Development of the disable software reporting system on the basis of the neural network

S Gavrylenko, O Babenko and E Ignatova

1,2 National technical university «Kharkiv Polytechnical Institute», Kharkiv, Ukraine
3 Intertek Ltd, Kharkiv, Ukraine

Abstract. The PE structure of malicious and secure software is analyzed, features are highlighted, binary sign vectors are obtained and used as inputs for training the neural network. A software model for detecting malware based on the ART-1 neural network was developed, optimal similarity coefficients were found, and testing was performed. The obtained research results showed the possibility of using the developed system of identifying malicious software in computer systems protection systems

Key words: PE-file structure, antivirus programs, computer systems, malicious software, signature method, heuristic method, ART-1 neural network

1. Introduction
The most notable event of the first summer month of 2017 was the Trojan.Encoder.12544 worm epidemic, which is mentioned in the media as Petya. A or Mbr locker 256. It is the first time in the history of Ukraine, that banks, gas stations, stores, sites of government agencies were struck in a few hours due to hacker’s attack. Even the sites of the Cabinet of Ministers and some of the largest media outlets were paralyzed. This malware infected computers of many organizations and individuals in 60 countries around the world. The damage from the virus attack is estimated at $ 8 million.

It is known that over a year, viruses cause damage to hundreds of billions of dollars, and about the same amount is indirect damage associated with the development of software and other measures to protect against viruses. It should be noted that this problem is compounded by the dynamic growth in the number of mobile devices, the general switch to cloud technology and the spread of the Internet technologies, which leads to an increase of the number of malicious software.

It becomes obvious that the analysis of such a number of malicious code and the formation of the entire spectrum of virus signature is an almost unrealized task. That is why the actual topic is the development of effective methods and technologies for counteracting computer viruses based on heuristic methods.

Analysis of literature [1-4], as well as research on the methods of heuristic analysis in antivirus programs showed a high variety of existing approaches and methods of heuristic analysis: intelligent subsystems based on the theory of artificial intelligence, methods of fuzzy logic, cluster analysis, the theory of neural networks, genetic algorithms and other. The main disadvantage of the heuristic method is the high frequency of false positives.
Statistical methods based on control cards (e.g., Shukhart control maps, CUSUM maps, EWMA cards, etc.) can also be used to solve the set tasks [5]. In addition, methods for statistical data processing, for example, BDS testing [6], can be used for further refine of obtained results.

These methods are based on the assumption that for a computer system (CS) there is a template for normal behavior and any significant deviations from it may be due to the influence of intruders. That is why the very important task is to select or form a template, which would reproduce the functional portrait of the CS and record its abnormal behavior with the given accuracy. At the same time, the more input data is analyzed, the more accurate the result of the evaluation. Meanwhile, if the model or evaluation criterion is chosen incorrectly, parametric methods lose their basic authority, which can lead to an increase in false positives.

The conducted studies have shown that the main way of eliminating these shortcomings is to improve the models of information technology and the reasoned choice of criteria for evaluating abnormal behavior of computer systems.

2. Results of the development and research

The conducted researches have shown that one of the perspective directions of heuristic analysis of computer viruses is the use of neural networks [7-16]. On the one hand neural networks, can behave as a deterministic machine, on the other as a fuzzy system, evaluating new data that did not participate in the formation of the neural network. Such a result is achieved by learning networks [17] and not by the formation of reaction rules, as is done in classical approaches. Ability of the networks to educate by examples makes them more attractive in comparison to systems that function according to a defined system of rules formulated by experts. The education process can be considered the architectural determination of the network and finding the coefficients of the connections between neurons. The neural network adjusts the weight of the connections depending on the existing training set and is a parallel computing device, since it is based on a set of simple computing elements - the parallel functioning neurons. Such a system is resistant to damage, that is, the network will work even if part of the neurons fails. In the education process, the neural network is capable to detect complex interdependencies between incoming data and output, as well as generalization. This means that in case of successful training, the network will be able to return the correct result based on the data that was missing in the training sample, as well as incomplete or partially distorted data.

It was decided to develop a heuristic analyzer based on the neural network, namely ART-1.

ART Network is a vector classifier. The input vector is classified depending on which of the many images, previously remembered, it is similar to. The ART network classifies its classification in the form of excitation of one of the neurons from recognition layer. If the input vector does not match any of the memorized images, a new category is created by memorizing the image, which is identical to the new input vector. If it is determined that the input vector is similar to previously remembered vectors in terms of a certain similarity criterion, the remembered vector will change (learn) under the influence of the new one in such a way as to become more similar to this input vector.

