The Impact of the Neural Network Structure by the Detection of Undesirable Network Packets

I. Halenár, A. Libošvárová

***Abstract*— Intrusion detection systems (IDS) are among the new trends in attack detection in communication networks. There exist several types of IDS, which are using several methods to detect network errors. The most progressive type of IDS belong a system with integrated artificial intelligence – neural network.**

The core of this work is an example of using the neural network for detecting false network packets having regard to architecture of the neural network and success of detection.

Also work includes a proposal of the system for data manipulation (checking and capturing traffic data) and defines the method of identifying data elements in the communication network. The next solves the transformation of parameters for input to the neural network and defines the type and the appropriate neural network architecture.

The proposal is realized and tested in several variants. Outputs from tests are shown in summary charts

***Index Terms*— communication, neural network, security**

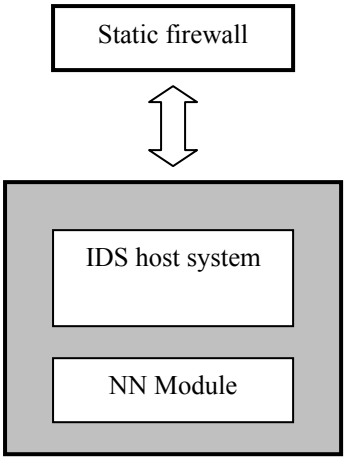
I. Introduction

IN present day is mostly used a network communication based on TCP / IP protocols. The trends in production systems are to combine types of industrial ethernet networks and traditional technologies [14]. The result of this is the creation of various additions to TCP/IP protocol, which are solving compatibility problems, but also bring to the automation network errors and security risks. The protection is happening on several levels and is performed by various systems such as firewalls, IDS (Intrusion Detection System), IPS (Intrusion Prevention System) devices or protocols [4]. With joining these elements of security and detection technology with technology of neural networks we can reach a separate intelligent network security systems - intelligent firewall [6], which contains knowledge about the potential security risks and threats in the computer network.

II. SUGGESTED SYSTEM

The suggested system (picture) is able to adapt dynamically to the possible failure of a communication, system attack or more complex forms of infiltration (exploits, out of band communications, the analysis of covert communication channels).

Manuscript received June 19, 2012; revised July 22, 2012.
I. Halenar is with the Institute of Applied Informatics, Automation and Mathematics, Slovak University of Technology, Trnava, Slovakia, phone:+421 918 646 021, e-mail: igor.halenar@stuba.sk
A. Libosvarova is with the Institute of Applied Informatics, Automation and Mathematics, Slovak University of Technology, Trnava, Slovakia, phone: +421 918 646 021, e-mail: adriana.libosvarova@stuba.sk



The whole proposed system is suitable to build on existing IDS. Of course the existing system must meet certain requirements:

- open system - it must be possible to program own extension modules
- possibility to capture packets - IP and TCP / UDP packets are a base of communication, it should be possible to capture these packets in the communication network
- packet saving - an important point because of the possibility of repeating cycles of learning a neural network
- existence of a communication interface - to manage the active part in data transmission network (firewall)

As a host system is suitable IDS Snort (GNU GPL licence) [17].

III. Data identification and transformation

For validation of correct data transfer is appropriate to begin tracing the values of individual fields in the data structure of TCP and IP protocols, thus easily identify the correctness of packets. Given the large number of protocols used and sub-protocols and their combinations, it is appropriate to choose the guidelines governing the selection of the parameters of data packets.

After preliminary analysis of options and the available literature [3], [15], we propose that the characteristic of data communication in this case will be specified with following parameters:

ISBN: 978-988-19252-4-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

--ID protocol - the protocol type associated with packet

--Source port - Number of TCP / UDP port of the source system

--Destination port - Number of TCP / UDP port of the target system

--Source Address - IP address of the source system

--Destination Address - IP address of the target system

--ICMP type - the type of ICMP packet

--Length of data transferred - the size of the data packet in bytes

--FLAGS - signs in the protocol header

--TCP window size - window size parameter of TCP packet

These data must be properly transformed as input to neural network [10]. According to available sources we can choose use of feed-forward neural network learning method with back-error propagation. This type of neural network provides sufficient flexibility and applicability to a wide range of tasks, where it is possible to use technology to minimize the objective function NN. To minimize the objective function we can use several optimization methods commonly used to minimize in the numerical mathematic [8]. The commonly used methods are including gradient methods, which disadvantage is the high number of iteration steps. Because this disadvantage we can use a variety of other, more efficient and faster optimization methods for adaptation of neural networks. In the available literature these methods are represented by the name "quickprop" (based on Newton's method) [2] or other numeric methods (method of variable steps, entropic normalization models, least squares, etc.). Alternatively, you can use modern methods to minimize using genetic algorithms or some of data analyzing methods as described in other literature [11], [16], [18].

