An Intrusion Prediction Technique Based on Co-evolutionary Immune System for Network Security (CoCo-IDP)

Mohammad Reza Ahmadi
Iran Telecom. Research Center (Email:m.ahmadi@itrc.ac.ir)
(Received Sep. 1, 2008; revised and accepted Feb. 17, 2009)

Abstract

Forecasting the unknown and detecting the known threats[1] and targeted attacks[2] are the most concern of network security especially in large scale environment. We have presented an intrusion[3] detection and prediction system using cooperative co-evolutionary immune system for distributed data networks. This is an intelligent technique based on genetic algorithm and co-evolutionary immune system where the detectors can discriminate the existing incidents[4] and predicting the new incidents in a distributed environment. We have prepared a prototype of CoCo-IDP[5] in a Jini platform running grid computing[6] in distributed systems. Evaluation results show that, the CoCo-IDP can adaptively converge for the best answer and can detect or predict the incidents in a selected boundary. Moreover, the system generates the flexible detectors with diversity in a variable threshold. In comparison with pure Immune System (IS), the obtained results show that the proposed system has simpler rules, more powerful detection and prediction capabilities with high accuracy metric. We have compared the probability of detection and false accuracy rate in KDD[7] database with several well known methods for proof and validation of our results.

Keywords: Cooperative co-evolutionary algorithm, grid computing, intrusion prediction system

1 Introduction

Intrusion detection systems seek to examine all the traffic in a network or system and determine if an attack is in progress. Intrusion prediction systems forecast the unidentified intrusions and try to prevent a compromise before any real damage can be done. As the technology has progressed, the lines between intrusion detection and prediction have blurred somewhat, because traditional detection systems have incorporated the capability not only to alert and advise but to take pro-active steps to prevent a compromise. Traditional detection solutions are necessary to prevent the transfer of malicious codes, but are not sufficient to address the new generation of threats and targeted attacks. Security solutions that proactively protect vital information assets in real time, without waiting for new signature creation and distribution, are needed. The definition of IDP that we are going to use is a system which has ability to detect the known attacks and predict the unknown attacks to prevent the new attacks from being successful [3, 7]. The traditional form of IDS and prevention systems are either signature-based or anomaly-based. Both require updates to maintain their signature database or they must have a period of time to develop a behavioral baseline to identify accurately "suspicious" or anomalous activities [8]. In a different approach, the neural networks have employed to train the intelligent detection systems for recognition of normal activities [15]. In similar research, immune system is a new approach for provisioning of network security with strong recognition capabilities [13]. Forrest, et. al. introduce an evolving models of immune system in solving the optimization problems [2, 4]. This method is closely related to genetic algorithms (GA) to discriminate between self[8] and non-self[9] in a limited scale environment. In a different approach, Potter has explored several CoCo algorithms that improve optimization functions. He presents a novel approach for concept learning in case that

[1] The possibility of trouble or danger.
[2] The process of accessing an uninvited visitor who snoops or vandalizes the system/web site.
[3] A malicious traffic.
[4] An individual event that might trigger an attack.
[5] A cooperative co-evolutionary intrusion detection & prediction system.
[6] A parallel processing approach in large scale computational problems over a network of multiple distributed computers.
[7] The fifth international conference on Knowledge Discovery and Data mining.
[8] A pattern that normally occurs in a protected system (body).
[9] A foreign pattern which is generated by viruses, bacteria or other components in an un-healthy body.

a specific model of co-evolutionary GA is used to create a limited scale immune system [10, 11]. Here, we introduce a new prediction system based on genetic algorithm with Co-Co model using grid computational technique in a distributed environment [1]. The motivation for this approach lies on integrating the immune system and cooperative co-evolutionary concept using an agent based grid computing in a distributed environment as an intelligent solution for tracking and predicting the un-wanted security threats in a large scale data networks. The rest of paper is organized as follows: Section 2 explains the CoCo-IDP model. Section 3 presents the system parameters. Section 4 focuses on prototype systems. Section 5 shows the performance evaluation. Finally, we conclude the paper in Section 6.

2 CoCo-IDP Model

In this section, we introduce architecture of our cooperative co-evolutionary IDP system. In continue the structure and evaluation parameters are discussed.

2.1 The System Architecture

Immune system is a biological model which is applicable to many networking and security problems. Genetic algorithms are successful method for optimization problems. In GA, the population repeatedly modifies with genetic operators in a search space and seeking for the answer with the best fitness. GA initializes a population to random individuals of digital values and over successive generations, the population "evolves" toward an optimal answer. On the other hand, the co-evolution algorithm is an extended version of GA with multiple groups of populations. Moreover, the cooperative co-evolutionary method includes several genetically isolated groups that evolve in a parallel model. The individual member from each group collaborates with other members through a representative population[10] and improves their fitness according to a specific objective function [1, 5, 6, 14]. We have proposed an architecture for CoCo-IDP using JiniGrid environment in a distributed media as it is shown in Figure 1. This system consists of several set of separated worker agents where each set manage an individual group and they are coordinating by a master agent. The number of computational process depends on network model and incident classification. Each CoCo-IDP class concentrates on a specific set of incident (e.g. teardrop, buffer overflow and so on) in DoS class. The implemented system is composed of a cluster of servers where all tasks are distributed in different hosts with sharing capability. Based on Figure 1, the initial population is stored in a population pool. This population is divided into several sub-populations, where each group evolves in its host and cooperates with other host's members. In the first level of process, a particular co-evaluation algorithm in each group cooperates with other groups under master agent management. Each group generates a set of representative which has the best fitness with all pre-selected incidents. The final set of representative will generate from the entire representative set in the second level of the process. As a result, based on final representative set, the best detectors for all known and unknown incidents are generated and they are stored in the master agent. This is the procedure for generating the detectors in CoCo-IDP system.

