

ANALYSIS AND COLLECTION DATA FROM IP NETWORK

Martin HASIN*, Martin CHOVANEC**, Jakub PALŠA*** Martin HAVRILLA****

*Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics,
Technical University of Košice, Letná 9, 042 00 Košice, Slovak Republic, Tel. +421 55 602 7648, E-mail: martin.hasin@tuke.sk

**Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics,
Technical University of Košice, Letná 9, 042 00 Košice, Slovak Republic, Tel. +421 55 602 7606, E-mail: martin.chovanec@tuke.sk

***Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics,
Technical University of Košice, Letná 9, 042 00 Košice, Slovak Republic, E-mail: jakub.palsa@tuke.sk

****Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics,
Technical University of Košice, Letná 9, 042 00 Košice, Slovak Republic, E-mail: martin.havrilla@tuke.sk

ABSTRACT

The rapid deployment of technologies that can share data brings, in addition to the positive aspects, also technologies that an attacker can use to misuse personal data. Vulnerabilities in the network can be divided according to the type of attack into horizontal attacks from the perspective of the attacker and vertical attacks from the perspective of the victim. An important goal of organizations is to successfully defend against such an attack. The search for attacks on the network infrastructure can be ensured by implementing machine learning that can capture the current attack. This work describes methods of searching for attacks using graphical representation of data and also using machine learning of the high count type at different sizes of the investigated segments.

Keywords: Cybersecurity, nfstream, database, ndpi, anomaly detection, machine learning

1. INTRODUCTION

One of the most common forms of cyber security, which is also perceived by ordinary computer users, is the use of anti-virus protection of personal computers and mobile devices. This area is only a small part of the security that needs to be met to achieve a minimum level of equipment protection. Achieving this level of protection is insufficient for corporate networks, and cyber security needs to be addressed on the network side as well. This network traffic needs to be monitored in real time and potential vulnerabilities evaluated.

When analyzing network traffic, you can select different methods by which network traffic can be analyzed. It is possible to implement various network analyzers that can thoroughly analyze network traffic at the L7 layer and categorize them according to predefined patterns. These applications include various IPS and IDS filters [1]. Another method is to analyze network traffic using information about given flows. This information provides a picture of the network, where it is possible to detect who is communicating in the network and by which devices the communication takes place. Based on this information, it is possible to look for anomalies in emerging connections in this network operation. Every single device on the network corresponds to certain patterns of behavior that can be recorded and analyzed in detail. Using these models, it is possible to define possible deviations from the assumed behavior model of the respective system. In practice, this means that if the system responds only to HTTP requests, responses to SSH requests from unknown networks may indicate a possible disruption of the system. These systems operate at two levels, the network level and the host level [2].

2. CYBERSECURITY IN NETWORKS

In order to ensure the security of the network infrastructure against intentional attacks and potential threats, co-operation is established with the private sector, academia

and government institutions. Within this connection, various cyber protection systems are being designed to ensure the credibility, integrity and availability of information systems. In Fig. 1 shows a conventional protection system that addresses various cyber vulnerabilities such as viruses, trojans, spams, botnets.

Network traffic is controlled by a network firewall, which contains additional layers such as spam filter, anti-virus. The network may also include a network intrusion detection system that detects possible vulnerabilities. The second part of cyber protection is host protection, which consists in the implementation of a local firewall, anti-virus or system to detect possible vulnerabilities in client devices. The conventional approach is solved by mechanisms designed in the firewall, which protects network clients and servers based on prescribed algorithms and rules. Protection consists of preventing viruses and other malicious applications from entering the network using simple rules. The antivirus system is designed to reduce the risk of theft of sensitive data in the operating system. These applications are installed on client computers or other critical infrastructure in the network, where it is necessary to ensure data protection on the client side [3].

