A Biogeography Inspired Approach for Security Audit Trail Analysis

M. Daoudi, A. Boukra, M. Ahmed-Nacer
Faculty of Electronics and Computer Science
Laboratory LSI, USTHB
BP 32 16111 El Alia
Bab-Ezzouar, Algiers
Algeria
{dmfinfo, amboukra}@yahoo.fr, anacer@cerist.dz

ABSTRACT: *The study of effective methods for intrusion detection in audit trail logs is an important part of the vast effort to improve Intrusion Detection Systems that constitute one of the primary approach in the problem of computer security. Different methods have been proposed including: Neural Networks, Immune Systems, and Genetic Algorithms. Security Audit trail Analysis can be accomplished by searching audit trail logs of user activities for known attacks. The later problem is NPComplete. Metaheuristics offer an alternative to solve this type of problem. In this paper, we propose to use Biogeography Based Optimization algorithm (BBO) as analysis engine. BBO is a population-based evolutionary algorithm well suited for constrained optimization problems. Experimental results for simulated attacks are reported. The effectiveness of the approach is evaluated by its ability to make correct predictions. It has proven effective and capable of producing a reliable method for intrusion detection.*

Keywords: Biogeography Based Optimization, Genetic Algorithms, Evolutionary Algorithm, Intrusion detection, Security audit

Received: 12 August 2011, Revised 19 September 2011, Accepted 25 September 2011

1. Introduction

Intrusion Detection System (IDS) is one of the primary approaches to the crucial problem of computer security in order to preserve the confidentiality, integrity and availability of data stored in computers [1], [2]. The study of effective methods for intrusion detection in an audit file is an important part of the vast effort to improve the security of computer systems. Different methods, like Neural Networks, Immune Systems, and Genetic Algorithms have been developed. Intrusion detection by security audit trail analysis can be performed by searching audit trail logs of user activities for predefined attacks. Because such search is NP-Complete [3], heuristic methods will need to be used as databases of events and attacks grow.

This paper presents a method to perform misuse detection using BBO algorithm [4], under the guidelines of previous work [5], [6] and extended in [7] - [9]. BBO is a population-based evolutionary algorithm (EA) based on the mathematics of biogeography which is the study of the geographical distribution of biological organisms. BBO is considered as an analytical engine that performs misuse detection. The effectiveness of our approach is evaluated by its ability to make correct predictions. Different measures are used in our numerical evaluation.

The rest of the paper is as follows: Section 2 outlines Intrusion Detection Systems. A formalization of Security audit trail analysis problem is given in section 3. BBO algorithm is presented in section 4. Our contribution using BBO for misuse detection is presented in section 5. Experimental results are reported in Section 6. Section 7 is conclusions.

2. Intrusion detection systems

An intrusion is defined as a series of actions that attempt to compromise the integrity, confidentiality or availability, or to bypass the security mechanisms of a computer or network. Any operation undertaken on the computer system results in a sequence of actions performed by the system called system activities. A system activity occurring at a point of time is called event. These events are recorded chronologically in a log file.

Intrusion Detection Systems usually process log records received from the operating system for a specific period of time in order to have a complete set of user activity and then perform analysis of the current activity [10], [11]. According to Intrusion Detection Working Group of IETF [12], IDSs include three vital functional elements: information source, analysis engine and response component. Five concepts are defined for their classification: the detection method, the behavior on detection, the audit source location, the detection paradigm and the usage frequency. The detection method is one of the principal characters of classification. When the IDS uses information about the normal behavior of the system it monitors (anomaly detection or behavioral approach), we qualify it as behavior-based. When the IDS uses information about the attacks (misuse detection or scenario approach), we qualify it as knowledge-based. However, both approaches may be complementary [13].

The behavioral approach was first proposed approach, introduced by Anderson [14], and extended by Denning [15]. It is to define a profile of normal activity of a user, and consider the significant deviations of the current activity of users, compared with normal patterns of behavior, as an anomaly. The techniques developed in behavioral approach include the expert systems, statistical models [16], and artificial immune systems [17], [18]. Other techniques like Bayesian parameter estimation [19] and clustering [20]-[23], Genetic Algorithms [24], [25] are also used.

In Misuse detection, IDS processes log records from the operating system for a specific period of time in order to have a complete set of user activity [16]. After that the IDS performs analysis of the current activity, using a rule base system, statistics, or a corresponding heuristic, in order to determine the possible occurrence of intrusion. The misuse mechanism [6]-[9] uses a predefined matrix of intrusion patterns, so the system knows in advance the appearance of the misuse and/or abuse. An intrusion can be specified by an array of activities to check, where each entry specifies the number of activities of a specific type that should occur in order to have an intrusion. Likewise, the results of the user records gathered can be seen as an array, where each entry specifies the total number of activities of that specific type performed by a user. If an intrusion array pattern is such that each entry of it is less or equal than each entry of the user's activity, then, it is possible that intrusion H has occurred. However, looking at some possible intrusions together, it is possible that one or several can occur, but not all together, because adding each corresponding entry, some results could be greater than the corresponding entry of the user activity vector. This is called a violation of the constraint [8]. Neural networks have been extensively used to detect both misuse and anomalous patterns [26]-[28].