[10] A set of the best individuals in a group.

2.2 The System Structure

In this part, we introduce the structure of our proposed method and mathematically analyze the system parameters based on cooperative co-evolutionary algorithm [1, 10, 11]. In co-co algorithm, when a detector evolves in one group, it cooperates with members in other groups. Thus, the selected representative members have the best fitness compared to other members. As a result, the system will converge to non-similar detectors in a population pool. This diversity is very important for concentrating on different types of incidents. In our prototype model, we have prepared a distributed system supporting several groups of traffic. Any group may experience a set of incidents. The goal is to generate a set of detectors which can trace the whole traffic and detect all the known and predict all the unknown incidents. In practice, each detector set focuses only on a particular class of incidents (i.e. DOS, Probing and so on) and ignores the rest of traffic. This process will repeat for all type of the existing classes. Collection of the detectors which are generated for detection forms the detector set. This step completes the generation phase and prepares the system for operational step. In operational process, the algorithm will trace the whole traffic and investigates for the events which have the best matching with existing detectors. The events which have an acceptable level of fitness with existing incidents form the detected incidents. On the other hand, the events with acceptable fitness which are not in the existing incidents form the predicted incidents. To implement this scenario, we have proposed a triple segment string schema in creation of detectors. A creator of detector (C-Detector) consists of 8 bits threshold, 64 bits pattern of binary, 64 bits mask field and 4 bits incident type. In order to create a detector, the mask filed applies to the pattern field where the value of "1" generates a corresponding bit and the value of "0" generates a don't care (X) bit. Thus we obtain the detectors with three fields including the threshold field, 64 bits pattern field and finally the incident type field. This structure will model a detector which binds a family of events with common characteristics. We consider an even pattern as a string of 64 bits. The content of string

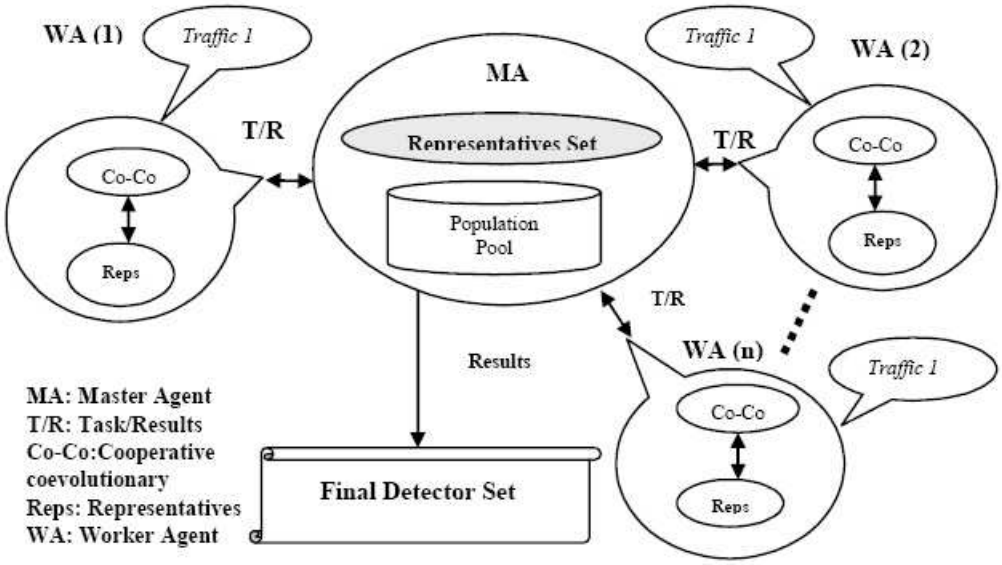

Figure 1: The architecture of CoCo-IDP in a distributed environment

includes a real traffic or existing incidents as follows:

$$\begin{aligned} T_a &= \{e_1, e_2, e_3, e_4, \dots e_a\} \\ T_b &= \{t_1, t_2, t_3, t_4, \dots t_b\} \\ Traffic &= T_a + T_b \\ &= \{e_1, e_2, e_3, e_4, \dots e_a\} + \{t_1, t_2, t_3, t_4, \dots t_b\} \end{aligned}$$

where T_a represents group of incidents e_i with "a" members, and T_b represents groups of normal traffic t_i with "b" members. Also, the d_{ij} represent a detector and the D_i represents a set of detectors:

$$D = [D_i]_{a\times n} = [d_{i,j}]_{a\times n}.$$

Here, "n" is the maximum number of detectors for each incident and this value may not be the same for all incidents. The goal is to find the best set of detectors (M) with m_i members (representative detectors) where the m_i has the best fitness in set of D_i and can obtain the maximum successful detection rate. To select the members with the best fitness, we should calculate the match-strength factor. We define S as the match-strength factor between two binary strings of x and y where $x \in D$ and $y \in (T_a$ and $T_b)$ with size of 64 bits. The value of S can simply obtain by comparison of similar position bits in x and y based on the following equation:

$$S(x,y) = \sum_{i=1}^{l} \begin{cases} 0 & if\ [x_i] \neq [y_i]\ or\ x_i \equiv X \\ 1 & else. \end{cases}$$

Where l=64 and X refers to the do not care values. In order to find the maximum fitness for incident e_j, we obtain a member which has the maximum match-strength based on the following equation:

$$S_{max_i}(D_i, e_i) = S(d_{ij}, e_i)|_{S(d_{ij},e_i)\geq S(D_{ik},e_i)}$$
$$\text{for } i = 1 \cdots a \ \&\ \text{for } j,k = 1 \dots n,\ k \neq j.$$

In set of detectors which are generated for existing incidents, representative members give a set of members which have the best fitness for each incident. We assume m_i has the maximum match-strength (best fitness) in set of detectors for incident e_i where:

$$M = \bigcup_{i=1}^{a} m_i.$$

In this process M defines a set of detectors with the best fitness.