Cyber security systems can be divided into several levels. The first level is to determine the input data most commonly used by the Libpcap library for Linux, Solaris BSM for SUN systems, and Winpcap for Windows operating systems. These libraries allow you to capture forensic data from current network traffic [4]. We can categorize the captured data as data from the network part and data from the host part. The data from the host part represents a set of commands executed on the end system by the user. These events include details such as access to information systems, sending e-mail messages, launching any software. Events that are categorized from the network part contain information about the used IP protocol. The subsequent process, after appropriate processing, enables the acquisition of other useful parameters from a given network flow.

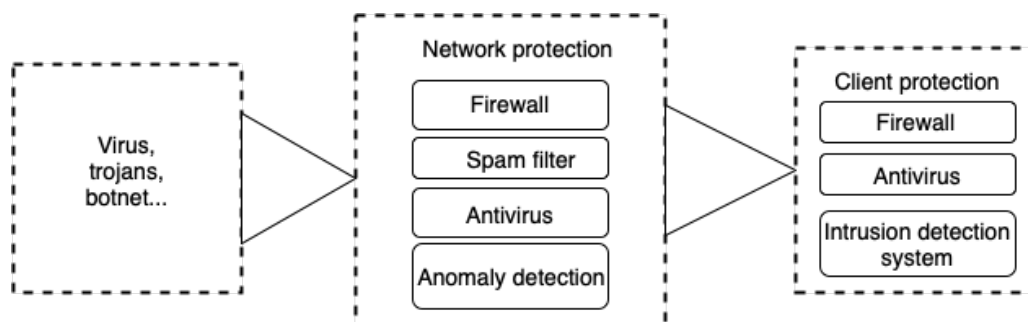


Fig. 1 Conventional system of cyber protection

Most often it is about finding the source and destination IP address, finding the communication protocol, counting the bytes transferred in the network. Various network intrusion detection methods can be implemented in the "analysis core" module. By examining the data with these algorithms, it is possible to detect anomalous network traffic that may signal the beginning of a cyber attack. As shown in Fig. 1 the core analysis module is the main part of the algorithm that forms the process of creating the adaptive capabilities of the cyber protection system. The cyber security solution must include a proactive and reactive side of network security.

Proactive approaches in the cyber system allow for rapid prediction and removal of vulnerable modules. For the correct functionality of these modules, user authentication using a password or biometrics is required. Following the login, the system restricts the users to the level of their assigned rights. This method of protection makes it possible to secure the system against the acquisition of unauthorized information, so it can be called data mining. Predictive approaches in systems are used as the first security line, where access levels to selected systems are defined. However, this model may already contain vulnerabilities in the selected applications, where it is possible to ensure special escalation of rights by special calls and then read data that is not accessible to regular users. Limiting the impact of poor software security is possible by using a firewall, which provides an additional layer of security.

The second line of security is reactive protection, which consists of reactive security solutions such as intrusion detection systems (IDS). These systems detect intrusions based on information about the protocols used and their characteristics of the monitored network. Using these techniques, it is possible to determine the extent of damage caused by an attacker, track down hackers or creators of the violation. This data can be used to prevent similar attacks in the future [5].

3. PACKET RECORDING USING NFSTREAM

The implementation of the NetFlow protocol, which is used to analyze network traffic, is possible using several options.

- Implementation of NetFlow protocol on central router devices.
- Implement NetFlow protocol on dedicated devices that perform only the probe task.

Both of the listed implementation options have their advantages. Using the NetFlow implementation option on central Router devices saves the cost of running additional network devices. The entire process of collecting and analyzing network traffic is performed by the Router, which then only sends the data of the required shape for processing by the Flow Collector. However, this option has limited capabilities for implementing the NetFlow protocol. By using an external dedicated probe, it brings the possibility of implementing additional methods for the analysis of incoming packets. The disadvantage of this connection is the need to use one or more devices involved in the infrastructure to collect the required data. This makes this connection model more expensive than using the built-in NetFlow protocol in the central Router device.

To implement the NetFlow protocol, it is possible to use the NFstream solution, which combines the NetFlow protocol and additional packet analysis using the nDPI library. This also outputs the NetFlow output to the additional network traffic information provided by the nDPI library. The nDPI library is an instruction set that can identify the relevant device, the type of applications used and other parameters based on the incoming packets [6].