The remembered image will not change if the current input vector doesn't look similar to it. Thus the dilemma of stability-plasticity is solved. The new image may create additional classification categories, however, it cannot force the existing memory to change itself. Several types of neural networks have been developed on the basis of adaptive resonance theory, in particular, the ART-1 network is designed to work with binary input images or vectors.

The architecture of the ART-1 neural network is shown in figure 1.

It has control elements G1 and G2 that provide control over the operation process [18]. In Figure 1, due to the large number of links between the Z and Y layers of the elements, only one generalized pair of weights, the connections between the interface neurons and the recognition neurons are indicated. Most of the links shown in Figure 1. are excitatory: from the input layer of Fa elements to the neurons R, G1, G2 and Fb, from the neurons G1, G2 according to the neurons of the Fb and Y layers. Braking signals transmit only the plurality of bonds from the interface elements to the R- A neuron, from the R neuron to the Y neurons and from the Y neurons to the G1 element. All ART-1 networks
communicate only binary signals 0 or 1. Each element in the interface or Y-layer of the ART-1 network has three sources of input signals. An arbitrary interface element can receive signals from elements of the input layer S, and elements of the Y-layer and from the neuron G1. Similarly, an element can receive signals from the interface elements, neurons R and G2. In order to translate the neurons of the interface or recognition layers into an active single state, the presence of incoming excitatory signals from two sources is required [20].

![Figure 1. An architecture of the neural network ART-1.](image)

It has control elements G1 and G2 that provide control over the operation process [18]. In Figure 1, due to the large number of links between the Z and Y layers of the elements, only one generalized pair of weights, the connections between the interface neurons and the recognition neurons are indicated. Most of the links shown in Figure 1 are excitatory: from the input layer of the Fα elements to the neurons R, G1, G2 and Fb, from the neurons G1, G2 according to the neurons of the Fb and Y layers. Braking signals transmit only the plurality of bonds from the interface elements to the R- A neuron, from the R neuron to the Y neurons and from the Y neurons to the G1 element. All ART-1 networks communicate only binary signals 0 or 1. Each element in the interface or Y-layer of the ART-1 network has three sources of input signals. An arbitrary interface element can receive signals from elements of the input layer S, and elements of the Y-layer and from the neuron G1. Similarly, an element can receive signals from the interface elements, neurons R and G2. In order to translate the neurons of the interface or recognition layers into an active single state, the presence of incoming excitatory signals from two sources is required [19].

If a prototype found with a certain accuracy specified by a special similarity parameter matches the input process, then it is modified to become more similar to the presented process. When the input process is not sufficiently similar to any of the existing prototypes, a new class is created on its basis. This is possible due to the fact that the network has a large number of redundant or unallocated elements that are not used until there is no need for it (if there are no undocumented neurons, then the input image does not cause network reactions) [19].

ART Neural networks - dynamic objects, which are described by systems of ordinary differential equations, so their training in the general case is rather laborious. However, ART network models can be simplified if we assume that the change in the output signals of the neurons occurs much faster than the change in the weight vectors of their bonds. Therefore, in the neural networks of the theory of adaptive resonance we can assume that after selecting for the learning of an acceptable Y-element (the resonance between the present and stored in the memory image), the output signals of all neurons remain unchanged for a long period of time, during which there are changes in connections’ weights. The algorithm for training the neural network is given in [20].

The input data for training the neural network are based on the analysis of the PE structure of the file.

In figure 2 it an example of PE structure of the file is shown and the areas for the further analysis are emphasized.
The PE structure of harmful and secure software was analyzed, namely:

- 290 Worm type files
- 1050 files such as Trojan,
- 1153 files of type Backdoor
- 1000 secure files

As a result of the analysis of the PE structure of the investigated files received:

- a table with API-functions and libraries in which they are included. There were 24945 records received.
- a table with strings. Found in total 175651 rows, their length varied from 6 to 70 characters.

The analysis of the data received from malicious and secure software allowed to highlight functions and rows, and to form the signs (Table 1), which are inherent to the considered viruses and to form a table of attributes. For further analysis it was decided to use 49 signs. These signs were later used as bitmaps for file analysis. The binaries of malicious files like Worm, Backdoor, Trojan and secure software were obtained as a result of searching for selected attributes in files (Figure. 3).

