Multi-criteria analysis and prediction of network incidents using monitoring system

Lukas MACURA[1,2,*], *Miroslav VOZNAK*[2]

[1]Institute of Informatics, Silesian University in Opava, Na Rybnicku 626/1 746 01 Opava, Czech Republic
[2]Dept. of Telecommunications, VSB - Technical University of Ostrava, 17. listopadu 2172/15, 708 33 Ostrava-Poruba, Czech Republic

*macura@opf.slu.cz
(Received: 13-February-2017; accepted: 24-April-2017; published: 8-June-2017)

Abstract. *Today, network technologies can handle throughputs up to 100Gbps, transporting 200 million packets per second on a single link. Such high bandwidths impact network flow analysis and as a result require significantly more powerful hardware. Methods used today concentrate mainly on analyzes of data flows and patterns. It is nearly impossible to actively look for anomalies in network packets and flows for a small amount of change of monitoring patterns could result in big increases in potentially false positive incidents. This paper focuses on multi-criteria analyzes of systems generated data in order to predict incidents. We prove that systems generated monitoring data are an appropriate source to analyze and enable for much more focused and less computationally intensive monitoring operations. By using appropriate mathematical methods to analyze stored data it is possible to obtain useful information. During our work, some interesting anomalies in networks were found by utilizing simple data correlations using monitoring system Zabbix. We concluded that it is possible to declare that deeper analysis is possible due to Zabbix monitoring system and its features like Open-Source core, documented API and SQL backend for data. The result of this work is a new approach to the analysis containing algorithms which allow to identify significant items in monitoring system.*

Keywords

Zabbix, Monda, ANN, SOM, MLP, Classification, Prediction.

1. Introduction

Network and security incidents can be seen as unrelated system events which correlate together by using mathematical models. The generally incidents start, continue and then end, and during this time there will be some system events and process changes that will correlate. In principle, it is inefficient to perform high-speed deep analysis of all communication. There is a better approach - to find correlations between processes and after this, do deep analysis only in small time window found as a result of processes analysis. Let's take an attack on the SMTP server and as an example: a standard network traffic analysis cannot accurately and entirely spot this attack. It can, however, be noticed by analyzing the SMTP server logs, correlate events with disk IOPS and CPU load. It is rather challenging for a network administrator to foresee all kind of incidents and defend against them. An attacker (black hat) just needs to succeed once, while security and network admins (white hats) have to succeed every time to protect an organization from successful cyber-attack. It is this inequality that calls for a new approach in using existing technology. Most networks are already equipped with monitoring systems capable of recording important system and network properties and logs, such as CPU utilization, processes running, disk load, utilization and errors on network ports, to mention just a few. This collected

data is stored in the database and network administrators can manipulate it to produce useful graphs or reports. Furthermore, monitoring systems allow for trigger management to allow for simple rules such as "if the port Eth0/0 is greater than 80% in 5 minutes, send an email".

This paper explains the new and efficient method to analyze monitoring systems data to predict anomalies and cyber-attacks. The advanced analysis employs neural networks and machine learning methods. A well trained neural network can predict known and unknown types of incidents with high probability, and warn administrators before these occur. Other approaches exist based on Artificial Intelligence that can be used for this purpose, for example, based on markovian model [1], [2] or based on swarm intelligence [3], [4]. It is also possible to find the real cause of the problem. For example, when indicator (e.g. free disk space) is out of range, but the actual cause is elsewhere (the attack on a service). The big advantage of using network and system monitoring tools is that the basic correlation rules are already in monitoring systems as these are typically set up to inform administrators about abnormal behavior that could impact system availability. This present a suitable way to train neural networks.

There are many different monitoring systems. All the principles written here are theoretically applicable to any monitoring tool. However, we selected Zabbix [5], an Open-Source project. The main selection reason is the proper organization of internal data and history in this system, the possibility of in-depth, focused and automated analyses directly using SQL and open API.