3 The System Parameters

For evaluation of our proposed system, we have introduced the probability of detection ratio (PDR) and false accuracy ratio (FAR) as the two evaluation parameters. Probability of detection shows the successful detection of a real incident where the probability of false accuracy ratio refers to the probability of selecting a normal traffic instead of an incident or vise versa. In order to calculate the probability of detection ratio, we have defined the following parameters:

T is a threshold level which is obtained by dividing the decimal value of threshold field in C-detector to 255. The result gives a real value between zero and one.

Hit is a detected target incident which refers to summation of all detected yes-incidents in set of T_a.

We obtain the hit value based on the following equation:

$$Hit = \Sigma_{i=1}^{a} \begin{cases} 1 & if[S_{max_i}(M, e_j)/z(m_i)] \geq T(m_{i)}; \\ 0 & else. \end{cases}$$

Here $T(m_i)$ is the threshold for member m_i and $Z(m_i)$ is obtained based on the following equation:

$$Z(m_i) \quad = \quad \Sigma_{j=1}^{64} \begin{cases} 1 & Maskbit(j) = 1 \\ 0 & else. \end{cases}$$

Probability of detection ratio is calculated based on Equation (1).

$$PDR \quad = \quad Hit/a. \tag{1}$$

On the other hand, the false rate (F) is sum of the non-incidents that are detected as the incidents based on the following equation:

$$F = \Sigma_{i-1}^{a} \Sigma_{j=1}^{b} \begin{cases} 1 & if[s(m_i, t_i)/Z(m_i)] \geq T(m_i) \\ 0 & else. \end{cases}$$

The probability of false ratio is calculated based on Equation (2).

$$EFR = F/b. \tag{2}$$

The false accuracy ratio (FAR) is obtained based on the following equation:

$$FAR = \frac{PFR}{PFR + PDR}.$$

Equations (1) and (2) are defined the two important evaluation parameters for detection procedure. For system evaluation, we have considered the successful ratio based on the following equation:

$$SR = PDR - FAR.$$

4 Prototype Systems

In this section, we have explained the technical specification of IS and CoCo-IDP prototype systems with more details. We also emphasis on structure of the detectors and describe how the detectors are generated.

4.1 The IS Method

In IS prototype model, the system has implemented based on genetic algorithms. The detectors are mapped into the same form of symbolic representation as CoCo-IDP system, while the IS system evolves in a fixed threshold. The detectors are generated and stores in the system during the initialization phase. In operational phase, the system traces the events which have the best fitness with generated detectors in a fixed threshold. When a detector matches or predict an incident, the detection process will activate. In a successful procedure, the results will store in the system; otherwise, it will remove within a time interval. This technique creates strong pressure for matching and discrimination between incidents and normal traffic. Moreover, the process distinguishes between the known and the un-known incidents. As a result, during the detection process, the detectors compete for successful detection/prediction where the best detectors will win the competition process.

4.2 The CoCo-IDP Method

In CoCo-IDP model, we have implemented the prototype system in a Jini-Grid platform running in a distributed environment. Table 1 shows the training and testing incidents which are using for detection and prediction processes. In training phase, the system generates necessary detectors to detect the existing training incidents. In the next step, the representative detectors are generated for detection and prediction in each class. The generated detectors apply to the system and prepare the system for detection and prediction procedure. It is possible to extend the system for any arbitrary incident types and concentrate on any known and unknown incidents. We have considered a set of 22 incidents with 31 features in data packets based on tables one and two. For implementation, the features and descriptions are interpreted into 64-bit binary string (event pattern). We have translated each feature issue to a binary coding where we have used the set of data in KDD database which are loaded in our local database [9]. In our assumption, we have considered four classes of incidents and the system generates a set of dedicated detectors for each individual incident. During the second phase of generation, among the dedicated detectors, the system generates a set of representative detectors for each class. It should be noted that the set of representative detectors have the best fitness among all detectors in the class. The representative detectors have the following characteristics: First, they can detect all the training incidents in related classes. Second, they can predict the testing incidents in each class. As a result, the set of representative detectors in all classes are able to detect the training incidents and predict the testing incidents in the system. For more details about two categories of incidents, please refer to Table 1.

4.3 Structure of Rules and Detectors

In order to detect the incidents, we should generate the detectors based on technical specification and features of each incident. In this section, we explain how the detectors are generated in CoCo-IDP and IS systems. In specification of CoCo-IDP system, each detector consists of different fields with related features (type of protocol, duration, service, etc). All the fields, features and descriptions are presented in Table 2. We have considered 31 fields with related features where each feature needs one, two, three, four, six, eight, or 16 bits for representation. Then we have converted the information to a separate rule-based representation using straightforward mapping. The CoCo-IDP system is operational in Jini-Grid platform based on the algorithm presented in Figure 2.

