Intelligent Cyber Defense System

Myroslav Komar[1], Anatoliy Sachenko[1,2], Sergei Bezobrazov[3] ,Vladimir Golovko[3]

[1] Ternopil National Economic University,
3 Peremoga Square, Ternopil, 46020, Ukraine

{mko, as}@tneu.edu.ua

[2] Silesian University of Technology,
26 Roosevelta st, Zabrze, 41-800, Poland

sachenkoa@yahoo.com

[3] Brest State Technical University,
256 Moskovskaja st, Brest, 224017, Republic of Belarus

gva@bstu.by, bescase@gmail.com

Abstract. In this paper a novel method for detection of network attacks and malicious code is described. The method is based on main principles of Artificial Immune Systems where immune detectors have an Artificial Neural Network's structure. The main goal of proposed approach is to detect unknown, previous unseen cyber attacks (malicious code, intrusion detection, etc.). The mechanism of evolution of the neural network immune detectors allows increasing the detection rate. The proposed Intelligent Cyber Defense System can increase the reliability of intrusion detection in computer systems and, as a result, it may reduce financial losses of companies from cyber attacks.

Keywords. Artificial Neural Networks, Artificial Immune Systems, Malicious Code Detection, Intrusion Detection, Intelligent System, Cyber Attacks, Cyber Defense Financial Losses

Key Terms. Software System, Research, Cyber Defense, Intelligent System

1 Introduction

The up-to-date computer system cannot be imagined without safety equipment. The work in the Internet is accompanying by high risks to be attacked by network intrusions and malicious code. As a result, cybercrime continues to do more financial damage to companies: a company's costs for preventing cybercrimes are estimated approximately $ 15 million per year. Thus, the costs of each company may vary from $ 1.9 million to $ 65 million per year. In absolute terms, the damage from cyber at-

ICTERI 2016, Kyiv, Ukraine, June 21-24, 2016

tacks increased by 82% in the last six years [1]. In 2011, the direct annual global losses from cybercrime were estimated $ 114 billion. Taking into account the financial losses of companies from cyber attacks and the costs of downtime and recovery, cybercriminal activity is worth of $ 388 billion per year to the world economy [2].

According to the opinion of Forbes experts [3], one of the high-profile crimes in the field of information security was the Anonymous attack on the MasterCard, Visa and Paypal payment systems in late 2010. Damage from this attack was worth of $ 5.5 million. The other high-profile cybercrime was an attack on Citibank in June, 2011. Hackers had stolen $ 2.7 million from the accounts of 3,400 customers of the bank. Breaking into Sony PlayStation Network took place in April, 2011. The total damage to the companies was estimated $ 171 million. As a result of hacking, there was a leakage of confidential information of 138,000 Internet users.

According to this, the costs for preventing cyber attacks are increasing. In average, the consequences of the attack can be eliminated during 46 days. Companies, participating in the research, in average, spend more than $ 1.9 million during this period. Thus, there is a growth of costs to 22% in comparison with 2014, when the amount of the costs has averaged to $ 1.5 million during 45 days [1].
As it states above an urgent problem is to have effective methods defending against cyber-attacks.

2 State-of-the-Art

Depending on used techniques, experts define four basic types of network attacks: denial of service attacks, user-to-root attacks, remote-to-local attacks and probe attacks, and several subtypes of these attacks [4].

Nowadays many methods for solving the problem of network attacks detection where developed. The essential part of these methods is based on artificial intelligence such as: artificial neural networks, methods of fuzzy logic, artificial immune systems.

Intrusion Detection Systems (IDS) on the ANN can be divided into four categories [5]. The first category (earlier studies) of IDS is based on Multi-Layer Feed Forward Neural Network (MLFF) [6, 7], such as the Multi-Layer Perceptron (MLP) [8, 9] and Back Propagation (BP). The second category of IDS is based on Cerebellar model Articulation Controller (CMAC) [10] neural networks and Elman neural networks [11, 12]. The third category of IDS is based on unsupervised learning the neural networks for classifying and visualizing the input data to recognize the normal behavior from abnormal one. Most systems in this category use the Kohonen Self-Organizing Map (SOM) [13]. The fourth category of IDS is based on hybrid neural networks [14, 15].

Intrusion detection models on fuzzy logic are using the fuzzy rules or fuzzy classifiers [16]. Dickerson et al [17] proposed a Fuzzy Intrusion Recognition Engine (FIRE) for the detecting the malicious activity in the network. Data portions are classified using static metrics and enabling to generate fuzzy rules for classification of the input network data. The main disadvantage of this approach is that the rules are cre-

ated manually, but not automatically. Moreover the process of rules generating is laborious, and it imposes serious constraints on system development.

In the field of Artificial Immune Systems several basic algorithms were proposed: Negative Selection algorithm [18] Clonal Selection algorithm [19, 20], Idiotypic Network [21, 22] and Dendritic Cell algorithm [23]. A. Perelson and S. Forrest in 1994 proposed the Negative Selection algorithm for solving the anomaly detection problems [18]. It's based on the process of lymphocytes maturation in the thymus – biological organ that plays the basic role in the human immunity [24, 25].

Despite its successful application, the negative selection algorithm has several serious weaknesses [22, 26]. The first, it needs to create a randomly-generated initial detector population. If the dimensionality of the future space increases, then a number of detectors is growing exponentially. The second, the definition of "normal" is not updated as the time progress. And the third, the negative selection algorithm can cause excessive numbers of false alerts.

