A Neural Network Based System for Classification of Attacks in IP Telephony

M. Voznak, J. Safarik and J. Slachta

***Abstract*—**This article deals with an application of artificial intelligence on classification of attacks in IP telephony. The proposed solution is based on the multilayer perceptron neural network. Current used solution of classification is typically based on statistical methods such as Hellinger-Distance, Holt-Winters or Brutlag's algorithm. The proposed solution MLP NN in the paper is used as a classifier of attacks in a distributed monitoring network of independent honeypot probes. The vulnerability of SIP elements to DoS attacks was examined in real infrastructure and evaluated their impact on a SIP server. We prepared a set of honeypots monitoring various aspects of nowadays VoIP infrastructure which bring valuable data about hacker's activity with no threat to the running system. Grouping single honeypots in one cloud solution enables to gather more data on hacker activities and to provide results with higher information value. Data about attacks on these honeypots are collected on a centralized server and then classified in the neural network. The paper describes inner structure of used neural network and also information about implementation of this network. The trained neural network is capable to classify the most common used VoIP attacks. With the proposed approach is possible to detect malicious behavior in a different part of networks, which are logically or geographically divided and use the information from one network to harden security in other networks.

***Keywords*—**attack classification, multilayer perceptron network, neural network, SIP attacks.

I. Introduction

THE main purpose of a honeypot is to simulate the real system and interact with anyone in the same way as the production system would. It watches the behaviour of anyone who interacts with it [1] and the information about attacks from a honeypot can be used for next processing [2].

This research has been supported by the Ministry of Education of the Czech Republic within the project LM2010005.

M. Voznak is an Associate Professor with Dept. of Telecommunications, VSB-Technical University of Ostrava (17. listopadu 15, 708 33 Ostrava, Czech Republic) and he is also a researcher with Dept. of Multimedia in CESNET (Zikova 4, 160 00 Prague 6, Czech Rep.), corresponding author provides phone: +420-603565965; e-mail: voznak@ieee.org.

J. Safarik is a PhD. student with Dept. of Telecommunications, Technical University of Ostrava (17. listopadu 15, 708 33 Ostrava, Czech Republic) and he is also a researcher with Dept. of Multimedia in CESNET, Zikova 4, 160 00 Prague 6, Czech Republic (phone: +420-596995848 and e-mail: kuba.safarik@gmail.com).

J. Slachta is a PhD. student with Dept. of Telecommunications, Technical University of Ostrava (17. listopadu 15, 708 33 Ostrava, Czech Republic) and he is also a researcher with Dept. of Multimedia in CESNET, Zikova 4, 160 00 Prague 6, Czech Republic (phone +420-596995847 and e-mail: jiri.slachta@gmail.com).

The paper deals with classification of attacks in IP telephony [3] and therefore data focusing on SIP protocol were acquired from honeypots.

The SIP (Session Initiation Protocol) is an open-source protocol which enables to establish, modify or terminate a general session [4], [5] and it is deployed for different kind of applications [6]. The IP telephony infrastructure based on SIP is very fragile to different types of attacks because of its similarity with HTTP and SMTP protocols. This can lead to loss of money, trust and other unpleasant consequences [7].

This situation could be solved partially with strict security rules, encryption and properly set VoIP servers. Even then is an attacker able to corrupt whole IP telephony network and stole sensitive information, eavesdrop calls, stole caller identities or deny the service (DoS attack) [8].

Monitoring of VoIP infrastructure, IDS/IPS (Intrusion Detection/Prevention Systems) or honeypots application can detect these attacks and malicious activity in the network. Some of these mechanisms can disrupt or mitigate certain types of attacks, but there is still much of remaining attacks, which can impact VoIP servers.

The information about SIP attacks from a honeypot application brings valuable source of network attacks. However analysis of data from these honeypots, especially in large or divergent network, cause unwanted overhead for network administrators.

With an automatic classification system is possible to automatically detect attacks on IP telephony from various set of honeypots and harden existing security mechanism. This honeypot concept is closely described in the section IV.

The paper is organized as follows. Section II presents results of experiments which we carried out on real SIP server and it brings clear evidence how efficient particular DoS attacks can be. At the end of section II, a proposed concept of security precaution with honeypot is introduced. Section III describes related works regarding mathematical methods and algorithms currently used for attack recognition. Section IV explains briefly the honeypot network concept applied for data set gathering. Section V deals with the proposed MLP neural network and its practical implementation is explained in section VI. Concluding section VII Last section describes contribution of this paper.

II. Impact of Attacks on SIP Server

We created a testing topology to measure DoS effectiveness and for further testing. It consists of SIP proxy server, hacker's PC and some endpoint devices. SIP proxy runs the Asterisk application, and the operation system implemented is Linux.

