A Genetic Based Intrusion Detection System

Binu Kaushik[1] and **Shashi Bhusan Vishwakarma**[2]
Department of Information Technology, IEC College of Engineering and Technology, 4 Knowledge Park I, Surajpur Kasna Road, Greater Noida 201306 UP India
binusharma25@gmail.com, shashibhusan076@gmail.com

Abstract : **Nowadays, it is very important to maintain a high level security to ensure safe and trusted communication of information between various organizations. But secured data communication over internet and any other network is always under threat of intrusions and misuses. So Intrusion Detection Systems have become a needful component in terms of computer and network security. There are various approaches being utilized in intrusion detections, but unfortunately any of the systems so far is not completely flawless. So, the quest of betterment continues. In this progression, we present an Intrusion Detection System (IDS), by applying genetic algorithm (GA) to efficiently detect various types of network intrusions.**

Keywords : Genetic Algorithms, Intrusion detection system, Mutation, Crossover , Fitness function

I. INTRODUCTION

WITH ever-increasing growth of computer networks and emergence of electronic commerce in recent years, computer security has become a priority. Since intrusions take advantage of vulnerabilities in computer systems or use socially engineered penetration techniques, intrusion detection (ID) is often used as another wall of protection. In addition, traditional security techniques as user authentication are not optimal method to protect data from any possible attack.That is due to the vastness of the network activity data and the need to regularly update our intrusion detection systems (IDS) to cope with new unknown attack methods or upgraded computing environments. Thus, ID is becoming one of the main technologies used to monitor network traffics and identify network intrusions. According to Matthew Berge: "*Network based intrusion detection attempts to identify unauthorized, illicit, and anomalous behavior based solely on network traffic"*.

A network IDS, using either a network tap, span port, or hub collects packets that traverse a given network. Using the captured data, the IDS system processes and flags any suspicious traffic. A computer system intrusion is seen as any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource. The introduction of networks and the Internet caused great concern about the protection of sensitive information and resulted in many computer security research efforts during the past few years. Though preventative techniques such as access control and authentication attempt to prevent intruders, these can fail, and as a second line of defense, intrusion detection has been introduced. Intrusion detection systems (IDS) are implemented to detect an intrusion as it occurs.

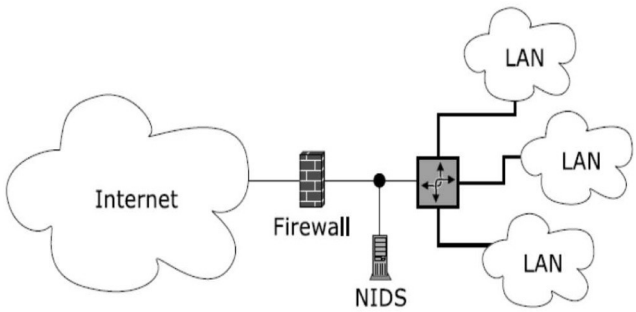

Figure 1. A Network Intrusion Detection System in a Network.

II. CATEGORIES OF ATTACK

There are four major categories of attack.

Denial of Service (DoS): A DoS attack is a type of attack in which the hacker makes a computing or memory resources too busy or too full to serve legitimate networking requests and hence denying users access to a machine *e.g.* apache, smurf, neptune, ping of death, back, mail bomb, UDP storm etc. are all DoS attacks.

Remote to User Attacks (R2L): A remote to user attack is an attack in which a user sends packets to a machine over the internet, which s/he does not have access to in order to expose the machines vulnerabilities and exploit privileges which a local user would have on the computer *e.g.* xlock, guest, xnsnoop, phf, sendmail dictionary etc.

User to Root Attacks (U2R): These attacks are exploitations in which the hacker starts off on the system with a normal user account and attempts to abuse vulnerabilities in the system in order to gain super user privileges *e.g.* perl, xterm.

Probing: Probing is an attack in which the hacker scans a machine or a networking device in order to determine weaknesses or vulnerabilities that may later be exploited so as to compromise the system. This technique is commonly used in data mining *e.g.* saint, portsweep, mscan, nmap etc.

III. CLASSIFICATION OF INTRUSION DETECTION

Intrusions Detection can be classified into two main categories [2]. They are as follow:

Host Based Intrusion Detection: HIDSs evaluate information found on a single or multiple host systems, including contents of operating systems, system and application files.

*Network Based Intrusion Detection***:** NIDSs evaluate information captured from network communications, analyzing the stream of packets which travel across the network.

IV. COMPONENTS OF INTRUSION DETECTION SYSTEM

An intrusion detection system normally consists of three functional components [2].

The first component of an intrusion detection system, also known as 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. The second component of an intrusion detection system is known as 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: Misuse/Signature-Based Detection: This type of detection engine detects intrusions that follow well-known patterns of attacks (or signatures). The main limitation of this approach is that it only looks for the known weaknesses and may not care about detecting unknown future intrusions. Anomaly/Statistical Detection: An anomaly based detection engine will search for something rare or unusual [26].