F. Burnet in 1959 proposed the Clonal Selection algorithm based on the Clone Selection theory [19, 20]. This theory explains the basic response of the adaptive immune system to an antigenic stimulus during proliferations of B-cells.

In our opinion, main problems of most AIS applications for data mining and anomaly detection tasks are the complex structure of immune detectors and no representatively matching methods. For example, in [18, 27] the binary structure of detectors is employed. Such structure of detectors requires the use of a contiguous bit matching method (r-contiguous bit [18], *r*-chunks [28]) that reduces the space and time complexity. Several works [26, 29] outline the unacceptable computational complexity of such methods by reason of the exponential relationship between the size of the data set(to be used) and the number of detectors that it is possible to generate. Also Gonzalez et al. [30] showed that matching rules between two binary strings cannot represent a good generalization of a self-space, and detectors demonstrate the insufficient good coverage of a nonself-space [31].

In comparison with mentioned methods above we propose the immune detectors based on neural network. An artificial neural network is an adaptive system that changes its structure based on external or internal information, and it flows through the network during the learning phase, and it's characterized by learning capability, generalizing ability and self-organization. Implementation of ANN permits to avoid the listed weaknesses above and increases the self-adaptation and self-evolution abilities of detectors in the tasks of data mining and anomaly detection.

In this paper we investigate the ability of immune detectors with neural network architecture to adapt to the changeable software environment and self-evolution in order to detect the unknown (before invisible) threat. The adaptability and self-evolution of detectors consists in modification of its structure for increasing the detection rate of unknown cyber attacks.

3 Generalized Architecture of Intelligent Cyber Defense System

The Artificial Immune System for Cyber-defense is the set of "intelligent" immune detectors and rules that describe their behavior. The structure of immune detectors, the algorithm of their training and evolution are described in [32, 33]. The system consists of modules that perform the control of immune detectors. The immune detectors are going through the different stages during the lifetime. There is creation, training, selection, detection etc. stages. Each stage can be represented as a module of the defense system. Thus, the developed system for computer attacks detection consists of several interacting modules. Figure 1 shows the architecture of the proposed system.

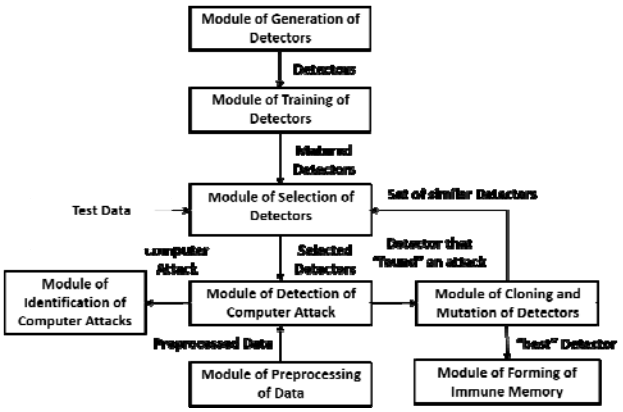

Fig. 1. Generalized architecture of intelligent cyber defense system

The module of generation of detectors produces the set of so-called pre-detectors that go through the several stages before they acquire the ability of correct classifications of objects. Every immune detector has the limited lifetime during which it "lives" in the system. At the end of the lifetime the detector is replaced by another new detector. This mechanism provide the system the continuously inflow of new immune detectors with different structure and different ability that can be more powerful than its precursors.

After creation, during the train stage, the immune detectors acquire the ability of correct classification of different objects and processes in computer environment and to detect the cyber attacks.

After training all immune detectors going through the selection stage where detectors pass the checking for correctness to minimize the erroneous work, while legitimate objects (files, processes, connections etc.) is considered as the threat. For this purpose, the preliminary created test sample – which consists only of legitimate objects – is given to detectors. If a *i*-th detector classifies one of test objects, as an attack, then it is destroyed and replaced by a new detector. If a *i*-th detector does not

generate the erroneous work during the test selection, then it is considered as a correct one and admitted to the analysis of computer environment. As a result the set of immune detectors for the analysis of environment is created, and it can be filled up due to the detectors of immune memory and generating new detectors after the end of their lifetime. The module of selection allows decrease the false alarm rate and increase the defense level.

All selected detectors can defense a computer system against cyber attacks. The set of active immune detectors forms a multi-agent system, where each immune detector is an intelligent agent with its own list of tasks. It selects the target of scanning, makes clones and evolves. At the functioning stage (or detection stage) all the information – which is getting by the computer – is primarily analyzed by immune detectors. If none of detectors found an anomaly, then data are processed by the operating system and the proper software. In addition, some period of life is given to each detector which analyses the environment during given period. If upon termination of given time the detector didn't find an anomaly, it is destroyed, and a new detector is created but on its place. If the any object is classified by immune detectors as an attack, then such detectors react on this attack. For example, they can block the proper connection, and, as a result, it is not processed by the operating system and software. The user receives the message simultaneously about the attack attempt on the computer system.

If detector found a threat then processes of cloning and mutation are activated. The goal of cloning and mutation module is to produces copies of the immune detector that found an attack. Such "clones" that are similar to the "parent" are very useful for example for defense against the family of malware, where each example have the similar malicious code. Such clones are capable to react on the found malware and check all the objects in the computer environment in a short period.