A. Attacks on server CPU using sipp

The Sipp programme is primarily used to simulate calls and to carry out SIP proxy stress tests, the original source code of SIPp was written by Richard Gayraud and modified by Olivier Jacques, but thanks to its rapidly growing popularity, it quickly became a community tool and many authors joined its development [10]. The application is an open source traffic generator that was designed specifically for testing purposes. Sipp is capable of simulation of both UAC and UAS and can also generate both signaling and media traffic.

During the development, the CLI was additionally extended by usage of XML files specifying the SIP communication scenarios and these scenarios were extended by the possibility of insertion of data from external CSV files. Another big advantage of SIPp is that it was (and still is) open source, written in C++. Therefore every user can easily implement any functionality according to his needs. Many patches for SIPp were created, extending its functionality in many ways. Thankfully, the maintainers of SIPp did not allow the tool to become a multi-branched unmanageable monster and restricted the way the patches were being adopted by the stable version of SIPp. The patches that did not make it to the stable version were made accessible in the project webpage for anyone to use or improve.

Sipp, with a simple upgrade of the call scenarios, can carry out malicious calls on SIP proxy. These calls are intended to overload server's CPU. Figure 1 shows the impact of these attacks on the server. The attack scenario applied was the same for each attack. Sending malicious packets started in 10 s and continued for 60 s. Another 30 s shows the time for which the server is still inhibited by the attack.

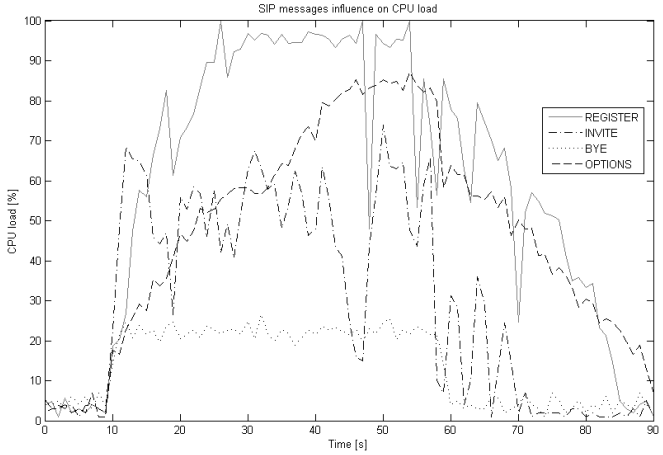

Fig. 1 The impact of different attacks on a server's CPU load

To enable the comparison of the efficiency of individual malicious SIP messages, the messages had been sent to the SIP server with the same rate (250 messages per second). For instance register flood attack is run by following syntax:

```
sipp -sn uac 192.168.0.10:5060 –sf reg_notag.xml -m 100000
-r 250 -s 1003
```

Clearly, the most effective SIP messages to attack a SIP server are REGISTER and OPTIONS. In the first case, the endpoint could not register or make calls, though running calls was not affected (the RTP stream only between endpoints). Sipp attacking scenario is shown below.

```
<?xml version=”1.0” encoding="UTF—8”?>
<scenario name=”SIP DOS REGISTER”>
 <send>
  <![CDATA[
   REGISTER sip:[service]@192.168.0.10:5060 SIP/2.0
   Via: SIP/2.0/[transport] [local_ip]:[local_port]
   From: <sip:[service]@[local_ip]:[local_port]>
   To: <sip:[service]@[remote_ip]:[remote_port]>
   Call—ID: [call_id]
   Cseq: 1 REGISTER
   Contact: sip: [service]@[local_ip]:[local_port]
   Max—Forwards: 70
   Subject: Performance Test
   Content—Type: application/sdp
   Content—Length: O
  ]]>
 </send>
</scenario>
```

Fig. 2 Example of xml attack scenario

Paradoxically was the scenario in Figure 2 one of the most effective register attack scenario. It's because of his simplicity. When SIP proxy doesn't have enough information in REGISTER message, it tries to find or guess missing data. This behavior leads to a further increase in computational load.

OPTIONS flood caused merely a delay in request processing, yet the situation deteriorated as the attack continued. In the end, not a single endpoint was able to register or make calls. The relatively long time necessary for the server to recover (in both cases) was rather surprising.

The delay in connection was evident in the attack performed by means of INVITE messages. Some calls failed to be connected at all. The attack was performed by a non–existing source user.

Attacks performed by means of BYE, CANCEL and ACK messages returned almost the same results, therefore Fig. 1 illustrates only the attack by means of the BYE message. During the attack, no call or registration was affected. BYE and CANCEL were not sent to end a particular call.

A. Security precautions

Security precautions against all these attacks include Snort rules tracking the number of messages sent to the SIP server from a particular source address. The blocking rules were

similar in most cases, like this Snort rule for blocking unwanted register flood.

```
alert udp $EXTERNAL_NET any ->
  $SIP_PROXY $SIP_PORT (msg:"SIP
  DoS attempt(registerflood)"; content:"REGISTER sip";
  detection_filter:track by_src, count 50, seconds 5;
  classtype:misc-attack; sid:1000001; rev:1; fwsam:src,
  10min;)
```

Fig. 3 The example of Snort rule.

