Network Security policy framework and Analysis

Suhas B. Chavan
MECSE (SEM III), Department of Computer Science & Engg.
Walchand Institute of Technology,
Solapur, India

L.M.R.J Lobo
Associate Professor, Department of Computer Science & Engg.
Walchand Institute of Technology,
Solapur,India

ABSTRACT

Improved genetic feedback algorithm based network security policy framework contains some drawback for security. This has motivated the need of a strong network security policy framework. In this paper a strong network model for security function is presented. The fitness function is examined for defining the gene of a network packet and a method to calculate fitness function is explained. In this model passive attacks are more difficult to deal as compare to the active attack. The basic attacks can be categorized as buffer overflow, array index out of bound, etc. We have dealt with passive attack, active attack, its types and brute force attack. These attacks are analyzed and security is provided. We finally find the best policy using a comparator. The main aim of this paper is threat detection, time optimization, performance increase in terms of accuracy and policy automation. We are experimenting using the real data set from the internet and local network. Our work is carried out on client server environment providing data confidentiality and authentication.

General Terms

Genetic algorithm, Network security, Crossover technique.

Keywords

genetic algorithm, network security policy framework, fitness function.

1. INTRODUCTION

The model given for genetic feedback network security policy framework was not defined properly; this made it difficult to be implemented. Genetic algorithm is an example of evolutionary computing models and optimization-type algorithm. Chromosomes, which are DNA strings, provide the abstract model for a living organism. Subsections of the chromosomes, genes, are used to define different traits of the individuals.

A policy based system has three basic steps in which it works i.e. creation, assignment and execution of policy. These depend on the network event type and requirement in terms of security.

In this paper a network security policy framework is given. In section 4 and 5 System architecture, fitness function and its parameters are also discussed. Section 7 presents the conclusion and future work.

2. RELATED WORK

[1] Have worked on improved genetic feedback algorithm based network security policy framework. This new model was much more simplified and implementable. Their future work included creating the network security framework and testing it on real data sets.

[2] Developed an evolutional network security policy framework based on genetic-feedback algorithm. Based on the historical security events, using genetic algorithm, they generated a rule base. When a new network event encountered, the analyzer judged whether the event was secure or not according to the rule base, and the policy system gave a policy decision too. Obviously, these two results differed. So the policies could be automatically adjusted, referring the genetic calculated results.

[4] Have worked on designing rule base for genetic feedback algorithm based network security policy framework using state machine. A genetic algorithm based policy management system judged the validity of network events according to the rules defined in the rule base. These rules were either IP address or some other parameters, such as port numbers etc. This paper discussed the design and benefits of rule base which is based on Finite State Machine. Since whenever a new network event came, the process of judge the event should be less time consuming. This paper introduced how FSM can be used for representing and managing the rule base of a genetic feedback algorithm based network security framework in an efficient way.

[5] Have worked on A Policy-based Management System with Automatic Policy Selection and Creation Capabilities by using a Singular Value Decomposing Technique. They described a novel method by which policies can be selected or created automatically based on events observed and knowledge learned. The relationships among the number of events and policies, orthogonal factors and cosine value are interesting topics for research.

[7] Have worked on An Artificial intelligence perspective on autonomic computing policies. They described three policy i.e. Action policy, goal policy, Utility function policy. In action policy dictate the action that should be taken whenever the system is in a given current state. Typically this takes the form of IF(Condition) THEN(Action), where Condition specifies either a specific state or a set of possible states that all satisfy the given Condition. Implementing utility function policy required optimization algorithms. Because utility functions are a function of states.

3. GENETIC ALGORITHM

A genetic algorithm (GA) is a computational model consisting of five components:

1) Starting set of individuals (chromosome), P
2) Crossover technique.
3) Fitness function.
4) Mutation algorithm.
5) Algorithm that applies the crossover and mutation techniques to P iteratively using the fitness function to determine the best individuals in P to keep. This algorithm replaces a predefined number of individuals from the population with each iteration and terminates when some threshold is met. As shown in Algorithm 1.

```
Input:
        P    //Initial population
Output:
        P'   //Improved population
Genetic algorithm:
  repeat
    N=|P|
    P'=Ø;
      repeat
      $i_1$ ,$i_2$ =select (P);
      $o_1$ ,$o_2$ =cross ($i_1$ , $i_2$);
      $o_1$ =mutate ($o_1$);
      $o_2$ =mutate ($o_2$ );
      P'=P'U {$o_1$ ,$o_2$ };
      until |P'|=N;
      P=P';
  until termination criteria satisfied;
```

Algorithm.1: Working of Genetic Algorithm

Initially a population of individuals, P is created. Although different approaches can be used to perform this step, they typically are generated randomly. From this population, a new population, P', of the same size is created. The algorithm repeatedly selects individuals from whom to create new ones. These parents i_1, i_2 are then used to produce two offspring, o_1 ,o_2 using a crossover process. Then mutants may be generated. The process continues until the new population satisfies the termination condition.