It should be noted that the detectors in IS system are also generated in a similar method. We have used four classes of attack in KDD data set for training the system. Moreover, in CoCo-IDP a variable detector-based threshold method is employed where in the IS system, all detectors have a fixed threshold value. To compare

Table 2: Feature specification and descriptions

Field	Feature Name	Description	No. Bits
1	duration	Length (N. of sec.) of a connection	3
2	protocol_type	Type of protocol, e.g. Tcp, Udp, etc.	2
3	service	Network service, e.g, http	6
4	src_bytes	No. of data bytes from S to D	16
5	dst_bytes	No. of data bytes from D to S	16
6	flag	Normal or error status	4
7	land	1 if connection is f/to the same host	1
8	wrong_fragment	No. of wrong fragments	3
9	urgent	No. of urgent packets	3
10	hot	No. of "hot" indicators	3
11	num_failed_logins	No. of failed login attempts	4
12	logged_in	1 if successfully logged in, else 0	1
13	num_compromised	No. of "compromised" conditions	3
14	root_shell	1 if root shell is obtained; 0 otherwise	1
15	su_attempted	1 if su root attempted, else 0	1
16	num_root	No. of "root" accesses	4
17	num_file_creations	No. of file creation operations	4
18	num_shells	No. of shell prompts	4
19	num_access_files	No. of operations on AC files	4
20	num_outbound	No. of outbound commands in ftp se.	4
21	is_hot_login	1 if login belong to the hot list, else0	1
22	is_guess_login	1 if the login is a"guest" login; else 0	1
23	Count	No. of connections to the same host	4
24	serror_rate	% of con. with "SYN" errors	4
25	same_srv_rate	% of con. with "REJ" errors	4
26	diff_srv_rate	% of con. to the same service	4
27	diff_srv_count	% of con. to different services	4
28	srv_count	N. of con. to the same services	4
29	srv_serror_rate	% of con. with SYN errors	4
30	srv_rerror_rate	% of con. that have "REJ" errors	4
31	srv_diff_host_rate	% of con. to different hosts	4

the complexity of detectors, Table 3 measures the average number of necessary detectors and rules in the two selected methods. The values in Table 3 are obtained by running 15 iterations for twenty two different incidents in KDD database system. The mean number of detectors in CoCo-IDP is consistently less than mean number in IS. Also the CoCo-IDP evolves for average number of 1.84 ± 0.59 detectors where for the IS system is 8.69 ± 0.52. In addition, to compare the structure of rules and detectors in both systems, we have considered the Smurf attack[11] as a typical example. A set of rules which are generated during this step can only detect one type of existing incident and ignores other attacks. In the second phase, for the set of generated detectors in each class, the system will generate a set of representative detectors that can detect the known incidents (training group) and predict the unknown incidents (testing group) in the same class.

[11]The Smurf attack is a way of generating a lot of computer network traffic to a victim host. It floods a target system via spoofed broadcast ping messages.

The procedure and rules for recognition of Smurf attack in CoCo-IDP and IS system are depicted in Figures 3 and 4 respectively.

Also technical parameters and features for the matching functions are given in appendix A. It should be noted that for detecting the Smurf attack in IS system, we need at least 6 detectors where in CoCo-IDP at least two detectors are necessary.

By comparison of the two rule-set in those Figures, it is obvious that the CoCo-IDP has less number of rules with significantly less complexity while they have more diversity compare to IS system. Also the generated rules in CoCo-IDP have more flexibility and can learn with more concise information compared to the IS system. As a result, the rule sets in CoCo-IDP are more successful in detection and prediction compare to the IS rules.

4.4 System Characteristics

We have implemented the prototype CoCo-IDP system using a jini-grid platform in a set of distributed severs run-

Table 3: Mean number of necessary detectors in CoCo-IDP and IS systems

Class	Incidents	Mean Number of Detectors	
		IS	CoCo-IDP
DoS	1-Smurf	6.7±0.41	2.2±0.60
	2-Land	7.1±0.45	1.4±0.51
	3-Neptune	9.6±0.60	1.63±0.67
	4- Teardrop	9.7±0.50	1.8±0.78
	5- Pod	5.2±0.25	1.8±1.03
	6-Back	8.1±0.35	1.7±0.13
R2L	7-Guess_passd	6.8±0.40	2.0±0.66
	8-Imap	9.5±0.90	2.3±0.94
	9-Multihop	11.0±0.6	2.2±0.91
	10- Ftp_write	9.6±0.63	1.1±0.31
	11-Spy	8.7±0.71	1.0±0.00
	12-Warezclient	10.7±0.5	1.8±1.03
	13-Warezmaster	8.1±0.45	2.1±0.23
	14-Phf	9.2±0.75	1.9±0.43
U2R	15-BuferOflow	4.3±0.25	1.8±0.63
	16-Landmodule	7.0±0.62	1.4±0.69
	17- Perl	8.8±0.65	1.5±0.52
	18-Rootkit	9.4±0.64	1.8±1.03
Probing	19-Nmap	10.2±0.66	1.9±0.87
	20- Portsweep	11.0±0.63	2.7±0.67
	21-Ipsweep	9.7±0.15	2.6±1.03
	22-Satan	10.7±0.5	1.8±0.92
Average		8.69±0.52	1.84±0.59