We investigate in what follows Security Audit Trail Analysis Problem, using an heuristic method which is BBO algorithm. We give in the next section the formulization of the problem, previously used in [5]-[9].

3. Security Audit Trail Analysis Problem

Formally, the Security Audit Trail Analysis Problem can be expressed by the following statement:

Let:

- N_e the number of type of audit events
- N_a the number of potential known attacks
- AE an N_e x N_a attack-events matrix which gives the set events generated by each attack. $AE_{i\,j}$ is the number of events of type i generated by the attack j ($AEi\ j \geq 0$)
- R a N_a-dimensional weight vector, where R_i ($R_i > 0$) is the weight associated with the attack i (R_i is proportional to the inherent risk in attack scenario i)

- *Oa* N_e-dimensional vector where O_i is the number of events of type i present in the audit trail (O is the observed audit vector)
- *Ha* N_a-dimensional hypothesis vector, where H_j = 1 if attack i is present according to the hypothesis and H_j = 0 otherwise (H describes a particular attack subset).

To explain the data contained in the audit trail (i.e. O) by the occurrence of one or more attack, we have to find the H vector which maximizes the product R.H (it is the pessimistic approach: finding H so that the risk is the greatest), subject to the constraint (AE.H)i $\leq$ Oi , 1$\leq$ i $\leq$ Na (Figure 1).

Because finding H vector is NP-Complete, the application of classical algorithm is impossible where N_a equals to several hundreds. The heuristic approach that we have chosen to solve that NP-complete problem is the following: a hypothesis is made (e.g. among the set of possible attacks, attacks i, j and k are present in the trail), the realism of the hypothesis is evaluated and, according to this evaluation, an improved hypothesis is tried, until a solution is found.

In order to evaluate a hypothesis corresponding to a particular subset of present attack, we count the occurrence of events of each type generated by all the attacks of the hypothesis. If these numbers are less than or equal to the number of events recorded in the trail, then the hypothesis is realistic.

To derive a new hypothesis based on the past hypothesis, we propose to use BBO algorithm we present in the next section.

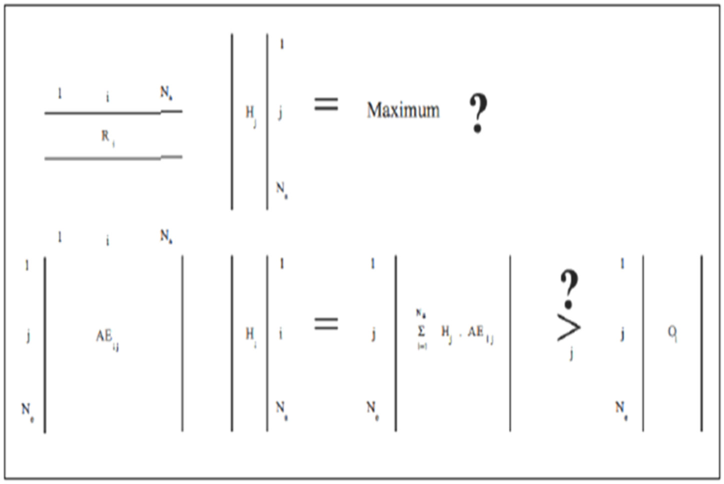

Figure 1. Security Audit Trail Analysis Problem

4. Biogeography Based Optimization overview

Biogeography studies the geographical distribution of biological organisms. It is due to the work of Alfred Wallace [29] and Charles Darwin [30] in the 19th century. This work had a descriptive and historical aspect. In 1960, Robert MacArthur and Edward Wilson have developed a mathematical model for the biogeography [31]. They were interested in the distribution of habitat species. A habitat represents any living space isolated from other spaces. The mathematical model that was developed, describes how some species migrate from one habitat to another. Habitats that are favorable to the residence of biological species are called high *habitat suitability index* (HSI). The parameters influencing the HSI may be rainfall, crop diversity, diversity of terrain ... etc. The variables describing the habitability are called *suitability index variables* (SIV).

Habitats are characterized by the following:

- A habitat with a high HSI tends to have a high number of species, while those with a low HSI tends to have fewer species.
- The habitats with high HSI are characterized by a high rate of emigration and low rate of immigration because they are saturated with species.
- The habitats with low HSI have a high immigration rate and a low emigration rate.

Immigration should result in the modification of the HSI of the habitat. If the HSI habitat remains too long without improving, the species that live there tend to disappear.

