Intrusion Detection System Using Genetic Algorithm

Salah Eddine Benaicha, Lalia Saoudi, Salah Eddine Bouhouita Guermeche, Ouarda Lounis
Computer Science Department
University of Mohamed Boudiaf of M'Sila
M'Sila, Algeria
Salah6234@yahoo.fr, Saoudi_l@yahoo.fr, Bouhouita@usa.com, Ouarda64@yahoo.com

***Abstract*—In this paper, we present a Genetic Algorithm (GA) approach with an improved initial population and selection operator, to efficiently detect various types of network intrusions. GA is used to optimize the search of attack scenarios in audit files, thanks to its good balance exploration / exploitation; it provides the subset of potential attacks which are present in the audit file in a reasonable processing time.**

In the testing phase the Network Security Laboratory-Knowledge Discovery and Data Mining (NSL-KDD99) benchmark dataset has been used to detect the misuse activities. By combining the IDS with Genetic algorithm increases the performance of the detection rate of the Network Intrusion Detection Model and reduces the false positive rate.

Keywords*—*intrusion detection system; genetic algorithm; NSL_KDD; fitness function; initial population

I. Introduction

Local networks and Internet are growing at an exponential rate in recent years. While we enjoy the advantage that the new technology has brought us, computer systems are exposed to increasing security threats that originate from external or internal hosts. Although the different mechanisms of protection, it is almost impossible to have a totally secure system. Therefore, intrusion detection technology becomes more and more important that monitors traffic and identifies network intrusion.

There are two major categories of the analyze techniques of IDS (Intrusion Detection System): the anomaly detection and the misuse detection. Anomaly detection uses the established normal profiles to identify any unacceptable deviation as the result of an attack. In a misuse detection system, also known as signature based detection system; well-known attacks are represented by signature. The Misuse approach uses several techniques grouped into three classes:

1) *The rule-based approaches or expert systems,*
2) *Approaches based on signature*
3) *Genetic Algorithms GA.*

In this paper, we present a GA approach for network intrusion detection.

Genetic algorithms are used to optimize the search of attack scenarios in audit files, thanks to its good balance exploration / exploitation; it provides the subset of potential attacks which are present in the audit file in a reasonable processing time. Our implemented system contains two modules where each operates at a different stage. In the training stage, a set of rules are generated from the audit data. In the stage of intrusion detection, produced rules are used to classify incoming network connections in real time. But the main goal is the optimization of the number of signatures in the audit file to minimize the search time and increase the detection rate of attacks.This system is tested using the Defense Advanced research Project Agency (DARPA) data set , the Network Security Laboratory-Knowledge Discovery and Data Mining (NSL-KDD) [1], which has become the standard test systems for intrusion detection.

The rest of the paper is organized as follows. Section2 shortly describes some previous works, section 3 gives an overview about intrusion detection system, Section 4 and Section 5 focuses on the genetic algorithm and its application in intrusion detection. Our intrusion detection mechanism is introduced into the section 6. The last section is the step of testing; it presents the environment as well as the experimental results.

II. Related Work

The effort of using GAs for intrusion detection can be referred back to 1995, when Crosbie and Spafford [2] applied the multiple agent technology and Genetic Programming (GP) to detect network anomalies.

Bridges et al. [3] have developed a method that integrates fuzzy data mining techniques and genetic algorithms to detect network anomalies. In this approach, a GA is used to find the optimal parameters of fuzzy functions as well as to select the most appropriate network functions.

Lu and Traore [4] used historical network dataset using GP to derive a set of classification. They used support-confidence framework as the fitness function and accurately classified several network intrusions. But their use of genetic programming made the implementation procedure very difficult and also for training procedure more data and time is required.

Li [5] described a method using GA to detect anomalous network intrusion. The approach includes both quantitative and categorical features of network data for deriving classification rules, but no experimental results are available.

Gong et all [6] followed Li [5] to implement its approach. Li has laid the foundation for the creation of a system using genetic algorithms analysis of DARPA data sets, and Gong proposed another implementation using Java-based Evolutionary Computation Research System (ECJ Evolutionary Computation in Java).

Goyal and Kumar [7] described a GA based algorithm to classify all types of smurf attack using the training dataset with false positive rate is very low (at 0.2%) and detection rate is almost 100%.