The whole process of data transformation is necessary to perform simply and fast. For this, we have to implement programs in some text processing language. For example in this case we are using AWK executable script for Bourne Shell system.

Adaptations must be carried out in several stages. The first step is extraction of the required values of the entire data package. We can afford to ignore the account records of the communications network service (ARP, RARP communication) and data, to us in terms of work, unattractive. The next phase is an appropriate representation of some elements [5].

Specifically, it is the following parameters:

--Variable IDPRO, where we make the following transformation

TABLE I
PROTOCOL TYPE TRANSFORMATION

IDPRO	Substitution
TCP	6
UDP	17
ICMP	1

--Variable ICMPT, where we make the following transformation

TABLE II
ICMP TYPE TRANSFORMATION

ICMPT	Substitution
ECHO	1
REQUEST	2
NULL	0

--Variable FLG, where we make the following transformation

TABLE III
FLAG TYPE TRANSFORMATION

FLG	Substitution
NO_FLAG	1
S	
RST	2
FIN	3
PSH	4
URG	5
SYN	6

The transformation process should be fast and automatic. After processing by the AWK program [1] we receive data represented by the following example.

```
1,1471751309,0,147175134254,0,0,0,44,1
6,62240183148,80,147175134254,2190,8514,4,1500,0
17,19416092,33239,233104778,1234,0,0,1500,0
6,147175130212,2246,19512215120,80,65535,1,40,0
17,13015680151,4262,2242127254,9875,0,0,224,0
1,147175130212,0,10254247189,0,0,0,36,2
```

Comma separated fields from first are IDPRO, SIP (source ip), SPO (source port), DIP (destination_ip), DEP (destination port), TW (size of tcp window), FLG (flag), SZ (size of packet), ICMPT (icmp type).

Given the expected use of neural networks (classification) is sufficient the distribution of patterns into two groups. This means to divide the patterns to correct packets (CLASS = 1) and those which will be subject to further testing (CLASS = 2). Designation of records is necessary for training neural networks. Records of class packets are stored in a separate table.

IV. IMPLEMENTATION OF NEURAL NETWORKS

With reference to the used literature [12], [13] is a neural network with one hidden layer (3 layer neural network) and a sufficient number of hidden neurons, capable of simulating each binary or continuous function with desired accuracy. In our case, we assume nine input neurons and two output neurons. Number of neurons in input layer is given by number of parameters for describing the communication. We are using neural network for classification to two groups. There can be only one neuron in output layer. But for software realization reasons we have to use two output neurons. Other parameters of neural network are next.

The activation function of hidden layer was used in all cases the standard sigmoid activation function:

ISBN: 978-988-19252-4-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

$$x_i = f(in_i) = \frac{1}{1 + e^{-cin_i}}$$

The activation function of output layer in the various experiments was a linear activation function:

$$x_i = f(in_i) = in_i$$

In experimental phases were made several attempts and tests with different numbers of neurons in the hidden layer.

For learning and testing the neural network we need large sample of captured traffic. Data can be from real environment or generated. For testing purpose in this case we are some mixture using data from simulated network communication in laboratory environment and real traffic.

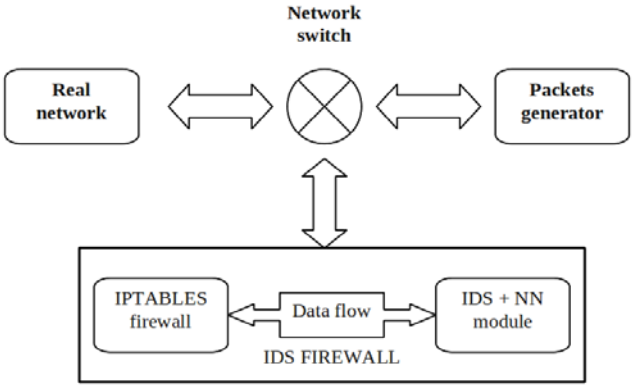


Fig. 1. The scheme of system for data recording.