For the correct functioning of the state identification system, an optimal coefficient of similarity has been selected experimentally.
Table 1. Features malicious signatures

| Functions/lines      | Var 1     | Var 2     | Quantity in the viruses |
|----------------------|-----------|-----------|-------------------------|
| callnexthookex       | -         | -         | 42                      |
| copyfile             | CopyFileA | CopyFileW | 132                     |
| createfile           | CreateFileA | CreateFileW | 178                    |
| enumcalendarinfo     | EnumCalendarInfoA | - | 43                      |
| findfirstfile        | FindFirstFileA | FindFirstFileW | 127                    |
| getcurrentprocessid  | -         | -         | 70                      |
|                      | ...       |           |                         |
| host                 | -         | -         | 10                      |
| admin                | -         | -         | -                       |
| hostname             | -         | -         | 9                       |
| localhost            | -         | -         | 19                      |
| sendmail             | -         | -         | 28                      |
| userhost             | -         | -         | 9                       |
| wininet              | -         | -         | 35                      |
| wnet                 | WNetEnumResourceA | WNetOpenEnumA | 55                      |
| getmodulefilename    | GetModuleFileNameA | GetModuleFileNameW | 206                    |
| getmodulehandle      | GetModuleHandleA | - | 247                      |
| loadlibrary          | LoadLibraryA | - | 135                      |
| setfilepointer       | SetFilePointerEx | - | 158                      |
| virtualalloc         | VirtualAllocEx | - | 158                      |

At the first stage, 100 signatures of the type Backdoor are taken for basic knowledge of the neural network. The initial similarity factor is 0.6, since it is the minimum allowable for the neural network. Signatures of 100 secure files were submitted to the input of the neural network for recognition. As a result of the program's operation, some of the input signatures were displayed on the console. Console-derived signatures were attributed to signatures of the type Backdoor, which was a mistake (Figure. 4) and required an increase of the coefficient of similarity to the optimal. The optimal coefficient is considered to be the coefficient for which there are no false recognitions. The results of the experimental selection of the optimal similarity coefficient for malicious software of the type Backdoor are given in Table. 2.

Table 2. Experimental match of the optimal coefficient of similarity for the viruses type Backdoor

| Coefficient of similarity | Quantity of false alerts for the safe signatures |
|---------------------------|-----------------------------------------------|
| 0.6                       | 18                                            |
| 0.7                       | 10                                            |
| 0.8                       | 1                                             |
| 0.84                      | 0                                             |
Thus, the absence of false positives for malicious software such as Backdoor managed to achieve with a similarity factor of 0.84.

At the second and third stage, 100 signatures of malicious software such as Trojan and Worm are taken for basic knowledge of the neural network. The results of the experimental selection of the optimal coefficient of similarity are given in Table 3 and Table 4.

**Table 3.** Experimental match of the optimal coefficient of similarity for the virus type Trojan

| Coefficient of similarity | Quantity of false alerts for the safe signatures |
|---------------------------|--------------------------------------------------|
| 0.6                       | 10                                               |
| 0.65                      | 3                                                |
| 0.7                       | 0                                                |

**Figure 5.** Optimal coefficients of the similarity grouped by malicious signature types

**Table 4.** Experimental match of the optimal coefficient of similarity for the virus type Worm

| Coefficient of similarity | Quantity of false alerts for the safe signatures |
|---------------------------|--------------------------------------------------|
| 0.6                       | 29                                               |
| 0.7                       | 14                                               |
| 0.8                       | 17                                               |
| 0.9                       | 10                                               |
| 0.97                      | 0                                                |

**Figure 5.** Optimal coefficients of the similarity.
In the future, 3 experiments with different input data and different similarity coefficients were carried out. The first one uses 100 signatures of the type Backdoor as the base of initial knowledge, 400 signatures are presented to the recognition input, which represent a mixture of all possible signatures (100 Backdoor signatures, 100 Trojan signatures, 100 Worm signatures, 100 signature files of secure files). The second one is the base of 100 signatures of type Trojan, the input is 400 signatures. The third is the base of 100 signatures of type Worm, the input is 400 signatures.

At the initial base of Backdoor malicious signatures, an additional group of malicious Trojan 28% signature was detected.

Thus, the absence of false positives for malicious software such as Backdoor managed to achieve with a similarity factor of 0.84.

At the second and third stage, 100 signatures of malicious software such as Trojan and Worm are taken for basic knowledge of the neural network. The results of the experimental selection of the optimal coefficient of similarity are given in Table 4 and Table 5.

At the initial base of Warm malicious signatures, only signatures from this group were 100%. At the initial base of Trojan malware, Trojan 100%, Backdoor -46%, Worm - 1% were detected.