In genetic algorithms, reproduction is defined by precise algorithms that indicate how to combine the given set of individuals to produce new ones. These are called crossover algorithm. Given two individuals parents from the population, the crossover technique generates new individuals (offspring or children) by switching subsequences of the strings. Table 1 illustrates the process of crossover.

Single crossover

Table 1: Reproduction (Crossover Algorithms)

000 | 000	000 |111
111 | 111	111 |000
Parents	Children

Multiple crossover

000 |000|00	000 |111 |00
111 |111|11	111 |000 |11
Parents	Children

The locations indicating the crossover points are shown in the Table 1 with the vertical lines. In Table 1 a crossover is achieved by interchanging the last three bits of the two strings. In part b the center three bits are interchanged.

As in nature however, mutations sometimes appear, and these may be present in genetic algorithms. The mutation operation randomly changes characters in the offspring. A very small probability of mutation is set to determine whether a character should change.

4. SYSTEM ARCHITECTURE

The network security policy framework and analysis consist of following components:-

Gene Designer- Gene designer is used to get input as a network packet. The properties can be source and destination IP address, port number, size of packet, in case of security breach the level of threat and damage caused, depending on type of security breach.

Genetic Operation Unit- In this component model genetic operations such as crossover, mutation, and selection are applied to the initial set of population selected by the administrator.

Gene Pool-In this component the all the gene selected during genetic operation based on their fitness score are stored along with their fitness value for future references.

Gene Comparator- In this component the gene generated by gene designer is compared with the gene present in the gene pool. If the gene is present in the gene pool then the output of the component is forwarded to Event action model. If the gene is not present then the fitness value of the gene is calculated in the fitness calculator.

Fitness Calculator- Here the fitness of gene is calculated and if the score is more than the threshold value decided for the fitness function then the gene is added in the gene pool and the output is sent to the network report generator.

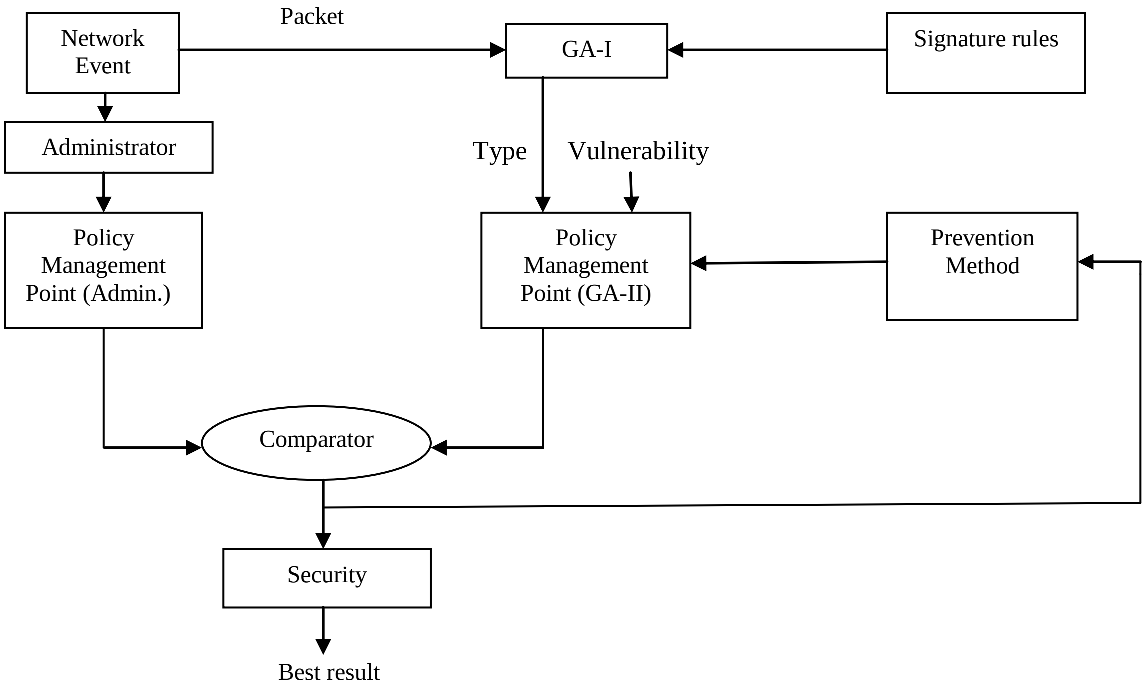

Fig. 3: Network security policy framework and analysis

Here we used genetic algorithm (GA-I) for packet analysis and (GA-II) for policy selection and prevention method.