III. Intrusion Detection Techniques

An intrusion detection system (IDS) is a mechanism to monitor the activity of a network or a given host to detect intrusion attempts and possibly react to this attempt. An IDS normally consists of three functional components [8]:

1) The event generator: is a data source. Data sources can be categorized into four categories namely Host-based monitors, Network-based monitors, Application-based monitors and Target-based monitors.

2) The analysis engine: This component takes information from the data source and examines the data for symptoms of attacks or other policy violations. The analysis engine can use one or both of the following analysis approaches:

Anomaly/Statistical Detection: These are based on observations of deviations from normal system usage patterns. They analyses system event streams, using statistical techniques to find patterns of activity that appear to be abnormal. The primary disadvantages of this system are that they are highly expensive and they can recognize a normal behavior as intrusive behavior because of insufficient data. This technique has had some success in detecting previously-unknown attack techniques.

The main advantage of behavioral IDS is to detect new types of attacks. Indeed, the IDS is not programmed to recognize specific attacks but to report any abnormal activity [9].

Misuse/Signature-Based Detection: also known as signature based detection system; well-known attacks are represented by signature. A signature is a pattern of activity which corresponds to intrusion. The IDS identifies intrusions by looking for these patterns in the data being analyzed. The accuracy of such a system depends on its signature database [10]. The main limitation of this approach is that it only looks for the known weaknesses and may not care about detecting unknown future intrusions.

The problem of false positives cause many commercial IDS offerings to focus on misuse detection

3) The response manager: will only act when inaccuracies (possible intrusion attacks) are found on the system, by informing someone or something in the form of a response.

IV. Genetic Algorithm

Genetic algorithms are a family of computational models inspired by natural evolution. It is based on Darwinian's principle of evolution and survival of fittest to optimize a population of candidate solutions towards a predefined fitness [5].

GA were originally invented by John Holland in the 1960s.They were developed by Holland and his students and colleagues at the University of Michigan in the 1960s and the 1970s [11].

GA uses an evolution and natural selection that uses a chromosome-like data structure and evolve the chromosomes using selection, recombination and mutation operators [5]. The process usually begins with randomly generated population of chromosomes, which represent all possible solution of a problem. An evaluation function is used to calculate the goodness of each chromosome according to the desired solution; this function is known as "Fitness Function".

Three factors will have vital impact on the effectiveness of the algorithm and also of the applications [6]. They are:

a) The fitness function;

b) The representation of individuals

c) The GA parameters.

The determination of these factors often depends on applications and/or implementation.

V. Genetic Algorithms Applied On IDS

Our goal is to implement a parser engine for an intrusion detection system based GA; the latter consists of two modules which each operate at a different stage. In training step, a set of rules is generated from the audit data using GA. In the step of online intrusion detection, the generated rules are used to classify the incoming network connections in real time.

A. NSL-KDD dataset:

NSL-KDD is a data set suggested to solve some of the inherent problems of the KDD'99 data set [12]. it can be applied as an effective benchmark data set to help researchers compare different intrusion detection methods. Furthermore, the number of records in the NSL-KDD train and test sets are reasonable. This advantage makes it affordable to run the experiments on the complete set without the need to randomly select a small portion. Consequently, evaluation results of different research work will be consistent and comparable [1].

Due to following reasons, NSL-KDD has become more popular dataset than KDD cup 99 dataset for intrusion detection purpose:

- Redundant records from the training set are eliminated.
- Duplicate records from the test set are removed to improve the intrusion detection performance.
- Use of NSL-KDD dataset for classification gives an accurate evaluation of different learning techniques.
- It is affordable to use NSL-KDD dataset for experiment purpose as it consists of reasonable numbers of instances both in the training set and testing set.

B. Data Set Pre-Processing

Pre-processing of original NSL -KDD dataset is necessary to make it as a suitable input for our GA The training dataset of NSL-KDD consist of approximately 4,900,000 single connection instances. Each connection instance contains 41 features such as the duration of the connection, the protocol type, etc.

Several attributes of a network connection can be chosen. In our approach, seven of these attributes were selected from the audit data network. TABLE I shows their characteristics and formats.

