A2 and … and x8 is A8 Then Attack is U2R

So that {x1…x8} is a subset of feature space F. It is illogical that we use fix length rules for all of attack. For example U2R attack may needs more condition than R2L attack. So in the first step we should put this approach to our genetic algorithm for finding the best rule set for each attack. For each attack we can use a bit for encoding the presence of each feature in the rules. In other words, if we have 41 features, we have a 41 bit number so that first bit represents the participation of first feature in the condition part of rule and this is the same for the rest of features. If the bit was 1, it means that feature is in condition part of the rule, otherwise it isn't.

6.2 Dynamic range for membership function

Ordinary methods of fuzzy intrusion detection system use triangular membership function such as figure 1.

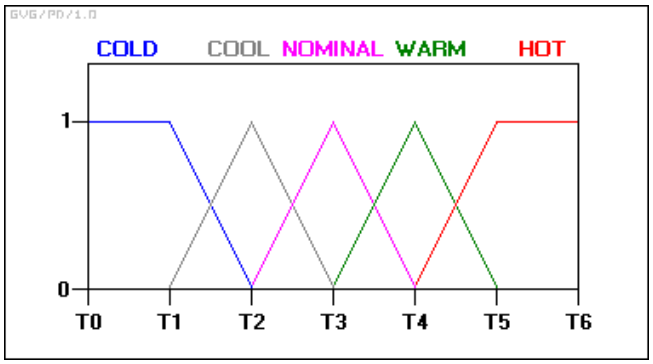

Figure: 1

These methods use equal triangles in membership functions for all features. So the values of T1…T5 are fixed. Dynamic membership function can help us to fit these triangles to their features values. Obviously, this scenario isn't valid for non-numerical features like UDP or TCP connection types. For these features, we will not use membership function at all. Figure 2 shows the bit sequence of T1…TM.

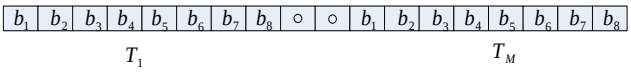

Figure 2

6.3 Population size:

We can start the GA with G=1000 population size.

6.4 Chromosome:

A chromosome is a binary sequence of rules properties like figure 3.

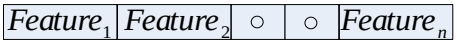

Figure 3

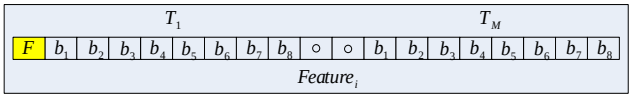

Figure 4

F value in each feature indicates the presence of that feature in the condition part of the rule.

6.5 Fitness function:

Fitness function is the classification accuracy of train samples.

$$Fitness = \frac{n-e}{n}$$

e is the error samples count.

6.6 New population generation:

The selection of parent chromosome is based on fitness function and roulette wheel selection algorithm. The cross over algorithm is two points crossover algorithm so that cross over points are selected randomly on chromosomes. The mutation is done by toggling a random bit of chromosome [13].

6.7 Termination condition:

The algorithm will end when the error rate reaches under the user defined threshold T or the epochs of iteration reaches to predefined value of epochs.

6.8 Scope and limitations

Based on the learning data, the usage of proposed method can be different. It can be used in the network or in individual stations. KDD99 has a diverse data collection gathered from many points of network. So the results of the proposed method can be indicative that the proposed method is a suitable solution for all points of network.

The limitations of proposed method are as follow:

- The speed of training is a negative aspect of proposed method. Since the speed of algorithm is not our primary goal, we put it aside for future works.
- The rule set may be increased at the outlier sections of feature space because in these sections, the data do not have a regular behavior. So, our proposed method tries to learn that disorder sections.
- Large rule set causes more intelligence in detecting the intrusion and less intelligence in detecting anomaly intrusions. Our proposed method has no policy to limit the rule set.

7. REFERENCES

[1] Dieter ollmann. Computer Security, Second Edition. Wiley, New Jersey, 2002.

[2] Edward G.Amoroso, "Intrusion Detection – An Introduction to Internet Surveillance, Correlation, Trace Back, Traps and Response", Intrusion.net Books, 1999.

[3] Kendall, K., "A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems, in C.S. 1998, Massachusetts Institute of Technology: Boston

[4] Pedro A.Diaz-Gomez, "optimization of parameters for binary genetic algorithms", Doctor of philosophy, University Of Oklahoma, 2007.

[5] Kafi I.Hassan, "Adaptive algorithm for obtaining in-phase (I) and quadrature-phase (Q) pseudo-noise (PN) sequences in CDMA", Doctor of philosophy, The City University Of New York, 2005.

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

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

[8] Chittur, A. "A Model Generation for an Intrusion Detection System Using Genetic Algorithms". http:// ww1.cs.columbia.edu/ids/publications/ gaids-thesis01.pdf. Accessed January, 2005

[9] Li, W. "A Genetic Algorithm Approach to Network Intrusion Detection". http://www.giac.org/practical/GSEC/Wei_Li_GSEC.pdf. Accessed January 2005

[10] W. Lu and I. Traore, "Detecting New Forms of Network Intrusion Using Genetic Programming", Computational Intelligence, vol. 20, pp. 3, Blackwell Publishing, Malden, pp. 475-494, 2004

[11] 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

[12] Marbin Pazos-Revilla, "FPGA based fuzzy intrusion detection system for network security", Master of science, The Faculty of Graduate School, Tennessee Technological University, May 2010.

[13] Iosif-Viorel Onut, "A fuzzy feature evaluation framework for network intrusion detection", Doctor of philosophy, the University of New Brunswick, April 2008.