When clones are creating, some changes in their structure are taking place. As a result, the clones are not exact copies of the parent but with small differences. This process is called mutation. It allows immune detectors acquire the new ability, adapt to new attacks and increase the detection rate. In our case, when detectors base on neural networks, each clone is training on information that is abstracted from the detected attack. It allows adapting to new attack with the purpose of increasing of quality detection.

During the detection and eliminating of attacks, it is expedient to save their parameters and samples with the purpose of further detailed analysis. The point is that immune detectors are trained on the limited set of data, which can not include all possible cyber attacks. Therefore at the samples of attacks, that was classified as an unknown, are saved and added to the trained sample. It enables to increase the authenticity of attacks detection and classification as well as provide the flexibility of system, so this process updates the information. Newly generated detectors will be trained already, on new data.

At the end, the best detector is chosen and transformed to "memory" detector. Memory detectors have unlimited lifetime and provide a quick reaction on repeated cyber attacks. Thus the set of memory detectors forms the "immune memory" and keeps the information about all met cyber attacks and provides the high level of reaction on repeated attempts of attacks.

Finally, the module of identification of threats is used for classification of detected threat. The knowledge about the class of the detected threat allows taking correct response.

4 Results and Discussions

4.1 Ability of Artificial Cyber Defense System Detecting the Network Attacks

In order to eliminate disadvantages of existing works and improve the reliability of detecting the network attacks authors developed the intrusion detection systems using the proposed methods above (Fig. 2).

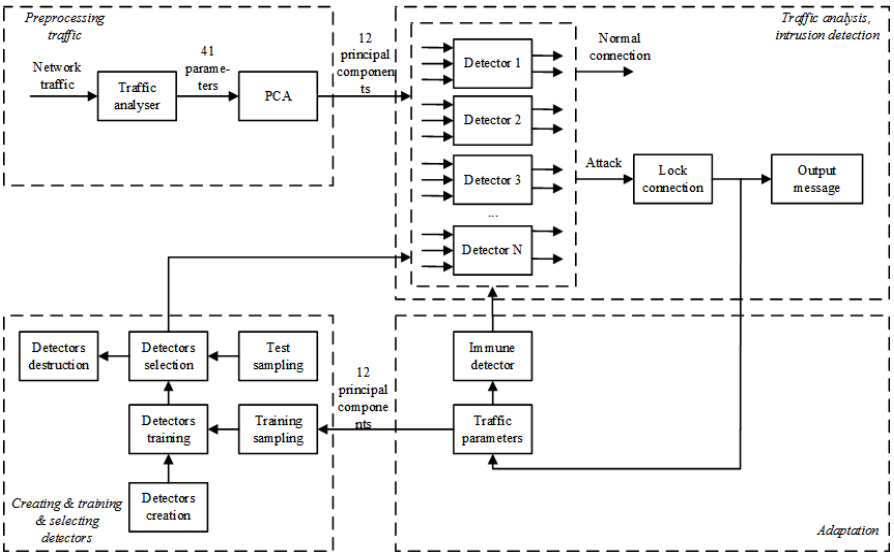

Fig. 2. Structure of Intrusion Detection System

Preprocessing traffic module is designed to represent traffic parameters in a convenient form for analysis. It consists of two modules – module capturing network traffic on the computer network and module of the principal components calculation from parameters of the captured traffic.

To capture the network traffic there used the specialized software called a sniffer – a software network analyzer of the traffic designed to capture and make the subsequent analysis of network traffic.

The network traffic – captured by sniffer – is analyzed. As a result there are extracted the 41 parameters of the network connection which characterize this connection and include time of the connection work, the protocol type, service type, a number of transferred bytes, etc.

Note the different parameters of the network connection have a different type of information, for example, the parameter "time of connection work" is set in seconds,

"protocol type" is set in a symbolic form, and "number of bytes from source to receiver" is set in bytes. Hence, in order to analyze such heterogeneous data, they have to be reduced to a general view.

To reduce the dimension of the analyzed data the unit PCA is used (see Fig. 2), performing the principal components selection and leading to improved quality of network attacks detection as well as increasing the analysis speed of network packets.

The PCA unit receives data from the sniffer – 41 parameters for network connections and, after all transformations and calculations, forms the 12 principal components. It was experimentally proved the 12 major components of network traffic parameters contain more than 99% information (Table 1).

A module of creating and training and selecting detectors (see Fig.2) is designed to create neural network immune detectors 1…*N*, which are considered as the basic elements of Intrusion Detection(each individual immune detector represents an artificial neural network).

Table 1. Distribution of information according to the components

Number of components	1	2	3	4	5	6
The amount of information, %	52,40	71,67	88,37	91,49	94,21	95,90
Number of components	7	8	9	10	11	12
The amount of information, %	96,96	97,71	98,27	98,73	99,00	99,18
Number of components	13	14	15	16	17	18
The amount of information, %	99,33	99,47	99,59	99,67	99,75	99,81
Number of components	19	20	21	22	23	24
The amount of information, %	99,87	99,90	99,93	99,94	99,95	99,96
Number of components	25	26	27	28	29	30
The amount of information, %	99,97	99,98	99,98	99,99	99,99	99,99
Number of components	31	32	33	34	35	36
The amount of information, %	99,99	99,99	99,99	99,99	99,99	99,99
Number of components	37	38	39	40	41	
The amount of information, %	99,99	100	100	100	100	

Unit of detectors creation is assigned to generate neural networks that are considered as the basis of the detectors. The input of this unit is supplied data such as: the number of neurons in the first layer and the number of neurons in the hidden layer. The unit then generates a neural network with the given parameters and initializes the weights coefficients between neuron elements according to a random distribution. Such neural network is going in the input of training detectors unit, whose task is to train how to classify images of normal network connections and network attacks. For this purpose there (in input of training detectors) are appeared data from the training set which are selecting in the random way. Therefore, such data are unique for each neural network that provides the great structural diversity of neural immune detectors.