Dan Simon introduced in 2008 a metaheuristic based on biogeography [4]. It uses the following analogy:

- A solution is analogous to a habitat
- The quality of the solution (fitness) is analogous to the HSI.
- The variables defining the solution are analogous to SIVs.
- A good solution is analogous to a habitat with a high HSI, and thus with a high number of species, a high rate of emigration and a low immigration rate.
- A bad solution is analogous to a habitat with a low HSI, a low number of species, a low emigration rate and a high immigration rate.
- A good solution tends to share characteristics with a bad solution to improve it (migration of SIVs). This is analogous to the migration of species between habitats. Sharing characteristics does not involve change in the characteristics of good solutions, because migration deals only with a sample of species, so that it does not affect the habitat.
- The bad solutions accept the characteristics of good solutions in order to improve their quality. This is analogous to the bad habitat that accepts immigration of species from other habitats.

Figure 2 illustrates a model of species abundance in a single habitat. It shows the immigration and emigration curves as straight a line (a simple model) which gives us a general description of the process of immigration and emigration:

- When the number of species increases, the rate of immigration (l) decreases and the emigration rate (ì) increases. This characterizes a good solution.
- When the number of species decreases the rate of immigration (l) increases and the emigration rate (ì) decreases. This makes a bad solution.

Improving the population is the way to solve problems in heuristic algorithms. The method to generate the next generation in BBO is by immigrating solution features to other habitats, and receiving solution features by emigration from other habitats. As described in the migration algorithm given below, suppose we have a population of candidate solutions to a problem, represented by vectors (Habitats H_i , i = 1…n). Each element of the vector is considered as an SIV value. In the migration process, the characteristics of good solutions replace the worst ones when using the immigration rate λ (1) and the emigration rate μ (2), according to a probability of change P_{mod}.

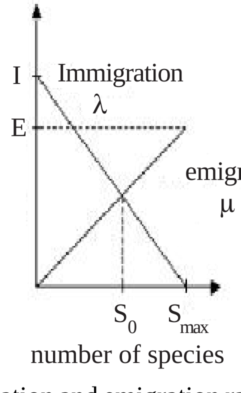

Figure 2. Evolution of immigration and emigration rates with the number of species

$$\lambda_k = I (1 - \frac{k}{N} \tag{1}$$

$$\mu_k = E * \frac{K}{N} \tag{2}$$

Where:

- λ_k is the immigration rate in a habitat with *k* species
- μ_k is the emigration rate in a habitat with *k* species
- E is the maximum rate of emigration
- I is the maximum rate of immigration
- N the maximum number of species
- P_{mod} is a value set by the user

In the migration process (algorithm 1), when a solution is selected to be changed, we use the rate of immigration λ to decide whether an SIV will be changed. In this case, we use the emigration rate μ to decide which good solution in the habitat will migrate its SIV.

```
Select $H_i$ with a probability $\alpha\lambda_i$
   If $H_i$ is selected
    Then for  j=1 to n
      Select $H_j$ with a probability $\alpha\mu_j$
        If $H_j$ is selected
           Then replace the SIV in $H_i$ with SIV of $H_J$
        End if
    End for
End if
```

Algorithm 1. Migration algorithm

In the mutation process (algorithm 2), mutation is performed for the whole or part of the population in a manner similar to the mutation in genetic algorithms (GAs). Changes in a habitat may arise. These modifications will change the HSI of the habitat. This is modeled in BBO when muting with a certain probability. The rate of change of habitat is given in formula (3) [4].

$$m(s) = \alpha \frac{1 - P_s}{P_{max}} \tag{3}$$

Where:

- m(s) : mutation rate of a habitat with s species.
- α : parameter defined by the user
- P_s : probability of having s species in the habitat (4).
- P_{max} : probability of having maximum species

And :

$$P_k = \begin{cases} \dfrac{\lambda_0\lambda_1..\lambda_{k-1}}{\mu_1\mu_2..\mu_k\,(1+\sum_{l=1}^{n}\frac{\lambda_0\lambda_1..\lambda_{k-1}}{\mu_1\mu_2..\mu_l})} & 1 \le k \le n \\ \dfrac{1}{(1+\sum_{l=1}^{n}\frac{\lambda_0\lambda_1..\lambda_{k-1}}{\mu_1\mu_2..\mu_l})} & k = 0 \end{cases} \tag{4}$$

Remark 1. Mutated habitats are those having a low HSI. This mutation introduces diversity and encourages poor habitats to improve.

Algorithm 2. Mutation algorithm

```
For j=1 $\alpha$ m
  use $l_i$ and $\mu_i$ to compute $P_i$
  Select SIV in $H_i$(j) with probability $\alpha P_i$
  If $H_i$(J) is selected
    Then replace $H_i$(j) with a random SIV
  End if
End for
```

In evolutionary strategies, global recombination is used to create new solutions, while BBO migration is used to change existing solutions. Global recombination in evolutionary strategy is a reproductive process, while migration in BBO is an adaptive process; it is used to modify existing habitats. As with other population-based optimization algorithms, some sort of elitism is typically incorporated in order to retain the best solutions in the population. This prevents the best solutions from being corrupted by immigration.