With data transformation was selected 5000 models of the ongoing communication in the test environment. Of this number, we have developed a selective choice of four groups containing 200 models for learning neural network and a group of 100, 200 and 3000 models for testing.

In experimental phases were made several attempts and tests with different number of neurons in the hidden layer. Specifically, the network involved in the 9-6-2, 9-10-2, 9-20-2 and 9-40-2 (input - hidden - output layer). As the algorithm to minimize the objective function was chosen quickprop method based on Newton's method [7].

The following charts are showing the results of learning neural networks in various configurations. The red line represents the absolute error tolerance, which is in our case 0,1. The graphs for every experiment are in linear and logarithmic scale.

A. Experiment one – six neurons in hidden layer

As you can see, in the first experiment (nn 9-6-2) the absolute error of neural network after one thousand iterations was 0,17905 . This value is stable and is not changing after 200 training cycles. But this value does not meet our marginal conditions.

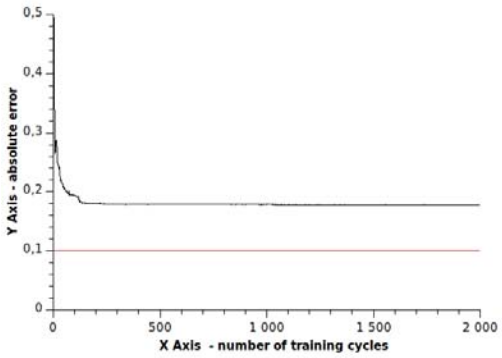


Fig. 2. Outputs from learning the neural network with 6 neurons in hidden layer, shown in linear scale

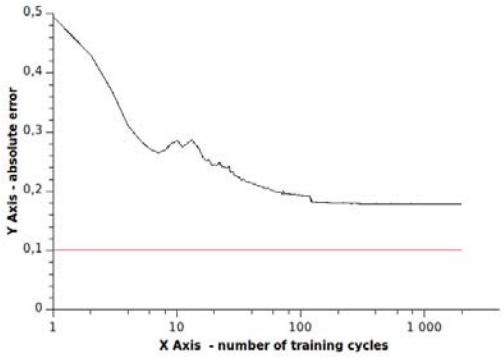


Fig. 3. Outputs from learning the neural network with 6 neurons in hidden layer, shown in logarithmic scale

B. Experiment two – ten neurons in hidden layer

The output from the neural networks with 10 neurons in hidden layer is similar to experiment one. The best value of the absolute error achieved in this experiment is 0,16937.

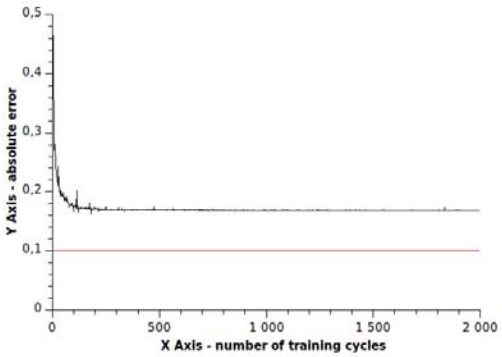


Fig. 4. Outputs from learning the neural network with 10 neurons in hidden layer, shown in linear scale

ISBN: 978-988-19252-4-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

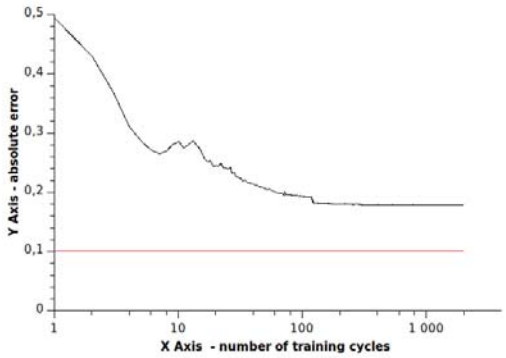


Fig. 5. Outputs from learning the neural network with 6 neurons in hidden layer, shown in logarithmic scale

C. *Experiment three –twenty neurons in hidden layer*

In the third case, the result obtained after one thousand iterations correspond to the absolute error is 0.06041. The number of patterns in training set, which have achieved the required tolerance, was increased to 96.67%.