To train the one neural network data are used, consisting of 64 connections from one of four classes of network attacks (that is, 80 percent of all training data for the neural network), and 16 connections belonging to the class of legitimate network con-

nections (representing 20 percent of the training sample). This relation of classes in the training set was obtained by experiment and showed the best results.

As a result, each neural network is trained on parameters of the 80 network connection. While a neural network is training, there is used the controlled competitive training in accordance with the rule of "winner takes all". I.e. data from the training set – formed exactly for this neural network – are supplied to its input sequentially. Then, in the output of the neural network the weights coefficients are adjusted depending on the coherence of submitted data. Trained neural networks must pass the verification process to prove their correctness at the classification of the various images for the network traffic. A verification function is running by the unit of selection detector (see Fig. 2).

The trained neural network is checked on a specially prepared test sample consisting of the parameters of legitimate network connections. The neural network (NN) is analyzing and classifying data of the test sample. If NN detects a network attack then such NN is considered as incorrect one (because the test sample contains the legitimate traffic only) and it's destroyed by detectors destruction unit (see Fig. 2). If NN does not detect network attacks in the test sample then it "passes" the selection phase and transforms into immune neural network detector and implements into module of the traffic analysis and intrusion detection.

Verification mechanism of NN functioning – passed a training stage – eliminates the beta errors i.e. those errors when the legitimate connection is classified as a network attack. Neural network immune detectors that have successfully passed the selection stage of training are considered as the basis of the module for the traffic analysis and intrusion detection. As it comes from above an individual NN immune detector is adjusting to detect network attacks of a particular type the set of such detectors provides intrusion detection belonging to any of the classes.

To analyze the network traffic the 12 parameters of network connection are going from the module of preprocessing network traffic to the input of each NN functioning immune detector in parallel. Detectors analyze these parameters and make a decision. If all the detectors identified analyzed connection as legitimate one (not attack), it is permitted for processing and implementation. If one of the detectors at least classified the current connection as a network attack, it is blocked and the message comes about network attack.

Each neural network immune detector has the so-called "lifetime", during which it can analyze network traffic. This limited period of detectors existence is needed to get rid of the "weak" detectors. Because there is no guarantee the detector is able to detect network attacks after the training and selection of neural network detectors. More precisely, it may be a situation where the detector will be classify an unknown image as an attack in a case when the image is exactly the same as the training sample.

This situation may arise due to the fact that the training sample is randomly generated per each detector. Then the situation may occur when the neural network – which is a base for the detector on the training data – cannot identify patterns in the parameters of compounds belonging to different classes.

A mechanism that limits the operation time gets rid of the "useless" detectors, so such detectors are destroyed if they did not recognize the attack within the specified

time. A new immune detector – which was just trained and selected – substitutes the destroyed one.

The modern system of the Intrusion Detection has to protect from known network attacks as well as from unknown ones which are not previously encountered. In other words the system must have the ability for self-adapting to changed "signatures" of network attacks and methods of their organization and implementation. Such kind of functions are provided by the adaptation module (see Fig. 2) based on a study of the detected network attack and ability of neural network immune detectors to additional training.

When the network attack is recognized by one of the detectors, there is a blocking of the network connection, and a message is generated to the user. Note, apart from the above actions, the system memorizes the characteristics of the network connection, classified as the attack by immune detector. Further, the parameters of this attack are compared with the parameters of the attacks that are in the data warehouse for detectors training. If such an attack or sufficiently similar characteristics already exists, nothing happens.

However, if an attack with such characteristics does not exist in the database or the characteristics of the detected attacks are quite different from those already known, then there are made the following operations:

1. A new detector – called the immune memory detector – is created on the basis of the neural network detector that detected a network attack.
2. Additional training of the new detector is running on the parameters of the new detected attack.
3. The new up-training detector is implementing into the module of the traffic analysis and intrusion detection.
4. Parameters of the new network attack are entered into the database that stores the data for training the neural network immune detectors.

The above described algorithm enables to analyze the detected network attack. If this attack is considered as the new one then the system is adapting to it by creating the immune memory detectors and entering the characteristics (signatures) of the new network attack into the training set for next new detectors. The results of conducted experiments exploring the summarizing properties of neural network immune detectors showed that trained detectors can detect and classify not only attacks – on which they are trained – but the new attacks as well. The authenticity of detection and classification of new attacks can approach 100 percent sometimes (Tables 2).