The nDPI library is developed by Ntop (2013) by modifying the open project OpenDPI. This project stood out as a library for implementation in Ipoque's Firewall solutions. When using the nDPI library, the packet strings in the measured network are compared with the samples it contains in the nDPI database. Advantages of nDPI:

- Supports detection of more than 170 current protocols.
- The SSL certificate decoding option allows encrypted connections to be detected.
- Ability to support other protocols using string correlation-based processes.

3.1. Experimental environment

The experimental environment consisted of creating a network probe that can use cisco SPAN technologies to analyze network traffic to the WEB server without disrupting the current network traffic to the server. Part of the experiment was the creation of a simulated attack on the WEB server, where the parameters of the attack were most similar to the real attack on the web server. The attack consisted of creating a network of attackers who, at various intervals,

opened connections to the WEB server or opened predefined websites. By increasing the number of connections, the functionality of the web server was prevented in extreme situations. Using a network probe, these data were measured, analyzed and the measurement results recorded in a non-relational database for additional analysis. The selected network probe has four separate processors, each with 12 cores. This system provides performance for additional analysis and implementation of the ML model. The UBUNTU operating system was chosen as the software, which supports the setting of parameters that can optimize the performance of the network card to its maximum [7]. Technologies required for network probe optimization set on the created probe:

- DMA (Direct Memory Access)
- RSS (Receive Side Scaling)
- RPS (Receive Packet Steering)
- RFS (Receive Flow Steering)
- GRO (Generic Receive Offload)
- LRO (Large Receive Offload)

By implementing these settings, it is possible to prevent unwanted dropping of packets received by the operating system. By implementing RSS load balancing technology, it is possible to ensure fast redistribution to processor cores when using multiple core or multi-processor systems, and thus ensure high packet opening speed [8].

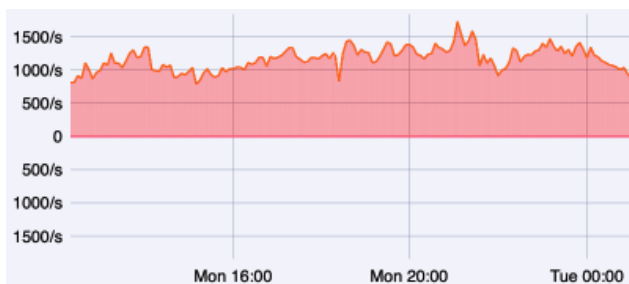


Fig. 2 Dropping network packets by the core of the operating system

In Fig. 2 shows the process of dropping network packets on network adapters when running measurements before the operating system optimization is implemented. Such packet dropping is caused by the functionality of the operating system kernel, which drops packets that are incorrectly shaped or corrupted. The discarding of network packets is also affected by the performance of the hardware with which we open the received packets.

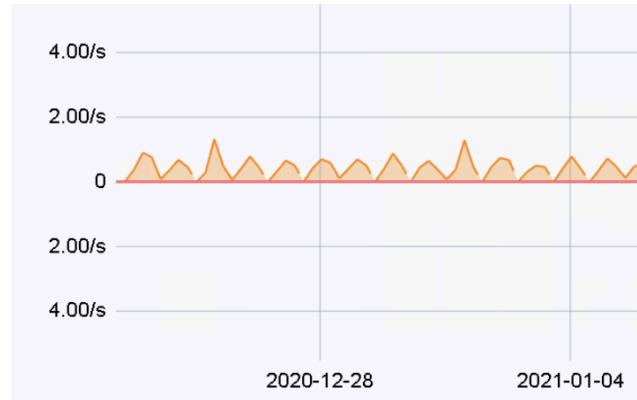


Fig. 3 Dropping network packets after optimization

In Fig. 3 shows the process of dropping network packets on the network adapter when running the measurement after the implemented optimization of the operating system and the network adapter. By optimizing the nfstream and operating system settings, packet dropping was reduced to less than 0.03 % for a given measurement. The number of packets in this measurement ranged from 200,000 to 500,000 packets per second.