Table 1: Description of training and testing incidents

	Training Incidents	Testing Incidents
Class	(Detection)	(Prediction)
DoS	1-Teardrop	1- Worm
	2-Land	2- Udpstorm
	3-Neptune	3- Processtable
	4-Pod	4- Mailbomb
	5-Back	5- Apache2
	6-Smurf	
Probing	7-Ipsweep	6- Saint
	8-Nmap	7- Mscan
	9-Portsweep	
	10-Satan	
R2L	11-Ftp_write	8- Named
	12-Geuss_passwd	9- Xlock
	13-Imap	10-Xsnoop
	14-Multihop	11-Sendmail
	15-Phf	12-Httptunnel
	16-Spy	13-Snmpguess
	17-Warezclient	
	18-Warezmaster	
U2R	19-Buffer_overflow	14-Snmpgetattack
	20-Loadmodule	15-Xterm
	21-Perl	16-Ps
	22-Rootkit	

ning Matlab software and supporting MDCE.[12] We have employed several Blade servers with windows operating system. Also, we have used the KDD database records with 31 fields and 132-bit length. The KDD records consist of five classes: Normal, DoS[13], R2L[14], U2R[15]and Probing[16]. In the database, the testing incidents have different probability of distribution from training incidents, and there are specific incidents in data test which do not exist in the training data.

We assume any un-known incident as a new event. This model creates a scenario very similar to a real application. In our prototype system, the data set contains 22 types of training incidents with additional 17 types of testing data. In the following section, we investigate the cooperation of members in GA for ruling the system to a converged solution. This evaluation proves that the CoCo-IDP algorithm has a solution for detection/prediction problem. Moreover, it is important to show the range of thresholds which are acceptable for the best detection results. In Co-Co algorithm, cooperation of the participated mem-

[12]Matlab Distributed Computing Engine

[13]Denial of Service attack is an attempt to make a computer resource unavailable to its intended users.

[14]Remote to Local is unauthorized access from a remote machine (e.g. guessing password).

[15]User to Root is unauthorized access to local super user privileges (e.g. various buffer overflows).

[16]Probing includes surveillance and other probing (e.g. port scanning).

```
Algorithm CoCo-IDP
Input: self /non-self set, Output: set of detectors (rules)
{
    Master agent initializes the worker agents;
    Send the self / non-self set to the worker agents;
    Create the detectors with random values [0, 1, x];
    for (each group S in worker agent)
     {
       /*P(s) =population of S, neg_sel =negative selection algorithm*/
        Initialize P(S) with neg_sel;
        Evaluate init fitness of P(S) detectors;
       }
      While (termination condition)
        {
         for (each group S)
           {
         /*run one step from genetic algorithm */
         Apply genetic operators to P(S), generate offspring detector;
         for (each detector #i in  S)
           {
         /* representative which have the best fitness*/
           Form collaboration set of #i and other representatives;
           Evaluate fitness by applying self and non-self set;
           Assign fitness collaboration to the #i detector;
           }
         }
        Master agent receives and synchronizes all the worker agents;
        Create rep-set from representations of all groups;
        Evaluate the fitness of rep-set by applying self and non-self sets;
        if(evaluation of rep-set has stagnated)
           {
            for (each group S )
             {
             Check contribution of P(s);
             if (S is unproductive)
             Remove S;
              }
           Create a new P(S);
           Initialize P(S), apply neg_sel and pruning algorithms;
           Transfer self and non-self to new worker agent;
           Evaluate fitness of new P(S);
           }
         }/*end of while*/
       Return (rep-set); /* all representative set*/
} End
```

Figure 2: The CoCo-IDP algorithm

```
Smurf-attack (incident)
{
Detector set = {
Rule 1=10111011100xxx ....1 0 xx 1 1 x 1 1 1 0 x001100
Rule 2= 0x 00x 11xx1001.....xx x x 0 x 1 x 1 0 x x 01 x 1x
Rule 3= xx xxx1111xxx11 .... x x 1 1 x 1 x x 0 x 011 100 1
Rule 4= xx x0x01111xxx .....1x 0 0 x1 1 0 0 x 1 01 x x1 x
Rule 5= xx 110100011x.x..... x x 0 x 1 1 x x 0 1 x x 11 1xx
Rule 6= x1 x00x00000xx..... x 0 x 1 x x x x 0 x 0 111 x xx
                                  };
  Threshold = 0. 7 (Fixed)
   for all rules in detector set
   {
    If Matching (Detector, Incident)>Threshold
       then Detection;
            If incident exists in testing incident;
              then Prediction;
                else Non-incident;
}
```

Figure 3: A rule-set for Smurf-attack in IS

```
Smurf-attack (incident)
{
Detector set = {
  Rule 1:1x0xxx010x0xx1100.....0x00011xx1110xx, T=0.56
  Rule 2:100x0xxx11xx11xx1..... 0x1x11100xx1xx0, T=0.87
                                  };
 for all rules in detector set
 {
  If Matching (Detector, Incident)>Threshold (i)
    {
      If incident exists in the Training Incidents
        then  Detection;
           If incident exists in testing incident;
             then Prediction;
               else Non-incident;
}
```

Figure 4: A Rule-set for Smurf-Attack in CoCo-IDP

bers in the process is an important procedure for system response. This method encourages the majority of population for cooperation and the system does not stagnate or deviate during the process [1].

