Analysis of an Off-Line Intrusion Detection System: A Case Study in Multi-Objective Genetic Algorithms

Pedro A. Diaz-Gomez*
Ingenieria de Sistemas, Universidad El Bosque
Bogota, Colombia
pdiazg@ou.edu

Dean F. Hougen
Robotics, Evolution, Adaptation and Learning Laboratory (REAL Lab)
School of Computer Science
University of Oklahoma, OK, USA
hougen@ou.edu

Abstract

A primary approach to computer security is the Intrusion Detection System (IDS). Off-line intrusion detection can be accomplished by searching audit trail logs of user activities for matches to patterns of events required for known attacks. Because such search is NP-complete, heuristic methods will need to be employed as databases of events and attacks grow. Genetic Algorithms (GAs) can provide appropriate heuristic search methods. However, balancing the need to detect all possible attacks in an audit trail with the need to avoid warnings of attacks that do not exist is a challenge, given the scalar fitness values required by GAs. A case study of a previously proposed GA-based IDS shows this difficulty with respect to its fitness function and proposes a new method to overcome it. Such analysis can be of benefit to the study of other multi-objective GAs.

Introduction

The need for automated audit trail analysis was outlined a quarter century ago (Anderson 1980) and is still present. This paper presents a case-study of an off-line intrusion detection system that uses GAs to search for matches in the audit trail (Mé 1998). Unfortunately, the parameters for its fitness function cannot be tuned to effectively detect all possible attacks in an audit trail while still avoiding false positives. Our work addresses this shortcoming.

GASSATA

A Genetic Algorithm as an Alternative Tool for Security Audit Trail Analysis (*GASSATA*) was introduced as an off-line intrusion detection system (Mé 1998) with fitness function

$$F(I) = \alpha + \sum_{i=1}^{N_a} W_i \cdot I_i - \beta * T^2 \qquad (1)$$

where I is the hypothesis vector, α maintains $F(I) > 0$ in order to retain diversity in the population (using proportional probability selection), N_a is the number of known attacks, W is the weighted vector that reflects the risk of each attack, β provides a slope for the penalty function, and T is the number of times for which $(AE \cdot I)_i > OV_i$, where *AE* is the attack-event matrix that shows which events are required for each attack, and *OV* is the observed vector of events.

Mé (1998) reports good results with *GASSATA* but our experience has been that the system often generates false positives and negatives (Diaz-Gomez & Hougen 2005).

Analysis of *GASSATA'S* Fitness Function

A genetic algorithm needs a scalar fitness function to work, and it appears natural that the one originally proposed (Mé 1998)—a combination of objectives into a single function using arithmetic operations—is appropriate. There are, however, problems with this approach. The first is that accurate scalar information must be provided on the range of objectives, to avoid one of them dominating the other. The second is the difficulty in determining the appropriate weights when there is not enough information about them. In this case, any optimal point obtained will be a function of the coefficients used to combine the objectives (Coello 1998).

The term $\sum_{i=1}^{N_a} W_i \cdot I_i$ guides the solution to have the maximum number of intrusions. However, this is good only until the correct set of intrusions are found. If more intrusions than that are hypothesized, the problem of false positives occurs. Similarly, the term $\beta * T^2$ decreases the fitness value but various intrusions can require the same event. When this happens, the counting of overestimates is wrong. See Figure 1.

In $\boxed{1}$ we have a first case: A type 5 intrusion was hypothesized, so one is added. This intrusion requires events 6, 7, and 17. For event 6, the hypothesis gives a number of events greater than the number of events that really happened, so one is subtracted. For events 7 and 17 there is no penalty—we observe $30 \leq 76$ and $62 \leq 94$. So, one was added because the attack was hypothesized and one was subtracted because for the 6th entry there were more events hypothesized than really happened.

In $\boxed{2}$ we have a second case: Intrusion type 21 was hypothesized which requires events of types 6 *again*, 17 *again*, and 23. For event types 17 and 23 there is no penalty because $(AE * I)_{17} \leq OV_{17}$ and $(AE * I)_{23} \leq OV_{23}$, and for event 6, there is no penalty either, because the penalty was *already* taken into account for intrusion type 5. In this case,

*Conducting research at the Robotics, Evolution, Adaptation, and Learning Laboratory (REAL Lab), School of Computer Science, University of Oklahoma.

