International Journal of
Emerging Research in Management &Technology
ISSN: 2278-9359 (Volume-3, Issue-7)

Research Article | **July 2014**

Network Intrusion Detection System Using Genetic Algorithm with Data Mining Approach

Shashank Lale* Millenium College Bhopal India

Susheel Tiwari Millenium College Bhopal India

Abhinav Gupta L&T Institute of technology, Mumbai India

Abstract—

As the transmission of data over the internet increases, the need to protect connected systems also increases. Intrusion Detection Systems are the latest technology used for this purpose. Although the field of IDSs is still developing, the systems that do exist are still not complete, in the sense that they are not able to detect all types of intrusions. Some attacks which are detected by various tools available today cannot be detected by other products, depending on the types and methods that they are built on.

Keywords -component; NIDS, Data Mining, Genetic Algorithm ,Intrusion, NIDSGA

I. INTRODUCTION

With the increase in network based attacks in general, and the world-wide access to computer networks and systems in particular, those responsible for network and computer system security need to utilize every tool available. Within this paper, the following will be examined, defined, and demonstrated. First, this paper will examine network intrusion detection. Then, genetic algorithms will be discussed. After this, the combining of genetic algorithms with intrusion detection will be reviewed. Finally, future steps will be discussed in the use of a genetic algorithm within intrusion detection

This document is a template. An electronic copy can be downloaded from the Journal website. For questions on paper guidelines, please contact the conference publications committee as indicated on the conference website. Information about final paper submission is available from the conference website.

II. NETWORK INTRUSION DETECTION

What is Network Intrusion Detection? 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. The goal of intrusion detection is to recognize attempts to sabotage in-place security controls (Berge). Specifically, network traffic is analyzed in search for system based breaches. Network breaches can take various forms. Next, an example intrusion is provided.

III. LITERATURE SURVEY

3.1 GENETIC ALGORITHMS

Genetic Algorithms are utilized in various areas of data analytics and problem solving. A branch of machine learning:Genetic algorithm is a family of computational models based on principles of evolution and natural selection. These algorithms convert the problem in a
specific domain into a model by using a chromosome-like data structure and evolve the chromosomes using selection, recombination, and mutation operators.
(Li, 2004, p. 1) Moreover, genetic algorithms (GA) 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 (Li, p. 2). Given the above definition of a GA, 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.

Table3.1: Firewall Log Entry Content

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 to
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
Service Protocol	Destination port or service of packet Usually layer 4 protocol of packet - TCP, UDP, etc.
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.

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.

Table3.2: Example

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 4 protocol of packet - 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.

3.2 SUSPECTED INTRUSION LOG RECORD

Firewall devices are typically the first point of entry within computer networks. Here (see Table 1), we will look at a typical content of a firewall log entry.

3.3 STRUCTURING 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 above 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 for why the rule may be a potential threat to the network and/or the systems that are nodes within a network.

Table 1.3 : Rule Definition

Attribute	Range of Values	Example Values	Descriptions
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 this 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	The 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

In order for a particular domain to be suitable for a genetic algorithm the domain must be converted into numeric values, either within the GA or as raw input. These numeric values are sometimes referred to as genes and are changed at random within a range during an evolutionary cycle. The set of chromosomes during a stage of evolution are called a population (Li, p. 1). A fitness function is used to calculate the "goodness" of each chromosome. We can convert the above example into a chromosome form, with each row as a "gene," as described below in Table 4 below:

Table 3.4: Each Row is gene

Source IP address	125.19.54.155 converted to 2006384639
Destination IP address	109.1.1.20 converted to 1996554516
Destination Port Number	8184
Protocol	5
Number of Bytes Sent by Originator	10500
Number of Bytes sent by Responder	2500000

Therefore, based on the above information, here is the example chromosome for the network intrusion detection GA:

Source IP	Destination IP	Destination Port	Protocol	Originator Bytes	Responder Bytes
2006384639	1996554516	8184	5	10500	2500000