In TABLE II we present the proportion of imported NSL-KDD data which was used in our experiment.

Our selected records represent the major part of the NSL-KDD data set. The immediate observation is that the training data taken approximately 90% of total training of the NSL-KDD dataset, and the test data reach 70% of the total NSL-KDD test. This proportion is so high that rarely used in literature for the validation of algorithms and approaches in the field of IDS. Our motivation was to build the most reliable for the detection models.

C. Initial Population:

The choice of the initial population is important because it can make more rapid convergence to the global optimum. In case it does not know anything of the problem, it is essential that the initial population be distributed over the entire area of research.

Our contribution was to propose an initial population of each type of attack. The size of the population was in equity, after testing, the size is set to 80 attacks per type; each type of attacks is randomly generated from the search space which is the NSL training data set, represented in TABLE II.

D. Chromosome Representation

The chromosome was designed on the basis of relevant attributes attacks [duration, protocol, service, flag, src_byte, dst_bayte, attackname].

Example: 0. TCP.-1, SF, -1, -1 , guess_passwd

(-1) indicates that any value may be used.

(SF) Special Frame flag

(TCP) Transmission Control Protocol

TABLE I. CHROMOSOME REPRESENTATION

Nom	Format	Nombre de gènes
Duration	Int	**1**
Protocol	Varchar(5)	**1**
Service	Varchar(10)	**1**
Flag	Varchar(10)	**1**
src_bytes	Int	**1**
dst_bytes	Int	**1**
Attack_name	Varchar(20)	**1**

E. Evolotion Process

Evolution is an iterative process conditioned by a maximum number of generations. The election is performed in two stages: the first guarantee the existence of rule. The existence of a rule is defined by the constraint "I have a fitness, I exist". The final election is made on the rules elected in the first election. Elitist rules are selected on the basis of the best value of fitness. In our study the best fitness is quantified by the constraint finess_elit> = 0.60.

Applying the constraint (fitness> 0.6) supported by a new constraint (count (-1) in the rule <= 3). These two constraints are the guide of the algorithm, as follows:

- (fitness> 0.6) favors the detection.
- (count_rule (-1) <= 3) promotes the reduction of false positive.

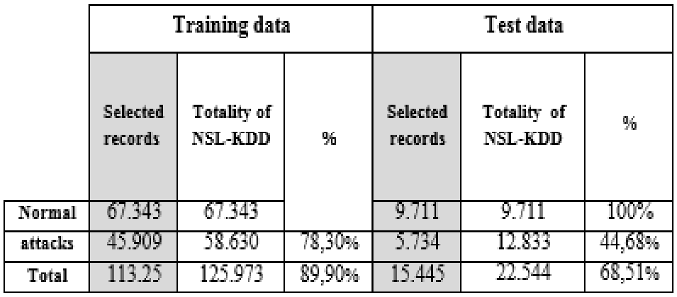
F. Fitness Function:

To determine the aptitude of a rule, the functions support-confidence [12] are used. If a rule is represented as: if A then B, therefore the accuracy (Fitness Function) is determined using the following equations:

Support = |A and B|/N
Confidence =|A and B|/|A|
Fitness=w1*support +w2 * confidence

Such as N is the total number of connections in audits data, | A | is the number of network connections, which corresponds to state A (condition), and | A and B | is the number of network connections corresponding to the rule If A then B. The weight w1 and w2 are used to control the balance between the two terms (support, confidence), in our case we take w1 = 0.2 and w2 = 0.8.

TABLE II. PROPORTION OF SELECTED NSL-KDD DATASET



	Training data			Test data		
	Selected records	Totality of NSL-KDD	%	Selected records	Totality of NSL-KDD	%
Normal	67.343	67.343		9.711	9.711	100%
attacks	45.909	58.630	78,30%	5.734	12.833	44,68%
Total	113.25	125.973	89,90%	15.445	22.544	68,51%

VI. DETECTION ALGORITHM ARCHITECTURE:

Fig.1 shows the architecture of our IDS model. We need to collect enough historical data that includes both normal and anomalous network connections. This data set is generated randomly (80 rules for each attack type) then analyzed by the network sniffers and results are fed into GA for fitness evaluation. Then the GA is executed and the rule set is generated. These rules are stored in a database to be used by the IDS.