We developed a new open-source tool, Monda [6]. Its primary purpose is a selection and preprocessing Zabbix data allowing the use of more sophisticated mathematical methods and procedures. The project is hosted on Github.com server and is accessible to the entire community. The project currently includes 6200 lines of source code. It has been designed for team collaboration and allows adding of new analyses.

2. Methods

There are a variety of methods to search for incidents in networks. Let us mention at least the most basic and most used techniques. Each method has its advantages and disadvantages and can be used for a specific type of incident to be more successful [7]. There can be some security incidents (like a compromise of the system, DOS attack) or regular incident (like the system is overloaded due to misconfiguration or lack of disk space). Each event leaves a footprint and can be found by using some analysis [8]. For a faultless operation of the network, it is very crucial to prevent any incidents. This can be achieved by proper configuration and backup services. But even in the well-configured network, there are some incidents which cannot be manually predicted by the administrator due to a big amount of event and state combinations. Therefore, some automatized prediction of incidents is important.

We can say that if we want to manage network uninterrupted, the monitoring system is a crucial part of it. We need to monitor and track most of the network equipment and servers to have a real footprint of the network. It is ineffective to record the network logs without monitoring them. It can occur very often that some data source (from some security probe) is missing due to failure. If we do not monitor this, the network seems to be without problems even if there is security incident on the background. There is yet another reason for network monitoring. If an attacker knows where security device is located and he knows its vulnerabilities, he can focus the first attack directly there. If this attack is successful, a security device is not functioning properly, and there is no monitoring enabled, the administrator will not know about this and next attacks.

3. State of The Art

There are a lot of tools to identify and classify network incidents, but there is no tool based on data from the monitoring system. We choose Zabbix and data from Silesian University to do

further analysis of data because of their availability and because of Zabbix features.

The correct choice of methods is crucial for any analysis. The best way seems to use neural networks and their self-organization. Current systems utilizing neural networks are usually specialized for one source of analyzed data. There are software and hardware platforms that can detect anomalies in network traffic by inspecting packets or streams [8]. Similarly, there are platforms which can analyze the log files [7]. Their disadvantage is a mostly narrow focus. Even if information from flows is important, it's usually not enough for deeper analysis because there is no following information, such as load for each server or network elements. Modern devices can classify traffic based on the days of the week and time of day to respect common usage in networks based on work hours and work days. It is even possible to use special probes as source of data for monitoring systems like VoIP attack analysis [9]–[11].

The generally, security must be carried very carefully to arise from the incident, so hierarchically. The local network is needed to set antispoof and general ban of unsafe services that are not used. Goodly configured network should not allow trivial attacks like faking MAC addresses, IP addresses or ARP. As an opposite, in carrier level network, there has to be the only limited amount of interventions. A typical example of the attacks on the carrier level where the attacks to some news sites in the Czech Republic. Even though the stream of data is flowing across most of the big operators, the real protection against attacks must occur at the server itself. Our goal was to elect interesting data from the monitoring system and use them for further analysis. There are a lot of data in the monitoring system.

4. Algorithms

In this part, we focus on issues concerning data selection, algorithms and data structures.

4.1. Data Selection

To be able to focus and orientate in a huge amount of data, pre-processing is needed. This part of the analysis is crucial. It would not be possible to do complex analysis of all data in the monitoring system. And it would not direct to the right results. Even for future, the preprocessing part will be the primary place for any optimization and improvements. Mathematical principles and formulas are strict, and their algorithms are known and well optimized. But preprocessing is data specific and has to be driven with a focus on features of inside data. There are a lot of data inside monitoring system, and there are many kinds of it. It can be a number, specifying the state of the interface which is an integer from 0 to 10, it can be a float as processor load or an integer which saves actual disk free space in bytes. Small change in one item is not important but same change in another item can mean a big problem in the network. Even more, data have some specific features like recurrence, statistical features, and some statistical associations [12]. From this reason, we have made our own Open-Source software Monda, hosted on GitHub, which is highly configurable and which does pre-processing part (but even more). Our goal was to create a framework and environment where every user can create its version of preprocessing strategies based on his setup. After we created and tested Monda, it became possible to do further analysis of data focused to Time Window, Host or network process. One big advantage of this software is that it can be automatized.