5. Biogeography-based Security Audit Trail Analysis Approach

We recall the approach aims to determine if the events generated by a user correspond to known attacks, and to search in the audit trail file for the occurrence of attacks by using a heuristic method (BBO algorithm) because this search is an NP– complete problem. The goal of the heuristic used is to find the hypothesized vector H that maximizes the product R.H, subject to the constraint $(AE.H)_i \leq O_i$, where R is a weight vector that reflects the priorities of the security manager, AE is the attack-events matrix that correlates sets of events with known attacks, $1 \leq i \leq N_a$, and N_a is the number of known attacks.

BBO is based on a stochastic optimization and on a biogeographic relatively simplistic vision. It operates on a set of individuals (islands or habitats) in a population (archipelago).The islands are characterized by a set of SIVs.

To use BBO, we must first find a good encoding for the solutions and then find a fitness function to evaluate them.

A potential solution of the problem to be solved is a vector H, expressed in a form of binary sequence (with a value 1 if the attack i is present in the audit file O and a value 0 otherwise). Consequently, in our case, the coding is straightforward: a habitat is composed with N_a SIV. Each SIV value is 0 (no attack) or 1 (presence of an attack) and N_a is the number of attacks. Each habitat in the archipelago corresponds to a particular H vector (Figure 3).

Further, as specified above, we have to search, among all the possible attack subsets, for the one which presents the greatest risk to the system. This results in the maximization of the product R.H. As BBO algorithm is an optimum search algorithm, finding the maximum of a fitness function, we can easily conclude that in our case this function should be made equal to the product R.H. So we have:

$$F = Max \sum_{1}^{n_a} R_j H_j$$

Where H is a habitat.

However, this selective function ignores the fact we deal with a constrained problem, which implies that some habitats among the 2^{Na} are not realistic (this is the case for some type i of events when: $(AE.H)_i > O_i$).

Solutions which do not satisfy the constraints will be penalized by setting their fitness to 0, migration rate μ to 0, and Immigration rate λ to1. As for the solutions satisfying the constraints, they will be sorted in decreasing order of the objective function. A number of species is associated to each solution. P (population size) species are associated to the best solution, (P-1) species to the next solution, and so on.

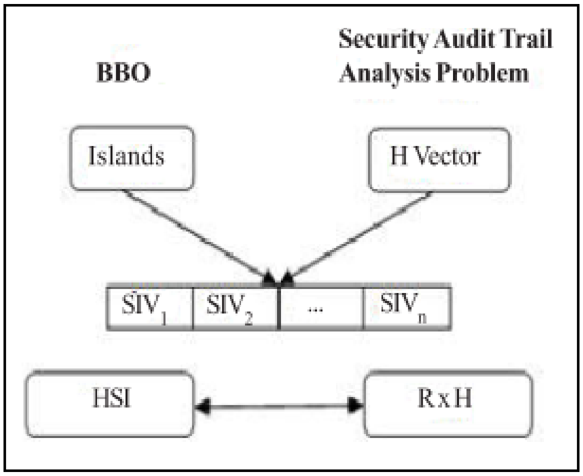

Figure 3. Coding of solutions and the objective function

In BBO algorithm, an initial population of several habitats (binary vector H) is randomly generated. It evolves through a migration (emigration and immigration) and a mutation processes to reach an optimal solution as shown in figure 4.

The migration process in the biogeography mechanism aims to modify one or more SIV, randomly chosen in each selected solution, allowing the exchange of information between the good and bad solutions. Mutation maintains diversity in the population and explores the search space, avoiding the algorithm to converge too quickly to a local optimum. Mutation comes after the migration process. It will be applied to the lower half of the population, with a very low probability (mutation rate: p_m = 0.005), for a great rate can cause destruction of valuable information contained in the solutions. During this process, the characteristics (SIV) of one or more individuals (habitats) of the population will be changed randomly. In other words, one or more characteristics (SIV) are modified to change from one solution to another solution with different value. However, the modified individual will not necessarily be better or worse, but it will bring additional opportunities that could be useful for creating good solutions. During the evolution of the population, it is likely that the best solutions are modified, and therefore lost after the migration process and mutation. To avoid such a situation, an elitism operator is adopted. It copies the p best individuals in the new generation (the value of p has to be fixed by simulation).

The process of mutation and migration are BBO mechanisms used to derive a new hypothesis based on the past hypothesis, exploring the search space, trying to improve the best solution in terms of its fitness value (that is the incurred risk). Changing an SIV in a H vector by means of migration and mutation processes causes an attack to appear or disappear. Algorithm 3 shows the different steps followed when analyzing the security audit trail using BBO algorithm.

6. Experimental results

Simulations were carried out on a 1.86 GHz Intel Celeron CPU 540 PC with 1GB memory. The results reported here concern tests performed on a (400 x 200)-Attack-Events matrix randomly generated. In the simulations, the set of attacks really present in the analyzed audit trail have to be known in advance. So, events corresponding to one or more attacks are included in the observed audit vectors. Each experiment performed can be characterized by a 4 – tuple (P, P_m, L, I_a), where P is the population size, P_m the mutation parameter, L the elitism parameter and I_a, the number of attacks actually present in the audit trail. All the following results are averaged over 10 runs performed for each value of the 4-tuple.

