Boundary Expansion of Expert Systems: Incorporating Evolutionary Computation with Intrusion Detection Solutions

Raymond C. Garcia
ray.garcia@gtri.gatech.edu
Georgia Tech Research Institute
Information Technology and Telecommunications Laboratory
Computer Science and Information Technology Division
347 Ferst Drive, Atlanta, Georgia 30332-0832

Dr. James Cannady
j.cannady@ieee.org
School of Computer and Information Sciences
Nova Southeastern University
Fort Lauderdale, Florida 33314

ABSTRACT

The work represented here utilizes evolutionary computation to improve upon intrusion detection techniques. Many intrusion detection techniques incorporate expert systems (e.g., ASAX, IDES, NIDES, DIDS, Hyperview, JiNao). Problems associated with expert systems are in how the rules are defined and matched against potential intruders. Going outside the rule set leaves minimal hope of detection. This work improves upon intrusion detection schemes that utilized expert systems by using an evolution strategy with combinations of attack signatures as individual characteristics. The overall strength is in viewing the rule-matching problem as an optimization problem.

I. INTRODUCTION

Intrusion detection has grown into a multi-million dollar business. Claims and counter-claims are a by-product of the growth in the industry. The one constant in the business of intrusion detection is the overwhelming need for it. Much work has been done in the area of intrusion detection.

The aim of intrusion detection is to detect attacks against computer systems and networks. Based on the intractable nature of providing provably secure information systems, intrusion detection systems are tasked with monitoring system usage. In particular, intrusion detection systems look for attempts in users/parties abuse of privileges or exploitation of security flaws.

II. EVOLUTION STRATEGIES

Evolution Strategies (ESs) are a sub-branch of computational intelligence inspired by phenotypical evolutionary processes (i.e. competition with others and dealing with one's environment [1]). One typical application is in the realm of function optimization. A common failure in most attempts to optimized functions is the avoidance of local extrema. Avoiding local extrema implies the objective of locating global extrema. Global extrema has its greatest chance of being located provided the following conditions are met [2]:

- A global extreme does not occur at a single point with an irrational value, i.e. a horizontal line with its maximum at an irrational point.
- The initial population is sufficiently large and distributed across the range of interest.
- No individual (guess) is initially a "fitter" (more extreme) individual in a local extreme region than others in a global extrema region.

The considerations made for this ES differ significantly from the one proposed in [3-5].

III. KNOWLEDGE-BASED IDS

There are several types of intrusion detection systems. Knowledge-based intrusion detection systems are based on expert systems (containing rules that describe an attack), signature analysis (seeking patterns of data in audit trails generated by the system), petri nets (graphical representations of complex signatures), or state transition analysis (based on attack descriptions as a set of goals and transitions). The use of evolutionary computation can potentially improve upon all knowledge-based intrusion detection systems but knowledge-based intrusion detection systems that are based on expert systems is what this work addresses.

Most current approaches to the process of detecting intrusions utilize some form of rule-based analysis. Rule-Based analysis relies on sets of predefined rules that are provided by an

0-7803-6748-0/01/$10.00 ©2001 IEEE

administrator, automatically created by the system, or both. Expert systems are the most common form of rule-based intrusion detection approaches, [6,7]. The early intrusion detection research efforts realized the inefficiency of any approach that required a manual review of a system audit trail. While the information necessary to identify attacks was believed to be present within the voluminous audit data, an effective review of the material required the use of an automated system. The use of expert system techniques in intrusion detection mechanisms was a significant milestone in the development of effective and practical detection-based information security systems, [8].

An expert system consists of a set of rules that encode the knowledge of a human "expert". These rules are used by the system to make conclusions about the security related data from the intrusion detection system. Expert systems permit the incorporation of an extensive amount of human experience into a computer application that then utilizes that knowledge to identify activities that match the defined characteristics of misuse and attack.