In Fig. 4 is a graph of packet dropping after starting nfstream measurement. The following parameters were set during testing:

- idle timeout = 120 second
- active timeout = 300 second

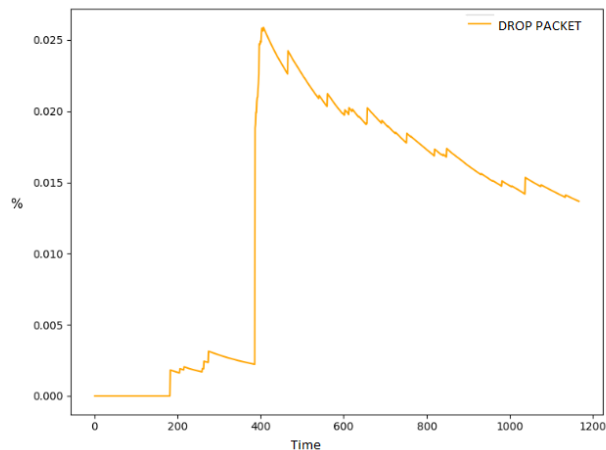


Fig. 4 Packet drop after start nfstream

The rapid increase in dropped packets is caused by the first launch of the application and the filling of the cache memory, which is regularly released over time. An algorithm for distributing performance across multiple processor cores has also been implemented in the system.

4. ANOMALY DETECTION

The nfstream application allows you to implement the created model of artificial intelligence into the analysis process. This model can classify data according to certain values and then mark them with the appropriate tag in the non-relational database [9]. In Fig. 5 the model of learning

the ML algorithm is shown directly from the data that were the output of the analysis of the given model, which means that artificial intelligence is already training on its results, into which data from external databases of already known threats enter [10].

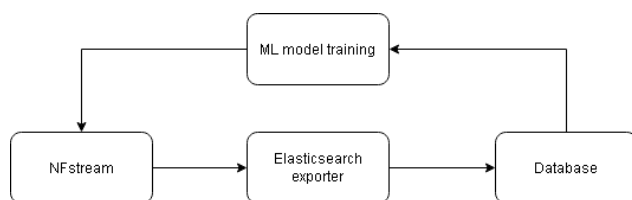


Fig. 5 Machine learning model

When analyzing data, it is possible to use machine learning methods as well as statistical methods that can describe network operation. Data classification allows the administrator to detect possible vulnerabilities that occur in a given system when communicating with other systems.

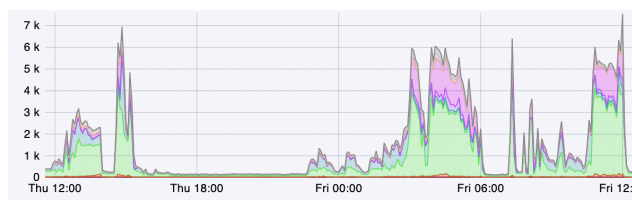


Fig. 6 Sample of attack to web services

In Fig. 6 the deviation of the system behavior by a sudden increase in the number of network packets is shown. This increase in communication represents an attack on a DDoS WEB service. By measuring the data using the proposed data collection model, it is possible to use statistical methods to determine the network deviation and thus the ongoing attack. By analyzing the attack on a given web server, it is possible to measure and record the increase in network traffic to the incoming WEB server. Using graphical methods, it is possible to analyze this data and predict the occurrence of an anomaly. The measurement on the network side is therefore independent of the information from the WEB server itself, which may not record all its activity due to errors. As a result of these findings, a network analysis model using the non-relational elasticsearch database and the use of machine learning functionality to predict network anomalies were chosen. In Fig. 7 the ongoing attack in the current measurement using the proposed nfstream system is shown. The figure describes the number of created records that represent the increased data traffic on the measured device. This increased data traffic represents a potential attack on the device.