Where the limit for messages was exceeded, the blocking rule was activated on the firewall by snortsam server.

The time of blocking malicious traffic was set to 10 minutes. After this interval is the blocked IP allowed to communicate with server. Snortsam automatically extend the time of blocking, when the attack is still detected. Whole solution than really effective against actual attacks and can simply converge to non-blocking state without human intervention when the attack ends.

The CPU load with the activated IPS (Intrusion Protection System) was about 9% during all these attacks as is depicted in Fig. 4 in case of the most efficient attack (REGISTER flood, see Fig. 1).

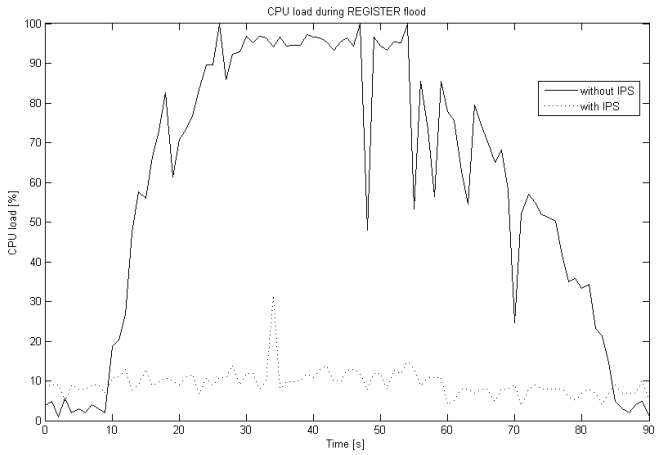

Fig. 4 The impact of an attack with IPS and without the protection.

TABLE 1: IMPACT OF ATTACKS ON THE SERVER

Attack type	VoIP server CPU impact	Threat	Communication with VoIP server
REGISTER flood	High CPU usage at a small rates	Very high	Impossible
INVITE flood	CPU load based on rates	High	Impossible
ACK,BYE, CANCEL flood	Low CPU load	Low	Very limited, delayed
OPTIONS flood	Gradually increasing CPU load	Very low	Impossible

The performed tests clearly indicate that SIP proxy is rather vulnerable to DoS attacks as is listed in Tab 1. As the server runs on a limited physical machine, only very basic protection mechanisms against certain DoS attacks can be implemented.

The most used precaution system based on linux and called as SSI consists of the following applications: Snort, SnortSam and Iptables [10]. The tests proved that the analysis of the server's traffic does not significantly affect server's performance.

The most dangerous attacks include flooding with REGISTER, INVITE and OPTIONS messages. No effective protection to be applied directly on the server exists against certain attacks. In this case, a more secure network topology is the only solution depicted in Fig. 5.

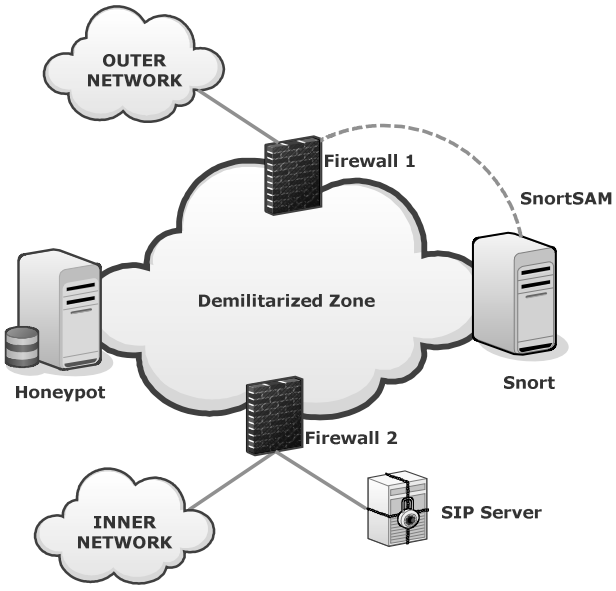

Fig. 5 The proposal of a safer topology.

The main change in this topology is the inclusion of a demilitarized zone (DMZ). It is located between two firewalls (inner and outer). The purpose of this zone is to separate the safe inner part from the rather dangerous outer part of the network. Both firewalls run SnortSam agents so rules can be dynamically applied on both machines.

The inner firewall (marked as Firewall 2) serves to protect the SIP server against the attacks from inside of the network. All traffic to the SIP server has to pass through at least one firewall. The safe inner network should be implemented as a matter of course. The potential attack from inside of the network would affect many users.

Using encryption, VoIP VLANs and methods such as ARP inspection and DHCP snooping should provide an adequate response to possible security breaches. The implementation of a QoS mechanism should further reinforce the protection. A honeypot located in the DMZ collects data on SIP events for further classification.