Unfortunately, expert systems require frequent updates by a system administrator to remain current. While expert systems offer an enhanced ability to review audit data, the required updates may be ignored or performed infrequently by the administrator. At a minimum, this leads to an expert system with reduced capabilities. At worst, this lack of maintenance will degrade the security of the entire system by causing the system's users to be misled into believing that the system is secure, even as one of the key components becomes increasingly ineffective over time.

Rule-based systems suffer from an inability to detect attacks scenarios that may occur over an extended period of time. While the individual instances of suspicious activity may be detected by the system, they may not be reported if they appear to occur in isolation. Intrusion scenarios in which multiple attackers operate in concert are also difficult for these methods to detect because they do not focus on the state transitions in an attack, but instead concentrate on the occurrence of individual elements. Any division of an attack either over time or among several seemingly unrelated attackers is difficult for these methods to detect.

Rule-based systems also suffer from a lack of flexibility in the rule-to-audit record representation. Slight variations in an attack sequence can affect the activity-rule comparison to a degree that the intrusion is not detected by the intrusion detection mechanism. While increasing the level of abstraction of the rule-base does provide a partial solution to this weakness, it also reduces the granularity of the intrusion detection device.

IV. RESULTS OF AN EXAMPLE SCENARIO

An example of a rule-based intrusion detection system is Snort. This open source program is a lightweight network intrusion detection system based on a libpcap packet sniffer and logger, (Roesch, 1999). Snort includes an expert system that uses logging to perform content pattern matching and detect a variety of attacks and probes. The detection engine is programmed using a simple language that describes per packet tests and actions. This capability simplifies and expedites the development of new exploit detection rules.

Snort was used in this research effort to evaluate the detection of denial of service attacks in a simulated network stream. Fifteen denial of service attacks were launched against a Linux-based host. The packets that were received by the host were collected using TCPDump. The libpcap file was then used as input to Snort. A denial of service rule-set was generated automatically by Snort to detect the most commonly found attacks. After processing the libpcap file Snort recorded no attacks in the simulated network stream. This experiment demonstrates a problem that is often encountered in applying expert systems to intrusion detection. The specificity of the rules in the Snort rule base limit the effectiveness of the approach unless the packets that are reviewed exactly match the specified rules. Unfortunately, this is not usually the case.

The evolution strategy used in this work essentially took a look at detecting anomalies associated with a denial of service attack (namely, the SNY flood attack). The main detail, which needs discussion is the cost function. Taking

notice of Figure 1, a unit cube is used to illustrate a SYN flood attack.

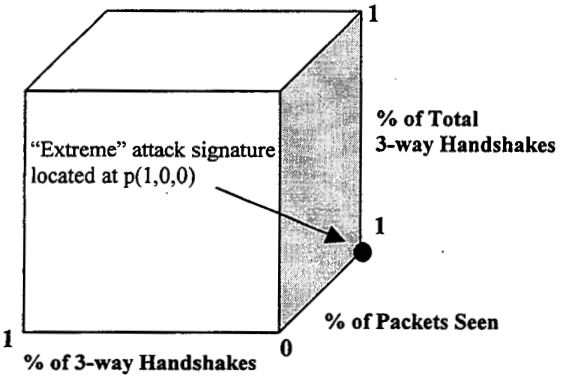

Figure 1. Cube Illustrating a SYN flood attack.

Figure 1 represents what a SYN flood signature is. Here in an extreme case, in terms of TCP-based traffic, a SYN flood can occur if an attacking machine represents all of the incoming packets where non of which are completed 3-way handshakes. From this cube it is easy to see the cost function as simply a point-distance calculation. What is left is a basic mapping of the traffic data to the cube. To this end, the coordinates are calculated based on the values computed from viewing a window of calculations. Figure 2 illustrates the results of the evolution strategy.

Taking notice of Figure 2, the top portion illustrates each client machine's total competitions won. This in turn represents the most likely anomalous source. The bottom portion represents candidates located relative to the attack vertice.