We are interested in analyzing the influence of each parameter, and more specifically on the quality of the detection. We define the ratios TNR, TPR, FPR, FNR, Accuracy and Precision as follows [33]:

- True negative rate (TNR): TN / (TN+FP)
- True positive rate (TPR) : TP / (TP+FN)
- False positive rate (FPR) : FP / (TN+FP)
- False negative rate (FNR): FN / (TP+FN)
- Accuracy: (TN+TP) / (TN+TP+FN+FP)
- Precision: TP / (TP+FP)

Algorithm 3. Security audit trail analysis process

```
Input: An instance with a (Ne x Na) - matrix AE and a Na - vector O;
Initialization:
        • Maximum number of generations (Nb_Generations)
        • population size (N)
        • mutation rate
        • Elitism parameter
        • probability of changing an individual = 1= Pmod
End of initialization
Begin
        • Generate a random population of islands (N subsets of attacks)
        • Test the validity of solutions generated
        • Evaluate the fitness (Calculation of the product R.H)
        • Sort the population from best to worst
        For I = 1 to Nb_Generations do
        Begin
                • Save the elected representatives of the current generation to the next generation
                • Calculate the number of species
                • Calculate emigration rate (α) and immigration rate (β)
                • Apply the Migration Process
                • Apply the Mutation Process
                • Test the validity of the solutions generated
                • Assessment of fitness (product R.H)
                • Sort population
                • Replace the bad solutions of the present generation by the
                  elected officials of the previous generation
        End
Output: A vector H which is the best hypothesis H found that maximizes the product
    RxH and satisfies the constraints.
End
```

Remark 2. We note that our proposed approach is subject to limitations such that:

- Attacks characterized by event absence are not taken into account.
- The multiple realization of a particular attack is not detected with a binary coding.
- Only independent attacks are considered.
- The events chronology is not taken into account

Where True negatives (TN) as well as true positives (TP) correspond to correct intrusion detection; that is, events are successfully labeled as normal and attacks, respectively. False positives (FP) refer to normal events being predicted as attacks; false negatives (FN) are attack events incorrectly predicted as normal events.

The tests are performed using BBO - default parameter values given in [4] (P = 50, Pm = 0.005, L = 2) and 20 injected attacks. Figure 4 shows the evolution of TPR and FPR versus the generation number (i.e. versus time). We observe that there is a good discrimination between present and absent attacks. Indeed, in the detection process, the initial population is randomly generated, so for generation 0, we have TPR HH 0.5 et FPR HH 0.5, and should have after a certain number of generations all the present attacks detected and no absent attack detected (TPR = 1 and FPR = 0). We note that the number of injected attacks has no influence on these results.

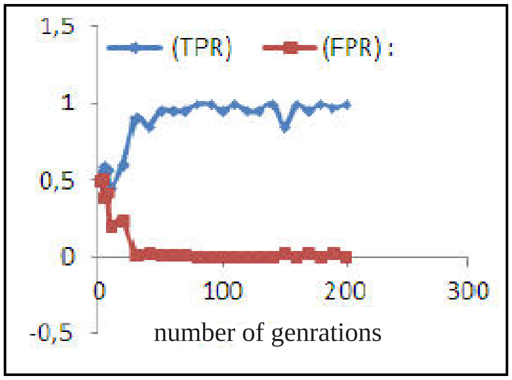

Figure 4. Evolution of TPR and FPR vs. generation

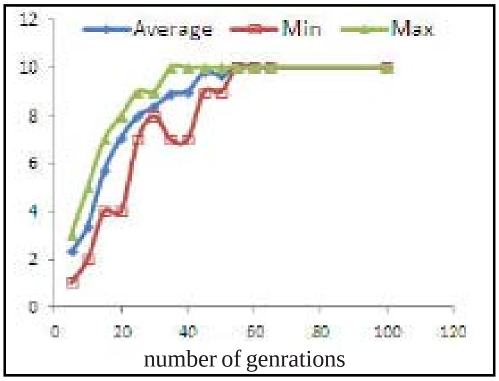

Figure 5. Average min, max and average fitness for 10 runs vs. generation

In order to appreciate the convergence of the population fitness and the time needed for detection, we compute, for each generation the minimum, maximum, and average whole population fitness values. Figure 5 shows that the average min, max and average fitness converge quickly on the optimum (after about 50 generations). The number of injected attacks has no influence on the results. However, it seems clear that the number of generations has to increase to keep the detection quality at the same level.

We observe in figures 6 and 7, that the detection's quality stabilizes (TPR = 1 and FPR =1) as the population size grows (P ≥ 80).

Figures 8 and 9 show the influence of the mutation rate P_m on the quality of the detection. Bad results are obtained for Pm ≥ 0.01. However, we observe that the system does not detect the attacks when the mutation process is not used. The mutation process is important for better performance of the system.