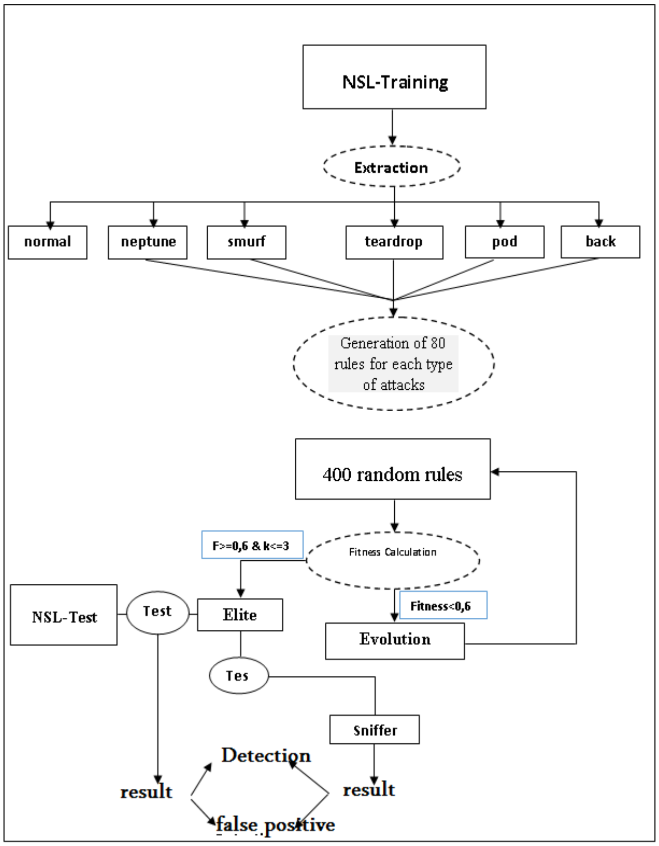


Fig. 1. Architecture of applying GA into IDS

VII. Implementation and Testing

After describing the theoretical foundations of our mechanism for the intrusion detection, we discuss the implementation aspects in order to validate our approach.

To implement this system, we used the Java language in the NetBeans environment, and to store the data we used the MySQL DBMS.

A. Analysis and Discussion

At this stage we are to validate the performance of our proposed detection model using genetic algorithm. The experimental results are so high, they can be described by the percentage 99.74% which makes even more surprising is that the false positive does not exceed 4% (3.74%) (Fig.2).

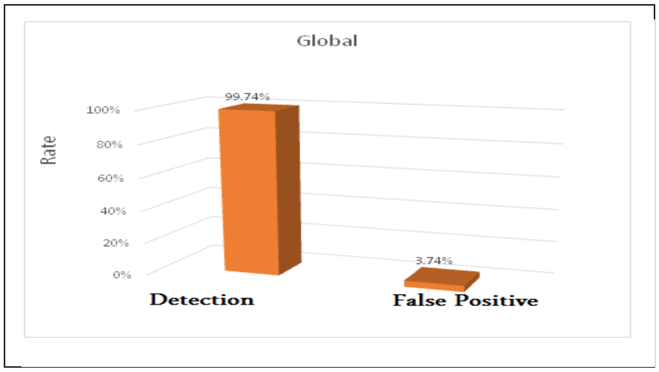


Fig. 2. Result of global detection

Analysis of experimental results for each type of attacks

For the three types of attacks (neptune, smurf, teardrop) the detection reaches 100% with a false positive [1.52%, 0.34, 1.72%], we note that it is so low. This confirms that the algorithm has perfectly designed the model of such attacks. For attack "back" type, detection is more than 97% with zero false positive (0%). the attack has been well modeled by the algorithm.

For the attack "pod" detection is around 88% still with a very low false positive (0.15%). Once again the algorithm shows its efficacy. As illustrated in Fig.3.