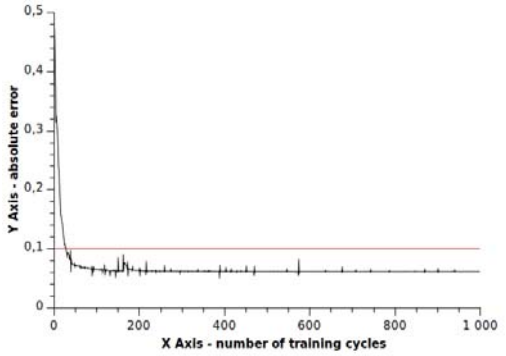


Fig. 6. Outputs from learning the neural network with 20 neurons in hidden layer, shown in linear scale

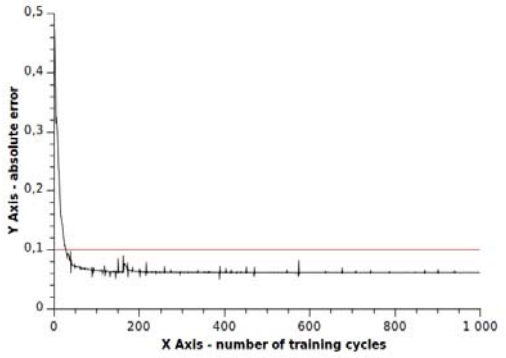


Fig. 7. Outputs from learning the neural network with 20 neurons in hidden layer, shown in logarithmic scale

D. *Experiment four –forty neurons in hidden layer*

In the fourth experiment was used neural network with 40 neurons in hidden layer. Such as in other experiments, the total number of specimens in the test group was one hundred. In this case we are getting best values from all. The number of samples in the test group which is satisfying the tolerance (0.10) is 98.56%.

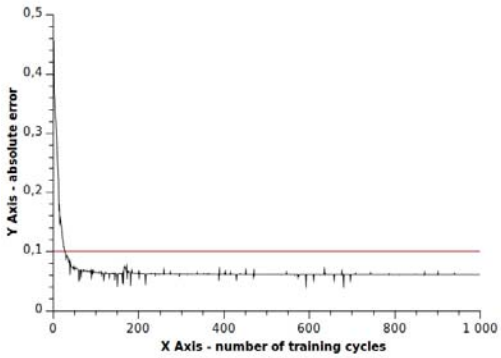


Fig. 8. Outputs from learning the neural network with 40 neurons in hidden layer, shown in linear scale

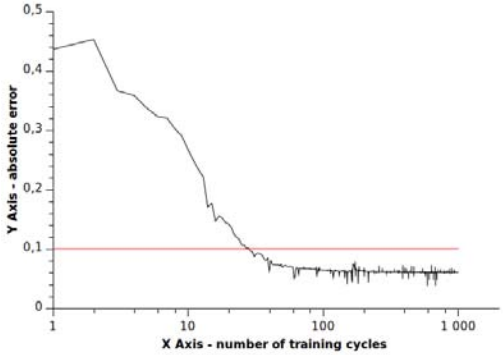


Fig. 9. Outputs from learning the neural network with 40 neurons in hidden layer, shown in logarithmic scale

From the above results is clear, that given tolerance (abs. error less than 0,1) meets neural network with 20 and 40neurons in hidden layer. But the best classification ability of the samples reaches neural network with 40 neurons in the hidden layer.

Other experiments with more types of neural networks have shown that more neurons in hidden layer (50, 60 and 100) have no significant effect.

V. Conclusion

We can see in those experiments that designed and trained neural network is able to sufficiently classify individual data packets. We have shown the possibility of application of artificial intelligence technologies in solving problems related to verification of data transfer in management and communications networks. We have made several experiments with neural networks with different structure. For testing and learning we need number of data. There is the scheme of the data generating and recording system with proposal of encoding communication protocol characteristics. Every neural network was trained with base of samples and outputs are clearly displayed in simply graphs.

In this case the suggested architecture of neural network

ISBN: 978-988-19252-4-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

is with 40 neurons in hidden layer. But this area is very complex and there is still a lot of work to do. This work represents just a part of the overall problem. In the next we can solve number of problems. For example propose a neural network, much larger, which would be able to separate incoming packets to more classes. Than we can to extend the model with other neural networks that would search in the data stream for other anomalies [9]. Probably we can modify this model to use other type of neural network, for example some type of the Kohonen network.