The primary goal of our work is in an innovative approach to selection and pre-processing of data using algorithms above. We used dimensionless quantity LOI (Level Of Interest), which is an integer. The bigger LOI is, the more interesting data are. Algorithms and formulas used will be explained later. A further mathematical analysis is based on LOI. When doing some complex computation, objects with highest LOI are selected first. If there are enough CPU, RAM, and disk size, it is theoretically possible to analyze all data inside or all data for specific Time Window only. But LOI will do a preview of data

inside Zabbix and selects most interesting data for subsequent analysis.

4.2. Data Structures

Data in Monda are oriented into Time Windows and Item statistics, see Fig. 1. The basic feature of Monda is that data in Zabbix are untouched. So computation is based on Zabbix database and results from it is saved into Monda database. Monda database only describes data in Zabbix and mark them with adequate LOI.

As shown in Fig. 1, Item which seems to be important in one Time Window can be uninteresting in another window. The typical example is free disk space. In most Time Windows, there is no change of it. But in the specific window where some attack affected it, change can be bigger, and Item can be interesting at this time. The algorithm used in pre-processing will prefer a combination of Item and Time Window if there are more changes. See below. Same to Time Windows, there can be interesting and uninteresting one. During work hours, there are a lot of changes in network metrics and these windows will be preferred. On the opposite side, night hours can be skipped because there were no interesting processes. Step by step algorithms follows.

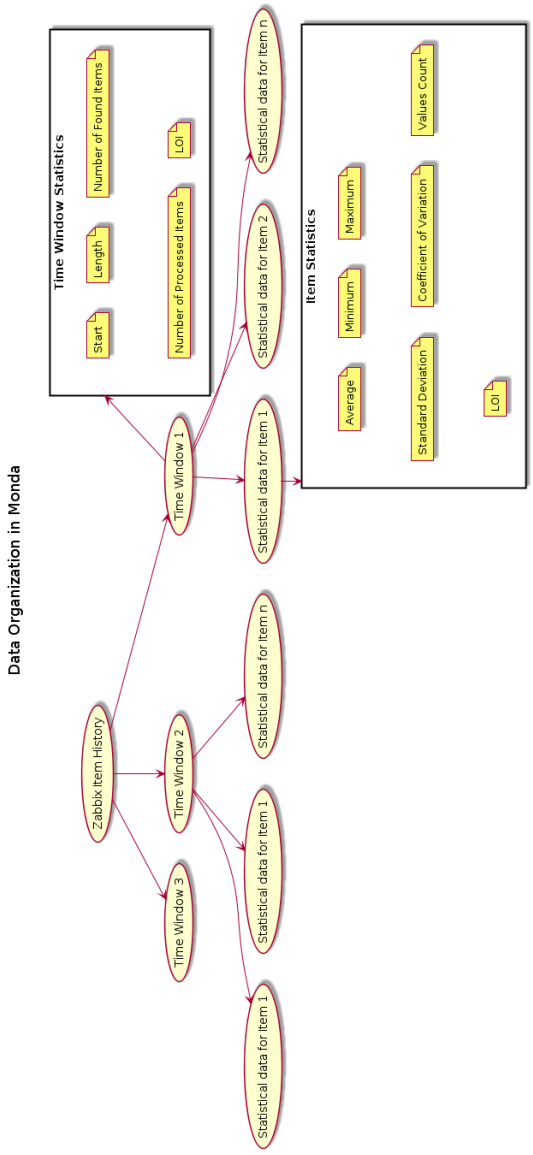

Fig. 1: Organization of data in Monda.

