A Genetic Algorithm-Based Intrusion Detection System

Han H. Lee, Duk Lee°, Hee S. Kim, Jong U. Choi

Markany

8-11, 3f, chungha b/d, chamwon-dong, seocho-ku, seoul, Korea

Fax: +82-2-3435-9195, E-mail: sslh2@markany.co.kr, juchoi@markany.co.kr, deli@ait.re.kr, nima79@hanmail.net

ABSTRACT

In this paper, a novel approach to intruder detection is introduced. The approach, based on the genetic algorithms, improved detection rate of the host system which has traditionally relied on known intruder patterns and host addresses. Rather than making judgments on whether the access is intrusion or not, the system can continuously monitor system with categorized security level. With the categorization, when the intruder attempts repeatedly to access the system, the security level is incrementally escalated.

In the simulation of a simple intrusion, it was shown that the current approach improves robustness of the security system by enhancing detection capability and flexibility. The evolutionary approach to intruder detection enhances adaptability of the system.

1. Introduction*

To protect computer resources from intruders, various techniques have been suggested to improve accuracy and timeliness of the intruder detection systems (IDS). The approaches suggested to IDS are classified into two techniques: host-based system and network-based system. Host-based system monitors transactions of a single system which is accessed by multiple users, while the network-based system exchanges information of the intrusion to each other. Network-based system is again classified into two categories: network-based intrusion detection (NID)[1][2] and distributed intrusion detection (DID).

The classification of host-based IDS largely depends on type of operating systems. Usually the host-based IDS is categorized into anomaly intruder detection (AID) [5] and misuse intrusion detection (MID) [3][4]. In this paper, a host-based IDS model is suggested based on genetic algorithm.

* This research was supported by a KOSEF(1999-2-511-001)

2. Self-Nonself Discrimination with Priority Value

The approach suggested in this paper mainly relies on MID technique that enables real-time monitoring through the difficulty to subvert and low overhead, genetic algorithm [8][9], and self-nonself discrimination [10]. Also, based on priority value assignment, the system categorizes risk of intrusion by 5 classes, rather than a single categorization.

2.1 Application of Self-Nonself Discrimination

Stephanie Forrest [10] attempted to detect virus and un-authenticated users based on self-nonself discrimination in IDS. With the data base of normal user's profile the system retrieves user profile data periodically or whenever necessary, compares user profile with current user's behavior, and detects existence of computer viruses by measuring the difference, or detect un-authenticated users

when the user's behavior are unknown.

In this paper, in database included are commands frequently used in attacks and ID and IP addresses listed as black users. In addition, the access time data assigned by host operators are categorized. In the GA-based detection model, the commands and access resources are searched by crossover to investigate compatibility with database. If the input data is compatible with existing data, it is judged as 'self', which means that the user belongs to 'dangerous user group'. The method is very similar to the virus detection conducted by T-Cell in immunology. The system determines security level with GA-based adaptive detection mechanism and Priority Value (PV) assigning mechanism.

2.2 Genetic Algorithm and Application of Priority Value

2.2.1 Priority Value

Priority value (PV) is assigned to each gene, ranging 1 to 5. The system assigns values to derive final security level by applying to fitness function described in the next section. PV has a table for each gene.

2.2.2 Chromosome structure

GA has been applied to optimization problems such as traveling sales person problem, job-shop assignment problem, or optimal network structure in communications network. In this research, GA was employed to solve problems of current MID systems so that the system can detect anomalies whose patterns are unknown. Changes of PV are triggered by the GA application makes it possible for the system to cope with new patterns of attacks. Each category of commands, access resources, and user Ids corresponds to each type of gene that has individual PV. PVs are initially set by operator, depending on the characteristics of host system.

Population size is not fixed. Instead, it continuously changes. If the monitoring objectives are added to the monitoring list with generation of mutation, the population size increases. The number of generation is not fixed, but determined by the population size.

Population Size=

$$G_{id} + \quad G_{resource} + \bullet\bullet\bullet + \quad G_{operation}$$

Maximum number of Generation=

$$G_{id} \times \quad G_{resource} \times \bullet\bullet\bullet \times \quad G_{operation},$$

where G indicates gene.