For example a detector 1 quite enough detects an attack <dos_back> where it was trained. In particular there were detected the 100 percent attacks of such kind at the low level of beta error equal 0.2 percent. Moreover this detector can detect new attacks as well, in particular 99,1%, of <dos_neptune> attacks and 100% of r2l_spy attacks and 88,9% of <u2r_loadmodule> attacks correspondingly. There were conducted experimental researches of adaptation neural network immune detectors to the new attacks using both cloning and mutation operations. For this purpose we selected a paternal detector which was studied on the example of <DoS_land> attack. This detector found out the two new attacks: <R2L_imap> and <Probe_portsweep>. As-

suming that such attacks are absent in the database, let's generate the two new detectors of A1 and A2, and add the parameters of found attacks in the trained sample for these detectors and thus train the proper detector <clonals> (Table 3).

Table 2. Detection of network attacks by detectors 1-3

Type of attack	Detector 1 (trained on DoS_back), %	Detector 2 (trained on Probe_Nmap), %	Detector 3 (trained on R2L_ftpwrite), %
DoS-attacks			
Back	100,0	99,1	0,3
Land	0,0	9,5	23,8
Neptune	99,1	99,9	0,0
Pod	0,0	12,9	1,9
Smurf	0,0	0,1	0,0
Teardrop	0,0	11,0	0,0
Probe-attacks			
Ipsweep	0,1	5,7	1,0
Nmap	80,5	100	0,0
Portsweep	2,1	30,8	0,1
Satan	13,3	96,1	2,1
R2L-attacks			
Ftp_write	0,0	0,0	100
Guess_passwd	0,0	0,0	5,7
Imap	0,0	0,0	0,0
Multihop	0,0	0,0	57,2
Phf	0,0	0,0	0,0
Spy	100,0	100	0,0
Warezclient	1,1	0,6	65,0
Warezmaster	0,0	0,0	90,0
U2R-attacks			
Buffer_overflow	0,0	0,0	83,4
Loadmodule	88,9	100	0,0
Perl	33,4	0,0	0,0
Rootkit	0,0	0,0	20,0

Table 3. Adaptation of the neural network immune detectors to new attacks

Type attacks	Detector 2 Sp(TNR)=99,0% Se (TPR), %	Detector of A1 Sp(TNR)=99,1% Se (TPR), %	Detector of A2 Sp(TNR)=98,9% Se (TPR), %
DoS-attacks			
Land	100,0	100,0	100,0
Pod	2,3	0,0	31,8
Probe-attacks			
Ipsweep	7,22	0,2	33,9
Portsweep	15,9	2,6	55,3
Satan	11,0	31,3	11,0
R2L-attacks			
Imap	83,3	91,7	83,3
Multihop	0,0	0,0	14,3
U2R-attacks			
Perl	0,0	66,7	0,0
Rootkit	0,0	20,0	0,0

As it can be seen from a table 4 the detector A1 began better find attacks of separate classes, in particularly <Probe_satan> – in 2,8 times, and <R2L_imap> on 8,4%, and it began also to find both the <U2R_perl> attacks and the <R2L_rootkit> attacks. On the other hand the detector A2 showed better results detecting the following attacks: Probe_ipsweep, DoS_pod, Probe_portsweep, R2L_multihop.

4.2 Adaptation Ability of Cyber Defense System Detecting the Malware

The goal of this experiment is to show the adaptation ability of the proposed artificial Cyber Defense system on the example of malware detection. Let us discuss briefly the experimental conditions. Initially, we generate several immune detectors. In the Table 4 there are five detectors D1… D5 that went through training and selection phase.

Table 4. Immune Detectors

Detectors	Learning set
D1	eventvwr.exe, dllhost.exe, eventvwr.exe, fixmapi.exe, **Trojan-Downloader.Win32. Bagle.f**
D2	finger.exe, eventvwr.exe, loadfix.com, proxycfg.exe, **Email-Worm.Win32. Brontok.q**
D3	control.exe, proxycfg.exe, systray.exe, regwiz.exe, **Trojan-Proxy.Win32.Lager.d**
D4	forcedos.exe, rspndr.exe, share.exe, lpq.exe, **Net-Worm.Win32.Bozori.k**
D5	regedt32.exe, redir.exe, loadfix.com, control.exe, **Trojan-Downloader.Win32. Small.dde**

The learning set for each detector is unique and consists of randomly chosen legitimate files and malicious code. After training and selection neural network immune detectors check the set of malware. A Table 5 shows the detection ability of each immune detector. P_T and P_F are interdependent values that characterize the membership of under-test file into legitimate or malicious class. P_T is the probability that checked object is legitimate. P_F is the inverse value and shows the belonging of checked object to the malware class. The equations 1 and 2 describe the calculation of P_T and P_F:

$$P_T = \frac{\overline{Y_1}}{L}, \qquad P_F = 1 - P_T = \frac{\overline{Y_2}}{L}, \tag{1}$$

$$\bar{Y}_1 = \sum_{k=1}^{L} Y_1^k, \qquad \bar{Y}_2 = L - \bar{Y}_1 = \sum_{k=1}^{L} Y_2^k \,. \tag{2}$$

where P_T – the probability of legitimate file; P_F – the probability of malware; Y_1 and Y_2 – the number of legitimate and malicious fragments of under-test file correspondingly; L – the total amount of fragments from under-test file, Y_{ik}- i-th output of immune detector for k-th input pattern.