1) Time Windows

For all Time Windows, Item statistics are computed. It means that for each Time Window, Zabbix history is searched, analyzed and computed for all Items found inside. Some Items are automatically removed at this part of the analysis because there is not enough data for them in given window. For example, for Item "disk free bytes" which is fetched each 20 minutes there is not enough data in 1-hour window (3 values) to do any proper analysis over it.

There are basic statistics computed for each Time Window, see below. All constants are configurable by Monda. This is the first place where data are reduced. Useless data (Items with small changes, Items without history or Items with small standard deviation) are not copied into Monda database.

2) Time Window Statistics

- found - overall number of items found in window
- lowcnt - items with low number of values
- lowavg - items with mean which is near to zero

- lowstddev - items with small standard deviation
- lowcv - items with small coefficient of variation
- avgcnt - average count of history data per item
- avgcv - mean of coefficient of variation
- Level of Interrest LOItw (1).

$$LOI_{tw} = 100\, avg\,(cnt)\; avg(cv)\, \frac{processed}{found} \quad (1)$$

3) Correlation Statistics

See equation (1). After marking Items and Time Windows with Loi, correlations are computed. From the principle described above, most interesting correlations are computed. It is not possible to compute all of them because the combination of all Items is wide. There are two kinds of correlations to compute. One is for correlation between Items in specific Time Window and correlation of the same Item in different Time Windows. The first type is to analyze the behavior of different values in the same time while second is to analyze the behavior of Item in different times. For example, to compare disk space usage in common hours of day. Pearson correlation coefficient is applied (2) and computed in two steps.

$$\begin{aligned} cov\,(X,Y) &= E\,(X - E\,(X))\,(Y - E\,(Y)) \\ &= E\,(XY) - E\,(X)\,E\,(Y) \end{aligned} \quad (2)$$

There are two steps of computation as is depicted in equation 2. This is due to reason that computations are SQL based and are computed directly on SQL server to be fast enough. Some databases can compute $cov\,(X,Y)$ directly, but because of compatibility reasons, we use two steps which work on almost any database engine.

4) Correlation within same Time Window

More Items are correlated in same Time Window. For example, how network interface load correlated with disk load at given Time Window.

5) Correlation within same hour of day

It is common that correlations can occur even between different Time Windows and same Item. For example, there can be a significant correlation of disk load on the backup server at backup hours each day. Similar correlations can be found for weekly backups in given day of the week. Instead of random processes which occur in Time Windows, these correlations represent in most situations recurrent operations in the network. Correlation does not imply causality. But it is not important at this phase of analysis. Most important is to know if two Items correlated in Time Window and if so, how much it was.

4.3. Monda

Monda [6] was designed and coded from scratch. It was designed to do most of the computations directly in SQL. This was crucial for speed up analysis. The result of analysis is stored back to SQL tables, so it is possible to do next quick operation within it. Zabbix server was configured not to delete any data. Instead of deleting history data it created partitions of SQL tables in regular intervals. Monda is used as a tool which concentrates to significant amount of data in Zabbix database and tries to find most interesting values and windows automatically. As mentioned, it is not possible to do complete analysis with over all data inside in real time. And in fact, it is not needed. A lot of data in monitoring system are not interesting. Monda never copies data from Zabbix. Instead of it, it uses algorithms and procedures which orientate inside data and copies statistical results into Monda database. At this time, Monda includes approximately 6200 lines of code. Overall design rule was not to affect Zabbix server availability or

length	found	processed	ratio	ignored	lowstddev	lowavg	lowcnt	lowcv
1 day	35863	4593	12%	31269	29281	779	105	1101
1 hour	35751	1079	3%	34671	30266	140	3794	468

Tab. 1: Time Windows Statistics Example

performance. Zabbix uses its tables very often and utilizes SQL server by itself. From this reason it was crucial to take care about all Monda operations to work in most situations in idle time of Zabbix server. Next, it was needed to set SQL timeout for Monda queries. If Monda analysis would take more than 10 minutes per query, it was stopped automatically.