2.2.3 Fitness Function

Based on the fitness function, the system can select a set of gene combination so that the Last Security Grade(LSG) of the gene set becomes the maximum value. In the fitness function, each gene has different effects on the function, depending on its weight. The following equation is the fitness function employed in this research.

Fitness function:

$$LSG = \max\left(\frac{\sum_{i} (Match(PV_i) * IR_i)}{M_s} \right)$$

(Equation 1)

In the equation-1, *IRi* weight of each gene, *i* is type of gene such as commands. Id, IP, etc., and *Ms* is the number of compatibility between user input information and gene information. In the equation, *Match()* means the process of gene matching with log data.

2.2.4 Mutation and Evolution

	Example	Weight	Detected Or Not	PV	LSG
ID	Hacker	0.3	O	4	1.2
IP	203.237.173.134	0.4	X	0	0
Login Time	20:00	0.3	O	1	0.3

Table 1. An Example of Mutation and Evolution

MID has a problem that it can detect well-known intrusion, but cannot detect unknown patterns of intrusion. The IDS model suggested in this research solves the problem. For example, let assume that three genes of ID, IP, and Login-Time are recorded as in the Table-1.

Table-1 describes that a user, Hacker whose IP is 203.237.173.134 accessed the host system at the time of 20:00. In the real-world system, IDS generates much more genes based on log file data. The value of 'O' in the column of 'detected or not' in Table-1 implies that matching gene was found with gene crossing. To the contrary, the value of 'X' in the column means that no gene is matched. In Table-1, the security level of the user is derived by the LSG (equation-1), and IP(203.237.173.134) is newly included in a gene of monitoring objective by setting PV=1.5. This means that a mutation is generated with a completely new gene, which leads to evolution. The evolution value (EV) becomes 0.1. Host system operator heuristically determines the range of change in EV. The evolution is determined by the following rule:

$$If\ (PV_{IP} > LSG)$$
$$PV_{IP} = PV_{IP} + EV. \qquad \text{(Condition 1)}$$

Based on the rule, security level of the user needs not change, but the PV changes from 1.0 to 1.1.

3. Experiment

The experiment was conducted in the environment of Linux 6.0. As the log data of the users are included in the experiment, no virtual difference exists in experimental environment between Linux and UNIX. The simulator built in the research has a storage in which intruder genes are stored, storage in which PVs are stored, and input nodes in which extracted log data are fed in. The simulation result shows that implementation of GA-based IDS and extraction of PVs can be done in real-time.