In packet capturing JGA (API) tool is used for implementation for this paper.

Authentication:-Authorize user can access the account. Logically it will be checking for three times. Otherwise incorrect username and password will be displayed on to your console.

Data confidentiality:-The protection of data from unauthorized disclosure.

Front end: - java

Back end: - mysql

5. FITNESS FUNCTION

There are many parameters can influence the effectiveness of the genetic algorithm. The evaluation function is one of the most important and difficult parameters in genetic algorithm. First we define a formula to calculate whether a field of the connection matches the pre-classified data set. In a chromosome header fields are taken. Data type can be character, integer, double or an object. If particular packet has been matched to a number of rules decided and we are calculating difference zero or one, then best delta value can be computed.

$$\text{Match\_Value}=\sum_{i=0}^{n} \text{match*weight}_i$$

Where "n" is the number of genes present in each chromosome. In our case gene means property that each network packet is to be checked for, here each network packet is equal to a chromosome.

Some of the properties which might be considered as gene for a network event are as follows:

1. Source IP address.
2. Destination IP address
3. Source port number
4. Destination port number
5. Size of packet
6. Number of hops between the source and destination.
7. Time to Live (TTL)
8. Packet type
9. Payload
10. Checksum
11. Sequence number

In time to live, we consider which OS packet has been sent for particular ms (i.e. millisecond). Otherwise it will be blocked. No. of hop means how many router or host we are considering? The above properties are used to decide the network event fitness value. With each instance of a property there will be some weight associated. The fitness score of the event will be decided by the adding the weights of all the properties. If the fitness value of the event is greater than equal to the threshold value decided, then the event is considered fit. We check if a particular packet has been sent or not in some millisecond.

Here, THREAT is taken as the fitness function. So the THREAT value of each event is been calculated and if it is above the threshold decided, then the packet is considered as dangerous.

$$\text{THREAT\_VALUE}=\sum_{i=0}^{n} \text{THREAT\_MATCH*WEIGHT}_i$$

In a third case we have taken m is number of packet and calculate attack value.

$$\text{Attack value}=\sum_{i=0}^{m} \text{TV}$$

TV indicates threat value.

We can optimize the time and improve the accuracy by applying network security framework and analysis. Policy automation is also done in this approach.

6. CONCLUSION AND FUTURE WORK

This paper introduces a new network security policy framework and Analysis. This new model is much more simplified and implementable. We have tested it on real data sets available on internet with some automated tool software. In our model approximately ten signature rules are predefined. As future expansion in this work, latest attack prevention can be dealt with.

7. REFERENCES

[1] "Improved Genetic Feedback Algorithm Based Network Security Policy Framework" by Atish Mishra, Arun Kumar Jhapate, Prakash Kumar Second International Conference on Future Networks on page 8-10.http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?armunber=5431893

[2] Chen Xiao-su Wu Jin-hua Ni Jun this paper appears in Wireless Communications, Networking and Mobile Computing, 2007 WiCom 2007.Internatinal Conference on page 2278-2281.

[3] "An Improved Interactive Genetic Algorithm Incorporating Relevant Feedback" by Hang-Fei Wang, Xu-Fa Wang and Jia Xue.

[4] "Designing Rule base for Genetic Feedback Algorithm Based Network Security policy Framework using State Machine" by Atish Mishra, Arun Kumar Jhapate and Prakash Kumar on pp 415-417 International Conference on Signal Processing Systems,2009.

[5] "A Policy-Based Management System with Automatic Policy Selection and creation capabilities by using a Singular Value Decomposition Technique" by Hoi Chan; Kwok, T.; IBM Thomas J. Watson Res. Center, Hawthorne, NY on June 2006.

[6] "Storing Scheme for State Machine Based Rule Base of Genetic Feedback Algorithm Based Network Security Policy Framework Depending on Memory Consumption" by Atish Mishra, Prakash Kumar on 2009 at IACSIT vol, 3 Singapore.

[7] "An Artificial Intelligence Perspective on Automatic Computing Policies" by Jeffrey O. Kephart and William E.Walsh IBM Thomas J. Watson Research Center Yorktown Heights, New York 10598.

[8] "Reinforcement Learning: A Survey" by L.P.Kaelbling, M. L. Littman, A. W. Moore.

[9] "Improved Algorithms for Finding Gene Teams and Constructing Gene Team Trees" by Biing Feng Wang, Chien-Hsin Lin.

[10] RFC 2573, "A Framework for Policy-based Admission Control", http://www.faqs.org/rfcs/rfc2753.html.