Figure 5.4 :Work Breakdown Structure

IV. DESIGN

4.1 ARCHITECTURE DESIGN

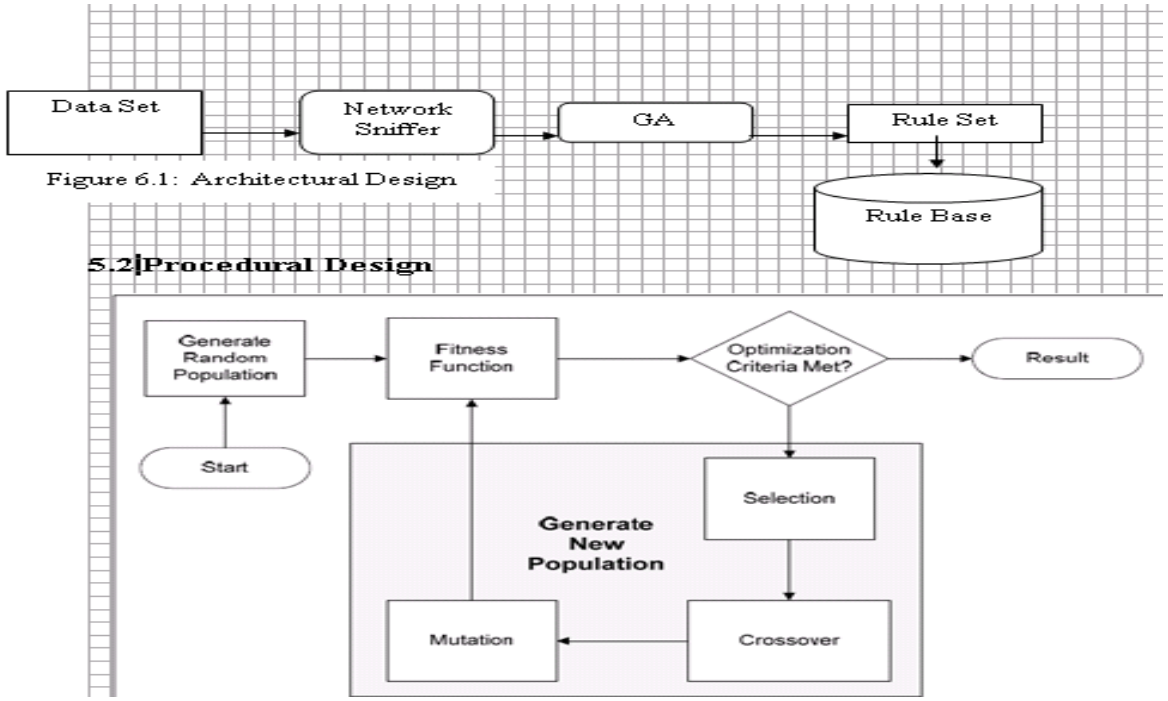


Figure 6.1 Procedural Design

4.2 ALGORITHM : GENETIC ALGORITHM USE IN RULE SET CREATION

Now, the structure of a potential GA within Network Intrusion Rule Set Creation will be detailed. The structure of a basic genetic algorithm. First, the GA creates a random population which is then evaluated concerning its level of fitness in a Fitness Function.

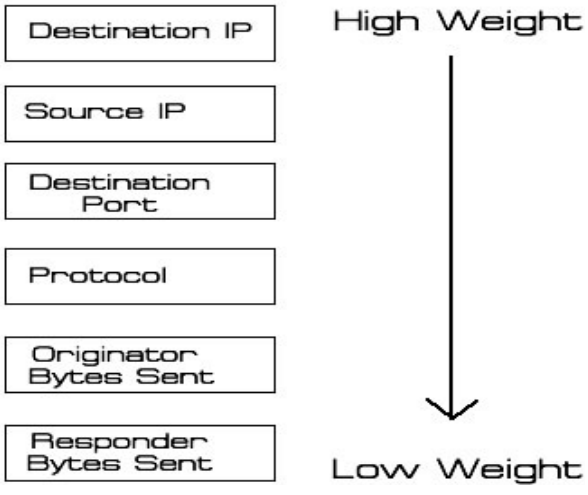
4.2.1 FITNESS FUNCTION

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 (See Figure 3).

Equation 1: Outcome

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

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 consist 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 (See Figure 4).



As a clarifying example, 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 weight of 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 (See Figure 5).

$$\Delta = | \text{outcome} - \text{suspicion\_level} |$$

Equation 2: Delta

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 be a 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) (See Figure 6). The "ranking" in the equation below indicates whether or not a network intrusion is easy to establish. Historical data should determine the value of the ranking. For example, given that Destination IP addresses of certain asset systems are well known by those within an organization, this ranking would be higher

$$\text{penalty} = \left(\frac{\Delta * \text{ranking}}{100}\right)$$

Equation 3: Penalty

Finally, the chromosome's fitness is then computed using the above penalty. The scope of the fitness result is between 0 and 1 (See Figure 7).

$$\text{fitness} = 1 - \text{penalty}$$

Equation 4: Fitness

4.2.2 SELECTION

Once the initial population (of chromosomes) is evaluated, the GA experiments with new generations and iteratively refines the initial outcomes so that those that are most fit are more probable to be ranked higher as results. The objective is to produce new generation of chromosomes to evaluate.

4.2.3 CROSSOVER

In essence, the crossover operation creates new chromosomes that share optimistic characteristics of the parent chromosomes while at the same time lowering the negative attributes in a child chromosome. Figure 8 below provides an example of a crossover of chromosomes from the parents to their offspring.