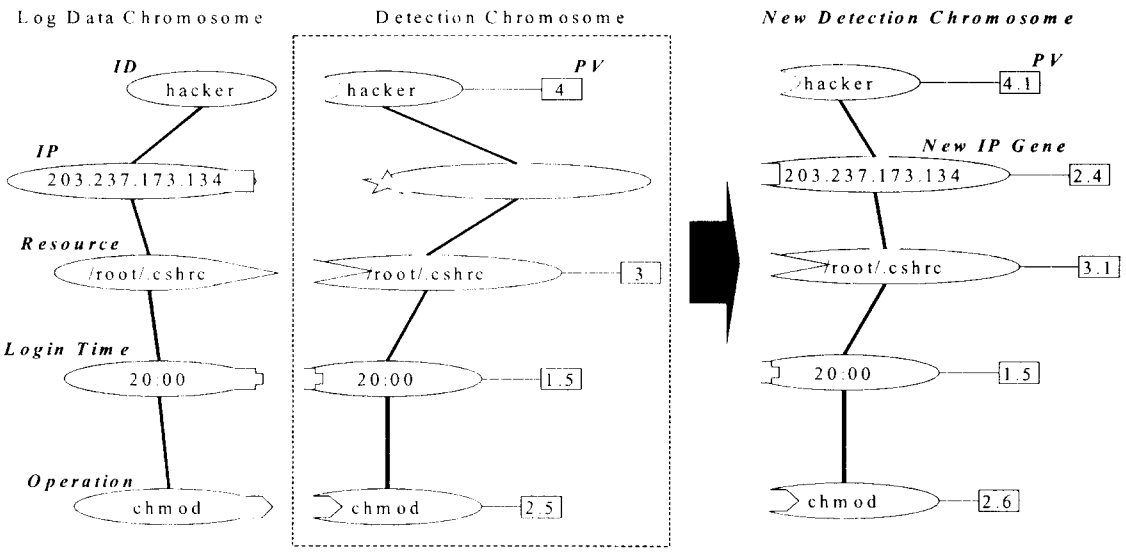

Figure 1. Example of Simple Experiment

Figure-1 depicts the processes internally generated when the simulator works. As depicted in the figure, there is no gene that only matches IP log data through crossover. In the process, a new mutation is generated, which is depicted as a new detection chromosome in the

right end of the figure. The IP new gene of the new detection chromosome can have PV as the LSG derived in Equation-1. The other genes go through evolution processes based on the condition-1. These processes are repeated in real-time with log-data. The above experiment describes a simple one. As revealed in the simulation, continuous attempts reinforce security system by making the system more robust than before. An attempt of new attack pattern is added to the monitoring list as a new gene, the system responds with a comprehensively counter-attack activities.

4. Conclusion & Future Research

MID is an IDS that enables a fast, real-time monitoring. The GA-based IDS model suggested in this research perfectly cured problems of existing MID systems. With the application of GA algorithm, the system becomes so flexible that it can appropriately handle new patterns of intrusions. Introduction of evolution process makes it possible for the system to strengthen security level in response to the repeated attempts to intrude. The categorization of the security level enables differentiated monitoring and effective host operation.

Current research, as the simulations were conducted with log file information, has some limitation. To handle sophisticated attacks requiring high-level programming, events generated by process generation should be expressed in gene. Then, much more robust system can be developed. The issue of how to determine the range of change in the evolution of PVs for each host should be studied in the future.

REFERENCE

[1] Stephen S. Yau and Xinyu Zhang, "Computer Network Intrusion Detecion, Assessment And Prevention Based on Security Dependency Relation", 0-7695-0368-3/99, 1999 IEEE..

[2] Staniford-Chen, S. S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levitt, C. Wee and R. Yip, D. Zerkle, "GrIDS-A Graph Based Intrusion Detecion System for Large Networks", NISSC 96. http://olympus.cs.ucdavis.edu/arpa/grids/welcome.html. 2000-1-15.

[3] Sandeep Kumar and Eugene H. Spafford, "A Pattern Matching Model For Misuse Intrusion Detection", http://www.certcc.or.kr/paper/index.html, 2000-1-7.

[4] Shiuh-Pyng Shieh and Virgil D. Gligor, " On a Pattern-Oriented Model for Intrusion Detecion", IEEE Transaction on Knowledge and Data Engineering, vol. 9, No. 4, July/August 1997.

[5] Jeremy Frank, "Artificial Intelligence and Intrusion Detection: Current and Future Directions", http://www.certcc.or.kr/paper/index.html, 2000-1-7.

[6] Debra Anderson, Thane Frivold, and Alfonso Valdes, "Next Generation Intrusion Detection Expert System(NIDES)", Technical Report, SRI-CSL-95-07.

[7] Debra Anderson, Teresa F. Lunt, Harold Javitz, Ann Tamaru, and Alfonso Valdes, "Detecting Unusual Program Behavior Using the Staticstical Component of the Next-generation Intrusion Detection Export Systems(NIDES)", Technical Report, SRI-CSL-95-06.

[8] Guy Helmer, Johnny Wong, Vasant Honavar and Les Miller, "Feature Selection Using a Genetic Algorithm for Intrusion Detection", Proceeding of the Genetic and Evolutionary Computation Conference, Volume 2, v.2, 1999-7-13.

[9] Ludovic MÉ, "GASSATA, a Genetic Algorithm as an Alternative Tool for Security Audit Trails Analysis", IEEE Symposium on Security and Privacy Oakland, 1996-5

[10] Stephanie Forrest, Alan S. Perelson, Lawrence Allen and Rajesh Cherukuri, "Self-Nonself Discrimination in a Computer", Proceedings of 1994 IEEE Symposium on Research in Security and Privacy.