4.4. Neural Networks

After pre-processing, interesting data was fed into neural networks. There was two kind of networks created - Self Organizing Maps and MLP network.

1) Self-Organizing Maps

SOM analysis did not produce strict results. Because of many kinds of inputs, it was hard to feed and learn network with right data. It is possible to focus on this analysis in future for specific kind of network devices or servers. But for generalized monitoring system data, it is not suitable. Next utilization of SOM could be a fingerprint of servers or network devices. Each network device has its own unique features in data, and some process could save this features or classes of data into the database. But it needs much more investigation over concrete data which was not our goal. Monda is prepared for SOM statistics so anybody in future can try it and do its analysis of his data.

2) MLP network

MLP network is suitable for data classification and prediction and that was the right mathematical method to use. We used Weka software for neural networks analysis. Algorithm used to feed MLP:

- Choose Trigger which was most active in given Time Window
- Find all Items which caused Trigger to evaluate
- Add Items which correlated in same Time Window
- Exports of data
- Exclude data which were not significant

3) Proposed Classifier

The classifier used for analysis is based on MLP feedforward ANN. Example can be found in Fig. 2. It is the network with five inputs, two classification outputs, and two hidden layers. It is the only example of network, real networks differ on each server because there were another number of inputs. Backpropagation was used to train network.

4) Training

We selected three servers which were most active during the analysis period. Two of them has another kind of utilization based on external events. Backup was selected because it was used in regular intervals and its run affected lot of other servers.

- IMAP - Mail server
- Horde - Webmail server
- Backup - Backup server

To classify or predict data we had to choose right time intervals to analyze. We took data being collected for seven days and divided into time intervals for 5 minutes, 30 minutes and 60 minutes. So we were able to classify/predict data according to these intervals.

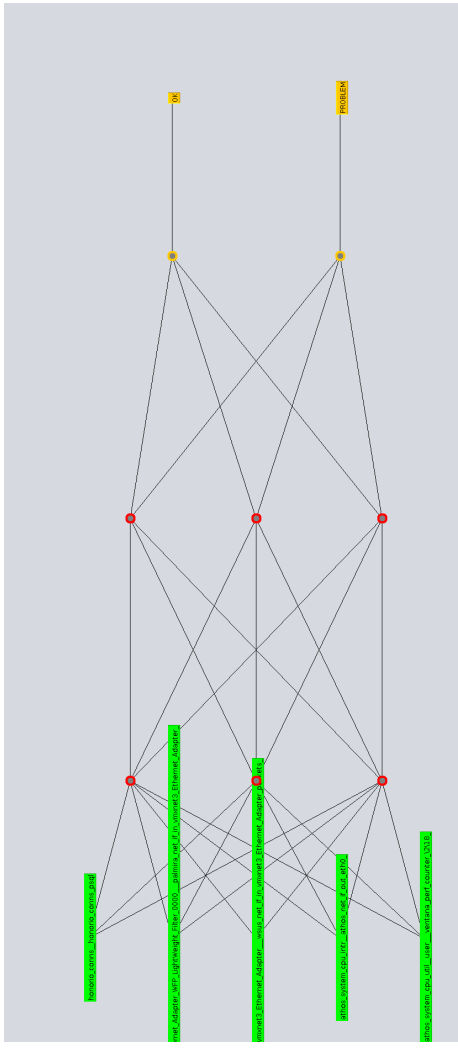

Fig. 2: MLP structure.

- 5 minutes - 2016 rows
- 30 minutes - 336 rows
- 60 minutes - 168 rows

Input data was divided in the ratio 70/30 (70% of train set to the 30% of verification set). Training was the first step and verification second one. Next, we used test set from another time window.

During verification, we defined a rule which made false positives more interesting. It is due to the fact that it is good for the administrator to know about the potential incident and recheck the status of the system. So success rate of prediction was the ratio between detected plus not-detected problems and not-detected problems in the same time window.