Experiments in graphical data analysis consisted of creating a data pattern with an attack resembling a real network of attackers and networks of simulated clients. The measurement and thus also the data contain a record of network activity on the measured server, where they contain attack times and also a simulation of legitimate network traffic. The measurement took place over 10 days, where 5,056,980,569 documents, which were 45 GB in size, were collected in the database. Graphical analysis methods using non-relational database algorithms were tested on this

sample. The goal is to create and verify algorithms for anomaly detection that can be implemented in the data analysis process. The anomaly search process can be automated and several machine learning options can be created to find anomalies in the measured operation. To detect anomalies, it is possible to use machine learning focused on:

- One metric with a high count function with a segment length of 15 minutes
- One metric with a high count function with a segment length of 5 minutes

The experiment consisted in the use of the high count function, where a second experiment took place with different segment lengths. A representation of the sample length of 5 minutes and 15 minutes was chosen among the credible ones. In Fig. 8 shows an experiment of finding anomalies in network traffic data with a DDoS attack lasting 15 minutes. The size of the segment that was analyzed by the high count task is also set to 15 minutes. The graph shows the course of detection of network attacks, where it is possible to see the probable ongoing attack. Due to the effect of the higher length of the attendance segments, the sharp increase in the average value of the metrics leads to a reduced possibility of anomaly detection. Despite the well-known ongoing DDoS attack, it is not possible to accurately detect an anomaly in the measured network using this task, because there are rapid changes in the increase in the number of records, which affects the used machine learning algorithm.

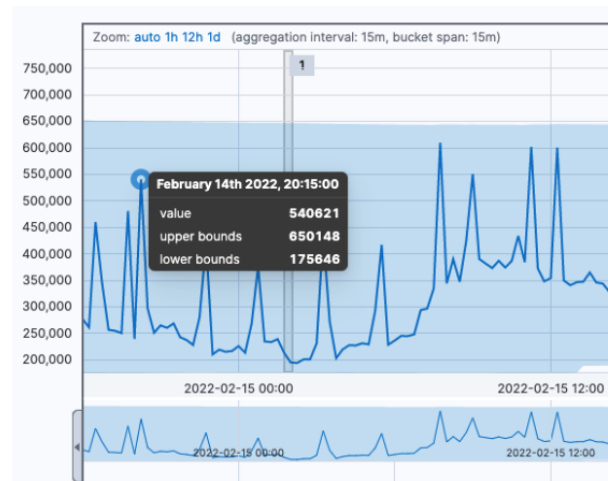


Fig. 8 Anomaly detection experiment 1

In Fig. 9 shows an experiment of finding anomalies in data using two machine learning tasks. The first job has an analyzed segment size of 5 minutes and the second job has an analyzed segment size of 15 minutes. By comparing both tasks in the figure, it can be seen that the segment size set to 5 minutes can detect even shorter and smaller anomalies in the network. Subsequently, these suspicious behaviors can be represented using points that determine the score of the given suspicious activity.

In Fig. 9 these points are represented by orange, yellow, blue and red points. Based on this predicted score, it is possible to detect the anomaly represented by the color of the given point. The redder the color, the higher the score,

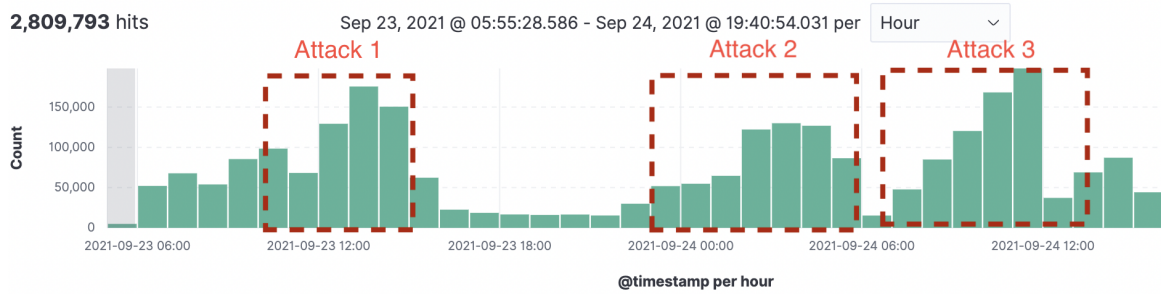


Fig. 7 Attact to web services

which determines the higher probability of the anomaly. Based on the results of this early one, we can conclude that the use of the second model makes it possible to detect even less significant anomalies, that is, attacks in the network.