III. Classification of Attacks and Related Works

Detection of SIP infrastructure attack is solved with different ways in a range of studies and papers. Some methods rely on IDS system as Snort and its features or implement new features for better attack detection [11]. Other ways use statistical methods to analyze attributes of the SIP traffic [12]. There are methods for attack recognition based on Hellinger-Distance [13], forecast methods like Holt-Winters [14] and Brutlag's algorithm [15] or variety of SIP traffic anomaly detections.

Hellinger-Distance is used to quantify the similarity between two probability distributions (1), where p is the distribution of data within training period and q the distribution of data within short period [13].

$$H^2(P,Q) = \frac{1}{2}\sum_{i=1}^{n}\left(\sqrt{p_i} - \sqrt{q_i}\right)^2 \quad (1)$$

Holt-Winters model is used for detection of anomaly using predictive approach, see relations (2) – (5), this model is called also as the triple exponential smoothing model and it is a well-known adaptive model used to modeling time series characterized by trend and seasonality [14].

$$\hat{y}_t = L_{t-1} + P_{t-1} + S_{t-T} \quad (2)$$

where L is a level component given by :

$$\mathrm{L_t} = \alpha(y_t - S_{t-T}) + (1-\alpha)(L_{t-1} + P_{t-1}) \quad (3)$$

P is trend component given by :

$$Pt = \beta(L_t - L_{t-1}) + (1-\beta)P_{t-1} \quad (4)$$

And S is seasonal component given by :

$$St = \gamma(y_t - L_t) + (1-\gamma)S_{t-T} \quad (5)$$

Holt-Winters method was used to detect network traffic anomalies by Brutlag in [15]. In his concept of confidence bands, parameters $\hat{y}_{\max_t}$ and $\hat{y}_{\min_t}$ were introduced and is possible to measure deviation for each time point in the seasonal cycle (6).

$$\begin{aligned} \hat{y}_{\max_t} &= L_{t-1} + P_{t-1} + S_t - T + m * d_{t-T} \\ \hat{y}_{\min_t} &= L_{t-1} + P_{t-1} + S_t - T - m * d_{t-T} \end{aligned} \quad (6)$$

Where d is predicted deviation given by (7):

$$d_t = \gamma \mid y_t - \hat{y}_t \mid + (1-\gamma)d_{t-T} \quad (7)$$

Where $\hat{y}_t$ is predicted value of variable in moment t and y_t is measured value of variable in moment t and T is time series period. Then α is data smoothing factor, β is trend smoothing factor and γ is the seasonal smoothing factor, finally m is the scaling factor for Brutlags confidence bands.

This paper proposes a SIP attack classification with MLP (multilayer perceptron) neural network from honeypot application Dionaea, the algorithm is described in chapter V.

IV. Honeypot Network Concept

The classification engine based on MLP network is only a part of IP telephony infrastructure protection. A single honeypot application brings valuable information, with a combination of honeypots running in different networks at different locations should provide even more detailed data.

Exceeding number of running honeypots causes higher requirements for their management and support. The proposed design of a distributed honeypot network, shown at Fig. 6, solves this problem with a centralized server for data gathering, analysis and honeypot monitoring.

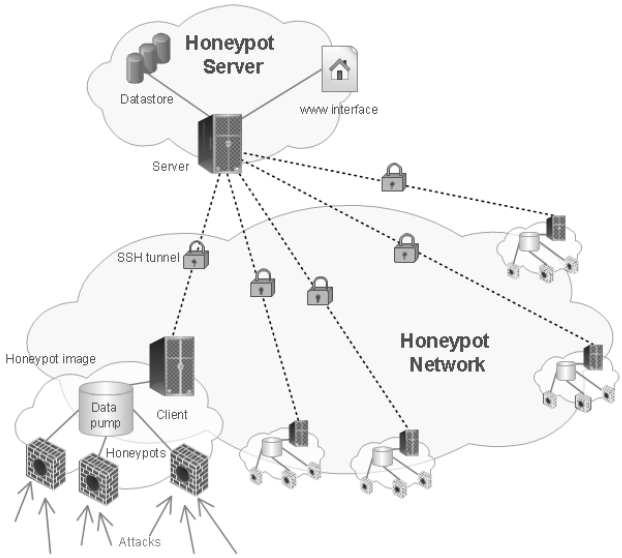

Fig. 6 Honeypot network concept

The main part of distributed network concept is honeypot image, which contains already prepared and preconfigured honeypot application, operation system and other software needed. This single node communicates with a centralized server via secured channel and periodically sends information about detected attacks to server.

Data gathered through data pump are converted into serialized objects. These objects are periodically sent to server for further operations and analysis. The honeypot client includes so-called dead man switch mechanism. With this functionality is possible to monitor honeypot status easily and there is no necessity of developing a specialized monitoring application. Clients send a simple UDP packet to server each 5

minutes, the dispatched information are checked on server by timers which are individually configurable for each image.