In figure 10, we observe that elitism value has not big influence on the quality of the detection

When tests are performed on smaller AE- matrix, the quality of the detection is good (TPR =1 and FPR = 0). On the benchmark used in [5], all the present attacks are detected and no absent attack is detected (TPR = 1 and FPR = 0) and the average min, max and average fitness converge quickly on the optimum (after about 18 generations). These results are better than obtained in [5] using Genetic algorithm (TPR = 0.91 and FPR = 0.027).

When tests are performed on smaller AE- matrix, the quality of the detection is good (TPR =1 and FPR = 0). On the benchmark used in [5], all the present attacks are detected and no absent attack is detected (TPR = 1 and FPR=0) and the average min, max and average fitness converge quickly on the optimum (after about 18 generations). These results are better than obtained in [5] using Genetic algorithm (TPR = 0.91 and FPR = 0.027).

7. Conclusion

In this paper, we proposed a new biogeography-based approach for a detection intrusion system. This later is based on a simplified analysis of the audit file. Experiments were performed to reach three main goals that are:

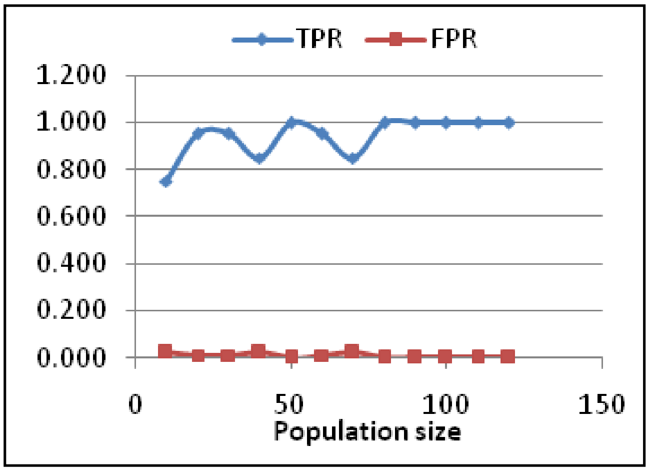

Figure 6. TPR and FPR vs. population size

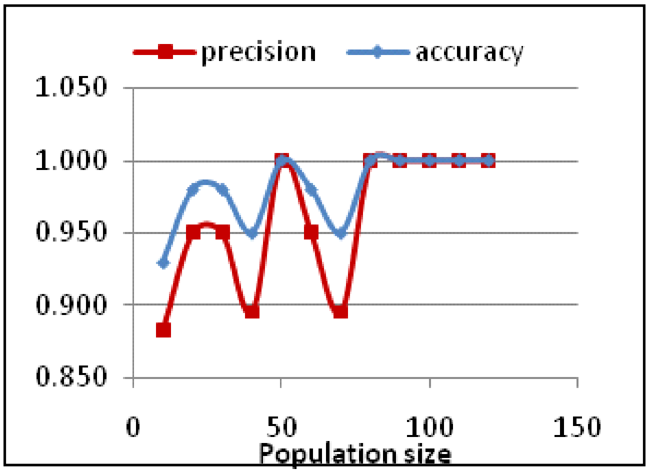

Figure 7. Precision and Accuracy vs Population size

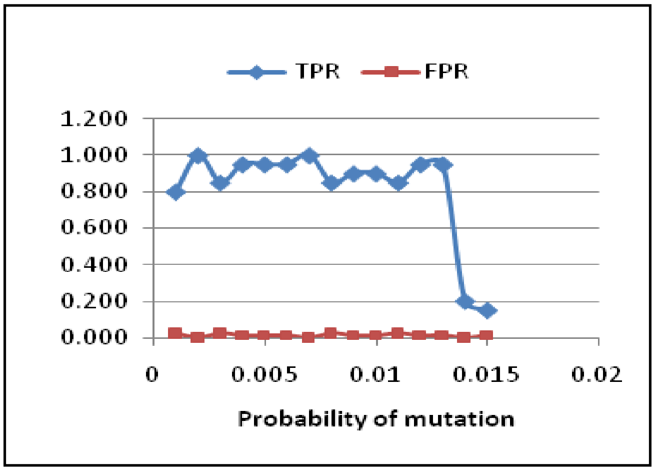

Figure 8. TPR and FPR vs. probability of mutation

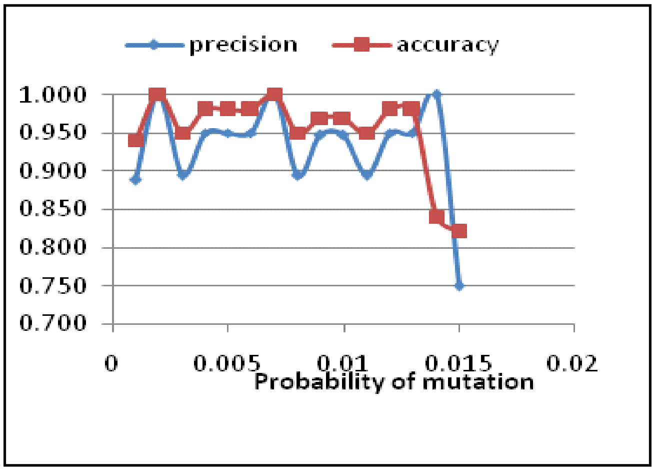

Figure 9. Precision and Accuracy vs. probability of mutation

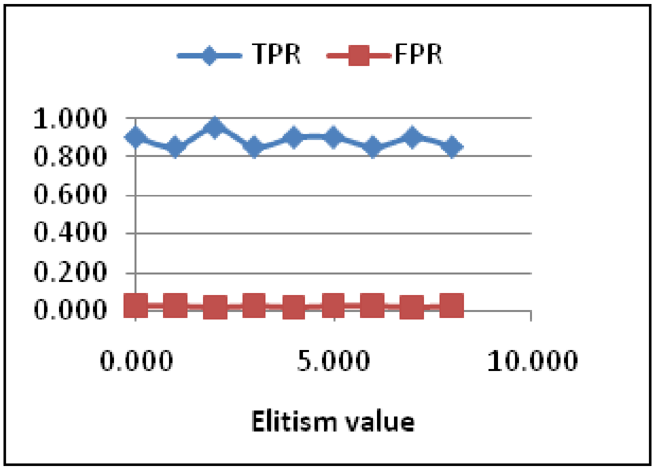

Figure 10. TPR and FPR vs. Elitism value

- The tuning of the metaheuristic's parameters,
- The new approach numerical evaluation consisting in the following measures of performance: True negative rate (TNR), True positive rate (TPR) ,False positive rate (FPR) ,False negative rate (FNR),Accuracy and Precision. The evaluation of these performance measures were performed on large instances that were randomly generated and on a benchmark issued from [5].

The results showed that the new approach has good performances. Another important result is the consistency of performance, regardless of the number of attacks actually present in the matrix of audit analysis. This means that if many attacks occur, the detection system performance is not deteriorated. Similarly, if no attack is launched, it does not detect anything. The processing time is very satisfactory. This allows us to consider our new BBO-based approach for intrusion detection system to be efficient and reliable.