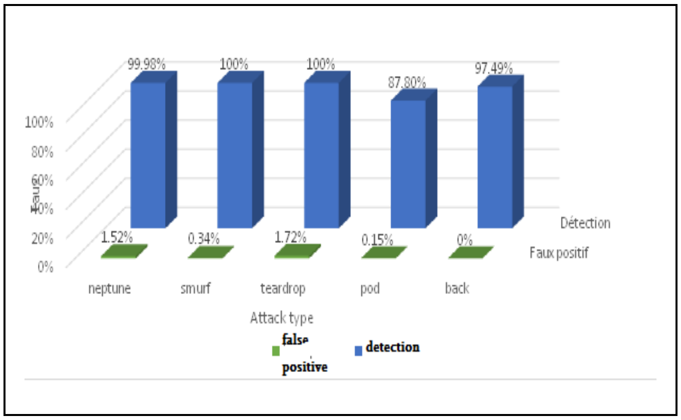


Fig. 3. Result of detection for each attack

B. The Influence of the Generation Factor on the Performance of the Algorithm

From the curve, the detection reaches is more than 15% without evolution (just with the rules chosen randomly), and a false positive of 2.21%. The evolution is launched during a period (5 generations), we note that the detection rate is accelerating at each period and stabilized at 99.73% in the 4th period (20 generations), while the false positive stabilizes in the second period (10 generations). As figured out in Fig.4.

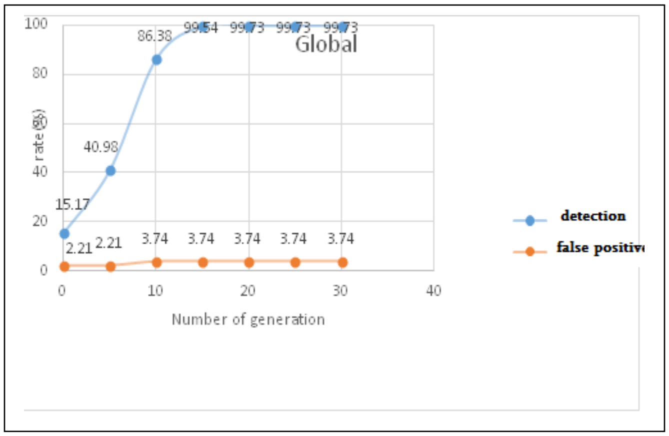


Fig. 4. The influence of the generation factor

VIII. Conclusion

In our research we are interested in intrusion detection based on genetic algorithms to improve the search time in the audit data without losing the performance of the system.

Satisfactory results are produced, in terms of very high detection rate (99%), reinforced by a low rate of false positives (3%). The results are obtained after several improvements of the approach used, such as the choice of the initial population for each type of attacks.

DARPA data set for intrusion detection is still the best corpus to the training and test phase, but many new protocols have been developed, and many types of attacks have been produced, hence the need for enrich this dataset.

REFERENCES

[1] NSL-KDD data set, available on : http://nsl.cs.unb.ca/NSL-KDD/ , September 2013

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

[3] S. Bridges and R. Vaughn, “Fuzzy Data Mining And Genetic Algorithms Applied To Intrusion Detection,” Proceedings of 12th Annual Canadian Information Technology Security Symposium, 2000.

[4] 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.

[5] W. Li, “A Genetic Algorithm Approach to Network Intrusion Detection,” SANS Institute, USA, 2004.

[6] R. H. Gong, M. Zulkernine, and P. Abolmaesumi, “A Software Implementation of a Genetic Algorithm Based Approach to Network Intrusion Detection,” IEEE, University Kingston, Ontario, Canada, 2005.

[7] A. Goyal and C. Kumar, “GA-NIDS: A Genetic Algorithm based Network Intrusion Detection System”, 2008.

[8] R. G. Bace, “Intrusion Detection”, Macmillan Technical Publishing. 2000.

[9] S. Sonawane, S. Pardeshi and G. Prasad, “survey on intrusion detection techniques”, World Journal of Science and Technology, 2(3):127-133, 2012.

[10] A. Kartit, A. Saidi, F. Bezzazi, and A. Radi, "A New Approach to Intrusion Detection System". Journal of Theoretical and Applied Information Technology, vol. 36, pages 56-68, 2012.

[11] M. Mitchell, "An Introduction to Genetic Algorithms", MIT Press, ISBN 0-262-13316-4 (HB), England, 1998.

[12] KDD cup 1999 available on:
http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html,October 2013.