They analyse 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 an intrusive behavior as normal behavior because of insufficient data. The third component of an intrusion detection system is the response manager. In basic terms, 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

V. GENETIC BASED MACHINE LEARNING

Genetic Algorithms are a family of computational models inspired by evolution. These algorithms encode a potential solution to a specie problem on a simple chromosome-like data structure and apply recombination operators to these structures as to preserve critical information. Genetic algorithms are often viewed as function optimizer, though the range of problems to which genetic algorithms have been applied are quite broad. An implementation of genetic algorithm begins with a population of (typically random) chromosomes. One then evaluates these structures and allocated reproductive opportunities in such a way that these chromosomes which represent a better solution to the target problem are given more chances to `reproduce' than those chromosomes which are poorer solutions. The 'goodness' of a solution is typically denied with respect to the current population [10].

The working principle of a GA is illustrated below. The major steps involved are the generation of a population of solutions, finding the objective function and fitness function and the application of genetic operators. These aspects are described briefly below. They are described in detail in the following subsection.

Algorithm GA

1. *formulate initial population*
2. *randomly initialize population*
3. *repeat*
4. *evaluate objective function*
5. *find*
6. *fitness function*
7. *apply genetic operators*
8. *reproduction*
9. *crossover*
10. *mutation*
11. *until stopping criteria*

An important characteristic of genetic algorithm is the coding of variables that describes the problem. The most common coding method is to transform the variables to a binary string or vector; GAs perform best when solution vectors are binary. If the problem has more than one variable, a multi-variable coding is constructed by concatenating as many single variables coding as the number of variables in the problem.

Genetic Algorithm processes a number of solutions simultaneously. Hence, in the first step a population having P individuals is generated by pseudo random generators whose individuals represent a feasible solution. This is a representation of solution vector in a solution space and is called initial solution. This ensures the search to be robust and unbiased, as it starts from wide range of points in the solution space.

In the next step, individual members of the population are evaluated to and the objective function value. In this step, the exterior penalty function method is utilized to transform a constrained optimization problem to an unconstrained one. This is exclusively problem specific. In the third step, the objective function is mapped into a fitness function that computes a fitness value for each member of the population. This is followed by the application of GA operators (*i.e.* mutation and crossover).

VI. IMPLEMENTATION

Algorithm My System: The system can be divided into two main phases: the precalculation phase and the detection phase. Listing 1 depicts major steps in precalculation phase, where a set of chromosome is created using training data. This chromosome set will be used in the next phase for the purpose of comparison.

Algorithm 1 : Initialize chromosomes for comparison
Input : Network audit data (for training)
Output : A set of chromosomes

1. *Range = 0.125*
2. *For each training data*
3. *If it has neighbouring chromosome within Range*
4. *Merge it with the nearest chromosome*
5. *Else*
6. *Create new chromosome with it*
7. *End if*
8. *End for*

Algorithm 2 depicts major steps of detection phase, where a population is being created for a test data and going through some evaluation processes (selection, crossover, mutation) the type of the test data is predicted. The precalculated set of chromosome is used in this phase to find out fitness of each chromosome of the population.

Algorithm 2 : Predict data/intrusion type (using GA)
Input : Network audit data (for testing), Precalculated set of chromosomes
Output : Type of data.

1. Initialize the population
2. CrossoverRate = 0.15, MutationRate = 0.35
3. While number of generation is not reached
4. For each chromosome in the population
5. For each precalculated chromosome
6. Find fitness
7. End for
8. Assign optimal fitness as the fitness of that chromosome
9. End for
10. Remove some chromosomes with worse fitness
11. Apply crossover to the selected pair of chromosomes of the population
12. Apply mutation to each chromosome of the population
13. End while

Implementation Procedure: In the pre-calculation phase, 10 groups of chromosomes were made according to training data. There were 10 (9+1) groups for each of attack and normal types presented in training data. Number of chromosomes in each group is variable and depends on the number of data and relationship among data in that group. Total number of chromosomes in all groups were tried to be kept at reasonable level to optimize time consumption in testing phase.

In the testing / detection phase, for each test data, an initial population is made using the data and occurring mutation in different features. This population is compared with each chromosome prepared in training phase. Portion of population, which are more loosely related with all training data than others, are removed. Crossover and mutation occurs in rest of the population which becomes the population of new generation. The process runs until the generation size comes down to 1 (one). The group of the chromosome which is closest relative of only surviving chromosome of test data is returned as the predicted type. Among the extracted features of the datasets, we have taken only the numerical features, both continuous and discrete, under consideration for sake of simplification of implementation.

VII. GENE REPRESENTATION

Genetic algorithms are good tools for acquiring optimized solutions and their use with determining rule sets for potential and actual network intrusions is both intuitive and potentially valuable. An example suspect log record that shows a potential network intrusion will be reviewed. Then, the structuring of the domain-problem based chromosome will be discussed. Finally, how the GA is used in assisting rule set creation for a potential network intrusion will be examined.