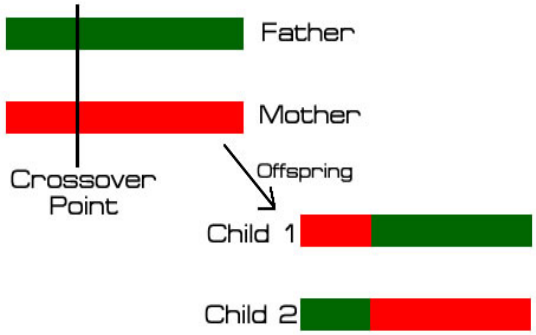


Figure 5.3.4: Crossover

Although this step is typical in most genetic algorithms, in the case of this project's chromosome (see Table 4 above) the crossover operation may not be beneficial. While a Source or Destination IP may be bound by upper and lower IP settings (as demonstrated in Table 3 above), a crossover of the IP octet values would probabilistically not be advantageous. For example, the crossover of the parental values of 209.103.51.134 and 101.1.25.193 could result in child IP addresses of 209.103.25.193 and 101.1.51.134. However, the probability that this offspring will be potential suspicious Source or Destination IP addresses is low.

4.2.4 MUTATION

The final step in the process of generating a new population is mutation. This phase randomly alters a gene's value to create a different one (Marakas, 2003, p. 143). Figure 9 below details how a gene's (or allele's) value is changed and thereby creating a new chromosome. Concerning the applicability of this step with the network intrusion chromosome, as was the case in the crossover step above the probability of useful outcomes is minimal.

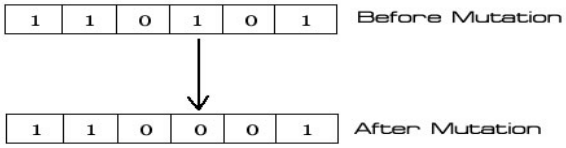


Figure 4.3.5: Mutation

4.2.5 THE RULE SET

Essentially, the rule set is produced from the output of the GA. For example, the input of Source IP = 1829975662 (which is an IPv4 address of 109.19.54.110)|
Destination IP = 1828782356 (which is an IPv4 address of 109.1.1.20) | Destination Port = 8184 | Protocol = 5 | Originator Bytes = 10500 | Responder Bytes = 250000 could produce the following rule:

if
{
the connection has following information: source IP 125.19.54.155; destination IP address: 119.1.1.17 ~ 119.1.1.21; 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
{
log the intrusion and stop the connection
}

Note that the Source and Destination IP input values above are a Java "double" primitive data type. This was needed to convert an IPv4 address to a Java primitive for its use in the GA.

V. APPLICATION

1. Artificial Intelligence.
2. Data Mining.
3. IDS monitors the interaction between user and application, which traces activity to individual users.
4. Measurements and analysis of typical and atypical user behavior. For example an anomaly based NIDS is capable of detecting high volume traffic flows, flash crowds, load imbalance in the network, sudden changes in demand of a port usage, sudden surge of traffic from/to a specific host, etc.
5. Enforcement of the security policies in a given network. For example a NIDS can be configured to block all communication between certain sets of IP addresses and or ports. A NIDS can also be used to enforce network wide access controls.

VI. CONCLUSION

The software development is very flexible and much functionality can be added to it, to enhance performance of this project titled "Intrusion Detection System In networking Using Genetic Algorithm". By using genetic algorithm, during run time the new
set of rules will added in the dataset. A brief overview of Intrusion Detection System, Genetic algorithm, and related detection techniques are discussed. This implementation of genetic algorithm is unique as it considers both temporal and spatial information of network connections during the encoding of the problem; therefore, it should be more helpful for identification of network anomalous behaviors.

REFERENCE

[1] Li, w. (2004). Using genetic algorithm for network intrusion detection. Proceedings of the united states department of energy cyber security group 2004 training conference, may 24-27, kansas city, ks.
[2] http://www.security.cse.msstate.edu/docs/publications/wli/doecsg2004.pdf
[3] Bezroukov, nikolai. 19 july 2003. "intrusion detection (general issues)." Softpanorama: open source software educational society. Nikolai bezroukov. Url: http://www.softpanorama.org/security/intrusion_detection.shtml (30 oct. 2003).
[4] Bridges, susan, and rayford b. Vaughn. 2000. "intrusion detection via fuzzy data mining." In proceedings of 12th annual canadian information technology security symposium, pp. 109-122. Ottawa, canada.
[5] Graham, robert. Mar. 21, 2000. "faq: network intrusion detection systems." Robertgraham.com homepage. Robert graham.
Url:http://www.robertgraham.com/pubs/network-intrusion-detection.html (30 oct. 2003).
[6] Crosbie, mark, and gene spafford. 1995. "applying genetic programming to intrusion detection." In proceedings of 1995 aaai fall symposium on genetic programming,pp. 1-8. Cambridge, massachusetts. Url: http://citeseer.nj.nec.com/crosbie95applying.html (30 oct. 2003).