The server identifies a non-active honeypot after three undelivered messages and creates a system warning which is reported in standard log file and eventually an administrator is notified by e-mail as well.

Neural network algorithm described in this paper is used in honeypot network hierarchy as a module on the centralized server for classification of SIP based attacks.

More information about distributed honeypot network covers previous paper [16].

V. MLP Neural Network

Neural networks try to model information processing capabilities of the nervous system of mammals. This nervous system is composed of millions of interconnected cells in a complex arrangement. The artificial neural network tries to model this design and it is also deployed for different prediction purposes in telecommunication systems [22]. The function of a single neuron is well known and serves as a model for an artificial one.

Even that we do not complete understand the complexity and massive hierarchical networking of the brain, with its incredible processing rate, artificial neural network handle complex problems by using different topologies. Many versions of these topologies are known today, each of them has its pros and cons.

The feed-forward MLP neural network was used for VoIP attack classifications. It consists of several layers, each containing the specific number of neurons called perceptron. These perceptrons in one layer are interconnected to each other in the following layer (this connection could be also called a synapse) [17].

The Fig. 7 shows the inner structure of used MLP network. The MLP network solution used for classification has two hidden layers, with one input and one output layer.

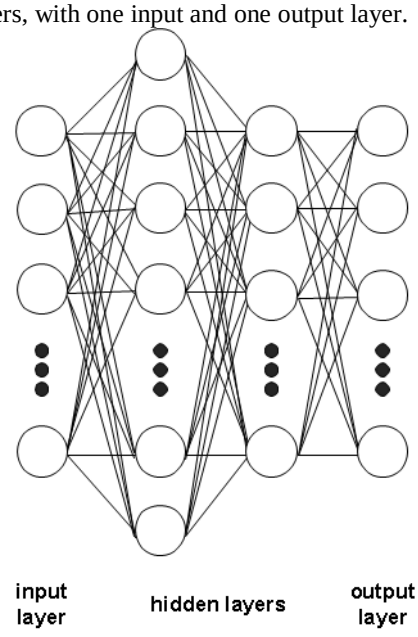

Fig. 7 MLP neural network topology

Each neuron in the input layer has a value based on input parameters. This layer has the same number of neurons as there are parameters in the input set. After the input layer continues two hidden layers and output layer. The output layer has the same number of neurons as the number of attack classes, so each neuron is then a single class of learned attack.

Number of neurons inside hidden layers depends on neural network configuration and are typically higher than the number of neurons in input or output layers.

A. Perceptron

The perceptron is a more general computational model than McCulloch-Pitts units. The main innovation is in including numerical weights for connections and a special interconnection pattern. The activation function for neuron – sigmoid impacts the final potential of a neuron. This result potential is then transmitted through connections to neurons in the next layer while afflicted by each connection weight. These weights serve as a memory for neural network. Inputs for the activation function are real inputs x_1, x_2,…, x_n from the previous layer, with the associated connection weights w_1, w_2,…, w_n.

The output of a neuron is between *0* and *1*, where *0* means inhibition and *1* excitation. The final value on the output of neuron (y) depends on its activation function. As was mentioned before, this function is a real sigmoid function (8) and (9).

$$S_C : \Re \to (0,1) \tag{8}$$

$$y = S_C(z) = \frac{1}{1+e^{-cz}} \tag{9}$$

The relation (10) shows parameter z, which is the sum of the output from previous layer neuron x and multiplies by corresponding connection weight w. Parameter c represents a skewness of the sigmoid function (typically it is *1.0*). Higher values of parameter c bring the skewness of a sigmoid function closer to the step function [18], [19].

$$z = \sum_{i=1}^{n} w_i x_i \tag{10}$$

B. Backpropagation Algorithm

As was mentioned before, the memory of neural network is saved in connection weights. The neural network learning mechanism – backpropagation is used to acquire these values. While classifying, the neural network is feed-forward mode and information is transferred from the input layer to the output layer. Backpropagation works as a reverse mechanism to feed-forward, with the specific set of data called training set. Training set has same the format as attack inputs for neural networks but contains also the final result of classification (or the class of the specific attack).

The core of a backpropagation algorithm and the neural network learning is a process of weight adaptation. It is done on the training set of inputs with known outputs. The solution of learning problem is a combination of weights with the minimal error function.

$$\delta_j = \sum_{k=1}^{n} \delta_k y_k (1 - y_k) c w_{jk} \quad (11)$$

The equation (11) shows computation the of backpropagation error (δ) for connection weight in one layer (indexed as *j*). It is counted as a multiplication of higher layer (indexed as *k*) backpropagation error, actual output, expected output and actual weight of the connection. Parameter *y* represents the output of neuron, *x* always its inputs. Parameter *c* is the expected output and *w* the connection weight. The backpropagation error is then used in weight adaptation equations (12) and (13).

$$\Delta w_{ij} = \eta \delta_j y_i \quad (12)$$

$$w_{ij} = w_{ij} + \Delta w_{ij} \quad (13)$$

Learn rate parameter (η) affects connection weight correction, used to lower the value of the error function (13). The learn rate parameter (η) serves to set a proper step of correction in one backpropagation iteration [18], [19].