Suspected Log Record: Firewall devices are typically the first point of entry within computer networks. In Table 1, we look at a typical content of a firewall log entry.

TABLE 1
EXAMPLE FIREWALL LOG ENTRY

Time	Local time on the management station
Action	Accept, deny, or drop. Accept=accept or pass the packet. Deny=send TCP reset or ICMP port unreachable message. Drop=drop packet with no error
Firewall	IP address or hostname of the enforcement point
Interface	Firewall interface on which the packet was seen
Product	Firewall software running on the system that generated the message
Source	Source IP address of packet sender
Destination	Destination IP address of packet
Protocol	Usually layer 4 protocol of packet - TCP, UDP, etc.
Service	Destination port or service of packet
Translation	If address translation is taking place, this field shows the new source or destination address. This only shows if NAT is occurring.
Rule	Rule number from the GUI rule base that caught this packet, and caused the log entry. This should be the last field, regardless of presence or absence of other fields except for resource messages.

A Firewall Log Analysis Primer for our purposes in the creation of input data for the Genetic Algorithm, we will look at the following example firewall log entry in Table 2.

TABLE 2 - A FIREWALL LOG ANALYSIS PRIMER

Source IP	Source IP address of packet sender
Destination IP	Destination IP address of packet
Destination Port	Destination port or service of packet
Protocol	Usually layer four protocol-TCP,UDP, etc
Bytes Sent by Originator	The number of bytes in the request from the source system.
Bytes Sent by Responder	The number of bytes returned from the responding or target system.

The Domain Problem Chromosome: A typical IDS rule would take the form of the following condition statement:

if {the connection has following information: source IP 125.19.54.155; destination IP address: 109.1.1.0 ~ 109.1.255.255; destination port number: 8184; the protocol used is FTP; the originator sent more than 10,000 bytes of data; and the responder sent more than 250,000 bytes of data } then {stop the connection} .

Given that the input value modeled in Table 2 is similar to a desired IDS Rule Set, the input rule will be the model for the chromosome-like data structure. This input rule within the GA will then be evolved into a fitter output, or as in this case, an IDS Rule. For a clearer view of the IDS rule, note Table 3 below. The Attribute column takes the contents of the above condition and provides labels. The Range of Values column shows the lower and upper bounds of the rule. The suspect source rule set is displayed in the Example Values column. The Descriptions column displays what each item is in the suspect rule and the justification.

Fitness Function: A GA Fitness Function typically has the following or similar steps. First, the general outcome is determined based on whether a gene (or allele) "matches" an existing data set of suspect log record that was obtained from a network device such as a firewall. Then, the function multiplies the "weight" of that field to the degree that the field value "matched" the suspect record field. Typically, the "match" value is either 1 or 0. Weight values are applied to the different genes as historically reported by network devices. For example, if the Destination IP gene historically demonstrates to be a consistent predictor of a network intrusion, its weight will be more than the other genes.Moreover, all particular genes types have the same weight value so all Protocol genes have a weight of 15, regardless of their degree of being a suspect record

$$Outcome = \sum_{i=1}^{6} Matched * Weight$$

in the case of a "gene" such as a Source IP address, let us suppose that historic data from an organization's border hardware devices such as its firewalls reveal that a Source IP address of 125.19.54.155 has attempted various intrusions targeting valuable assets such as a cluster of database systems. If the weightof a Source IP was 10, and given that the historic data supports a "match" value of 1, the outcome of the Source IP gene is 10 (10 = 1 * 10). Next, the delta value or absolute difference between the "outcome" of the chromosome and the suspicion_level is then computed using the following equation.

$$Delta = Outcome - Suspicion\_Level$$

The suspicion_level is a value that indicates if the historical gene value and the suspicious gene value are considered a "match" from historic log data. Continuing with our previous example, given that the Source IP of 125.19.54.155 was determined to bea suspicious IP address, the suspicion_level value would be higher with a value such as 8. Therefore, the delta result is a low number of 2 (2 = |10 – 8|).

If the delta level is high enough, a penalty value is calculated using this delta (or absolute difference). The "ranking" in the equation below indicates whether or not a network intrusion is easy to establish. Historical data should determine the value of

TABLE 3
CLEARER VIEW OF THE IDS RULE

ATTRIBUTE	RANGE OF VALUE	EXAMPLE VALUES	DESCRIPTION
Source IP Address	125.0.0.0 ~ 125.150.255.255	125.19.54.155	125.19.54.155 is a suspect IP address.
Destination IP Address	119.0.0.0 ~119.150.255.255	119.1.1.20	IP Address 119.1.1.17 ~119.1.1.21 are database servers.
Destination Port Number	0 ~ 65535	8184	Destination port number, indicates his is a http service. 8184 is for internal data access
Protocol	1 ~ 20	5	The protocol for this connection FTP. 5 = FTP.
Number of Bytes Sent by Originator	0 ~ 250 KB	10.5 KB	originator sends 7,500 bytes of data
Number of Bytes sent by Responder	0 ~ 1 MB	2.5 MB	The responders sends 250,000 bytes of data