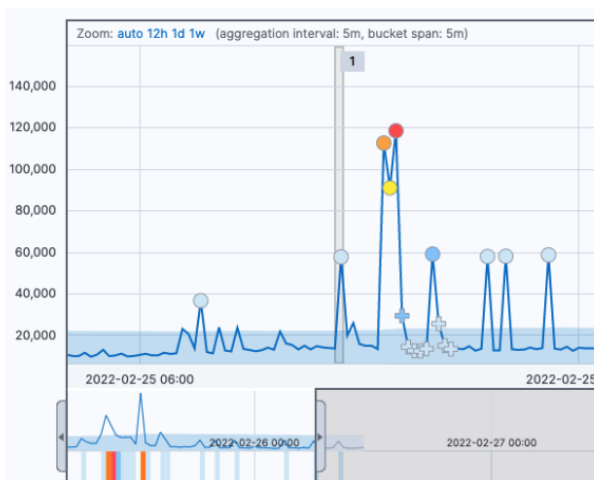


Fig. 9 Anomaly detection experiment 2

The first model in this experiment was able to detect only two larger attacks, while the other smaller ones were not detected as an anomaly.

5. RESULTS

The result of the work is the possibility of implanting an analysis with a suitable machine learning model that can reliably detect possible penetrations. When detecting these intrusions, a group of packets is analyzed at the same time, which, however, in real networks may mean the possibility of losing one or more packets in a given measurement. Such packet loss affects the accuracy of analyzes and the evaluation of the state of a given network traffic. The aim is to experimentally verify the model with the smallest possible share of falsely detected threats in the measured network, also the aim of the work is to confirm the functionality of the model in the experimental environment and compare it with real network traffic. In order to prevent packet loss, optimization methods of the operating system kernel were designed and implemented, as well as technologies necessary for network card optimization. After applying these

optimization processes, a reduction of the packet loss rate from values exceeding 1000 packets per second to a loss rate reaching less than 2 packets per second was achieved. In order to reduce the resulting loss rate, it is possible to implement special hardware that, using ISIK modules, can quickly and efficiently work with packets and then transfer them to the RAM memory. By reducing loss, network threats can be more accurately detected. In the created experiments, two methods of machine learning, which can be used to search for anomalies, were measured. A machine learning method based on the detection of an increased number of high count records where it is possible to change the sample size that the model uses in calculations. The measurement consisted of two anomaly detection experiments using a segment length metric of 5 and 15 minutes. By measuring, it was found that by setting the sample size to 5 minutes, it was able to detect a greater number of anomalies than it was when the sample size was set to 15 minutes. Using this measurement, it was also possible to detect attacks that were not more extensive and lasted shorter. By using a segment size of 15 minutes, the average value of the metric increased as weaker attacks were represented in the model as legitimate network traffic. Setting the sample size low to 1 minute resulted in an increased number of falsely detected attacks due to the rapidly changing network infrastructure. With such a low setting of the segment size, legitimate network activity is detected, which consists in opening several pages and clicking through this page, as a possible anomaly. The optimal model setting for this measurement should be chosen for 5 minutes, where it is possible to detect with high accuracy an ongoing attack on the web service.