One iteration of backpropagation learning uses all records from the training set. The last parameter w_{ij} is the connection weight from the previous layer (*i*) to the actual layer (*j*) as shown Fig. 8.

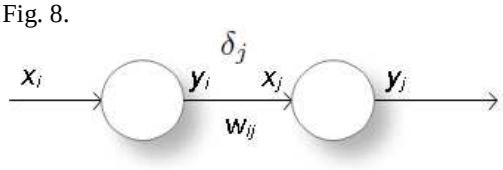

Fig. 8 Indexing between layers

VI. Practical Implementation

In previous research covered in [19] was used MLP neural network for detection of six basic SIP attacks. Because there are changes in input vector and attack classes, inner structure of MLP network was changed.

A. MLP neural network configuration

The new final neural network contains 10 input layer neurons as the previous generation. The two hidden layers contain 30 and 24 neurons, the last and output layer 8 neurons. The previous generation was not able to detect correctly one class of attacks, so it detects only five types of SIP attack. With the change to eight classes, there is more robust and accurate detection of attacks.

The inner structure of previous generation network was based on test of convergence of 100 backpropagation iterations. These tests proved different mean times of learning for different inner structures, see Fig. 9 (timesXXYY – XX means number of neurons in first hidden layer, YY – number of neurons in second hidden layer). The impact of the structure is evident only for backpropagation learning time.

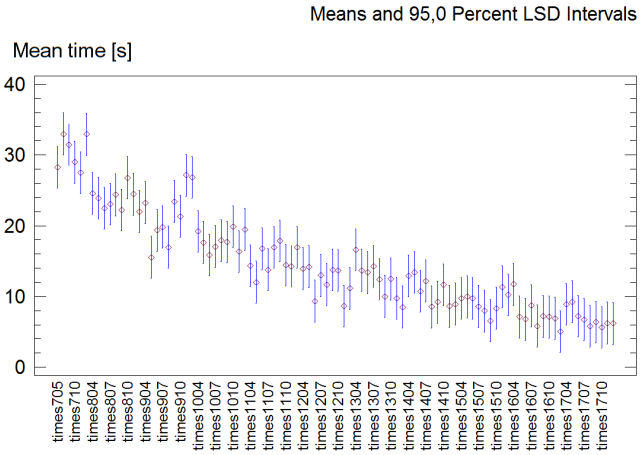

Fig. 9 Mean times of 100 backpropagation iteration with different inner structure

With higher numbers of neurons in hidden layers, the mean time of backpropagation learning decreases. On the other hand, raise memory and computational requirements of MLP neural network. After proper learning, inner structure has no more statistically significant impact on attack classification. The final neural network structure for new generation network contains 30 and 24 neurons in hidden layers, because of conducted investigations and tests.

Both generations of MLP neural networks are learned to the same confidence interval. The successfulness on the training set is always lower than 5%. This ensures statistically significant classification capability on the training set. Both generations use the same configuration of skewness (*1.0*) and learn momentum (*0.8*).

B. Data source for classification & input vector parameters

All attack information is gathered through multi-service oriented honeypot application Dionaea. We choose Dionaea for its features tested in previous research. It emulates and monitors traffic of a SIP PBX (Private Branch eXchange). However only specific set of information is saved to the internal database e.g. used SIP message, IP addresses, ports or specific SIP header values.

Dionea honeypot contains strict information about malicious traffic. All running honeypots are accessible through internet on public IP addresses (IPv4). No legitimate calls or devices connect to this honeypot, so only malicious traffic or misconfigured devices communicate with it.

All attack data save Dionaea to sqlite database. The database contains several tables for specific protocols and functions. But all tables have a single pointer to a table connection, which contains basic information about attack.

Information about SIP attacks is distributed in five tables, each with different information. Selecting only single lines from these tables for classification is valueless, partly because of request/response behavior of SIP protocol. All data for final classification are aggregated from selected tables to an array with 10 attributes.

These individual 10 attributes then serve as an attack vector (or neural network input). The aggregation depends on attack origin and also time of last message occurrence (there is 5 minute sliding window after last message detection). Attributes are in the following order: attack time duration; connection count; REGISTER message count; INVITE msg. count; ACK msg. count; BYE msg. count; CANCEL msg. count; OPTIONS msg. count; SUBSCRIBE msg. count; connection rate. The connection count attribute holds the number of connection from a single source on honeypot. The connection rate is the ratio of all received SIP messages to connection count.