5) Results

IMAP server did not show bigger dependencies between its items and there were no other data to use for prediction. If monitoring system contains more pieces of information (like the number of sent emails, discarded emails, logins etc.), the analysis could be much better as is depicted in Tab. 2. It is shown that none of the networks could be trained and verified with acceptable precision. This is due to external influences which came from random access to email server from many users. Server backup had better success rate than IMAP. The classifying problem with a high success rate for 30 minute intervals was possible. In 5 minute intervals, the success rate was lower, see Tab 3. Webmail server showed best results. It was possible to predict problem within 30 minute periods with success rate as high as 93%. as is depicted in Tab. 4.

T	W	HL	U
300	same	1	X
300	same	2	X
300	same	3	X
300	next	1	X
300	next	2	X
300	next	3	X
1800	same	1	X
1800	same	2	X
1800	same	3	X
1800	next	1	X
1800	next	2	X
1800	next	3	X
3600	same	1	50
3600	same	2	X
3600	same	3	X
3600	next	1	X
3600	next	2	X
3600	next	3	X

T Window length in seconds
W same - classification within window, next - prediction
HL Number of hidden layers
U Success rate (X=verification unsuccessful)

Tab. 2: Result of IMAP server

T	W	HL	U
300	same	1	70
300	same	2	70
300	same	3	X
300	next	1	70
300	next	2	70
300	next	3	X
1800	same	1	77
1800	same	2	100
1800	same	3	X
1800	next	1	X
1800	next	2	X
1800	next	3	X
3600	same	1	75
3600	same	2	75
3600	same	3	X
3600	next	1	X
3600	next	2	X
3600	next	3	X

Tab. 3: Result of Backup server

T	W	HL	U
300	same	1	80
300	same	2	80
300	same	3	X
300	next	1	30
300	next	2	X
300	next	3	X
1800	same	1	93
1800	same	2	93
1800	same	3	X
1800	next	1	X
1800	next	2	X
1800	next	3	X
3600	same	1	91
3600	same	2	91
3600	same	3	X
3600	next	1	X
3600	next	2	X
3600	next	3	X

T Window length in seconds
W same - classification within window, next - prediction
HL Number of hidden layers
U Success rate (X=verification unsuccessful)

Tab. 4: Result of Webmail server

5. Conclusions

Interesting results were found during analysis. A new approach to identify network incidents was invented. We created software Monda which is Open-Source and it can be used by anybody to following analysis in Zabbix. Verification of methods was done on Silesian University data stored in the monitoring database. Data in monitoring system are interesting for next analysis. Even if it is relatively complex to choose right data and right intervals, data are suitable for prediction of some incidents. Monda can do preprocessing part very quickly and effective way directly within SQL server. Anybody can write its own analysis module to focus on specific incident or time. Algorithms used here are mainly based on logical assumptions which are derived from knowledge of monitoring system and its data.

Next assumption is that to do better analysis and prediction of incidents, the monitoring system must have more inputs about incidents on the network. In other words, more inputs related to security and statistics of systems, better analysis and prediction of incidents. We verified that we can achieve good results using MLP networks. The prediction could be even better if we save fingerprints of hosts. This means, to save vital statistical, correlation data and trends per each host and time interval. Deviation of this data could be used to better prediction. Next step is to interconnect Zabbix with logging server. It is theoretically possible to write a new module for Monda to do so.

5.1. Future improvements

First place for optimization is pre-processing of data. More information about stored data and their source mean better pre-processing of data. One of the improvements could be a manual description of Items inside Zabbix so preprocessor could know right ranges for given Items. Next, it would be nice if Zabbix could do data approximation on historical data. Zabbix deletes data from history after configured amount of days and computes Trends from it. So we can see minimum, maximum and average in hour intervals.