4.5 Acceptable Affinity/Deviation (Variance δ) Rate

In fact, the acceptable affinity rate limits the system to an acceptable deviation rate where the maximum deviation refers to minimum matching boundary. The traditional intrusion detection system focuses on an exact signature matching or zero deviation ($\delta = 0$) for detection procedure. As the system becomes more intelligent, the acceptable deviation rate affects the decision criteria where the certain levels of similarity replace with exact matching condition. Thus, the intelligent system has capability to detect the incidents within a reasonable boundary limited to maximum acceptable deviation. In the IS system, the acceptable area is limited within a fixed threshold. The value of threshold is constant for all the incidents where outside this boundary the system is not able to concentrate for a successful detection. On the other hand, for CoCo-IDP the scenario is different and the threshold is more flexible. The threshold level is wider while the maximum level might be different for each type of incident. This is the result of technical specification and constraints in generating the system rules. As a result, the detectors in CoCo-IDP have more diversity, more capability in training and more flexibility compared to IS rules. Moreover, the detectors in CoCo-IDP are able to trace a suspicious incidents with less degree of similarity to the reference patterns compared to the IS system. This characteristic expands the border of activity for the detection algorithms and increases more opportunity for tracking the existing incidents. To show the acceptable area, we have prepared a prototype system similar to the previous model.

Figure 5 shows the maximum border (zero means 100% matching and maximum refers to limit of acceptable mismatching) for most predictive accuracy in Co-Co-IDP prototype system where the border is limited to 0.09 in IS system. The results show that CoCo-IDP has more flexibility and can concentrate on the events with less degree of similarity as well as more prediction accuracy compare

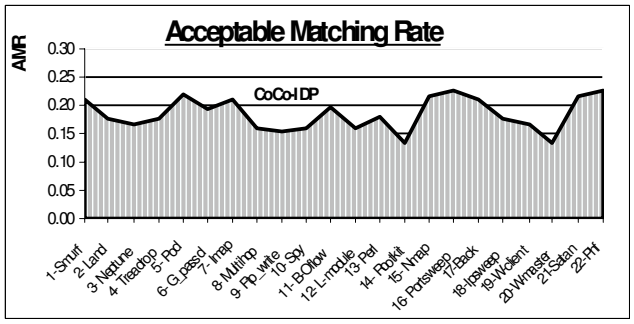

Figure 5: AMR for 22 incidents in CoCo-IDP

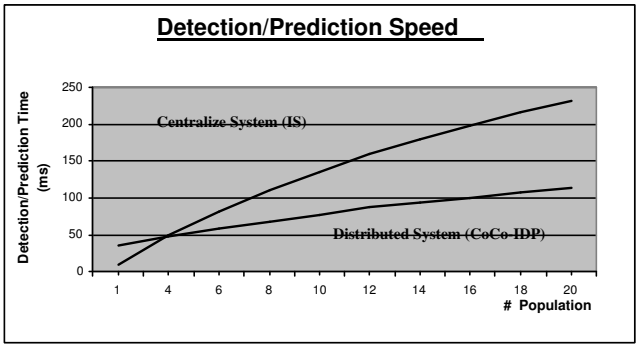

Figure 6: Comparison of detection speed

to the IS system.

4.6 Average Detection/Prediction Process Speed

The detection processing time is a key issue in any intrusion detection/prediction system. Detection period depends on several factors such as algorithm agility, processing method, accuracy rate and so on. In any real time detection system, the detection period should be negligible compared to system initialization and training phase. On the other hand, the processing technique and compromise between accuracy and agility are noticeable. In our evaluation, we have considered the IS system running by one agent in a single processor system and CoCo-IDP system running by multi agents in a multi host system (distributed environment). Figure 6 compares the detection/prediction speed for the both systems.

As it is shown in Figure 6, in a limited number of population, the centralize system has more agility for detection. On the other hand, by increasing the number of population, the CoCo-IDP detection/prediction period will increase smoothly while it will increase sharply in the IS system. This analysis confirms the effect of CoCo-IDP method for implementation in a distributed environment.

4.7 System Parameters and Actual Operational Procedure

To represent the profile, attack scenario and parameters of the system; based on all features and specifications of the known incidents (Table 2), the system creates the dedicated rules and detectors in the predefined structure and formats (Subsections 2.2 and 4.3). The set of rules and detectors are corresponding to all expected known attacks in different class of incidents. Then, the system generates representative detectors for all existing classes. With this preparation, the system is ready for actual detection/prediction process in the real network environment. The generated detectors apply to the system and prepare the algorithm for detection/prediction in the network. The system investigates in the existing traffic to find the minimum acceptable similarity between the events and detectors; in this case, the events will candidate as the suspicious incidents. Thus, the algorithm concentrates to find the best fitness member through the rule-set procedure in consecutive process of GA algorithm. Finally, the system decides a decision of detection, prediction or Non-incident based on structure of the CoCo-IDP algorithm (Subsection 4.3).

5 Performance Evaluation

In this section we have evaluated the capability of IS and CoCo-IDP systems for detection of known and prediction of the un-known incidents. In continue, we have compared the CoCo-IDP system with several well known detection methods for validation of the results.

5.1 System Evaluation

In this part, we have evaluated the capability for detection/prediction of IS and CoCo-IDP systems in operational mode. We have implemented the system based on Section 2, generated the rules and detectors based on Section 4. The final representative detectors are obtained from a set of dedicated detectors which are generated for training incidents. The selected representative detectors can cover 4 classes of incidents including the DOS, R2L, U2R, and Probing based on Table 1. The representative detectors are able to concentrate on detecting the training or predicting the testing incidents. The systems are operational in a set of distributed servers for CoCo-IDP and in a centralize server for IS model. We have applied the same data to both systems and measured the number of detected/predicted incidents.