References

[1] Cole, E., Krutz, R.L., Conley, J. (2005). Network Security Bible, Wiley Publishing.

[2] Tjaden, B. C (2004). Fundamentals of Secure Computer Systems. Franklin and Beedle & Associates.

[3] Mé, L. (1994). Audit de Sécurité par Algorithmes Génétiques. Thèse de Doctorat de l'Institut de Formation Supérieure en Informatique et Communication DE Rennes.

[4] Simon, D. (2008). Biogeography-Based Optimization. *IEEE Trans. On Evol. Comput*. 12 (6)712-713.

[5] Mé, L. (1995). Un Algorithme Génétique pour détecter des Intrusions dans un Système Informatique.VALGO, 95 (1) 68-78

[6] Mé, L.(1998). GASSATA, a Genetic Algorithm as an Alternative Tool for Security Audit Trails Analysis. *In*: Proc. of the 1st International Workshop on the Recent Advances in Intrusion Detection (RAID 98). Louvain-la-Neuve, Belgium, p. 14–16.

[7] Diaz-Gomez, P. A., Hougen, D. F. (2005). Improved Off-Line Intrusion Detection using a Genetic Algorithm. *In*: Proc. of the seventh International Conference on Enterprise Information Systems, p. 66-73.

[8] Diaz-Gomez, P. A., Hougen, D. F. (2006). A genetic algorithm approach for doing misuse detection in audit trail files. *In*: Proc. of the (CIC-2006) International Conference on Computing(CIC-2006), p. 329-335.

[9] Ahmim, A., Ghoualmi, N., Kahya, N. (2011). Improved Off-Line Intrusion Detection using a Genetic Algorithm and RMI. *In*: Proc. of the International Journal of Advanced Computer Science and Applications (IJACSA), V 2 (1).

[10] Bace, R., Mell, P. (2001). NIST Special Publication on Intrusion Detection Systems.

[11] Roberto, P., Luigi V, M. (2008). Intrusion Detection Systems.

[12] Debar, H., Dacier, M., Wespi, A. (2000). A Revised Taxonomy for Intrusion Detection Systems. *Annales des Telecommunications*, 55 (7-8).

[13] Tombini, E. (2006). Amélioration du Diagnostic en Détection d'Intrusions: Etude et Application d'une Combinaison de Méthodes Comportementale et par Scénarios. Thèse de Doctorat de l'Institut National des Sciences Appliquées de Rennes.

[14] Anderson, J. (1980). Computer Security Threat Monitoring and Surveillance. Technical Report 79F296400, James, P. Anderson, Co., Fort Washington, PA.

[15] Denning, D. (1987). An Intrusion-Detection Model. *In:* Proc. of the 1986 IEEE Symposium on Security and Privacy, pages 118-131.