If Zabbix uses approximation function, it is possible to describe data at summarized intervals better. It is possible to use SOM in future for better fingerprinting of Hosts. But it needs more investigations and more data of separated Zabbix servers to do so. Some processes on the network are under resolving power of monitoring system. To be able to catch, analyze or predict them, it is needed to feed them either asynchronously to Zabbix or use smaller time intervals to fetch data.

Acknowledgment

The research received a financial support from the SGS grant No. SP2017/174, VSB - Technical University of Ostrava, Czech republic. Authors would like to thank to Silesian University for Zabbix server data availability.

References

[1] FAZIO, P., M. TROPEA. A New Markovian Prediction Scheme for Resource Reservation in Wireless Networks With Mobile Hosts. *Advances in Electrical and Electronic Engineering*. 2012, vol. 10, iss. 4, pp. 204–210.

[2] FAZIO, SP., M. TROPEA and S. MARANO. A distributed hand-over management and pattern prediction algorithm for wireless networks With mobile hosts. In: *Proc. 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*. Sardinia, 2013, pp. 294–298.

[3] DE RANGO, F., M. TROPEA, A. PROVATO, A. F. SANTAMARIA, S. MARANO. Multi-Constraints Routing Algorithm Based on Swarm Intelligence over High Altitude Platforms. *Studies in Computational Intelligence*. 2007, vol. 129, pp. 409–418.

[4] DE RANGO, F., M. TROPEA, A. PROVATO, A. F. SANTAMARIA, S. MARANO. Minimum Hop Count and Load Balancing Metrics Based on Ant Behavior over HAP Mesh. In: *Proc. IEEE GLOBECOM 2008*. New Orleans, 2008, pp. 1–6.

[5] Open-Source tool ZABBIX: the network monitoring SW. Available at: `http://www.zabbix.com/`.

[6] Open-Source tool MONDA: data analyzing in monitoring system Zabbix. Available at: `https://github.com/limosek/monda/`.

[7] SINGH, N., A. JAIN, R.S. RAW, R. RAMAN. Detection of Web-Based Attacks by Analyzing Web Server Log Files. In: *Networking, and Informatics. Advances in Intelligent Systems and Computing*. Springer, 2014, vol. 243.

[8] CELEDA, P., M. KOVACIK, T. KONICEK, et al. *FlowMon Probe*. Networking Studies, 2006.

[9] SAFARIK, J., M. VOZNAK, F. REZAC, L. MACURA. IP telephony server emulation for monitoring and analysis of malicious activity in VOIP network. *Komunikacie*. 2013, vol. 15, iss. 2A, pp. 191–196.

[10] SAFARIK, J., P. PARTILA, F. REZAC, L. MACURA, M. VOZNAK. Automatic classification of attacks on IP telephony. *Advances in Electrical and Electronic Engineering*. 2013, vol. 11, iss. 6, pp. 481–486.

[11] SAFARIK, J., M. VOZNAK, F. REZAC, L. MACURA. Malicious traffic monitoring and its evaluation in VoIP infrastructure. In: *Proc. 35th Int. Conference on Telecommunications and Signal Processing*. TSP, 2012, iss 6256294, pp. 259–262.

[12]] DAVID, N., N. RESHEF, A. YAKIR A. et al. Detecting Novel Associations in Large Data Sets. *Science*. 2011, vol. 334, iss. 6062, pp. 1518–1524.

About Authors

Lukas MACURA is with the Institute of Informatics, Silesian university in Opava, the

Czech Republic. He graduated from the Faculty of Electrical Engineering and Computer Science, VSB-TU Ostrava and delivered his PhD. thesis in field of network security. His professional interests focus on computer networks, their security issues and network monitoring systems.

Miroslav VOZNAK obtained his PhD. in Telecommunications from the Faculty of Electrical Engineering and Computer Science in 2002. He is an IEEE Senior member, actively engaged in numerous IEEE conference committees and has served as a member of the editorial board for several journals. His research interests focus generally on information and communications technology, particularly on quality of service and experience, network security, wireless networks and in the last couple years also on Big Data analytics in mobile cellular networks.