V. CONCLUSIONS

Figure 3 represents the expert system contained within the intrusion detection system. Figure 4 represents the evolution strategy replacing it. The rule matching is modeled parametrically. This system incorporates the rules of the expert system and measures fitness of the individuals based on a point-distance calculation. This way, simpler rules can be used thereby simplifying the entire system. An additional major benefit of using an evolution strategy is its resistance to missed detection based on a lack of an exact match. Future work needs to be done with respect to robustness and resistance to false positives.

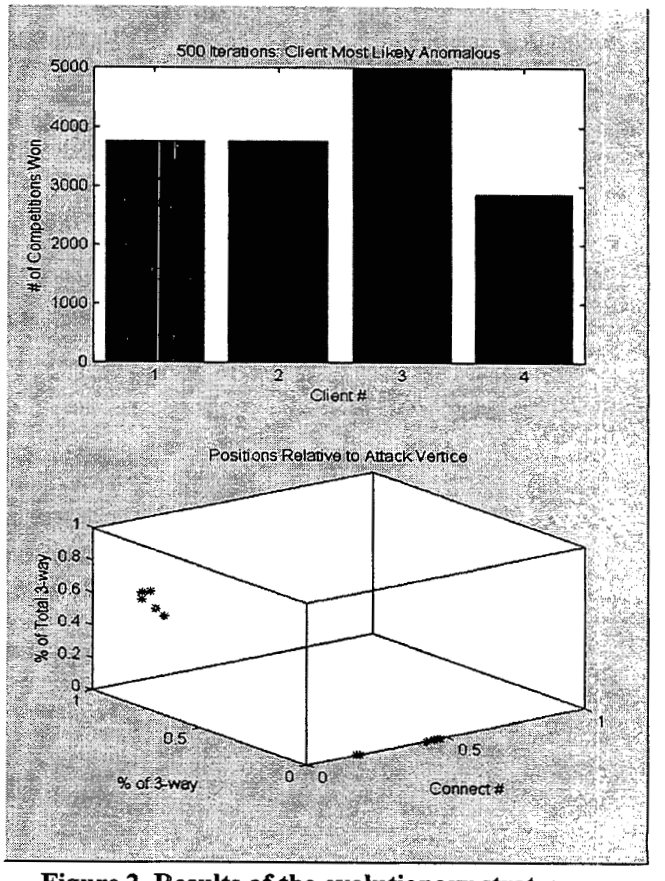

Figure 2. Results of the evolutionary strategy.

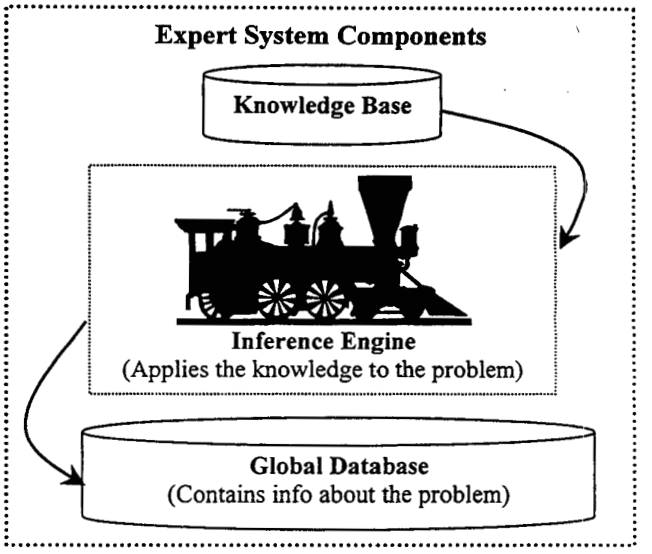

Figure 3. Relationship between components.