ACKNOWLEDGEMENT

This publication has been published with the support of the Operational Program Integrated Infrastructure within project: Research in the SANET Network and Possibilities of Its Further Use and Development (ITMS code: 313011W988), co-financed by the ERDF. Development and innovation of TUKE simulation environment in the field of technical sciences (004TUKE-2-1 / 2021)

REFERENCES

- [1] PAPADAKI, M. – FURNELL, S.: “IDS or IPS: What Is Best?” *Network Security*, grade. 2004: *doi.org*, 7 7, No. 2004 (15–19) [https://doi.org/10.1016/S1353-4858\(04\)00106-0](https://doi.org/10.1016/S1353-4858(04)00106-0)
- [2] GHAFAR A. JAAFAR et al. “Review of Recent Detection Methods for HTTP DDoS Attack”. *Journal of Computer Networks and Communications*, grade. 2019: *doi.org*, 18 1, No. 2019 (1–10) <https://doi.org/10.1155/2019/1283472>
- [3] GALINEC, D. et al. “Cybersecurity and Cyber Defence: National Level Strategic Approach”. *Automatika*, grade. 58, č. 3 (Crossref): *doi.org*, 18 7, No. 2017 (273–286) <https://doi.org/10.1080/00051144.2017.1407022>
- [4] VON SOLMS, R. – VAN NIEKERK, J.: “From Information Security to Cyber Security”. *Computers Security*, grade. 38: , 18 10, No. 2013 (97–102) <https://doi.org/10.1016/j.cose.2013.04.004>
- [5] KILINCER, I.F. et al. “Machine Learning Methods for Cyber Security Intrusion Detection: Datasets and Comparative Study”. *Computer Networks*, grade. 188: , 18 4, No. 2021 (107840) <https://doi.org/10.1016/j.comnet.2021.107840>
- [6] AOUMINI, Z.– PEKAR, A.: “NFStream”. *Computer Networks*, grade. 204: , 18 2, No. 2022 (108719) <https://doi.org/10.1016/j.comnet.2021.108719>
- [7] PEKÁR, A. et al. “Adaptive Aggregation of Flow Records”. *Computing and Informatics*, grade. 37: , 1 1, No. 2018 (142–164) https://doi.org/10.4149/cai_2018_1_142
- [8] GUAN, H. et al. “Performance Enhancement for Network I/O Virtualization with Efficient Interrupt Coalescing and Virtual Receive-Side Scaling”. *IEEE Transactions on Parallel and Distributed Systems*, grade. 24: , 6 6, No. 2013 (1118–11128) <https://doi.org/10.1109/TPDS.2012.339>
- [9] GYORÖDI, C. et al. “A Comparative Study of Relational and Non-Relational Database Models in a Web- Based Application”. *International Journal of Advanced Computer Science and Applications*, grade. 6: , 18 11, No. 2015 (78–83) <https://doi.org/10.14569/IJACSA.2015.061111>
- [10] VAJDA, D. et al. “Towards Machine Learning-based Anomaly Detection on Time-Series Data”. *Infocommunications journal*, grade. 13: , 18 1, No. 2021 (35–44) <https://doi.org/10.36244/ICJ.2021.1.5>

Received April 29, 2022, accepted August 20, 2022

BIOGRAPHIES

Martin Hasin is a PhD student at the Department of Computers and Informatics at Faculty of Electrical Engineering and Informatics, Technical University of Košice since 2020. His research interests data analysis and artificial intelligence for cybersecurity.

Martin Chovanec received his Engineering degree in Informatics in 2005 from the Faculty of Electrical Engineering and Informatics, Technical University of Košice. In 2008 he received his Ph.D. degree at the Department of Computers and Informatics of the Faculty of Electrical Engineering and Informatics of the Technical University of Košice and his scientific research was focused on network security and encryption algorithms. Currently, he is Director of the Institute of Computer Technology of the Technical University of Košice.

Jakub Palša is a PhD student at the Department of Computers and Informatics at Faculty of Electrical Engineering and Informatics, Technical University of Košice since 2019. His research interests on detection of malware samples using machine learning algorithms.

Martin Havrilla has been a PhD student at the Department of Computers and Informatics at the Faculty of Electrical Engineering and Informatics of the Technical University in Košice since 2020. His area of research is focused on autonomous mechanisms serving to change the configuration of network devices based on the nature of computer networks.