The suggested system is working correctly within specified conditions in our laboratory environment.

REFERENCES

[1] A. Robbins, "*GAWK Effective AWK Programming: A User's Guide for GNU Awk, for the 3.1.7 (or later) version of the GNU implementation of AWK The GNU*". Cambridge: O'Reilly Media, 2001. 456 p. ISBN 978-0-596-55594-8

[2] D. Tveter, "*The Pattern Recognition Basis of Artificial Intelligence*", Washington: Wiley-IEEE Computer Society Press, 1998. 388 p. ISBN 978-0-8186-7796-0

[3] M. Thottan and Ch. Ji, "Anomaly Detection in IP Networks" *Journal of Network and Systems Management : IEEE Transactions on signal processing, vol. 51, no. 8.* New York: Springer, 2007 p. 267 – 283. ISSN 1573-7705

[4] S. Axelsson (2009) Intrusion Detection Systems: A Taxonomy and Survey. Dept. of Computer Engineering, Chalmers University of Technology, Sweden. [Online]. Available: http://www.ce.chalmers.se/staff/sax/taxonomy.ps

[5] E. Denning, "An Intrusion Detection Model" *Proceedings of the Seventh IEEE Symposium on Security and Privacy May 1986* Oakland Cal. :ACM Press, Ltd. , 1996, p. 119-131

[6] J.B. Joyce (2009) Methods and apparatus for heuristic firewall . [Online]. Available: http://www.wipo.int/pctdb/en

[7] M.R. Hirstev (2009,august) The ANN Book. Edition 1 [Online] Available: ftp://ftp.funet.fi/pub/sci/neural/books

[8] P. Sinčák and G. Andrejková, "*Neurónové siete Inžiniersky prístup (1. diel)*", Košice, Slovakia, Technická Univerzita Košice, 2005.

[9] P. LichodzijewskiI and M. Heywoor, "Dynamic intrusion detection using self organizing maps",*Proceedings of the 2002 IEEE World Congress on Computational Intelligence*, New Jersey: IEEE Computer Society, 2002. p. 412 – 419.

[10] J.Cannady, "Neural Networks for Misuse Detection", *Proceedings of the 1998 National Information Systems Security Conference* (NISSC'98). Berkeley: USENIX Association, 1999. p. 443–456.

[11] A. Trnka and P. Tanuška, "Datawarehousing and datamining methodologies as a one of the possible benefit in control process", *Aktual'nye problemy i innovacii v ekonomike, upravlenii, obrazovanii, informacionnych technologijach : materialy meždunarodnoj naučnoj konferencii* (12-15 maja 2009 goda, Stavropo=B5-Kislovodsk). ISSN 2074-1685, p. 84-87.

[12] S. Haykin "*Neural Networks and Learning Machines, third edition*", Canada: Prentice Hall, 2009. 936 p. ISBN 01-31471-39-2

[13] V. Kvasnička et al "*Úvod do teórie neurónových sietí*", Bratislava: IRIS, Bratislava, 1997. 140 p. ISBN 80-8878-30-1

[14] K. Tan, "The Application of Neural Networks to UNIX Computer Security", *Proceedings of the IEEE International Conference on Neural Networks*, Vol.1. Melbourne, 1995, p. 476-481.

[15] G. Conti and A Kulsoom, "Passive visual fingerprinting of network attack tools", *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security table of contents*. Washington DC, USA: ACM, Washington DC, 2004, p. 45-54.

[16] R. Halenar "Contribution of Near Real Time ETL" *2011 International Conference on Database and Data Mining (ICDDM 2011)* .Proceedings / editors: Steve Thatcher and Liu Guiping. - Sanya : IEEE, 2011. ISBN 978-1-4244-9610-5. p. 243-247

[17] M. Roesch and Ch. Greem (2009), "SNORT Users manual 2.8.5, The Snort Project". [Online]. Available: http://www.snort.org/docs

[18] M. Kebísek and P. Schreiber "The possibility of utilization of neural networks at the data mining", *CO-MAT-TECH 2004 : International Scientific Conference*. Trnava, Slovak Republic, 14-15 October 2004. ISBN 80-227-2117-4. - p. 589-595