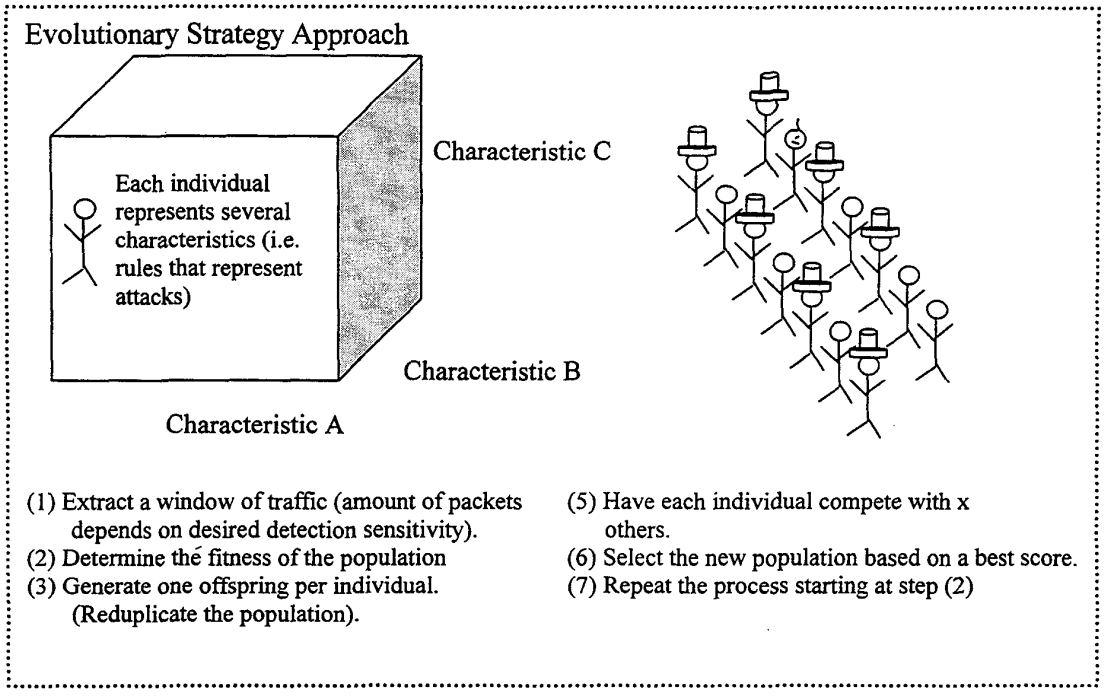

Figure 4. Illustration of the evolution strategy.

VI. REFERENCES

[1] R. N. Brandon, *Concepts and Methods in Evolutionary Biology*, Cambridge Press, New York, New York, 1996.

[2] R. C. Garcia and M. N. O. Sadiku, "Using Evolution Strategies to Solve Laplace's Equation," Symposium Digest, 15th Benjamin Franklin Symposium, 5/97, pp. 18-21.

[3] D. B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, IEEE Press, New York, New York, 1995 pp. 169-170.

[4] I. Rechenberg, "Evolution Strategy", *Computational Intelligence: Imitating Life*, IEEE Press, New York, New York, 1994, pp. 147-159.

[5] J. H. Minster, N. P. Williams, T. G. Masters, J. F. Gilbert, and J. S. Hasses, "Application of Evolutionary Programming to Earthquake Hypocenter Determination," Proc. 4th Con. On Evol. Prog.1995, pp. 3-17.

[6] Denning, Dorothy. (February, 1987). An Intrusion-Detection Model. *IEEE Transactions on Software Engineering, Vol. SE-13, No. 2.*

[7] Sebring, M., Shellhouse, E., Hanna, M. & Whitehurst, R. (1988) Expert Systems in Intrusion Detection: A Case Study. In *Proceedings of the 11th National Computer Security Conference.*

[8] Anderson, D., Frivold, T. & Valdes, A (May, 1995). Next-generation Intrusion Detection Expert System (NIDES): A Summary. *SRI International Technical Report SRI-CSL-95-07.*

[9] Roesch, Martin. (1999). Snort - Lightweight Intrusion Detection for Networks. In *Proceedings of the USENIX LISA '99 Conference.*