[16] Bace, R. G .(2000). Intrusion Detection Systems. Mac Millan Technique Publication, USA.

[17] Haidong, Yang., Jianhua, Guo., Feiqi, Deng. (2011). Collaborative RFID Intriusion Detection with an Artificial Immune System. *Journal of Intelligent Information System*, 36 (1) 1-26

[18] Bankovic, Z., Álvaro, A., de Goyeneche, J. M. (2009). Intrusion Detection in Sensor Networks Using Clustering And Immune systems. Lecture Notes in Computer Science, V. 5788, Intelligent Data Engineering and Automated Learning - IDEAL 2009, p. 408-415.

[19] Cho S. Cha S, SAD. (2004). Web Session Anomaly Detection Based on Parameter Estimation. *Computers & Security*, 23 (4) 265-351.

[20] Xu, B., Zhang, A. (2005). Application of Support Vector Clustering Algorithm to Network Intrusion Detection. *In*: Proc. of the International Conference on Neural Networks and Brain, ICNN&B '05. V. 2, 1036 – 1040.

[21] SH, O., WS, L. (2003). An anomaly Intrusion Detection Method by Clustering Normal User Behavior. Computers & Security, 22 (7) 596-612.

[22] Leon, E., Nasraoui, O., Gomez, J. (2004). Anomaly Detection Based on Unsupervised Niche Clustering with Application to Network Intrusion Detection. *In* Proc. of the IEEE Conference on Evolutionary Computation (CEC), p.. 502-508.

[23] Guan, Y., Ghorbani, A., Belacel, N. (2003). Y-MEANS: a Clustering Method for Intrusion Detection. *In:* Proc. of the Canadian Conference on Electrical and Computer Engineering, p. 1083-1086.

[24] Saniee Abadeh, M., Habibi, J., Lucas, C. (2007). Intrusion Detection Using a Fuzzy Genetics-Based Learning Algorithm. *Journal of Network and Computer Applications,* 414-428.

[25] Ozyer, T., Alhajj, R., Barker, K. (2007). Intrusion Detection by Integrating Boosting Genetic Fuzzy Classifier and Data Mining Criteria for Rule Pre-Screening. *Journal of Network and Computer Applications* , p. 99–113.

[26] Kumar, G.,Kumar, K., Sachdeva, M. (2010). The Use of Artificial Intelligence Based Techniques for Intrusion Detection: A Review. Artificial Intelligence Review, 34 (4) 369-387.

[27] Castellano, M.,Mastronardi, G., Tarricone, G. (2009). Intrusion Detection Using Neural Networks: A Grid Computing Based Data Mining Approach. Lecture Notes in Computer Science, V. 5864, Neural Information Processing, p. 777-785

[28] Mukkamala, S., Xu, D., H. Sung, A. (2006). Intrusion Detection Based on Behavior Mining and Machine Learning Techniques. Lecture Notes in Computer Science, 2006, V. 4031. Advances in Applied Artificial Intelligence, p.619-628

[29]Wallace, A. (2005).The Geographical Distribution of Animals (Two Volumes). Boston, MA: Adamant Media Corporation.

[30] Darwin, C. (1995). The Origin of Species. New York: Gramercy.

[31] MacArthur, R., Wilson, E. (1967). The Theory of Biogeography. Princeton, NJ: Princeton Univ. Press.

[32] Wu, S., Banzhaf, W. (2008). The Use of Computational Intelligence in Intrusion Detection Systems: A review. Computer Science Department, Memorial University of Newfoundland, St John’s, NL A1B 3X5, Canada.

Author biographies

Mourad Daoudi (Tizi-Ouzou, August 1954) is lecturer at USTHB (Algier’s University) and a researcher at the LSI laboratory (USTHB). He was Bachelor of Science in Mathematics (D.E.S in Analysis and Geometry) in 1976. He received the MSc degree in Operations Research at the university of Southampton (England) in 1980 and the Magister in Operations Research at USTHB University in Algiers (1985). He worked for several years in different areas. His current research interests include the use of metaheuristics in various domains involving combinatorial optimization problems.

Abdelmadjid Boukra (Algiers, September 1962) is lecturer at USTHB (Algier’s University). He was engineer in computer science in 1986. He received the Magister (PhD) in 1989 from USTHB (Algiers’sUniversity). He obtained his (Doctorat d’Etat) in 2007 from USTHB (Algiers’sUniversity). He is a Research Master in the LSI laboratory (USTHB). His current research interests include data warehousing, metaheuristics and intrusion detection.

Mohamed Ahmed-Nacer (Algiers, November 1957) is a full Professor at USTHB (Algier’s University). He received the PhD degree in Computer Science from the Polytechnic National Institute (INPG) of Grenoble, France in 1994 and received his research habilitation (Doctorat d’Etat) at Algier’s University (USTHB) on software engineering in 1997. He is a research director and is in charge of the software engineering team at the Computer Engineering Laboratory of USTHB. He published extensively and his current research interests include process modeling, software architecture based components, service web development and software databases.