C. Backpropagation configuration

MLP neural network use backpropagation algorithm for learning. SIP attack classification MLP network is evaluated as learned, if there correctly identify more than 95% of items in the training set (so the confidence interval is always lower than 5%). Before backpropagation starts, all connection weights are set randomly from range (*-1,1*). After first initialization of weights starts backpropagation iteration. Each iteration cycle uses all items from the training set. After specific number of iteration cycles (100) is automatically checked successfulness of classification. If the successfulness on the training set is lower than specified threshold, backpropagation runs another iteration cycle. When the successfulness is higher than 0.95, the neural network learning is done. To avoid the possibility of stuck in local extreme, system automatically reinitialize all connection weights after 2 500 000 backpropagation cycles.

D. Training set

The training set, along with neural network input, is one of key parts of neural network. If the items in the training list are not specific representative of an attack group, the attack cannot be successfully classified. This misclassification has a negative influence on learning and prevents successful learning of attacks.

As a source for the training set was always used real attack traffic. This traffic is aggregated and then classified by hand. From this classified set is prepared training set with attack groups. The new MLP neural network training set consist of 104 items, 13 items for each attack group. These classes are options tests; options scanning; call testing; unknown protocol; register and call; registration test, registration flooding; register attempt. This set of attack classes corresponds to detected types of attack from a period of two months.

E. Reference set

As a reference set serves an aggregated set attack vector detected on various honeypots. All runs Dionaea application but on different hardware, IP addresses and in different geographical locations. One honeypot is masquerading behind a set of IP addresses for raising the malicious traffic. Each honeypot application runs for other period because they are not started together, so they provide a diverse group of detected attacks. All attacks detected on these honeypots were classified by hand, so we cannot eliminate human factor error.

Tests of new MLP neural network on three reference data set bring exciting information. These set do not contain data from the training set. Result of analyses with MLP networks has following successfulness: 94.94%; 79.85% and 97.54%. Totally were detected 1631 attack groups and 57752 SIP messages (data for three months period). The lowest classification precision 79.85% was caused by new call attack, which was not included in the training set.

VII. Conclusion

SIP protocol is an open-source protocol and becomes one of the most used protocols for handling of VoIP services. There is even estimated rise of SIP devices in the future and the security of SIP device and PBX will become a crucial question. This situation will lead to a higher exposition to various types of attacks and even misconfigured devices. These factors will have a negative impact on VoIP service. Previous research in our laboratory confirmed high vulnerability of SIP servers to various attacks [20].

The proposal distributed honeypot network in combination with neural network classifiers serves as another security level. But the potential lies not only in detection capability and attack research. With the possibility to change firewall rules or network routing, whole system can prepare precaution mechanisms against attacks even when it do not influence the target network. The proposal of such monitoring system is distributed honeypot network.

As we found out, similar research on using neural networks with for identification and classification of attacks in VoIP has been carried out in [21], where a feedforward artificial neural network was applied as well. Nevertheless, this research was focused mainly on identification of DoS attacks and can not be compared with our results. Since we exploit data from honeypots, where only malicious traffic is directed, our research is strictly oriented on classification of attacks. The trained classifier was able to distinguish the particular type of attack with high reliability.

Classification by human is very precise, but time consuming and expensive. It is typically conducted after the damage is done. Automatic classification mechanism brings a solution for VoIP classification and simplifies the analysis of attacks. The biggest disadvantage of this solution is it strong bindings on the training set. The MLP neural network cannot adapt to new attack classes or scenarios.

Test of the new MLP classification network on reference sets prove its quality but shows its limits and new ways for

improvements. The future plans for neural network classification cover improving the accuracy of existing solutions and implementation of other evolutionary and statistical algorithms for attack classification. One of challenges is also in detection of attacks in legitimate VoIP traffic.

REFERENCES

[1] L. Spitzner, *Honeypots: Tracking Hackers*, Addison-Wesley Proffesional, 2002.

[2] N.A. Quynh, Y. Takefuji, "A novel stealthy data capture tool for honeypot system," *WSEAS Transactions on Computers*, Volume 5, Issue 1, January 2006, pp. 209-215.

[3] F.De Rango, M.Tropea, P.Fazio, S.Marano, "Overview on VoIP: Subjective and Objective Measurement Methods", *Int. Journal of Computer Science and Network Security*,Vol.6, N°1, Jan. 2006, pp.140-153.

[4] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, "SIP: Session Initiation Protocol", IETF RFC 3261, June 2002.

[5] L. Macura, M. Voznak, K. Tomala, J. Slachta, "Embedded multiplatform SIP server solution", In *Proc. TSP 2012, art. no. 6256295*, 2012, pp. 263-266.