Table 5. The Detection Ability of Detectors

Malware	D1,P_T/P_F	D2,P_T/P_F	D3,P_T/P_F	D4,P_T/P_F	D5,P_T/P_F
Worm.Brontok.q	**0,78/0,22**	0,83/0,17	0,83/0,17	0,85/0,15	**0,78/0,22**
Worm.NetSky.q	**0,74/0,26**	0,95/0,05	0,97/0,03	1,00/0,00	0,90/0,10
Worm.Rays	0,96/0,04	0,86/0,14	0,85/0,15	**0,79/0,21**	0,82/0,18
Worm.Bozori.a	**0,78/0,22**	0,93/0,07	0,95/0,05	0,99/0,01	0,88/0,12
Worm.Bozori.k	**0,77/0,23**	0,92/0,08	0,93/0,07	0,96/0,04	0,88/0,12
Packed.Tibs	**0,77/0,23**	0,96/0,04	0,97/0,03	0,99/0,01	0,92/0,08
Trojan.Dialer.eb	0,89/0,11	0,81/0,19	0,80/0,20	0,83/0,17	**0,79/0,21**
Trojan.Bagle.f	0,83/0,17	0,87/0,13	0,89/0,11	0,91/0,09	0,89/0,11
Trojan.INS.bl	0,86/0,14	0,86/0,14	0,84/0,16	0,86/0,14	0,81/0,19
Trojan.INS.gi	0,86/0,14	**0,79/0,21**	**0,75/0,25**	**0,73/0,27**	**0,75/0,25**
Trojan.Ladder.a	0,89/0,11	0,92/0,08	0,94/0,06	0,96/0,04	0,93/0,07
Trojan.Small.da	0,80/0,20	0,94/0,06	0,95/0,05	0,99/0,01	0,90/0,10
Trojan.Small.dde	**0,77/0,23**	0,95/0,05	0,96/0,04	1,00/0,00	0,90/0,10
Trojan.Small.dg	0,86/0,14	0,97/0,03	0,99/0,01	1,00/0,00	0,97/0,03
Trojan.Daemon.a	0,89/0,11	0,89/0,11	0,89/0,11	0,93/0,07	0,88/0,12
Trojan.Lager.d	0,83/0,17	0,88/0,12	0,75/0,25	0,93/0,07	**0,79/0,21**
Trojan.Mitglied.o	0,90/0,10	0,87/0,13	0,87/0,13	0,91/0,09	0,84/0,16
Trojan.Small.a	0,89/0,11	0,96/0,04	0,98/0,02	1,00/0,00	0,97/0,03
Virus.Bee	0,97/0,03	**0,79/0,21**	**0,77/0,23**	**0,77/0,23**	0,80/0,20
Virus.Neshta.a	0,90/0,10	**0,74/0,26**	**0,72/0,28**	**0,72/0,28**	**0,72/0,28**
Virus.VB.d	0,93/0,07	**0,69/0,31**	**0,65/0,35**	**0,65/0,35**	**0,69/0,31**

The fragments of files appear when we divide the checked file into the chunks with the size equals to number of input neurons of neural network immune detectors. For example, if the file size equal 16 Kbyte and we use 128 input neurons in neural network immune detector the number of fragments equal

$$L = 16*1024/128 = 128 \tag{3}$$

We are using the threshold that defines the belonging of checking objects to class of malware. If $P_T > 0.8$ ($P_F < 0.2$ correspondingly) then the checked object is legitimate. Conversely if $P_T < 0.8$ ($P_F > 0.2$ correspondingly) then the checked file is malicious.

A Table 6 demonstrates the ability of detectors to recognize the new unknown malware. In Tables 5 and 6 the gray cells indicate the detection of malware. As can be seen each immune detector is capable to detect not only the known malware (known malwiare is the malware that is from training set, for example, Trojan-Proxy.Lager.d for D3 neural networks immune detector) but also the unknown malware (for example, there are Trojan. INS.gi, Virus. Bee and Virus.VB.d. for D3 detector).

Table 6. The Detection Ability of Clones

Malware	C1, P_T/P_F	C2, P_T/P_F	C3, P_T/P_F	C4, P_T/P_F
Worm.Brontok.q	0,86/0,14	0,88/0,12	0,87/0,13	0,85/0,16
Worm.NetSky.q	0,95/0,05	0,93/0,07	0,94/0,06	0,90/0,10
Worm.Rays	0,87/0,13	0,86/0,14	0,87/0,13	0,85/0,15
Worm.Bozori.a	0,97/0,03	0,95/0,05	0,97/0,03	0,94/0,06
Worm.Bozori.k	0,94/0,06	0,91/0,09	0,92/0,08	0,90/0,10
Packed.Tibs	0,96/0,04	0,98/0,02	0,95/0,05	0,91/0,09
Trojan.Dialer.eb	0,89/0,11	0,90/0,10	0,91/0,09	0,81/0,19
Trojan.Bagle.f	**0,79/0,21**	**0,78/0,22**	**0,79/0,21**	**0,77/0,23**
Trojan.INS.bl	**0,77/0,23**	**0,79/0,21**	**0,78/0,22**	**0,76/0,24**
Trojan.INS.gi	**0,64/0,36**	**0,66/0,34**	**0,65/0,35**	**0,68/0,42**
Trojan.Ladder.a	**0,75/0,25**	**0,76/0,24**	**0,77/0,23**	**0,77/0,23**
Trojan.Small.da	0,83/0,17	0,81/0,19	**0,79/0,21**	0,80/0,20
Trojan.Small.dde	0,84/0,16	**0,79/0,21**	0,85/0,15	0,80/0,20
Trojan.Small.dg	**0,77/0,23**	**0,77/0,23**	**0,78/0,22**	**0,76/0,24**
Trojan.Daemon.a	**0,73/0,27**	**0,79/0,21**	**0,78/0,22**	**0,74/0,26**
Trojan.Lager.d	**0,78/0,22**	**0,78/0,22**	**0,78/0,22**	**0,77/0,23**
Trojan.Mitglied.o	**0,78/0,22**	**0,77/0,23**	**0,77/0,23**	**0,78/0,22**
Trojan.Small.a	0,81/0,19	0,82/0,18	0,82/0,18	**0,78/0,22**
Virus.Bee	0,91/0,09	0,89/0,11	0,90/0,10	0,87/0,13
Virus.Neshta.a	0,85/0,15	0,85/0,15	0,86/0,14	0,81/0,19
Virus.VB.d	**0,78/0,22**	0,81/0,19	0,88/0,12	**0,79/0,21**