The results of the identification system showed that when training the neural system by the Backdoor sample, the system also begins to identify harmful Trojan (28%) signatures, since these types of signatures have a high similarity coefficient, since they perform similar actions from the point of view of the operating system.

When learning a worm-patterned system, only a signature of this type is detected, due to a relatively high optimal similarity factor of 0.97, the tokens of this type are abandoned from Signature Backdoor, Trojan, and secure software.

Also, when training a neural system with Trojan sampling, in addition to this sample, the system recognizes backdoor signatures (46%) and insignificant number of worm-signatures (1%).

This percentage of signature recognition (Backdoor - Trojan and vice versa) is due to the fact that these types of viruses have similar actions from the point of view of the operating system namely harm to it: the desire to obtain unauthorized access to data or remote control of the operating system and the computer as a whole; collection of information and its transmission to the attacker, its destruction or malicious change, the disruption of the computer; the use of computer resources for other purposes.

3. Conclusions.

In this article, the methods of constructing antivirus programs are analyzed. The PE structure of harmful and secure software has been analyzed, API functions and rows are found, inherent to these files, and selected part of them for further analysis. The result of the search for highlighted features in files is binary vectors of malicious software such as Worm, Backdoor, Trojan and secure software, and used as inputs for training the neural network.

The software model of the heuristic analyzer on the basis of the ART-1 neural network has been developed, the optimal similarity coefficients were found, and the tested computer virus detection system was tested.

The test results showed that when training the neural system with the Backdoor sample, the system also begins to identify 28% of the malicious Trojan type signatures, since these types of signatures perform such actions from the point of view of the operating system. When learning the system, the worm sample is only susceptible to virus of this type.

The obtained research results showed the possibility of using the developed system of identifying malicious software as an auxiliary method in information security systems of the COP

References
[1] Semenov S, Gavrylenko S and Davydov V 2014 Data Protection in Computer-Aided Control Systems (“LAP Lambert Academic Publishing” Germany)
[2] Lukatsky A 2001 Attack Detection (St. Petersburg: VHV-Petersburg)
[3] Shelukhin O I, Sakalema D Zh and Filinov A S 2013. Intrusion Detection into Computer Networks (Moscow: Hot line-Telecom) p 220
[4] Gavrylenko S and Semenov S 2015 Approximating computer system operation technologies under external action through the brusselator model with perturbation in the form of dynamic chaos Industrial Engineering Journal 1 (44) p 36-40 Romania
[5] Gavrylenko S, Chelak V and Hornostal O 2016 Metrology and metrology assurance” pp 342-347 Bulgaria
[6] Gavrylenko S, Semenov S and Chelak V 2016 Bulletin of NTU "KhPI”. Informatics and modeling vol 21 p 18-125 Ukraine
[7] Grishin A V 2011 Information technologies and computing systems vol 1 p 53-64
[8] Yemelyanova Yu G 2011 Software systems: theory and applications vol 3 (7) p 3-15
[9] Izrailov K E 2012 Vestnik Inzepona. Series: Engineering 8 (59) p 150-153
[10] Nesteruk F G, Nesteruk L G and Nesteruk G F 2009 systems Automation and Remote Control 70 p 491-501
[11] Kochetkova A S 2007 Monitoring Bulletin of Volgograd State University Series 9: Research of Young Scientists 6 p 163-166
[12] Magnitsky N A 2008 Dynamics of Non-uniform Systems p 200-205
[13] Marchenko O O, Matvienko S V and Nesteruk F G 2007 Scientific and Technical Bulletin of Information Technologies, Mechanics and Optics 39 pp 83-93
[14] Skorik F A and Sayenko I B 2013 Innovations in science (materials of XVI international correspondence scientific-practical conference) p 151-156
[15] Tumoyan Y P 2009 Development of the method of simulation of network attacks on the basis of neural networks and probabilistic graphs, News of Southern federal university. Technical sciences 90 p 148-153
[16] Novikov O O and Krasnopevtsev A A 2012 Safety of information technologies vol 1 p 47-50
[17] Bezobrazov S V and Golovko V A 2010 Neuroinformatics vol 7 p 273-288
[18] Dmitrienko V D, Zakovrotny A Yu, Noskov V I and Mezentsev M V 2014 Fundamentals of neurocomputing: educational and methodical manual for practical classes p 140 Ukraine
[19] Ossovsky S 2002 Neural Networks for Information Processing (Moscow: Finance and Statistics) p 344
[20] Gavrylenko S Yu, Semenov S G and Babenko O S 2015 Systems of information processing 10 (135) p 126-129 Ukraine