[6] F. De Rango, P. Fazio, F. Scarcello, F. Conte, "A new distributed application and network layer protocol for VoIP in mobile ad hoc networks", *Mobile Computing, IEEE Transactions on*, vol. 13, Issue: 10, Oct. 2014, pp. 1536-1233, DOI 10.1109/TMC.2014.2307315.

[7] M. Voznak, J. Safarik, F. Rezac, "Threat prevention and intrusion detection in VoIP infrastructures," *International Journal of Mathematics and Computers in Simulation*, Volume 7, Issue 1, 2013, pp. 69-76.

[8] M. Li, M. Li, X. Jiang, "DDoS Attacks Detection Model and its Application," *WSEAS Transactions on Computers*, Issue 8, Volume 7, August 2008, pp. 1159-1168.

[9] M. Voznak, J. Rozhon, "SIP end to end performance metrics," *International Journal of Mathematics and Computers in Simulation*, Volume 6, Issue 3, 2012, Pages 315-323

[10] R. Chi, "Intrusion detection system based on snort," *Lecture Notes in Electrical Engineering*, Volume 272 LNEE, Issue 3, 2014, pp. 657-664.

[11] J. Gomez, C. Gil, N. Padilla, R. Banos, C. Jimenez, "Design of a Snort-Based Hybrid Intrusion Detection System", *Lecture Notes in Computer Science*, Volume 5518, 2009, pp 515-522.

[12] H. J. Kang, Z. Zhang, S. Ranjan, A. Nucci, "Sip-based VoIP Traffic Behavior Profiling and Its Applications", In *Proc. MineNet'07*, 2007.

[13] H. Sengar, H. Wang, D. Wijesekera, S. Jajodia, "Detecting VoIP Floods Using the Hellinger Distance", *IEEE transactions on parallel and distributed systems*, Vol. 19, No. 6, June 2008.

[14] M. Szmit, A. Szmit, "Usage of Modified Holt-Winters Method in the Anomaly Detection of Network Traffic: Case Studies, *Journal of Computer Networks and Communications*, Volume 2012, 2012, Article ID 192913.

[15] J. D. Brutlag, "Aberrant Behavior Detection in Time Series for Network Monitoring", In *Proc. 14th System Administration Conference*, New Orleans, 2000, pp. 139-146.

[16] J. Safarik, J., Voznak, M., Rezac, F., Partila, P., Tomala, K., "Automatic Analysis of Attack Data from Distributed Honeypot Network", In *Proc. Mobile Multimedia/Image Processing, Security, and Applications 2013*, 875512, Baltimore, May 28, 2013.

[17] R. Rojas, *Neural Networks*, Springer-Verlag, 1996, ISBN 978-3540605058.

[18] J. Heaton, *Introduction to Neural Networks for JAVA*, 2nd Edition", Heaton Research, 2008, ISBN 978-1604390087.

[19] J. Safarik, P. Partila, F. Rezac, L. Macura, M. Voznak, "Automatic Classification of Attacks on IP Telephony", *Advances in Electrical and Electronic Engineering*, Vol. 11, Issue 6, 2013, pp. 481-486, ISSN 1336-1376.

[20] F. Rezac, M. Voznak, K. Tomala, J. Rozhon, J. Vychodil, "Security Analysis System to Detect Threats on a SIP VoIP Infrastructure Elements", *Advances in Electrical and Electronic Engineering*, Vol. 9, No. 5, 225-232, 2011, ISSN 1336-1376.

[21] N. Shekokar and S. Devane, "Anomaly detection in VoIP system using neural network and fuzzy logic," *Communications in Computer and Information Science*, Volume 250 CCIS, 2011, pp. 537-542.

[22] P. Fazio, F. De Rango, I. Selvaggi, "A novel bandwidth reservation algorithm based on neural networks path prediction in wireless environments", *Int.Symposium on Perf. Evaluation of Computer and Telecommunication Systems (SPECTS '10)*, Jul 2010, Ottawa.

Miroslav Voznak holds position as an associate professor with Department of Telecommunications, Faculty of Electrical Engineering and Computer Science (FEECS) VSB-Technical University of Ostrava, Czech Republic. He received his M.S. and Ph.D. degrees in telecommunications, dissertation thesis "Voice traffic optimization with regard to speech quality in network with VoIP technology" from the Technical University of Ostrava, in 1995 and 2002, respectively. The topics of his research include next generation networks, IP telephony, speech quality and network security. He is a member of the editorial boards of several journals and conference committees of international scientific conferences.

Jakub Safarik received his M.S. degree in telecommunications from VSB – Technical University of Ostrava, Czech Republic, in 2011 and he continues in studying Ph.D. degree at the same university. His research is focused on IP telephony, computer networks and network security.

Jiri Slachta received his M.S. degree in telecommunications from VSB – Technical University of Ostrava, Czech Republic, in 2014 and he continues in studying Ph.D. degree at the same university. His professional activities are focused on embedded systems, networks and application development for mobile systems.