As an example of adaptation ability of detectors let's chose the detector D3 that detects several malware: Trojan-Proxy.Lager.d, Trojan.INS.gi, Virus.Bee, Virus.Neshta.a and Virus.VB.d. After detection of the first malware Trojan.Win32.INS.gi detector D3 undergoes on the cloning process and as a result several detectors-clones Ci are generated. Every clone goes through the relearning process where data from the detected malware Trojan.Win32.INS.gi are included into the learning sample.

The Table 6 above demonstrates the detection ability of four clones Ci. As it can be seen each clone Ci detects Trojan.Win32INS.gi with higher rate than detector D3. In

addition the detectors-clones demonstrate the ability to detect such the new unknown malware (Trojan.Bagle.f, Trojan.INS.bl, Trojan.Ladder.a, Trojan.Small.da, Trojan.Small.dde, Trojan.Small.dg, Trojan.Daemon.a, Trojan.Mitglied.o, Trojan.Small.a) that are stayed undetectable in the case of detector D3 scanning. The process of retraining of the immune detectors where data from detected malware are used allows immune detectors to improve the detection ability, increase detection rate and permits the whole system to adapt to changeable environment by means of evolution. However while detectors-clones C_i acquire the ability to detect Trojans with high rate they practically lost the ability to detect malware from other classes. It is the back side of the processes of cloning and mutation – the detectors-clones are tuned in to the specific malware or series of specific malware.

Thus the detected new malware is added to the training data set in order to train new neural network immune detectors and it enables obtaining the new detectors with the different structure, tuned into new malware. These two mechanisms, namely cloning and training dataset modification, allow increasing the detection quality and adapting detectors to the new unknown malware. Obviously, the use of the proposed Intelligent Cyber Defense System in a number of applications mentioned in the Introduction above, will improve the reliability of detecting intrusions into computer systems, and as a result – can lead to a significant reduction in financial losses of companies from cyber attacks.

5 Conclusion and Future Work

In this paper we proposed the system for cyber defense which can detect not only already known network attacks but previously unknown, new cyber threats. This system is characterized as the intelligent, adaptive, evolutionary and self-organizing one. We examined the ability of immune detectors with neural network structure to adapt to changeable cyber-attacks trend and as result to evolve with the scope increasing the rate of unknown cyber attacks. We have run experiments proofing that detectors can adapt to the new threat.

The ability of the immune detectors to evolve – by exploring of the new malicious material - allows the intelligent security system to adapt to the new threat and provide an effective defense against the known and unknown cyber attacks. The adapted detectors acquire the ability to detect some new attacks with a higher quality. The new detected attack is adding to the training sample that increases the difference in immune detectors and enables detecting unknown attacks.

Thus, it is experimentally confirmed, that proposed neural network immune detectors are able to discover the unknown types of attacks and adapt to them.

The proposed Intelligent Cyber Defense System can increase the reliability of intrusion detection in computer systems and therefore it may reduce financial losses of companies from cyber attacks.

In the nearest future we expect to implement a part of the cyber defense system units using programmable logic arrays. Such solution, unlike the software protection, will allow eliminate the influence of the software intrusions on the cyber defense

system. To make decisions about counter invasion methods it is expected to use the Mamdani's fuzzy inference rules.

Acknowledgements

This work is running under a grant by the Ministry of Education and Sciences, Ukraine, 2016–2017. This work is supported by the Belarusian State Research Program “Informatics and Space”, 2011–2015.