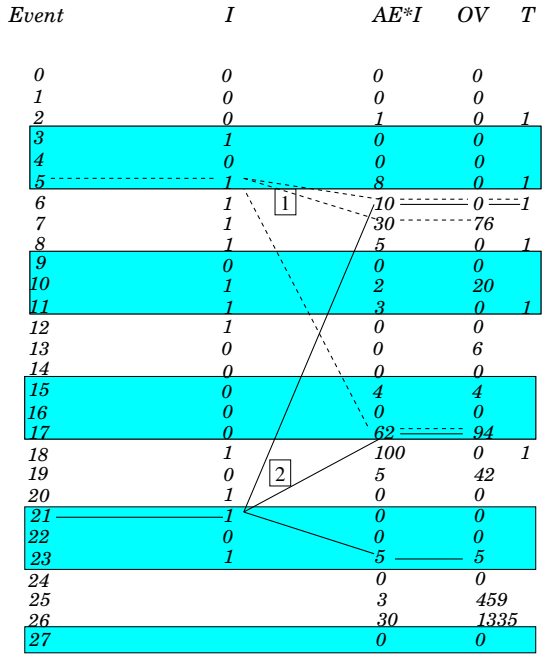

Figure 1: The fitness evaluated as per *GASSATA*.

where there should be a penalty, there is no penalty at all.

Table 1 gives the *AE* matrix, an individual I hypothesized in the last generation, and the counts of overestimates for that individual.

New Fitness Function Proposal

The solution proposed has two parts: (1) remove the term $\sum_{i=1}^{N_a} I_i$, and (2) count overestimates as follows: if two intrusions require the same event each resulting in an overestimate of the number of events hypothesized, then count them twice, and so forth. Call this T'.

With this in mind, the new fitness function suggested is

$$F(I) = N_e - T' \tag{2}$$

Now, the better the hypothesized vector I, the smaller T' is, and of course, $F(I) \rightarrow N_e$, the maximum. To avoid false negatives, we add a mechanism that takes the union of all newly hypothesized attacks that are consistent with the existing aggregate solution set.

The results found with the new fitness function and mechanism are shown in Table 2. As can be seen, with the new method *there are no false positives* and the number of *false negatives decreases dramatically* compared to the results we saw previously (Diaz-Gomez & Hougen 2005). This time 70 runs were performed—10 repetitions each for 7 different cases—and only one time a false negative was present.

Conclusions & Future Work

This paper shows some difficulties in providing accurate values to parameters in the fitness function suggested in *GASSATA* (Mé 1998) and proposes a solution independent of variable parameters making the fitness function to solve this particular problem quite general and independent of the audit trail data. This approach can be generalized to similar multi-objective fitness functions for genetic algorithms. We will compare this approach to other approaches such as replacing proportional probability selection with another selection method, such as tournament selection.

ATTACK #

EVENT #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	I	AE*I	OV	T	T'
0	3																								0	0	0		
1					1																				0	0	0		
2				1																					0	1	0	1	1
3		3																							1	0	0		
4			3																						0	0	0		
5		3							8																1	8	0	1	1
6						5														1		5			1	10	0	1	2
7						30																			1	30	76		
8							5																		1	5	0	1	1
9										3															0	0	0		
10											2														1	2	20		
11												3													1	3	0	1	1
12															10	1									1	0	0		
13																1									0	0	6		
14																1									0	0	0		
15																								4	0	4	4		
16																				1					0	0	0		
17		3				35	5		8	3	2	3			10	3		300		2		5		4	0	62	94		
18																			100						1	100	0	1	1
19								5																	0	5	42		
20														10											1	0	0		
21																				1					1	0	0		
22																									0	0	0		
23																						5			1	5	5		
24																							1			0	0		
25					1																3					3	459		
26													30													30	1335		
27																	50									0	0		

Table 1: Example AE, I, OV, T, and T'.

User	Average Count			Average %		
	False +	False -	Detected	False +	False -	Detected
2051_7	0	0	3	0	0	100
2051_11	0	0	4	0	0	100
2506_15	0	0	4	0	0	100
Zero Vector	10	0	0	0	0	100
One Intrus.	0	0.1	0.9	0	10	90
Two Intrus.	0	0	2	0	0	100
Three Intrus.	0	0	3	0	0	100

Table 2: Results with fitness function $F(I) = N_e - T'$.

References

Anderson, J. P. 1980. Computer security threat monitoring and surveillance. Technical Report 79F296400, James P. Anderson, Co., Fort Washington, PA.

Coello, C. A. C. 1998. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems* 1(3):269–308.

Diaz-Gomez, P., and Hougen, D. 2005. Improved off-line intrusion detection using a genetic algorithm. In *Proceedings of the Seventh International Conference on Enterprise Information Systems*. To appear.

Mé, L. 1998. GASSATA, a genetic algorithm as an alternative tool for security audit trail analysis. In *First International Workshop on the Recent Advances in Intrusion Detection*.