It should be noted that the training incidents in each class can be detected by dedicated detectors where the representatives detectors are able to detect and predict all the training and testing incidents. Figure 7 compares the successful detection rate in IS and CoCo-IDP systems for 22 training incidents. On the other hand, Figure 8 compares the set of successful prediction rate in CoCo-IDP and IS systems for 17 testing incidents.

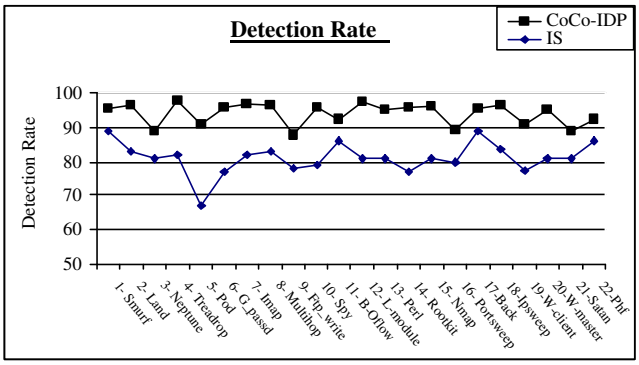

Figure 7: Comparison between detection rate in IS and CoCo-IDP systems

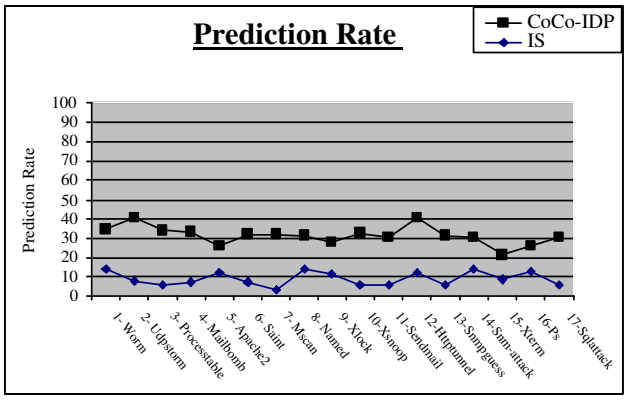

Figure 8: Comparison between prediction rate in IS and CoCo-IDP systems

The results in both evaluation show that the CoCo-IDP detectors are more successful in both detection and prediction compared to IS system. In this evaluation, the following issues are noticeable: the generated rules in CoCo-IDP have more flexibility in value of thresholds and the system can investigate in more events which have less fitness value compare to IS system. Moreover, the CoCo-IDP concentrates on members with the best fitness and selects its candidates through representative members rather than the ordinary members. Also the CoCo-IDP system can remove the non cooperative members during the detection period and does not stagnate in the recursive loops. On the other hand, the capability for cooperation in distributed environment speed up the detection process for more successful decision. As a result, for both evaluation factors, the CoCo-IDP system is relatively more successful compare to the IS system.

5.2 Confirm the Evaluation Results

Evaluation results show that the CoCo-IDP method is a successful technique for intrusion detection and prediction in a distributed system. To confirm validation of the results in detection using a standard confirmation technique, we have prepared a comparison scenario with several well known detection methods using predictive accuracy metric as well as the detection method in [1]. The predictive accuracy and false accuracy rate are two important parameters for precise evaluation of the prototype systems. In validation procedure, we have used 10-fold cross-validation technique. The technique involves randomly dividing the complete data set into 10 disjoint sets with equal size where we use one subset as a test set and the rest as the training set. We have given the predefined parameters for the random traffic and execute the procedures for several consecutive trials. In initial process, the CoCo-IDP uses a training data to learn and generate the detectors with a specific detector-based threshold. Once the process is completed, the generated detectors are applied to the system and the successful detections are measured. In experiment, each class of detector has to recognize the related incidents in a class (i.e. Smurf, Teardrop, $\cdots$ in DOS class) and ignores the other incidents as well as the normal traffic. The CoCo-IDP executes the procedures for 100 consecutive generations. Here, we execute all the selected systems for 100 trials and applied the testing data to calculate the predictive accuracy values for CoCo-IDP and other reference methods.

Tables 4 and 5 compare the predictive accuracy and false accuracy rate for several well known methods in a similar scenario based on KDD data base [12]. We have compared the probability of detection as well as the false alarm rate with other methods. The concept of our greedy technique for tracking the incidents with low matching rate will increase the suspicious events and improves the successful detection rate. It certifies the robustness of our system for detecting the incidents with relatively small fitness compare to the other available techniques.

6 Conclusions

Intrusion Detection and Prediction provide the capability of both detecting and predicting against any security threats. Detection system monitors abnormal traffic pattern and reports the suspicious events; however, it is unable to predict any unknown incident. The prediction is an intelligent process that learns and adapts the system for distinguishing the unknown threats. We have presented a detection/prediction system based on a cooperative co-evolutionary immune system and a grid computing technique in a distributed data networks. We have implemented a pure immune system (IS) and a CoCo-IDP system as the prototype models and compared the key parameters of both systems. The results show that the CoCo-IDP system is more successful in both detection and prediction with higher accuracy metric compared to the IS system. Also the system has learning capability to recognize the suspicious events with less fitness compare to the IS system. The probability of detection and the false accuracy rate in the proposed system are compared

Table 4: Comparison of the PD parameters in several methods

%	GAU	NEA	RBF	LEA	HYP	ART	C0Co-ISD	CoCO-IDP
DOS	82.4	97.1	73.0	97.2	97.2	97.0	98.2	98.41
R2L	9.6	3.4	5.9	0.1	0.1	3.7	6.8568	6.22
U2R	22.8	22	6.1	6.6	8.3	6.1	8.368	9.14
Probe	90.2	88.8	93.2	83.8	84.8	77.2	98.7417	98.73

Table 5: Comparison of the FAR parameters in several methods

%	GAU	NEA	RBF	LEA	HYP	ART	C0Co-ISD	CoCO-IDP
DOS	0.9	0.3	0.2	0.3	0.3	0.4	0.07391	0.04123
R2L	0.1	0.01	0.3	0.003	0.005	0.004	0.088	0.079
U2R	0.5	0.006	0.04	0.03	0.009	0.001	1.8433	0.98
Probe	11.3	0.5	18.8	0.3	0.4	0.2	0.3449	0.3145

with several well known methods where those comparisons confirm the advantage of CoCo-IDP system.