References

1. Cybercrime in the world, http://www.tadviser.ru/index.php (In Russian)
2. Symantec it estimated the annual losses from cybercrime at $ 114 billion, http://www.companion.ua/articles/content?id=162264 (In Russian)
3. Forbes experts have chosen the most high-profile cyber attacks recently, http://www.securitylab.ru/news/444700.php (In Russian)
4. Tavallaee, M., Bagheri E., Lu, W., et al.: A detailed analysis of the KDD CUP 99 data set, In: Proceedings of the IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA’2009), 1–8. DOI: 10.1109/CISDA.2009.5356528 (2009)
5. Laheeb, M. I.: Anomaly network intrusion detection system based on distributed time-delay neural network (DTDNN). Journal of Engineering Science and Technology, 5 (4), 457-471 (2010)
6. Grediaga, A., Ibarra, F., García, F., Ledesma, B., Brotons, F.: Application of neural networks in network control and information security. LNCS, (3973), 208-213 (2006)
7. Zhang, C., Jiang, J., Kamel, M.: Comparison of BPL and RBF network in intrusion detection system. LNCS (LNAI), (2639), 460-470 (2004)
8. Lorenzo-Fonseca, I., Maciá-Pérez, F., Mora-Gimeno, F., Lau-Fernández, R., Gil-Martínez-Abarca, J., Marcos-Jorquera, D.: Intrusion detection method using neural networks based on the reduction of characteristics. LNCS, (5517), 1296-1303 (2009)
9. Kang, B.D., Lee, J.W., Kim, J.H., Kwon, O.H., Seong, C.Y., Park, S.M., Kim, S.K.: A mutated intrusion detection system using principal component analysis and time delay neural network. LNCS, (3973), 246-254 (2006)
10. Cannady, J.: Applying CMAC-based online learning to intrusion detection, In: Proceedings of the International Joint Conference on Neural Networks, (IJCNN’2000), IEEE-INNS-ENNS, (5), 405-410 (2000)
11. Debar, H., Becker, M., Siboni D.: A neural network component for an intrusion detection system, In: Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy, 240-250 (1992)
12. Cheng, E., Jin, H., Han, Z., Sun, J.: Network-based anomaly detection using an Elman network. Networking and Mobile Computing, Lecture Notes in Computer Science, Springer, Berlin/Heidelberg, (3619), 471-480 (2006)
13. Höglund, A.J., Hätönen, K., Sorvari, A.S.: A computer host-based user anomaly detection system using the self-organizing map, In: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN’00), (5), 411-416 (2000)
14. Jirapummin, C., Wattanapongsakorn, N., Kanthamanon, P.: Hybrid neural networks for intrusion detection system, In: Proceedings of the International Technical Conference on Circuits/Systems, Computers and Communications, Thailand, 928-931 (2002)

15. Horeis, T.: Intrusion detection with neural networks – Combination of self-organizing maps and radial basis function networks for human expert integration, Tech. report, University of Passau, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.106.191&rep=rep1&type=pdf
16. Chimphlee, W., Abdullah, A.H., Sap, M.N.M.: Anomaly-based intrusion detection using fuzzy rough clustering, In: Proceedings of the International Conference in Hybrid Information Technology (ICHIT'06), (1), 329-334 (2006)
17. Dickerson, J.E., Juslin, J., Koukousoula, J., Dickerson, J.A.: Fuzzy intrusion detection, In: Proceedings of the 20th International Conference of the North American Fuzzy Information Society (NAFIPS'01) and Joint the 9th IFSA World Congress, 3, Vancouver, Canada, (3), 1506-1510 (2001)
18. Hofmeyr, S., Forrest, S.: Immunity by design, In: Proceeding of the Genetic and Evolutionary Computation Conference (GECCO'1999), 1289-1296 (1999)
19. Burnet, F.: The Clonal Selection Theory of Acquired Immunity. Cambridge University Press (1959)
20. Burnet, F.: Clonal selection and after. Theoretical Immunology, Marcel Dekker Inc, 63-85 (1978)
21. Jerne, N.: Towards a network theory of the immune system. Ann. Immunology (Inst. Pasteur), (125C), 373-389 (1974)
22. Greensmith, J., Whitbrook, A., Aickelin, U.: Artificial Immune Systems. Handbook of Metaheuristics, 2nd edition, Springer, chapter 14, 421-448 (2010)
23. Greensmith, J., Aickelin, U.: The deterministic dendritic cell algorithm. In: Proceedings of the 7th International Conference on Artificial Immune Systems (ICARIS'2008), Springer, 291-302 (2008)
24. Coico, R., Sunshine, G., Benjamini, E. Immunology: A Short Course. Wiley-Liss (2003)
25. Murphy, K., Travers, P., Walport, M. Janeway's Immunobiology. Garland Science, 7th edition (2008)
26. Stibor, T., Mohr, P., Timmis, J., Eckert, C.: Is negative selection appropriate for anomaly detection? In: Proc. GECCO'2005, Springer, pp. 321--328 (2005)
27. Harmer, P., Williams, P., Gunsch, G., Lamont, G.: An artificial immune system architecture for computer security applications. IEEE Transactions on Evolutionary Computation, 6(3), 252--280 (2002)
28. Balthrop, J., Esponda, F., Forrest, S., Glickman, M.: Coverage and generalization in an artificial immune system, In: Proc. GECCO'2002, pp. 3-10 (2002)
29. Stibor, T., Bayarou, K., Eckert, C.: An investigation of r-chunk detector generation on higher alphabets, In: Proc. GECCO'2004, pp. 299-307 (2004)
30. Gonzales, F., Dasgupta, D., Gomez, J.: The effect of binary matching rules in negative selection, In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2003), 198-209 (2003)
31. Kim, J., Bentley, P., Aickelin, U., Greensmith, J., Tedesco, G., Twycross, J.: Immune system approaches to intrusion detection – A review. Natural Computing, Springer, 6 (4), 413-466 (2007)
32. Komar, M., Golovko, V., Sachenko, A., Bezobrazov S.: Intelligent system for detection of networking intrusion. In: Proceedings of the 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS-2011), Prague, Czech Republic, (1) 74-377 (2011)
33. Komar, M., Golovko, V., Sachenko, A., Bezobrazov, S.: Development of neural network immune detectors for computer attacks recognition and classification, In: Proceedings of the 7th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS-2013), Berlin, Germany, (2) 665-668 (2013)