Acknowledgement

The author acknowledges the assistance of Mr. Davood Maleki.

References

[1] M. R. Ahmadi, and D. Maleki, "An intrusion detection technique using Co-Co immune system for distributed networks (CoCo-ISD)," *International Journal of Computer Science and Network Security*, vol. 8, no. 4, pp. 160-169, Apr. 2008.

[2] J. Balthrop, S. Forrest, and M. R. Glickman, "Revisiting LISYS: Parameters and normal behavior," *Proceedings of the 2002 Congress on Evolutionary Computation, CEC-2002*, 2002.

[3] L. Feng, X. Guan, S. Guo, Y. Gao, and P. Liu, "Predicting the Intrusion Intentions by Observing System Sequence," *Computer & Security Journal Elsevier*, pp. 241-252, 2004.

[4] M. Glickman, J. Balthrop, and S. Forrest, "A machine learning evaluation of an artificial immune system," *Evolutionary Computation Journal*, vol. 13, no. 2, pp. 179-212, 2005.

[5] I. Foster and C. Kesselman, *The Grids: Blueprint for a new Computing Infrastructure*, Elsevier/ Morgan Kaufmann Publishers, 2004.

[6] N. Guelfi, E. Astesiano, and G. Reggio, "JGrid: Exploiting jini for the development of grid applications," FIDJI 2002, pp. 132-142, 2003.

[7] K. Haslum, A. Abraham, and S. Knapskog, *Fuzzy Online Risk Assessment for Distributed Intrusion Prediction and Prevention Systems*, pp. 216-223, IEEE Computer Society Press, USA, ISBN 0-7695-3114-8, 2008.

[8] K. Hwang, M. Cai, and Y. Chen, "Hybrid intrusion detection with weighted signature generation over anomalous internet episodes," *IEEE Transactions on Dependable and Secure Computing*, vol. 4, no. 1, Jan.-Mar. 2007.

[9] *KDD Cup 1999 Data*, The UCI KDD Archive Information and Computer Science University of California, Irvine, Oct. 1999.

[10] M. A. Potter, and K. A. De Jong, "The design and analysis of a computational model of cooperative coevolution," *12th European Conference on Artificial Intelligence*, Unpublished doctoral dissertation, George Mason University, 1997.

[11] M. A. Potter, and K. A. De Jong, "Cooperative coevolution: An architecture for evolving coadapted subcomponents," *Journal Evolutionary Computation*, vol. 8, no. 1, pp. 1-29, MIT Press, 2000.

[12] M. Sabhnani, and G. Serpen, "Application of machine learning algorithms to KDD intrusion detection dataset within misuse detection context," *Proceedings of the International Conference on Machine Learning Models, Technologies and Applications*, pp. 209-215, 2003.

[13] A. Siraj, R. B. Vaughn, and S. M. Bridges, "Decision making for network health assessment in an intelligent intrusion detection system architecture," *International Journal of Information Technology & Decision Making*, vol. 3, no. 2, pp. 281-306, World Scientific Publishing Company, 2004.

[14] R. P. Wiegand, and K.A. DeJong, *An analysis of Cooperative Coevolutionary Algorithms*, A Dissertation for the degree of Doctor at George Mason University, 2003.

[15] C. Zhang, J. Jiang, and M. Kamel, "Intrusion detection using hierarchical neural networks," *Pattern Recognition*, pp. 779-791, Elsevier, 2005.

Appendix A

CoCo-IDP Detector Function:
Matching(event pattern, incident)
Protocol-type = ICMP|UDP;
Service=|ecr_i|domain_u|finger|whois|domain|hostnames|
name|time|echo|;
Src_byte = Bit(1,2,3,5)=1; Wrong_fragment=1;
Is_hot_login=1;
THEN Smurf-attack.

IS Detector Function:
Matching(event pattern, incident)
Protocol-type = ICMP|UDP;
Service=|eco_i|finger|domain_u|time|; Flag = S1|So;
Src_byte = Bit(3,4,6) = 1; Wrong_fragment = 1;
Logged_in = 1;
Root_shell = 1; Su_attemped = 1; Is_hot_login = 1;
Is_guess_login = 1; Count >= 100; Rerror_rate >= 0.5;
Dst_host_rerror_rate >= 0.5;
THEN Smurf-attack.

Mohammad Reza Ahmadi received the B. Sc. and M. Sc. degrees in Electrical Engineering and Communication Systems from K.N.T. University of Technology in 1986 and 1990 respectively. He received his Doctor degree in Communication Networks from Tokyo Institute of Technology, Tokyo, in 1997. Currently he is the project manager and researcher in IT department of Iran Telecommunication Research Center (ITRC). His research interests are network security focus on intrusion detection/prediction systems, Immune systems focus on network applications and data center design and implementation.