INTRUSION DETECTION SYSTEMS IN A HETEROGENOUS NETWORK ENVIRONMENT

Martin ŠTANCEL, Martin CHOVANEC, Eva CHOVANCOVÁ
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 4220,
E-mails: martin.stancel@tuke.sk, martin.chovanec@tuke.sk, eva.chovancova@tuke.sk

ABSTRACT

This paper deals with a design and implementation of an intrusion detection system in heterogeneous network environment. The work includes work out characteristics of Intrusion detection systems, it`s base classification, development of problematic, building structure, used strategies of analyses in intrusion detectio and studies of placement the detectors of system. The scope of this work is to design and to implement the Intrusion detection system with uses the anomaly detection analyses, concrete statistical methods, in target of the third layer of ISO/OSI model. In implementation of the work is using the knowledge from area of the programming in C language and the modular programming, and the knowledge from area of the computer networks mainly from the building a networks traffic up to third layer of ISO/OSI model. The work includes the classification of implemented system and the experimental verification of the system functionality on the real specimen of network traffic from different network environments.

Keywords: *computer security, intrusion detection system, networking*

1. INTRODUCTION

Current research in the field of information systems and technology has resulted in the development of diverse applications in all areas of business and human life. Data as an object of information has become a critical property of organizations and therefore effective access, sharing, evaluation and use of this data has become a very important part of every organization. Therefore, the directions of efficient data storage, access to and protection of distributed servers have become important research points.

Whatever the method of data acquisition, a secure system must protect the data from deterioration or misuse. But is there a sufficiently secure way to protect data? With username and password authentication, which is often the only way to protect data, there are ways to bypass this protection, either by worms or other intrusion techniques that are already quite advanced. However, the goal of the attack is not only to gain access to the data, but to gain access to it even more, without the data owners knowing that it has been compromised. Therefore, the intruder uses techniques that are relatively difficult to track, and rarely leaves any traces behind. Much more often we encounter masked attacks where their activity on a given system cannot be easily detected.

The security situation is exacerbated mainly by viruses and worms that do not need human surveillance for their destructive activity and are capable of replicating and spreading independently in systems where they have been released. This is all the more worse by the fact that, over time, the place of initial infection, as well as its further spread, cannot be determined, which provides sufficient protection for the agent. Another threat is a Trojan horse, which has some destructive activity defined in its code.

Several computer systems rely on access control mechanisms as one of the main means of protection. This model limits access to system objects, but on the other hand does not prevent what an entity can do with the object to which it has access. This model does not prevent an unauthorized flow of information on the system because the flow may be related to authorized access.

The security of software applications is also closely related to their life cycle, which is getting shorter. These are the problems of creating flawless software, where errors in the software manifest themselves as security deficiencies and along with the close linking of the program life cycle, the program becomes insecure for some time.

Intrusion detection systems [1] control the activity of a computer system or the activity of a protected system environment, and when failures are detected, they can create predefined actions to protect critical data. Their system protection options are closely related to internal environment protection, where, if the authentication module is compromised, or captured, data can compromise without the system detecting an intrusion.

This paper describes the creation of an IDS for enhancing the security of a monitored system without limiting its performance. However, what should be done at the current state, when the amount of data transmitted in the network is constantly increasing? The performance of each security system depends directly on the amount of resources used in the system and the security of the system and its accessibility decreases in proportion to the amount of system resources that the security system needs for its operation. To improve performance, it is necessary to limit the resources used during the execution of the detection, which can be easily implemented as a prioritization of the system, to define which information must necessarily be subjected to in-depth analysis. The created IDS will be the source of this information on the need to conduct an in-depth analysis.

2. GENERAL CLASSIFICATION OF INTRUSION DETECTION SYSTEMS

IDS can be classified according to several criteria. Significant classifications of IDS systems include classification based on intrusion detection system response, where IDS is classified as passive and active [2]. This classification is important for the general user in terms of the necessary knowledge in securing the system, because in the selection of the passive system it is assumed to have a higher knowledge in the field of security, compared to the active IDS, where the need for knowledge of system security is not so important.

Passive systems are those whose sensors detect intrusion attempts, but their response is passive in the sense that the system does not attempt to respond to the attacker, or to remove or minimize the damage done by the attack. Their purpose is to create an intrusion report or make an attempt to intrude a system.

In contrast, active systems directly intervene in detecting an intrusion against an attacker in an attempt to minimize the damage done by the intrusion. Based on the way we respond to the attacker, we can divide them into:

- those that apply control through an attack channel, for example by modifying the current state of the shutdown system or mitigating the effects of an attack,
- those that exercise control over an attacking system, e.g. turn an attack against an attacker and attempt to shut down the attacking system.

In current systems, one channel from network connections responds to suspicious attacks and one blocks suspicious system calls, terminating processes when this option fails. This method of defense is generally complex, and therefore opens the system with the usual attack, such as denial of service (DOS, DDOS).

Another important criterion for IDS distribution is frequency analysis, where IDSs are divided into real-time IDSs and IDSs that run periodically. IDS that run periodically do not check the current state of the system, but is the check is triggered only once at a predefined time interval where their task is to determine whether or not the protected system has been compromised. The IDS system creates a log file containing all the necessary information to perform the detection throughout the entire monitoring period. Real-time analysis, on the other hand, tries to check the current state of the system and detects whether or not the system is under attack.

Requirements for a properly functioning general IDS.

- Performance (in this concept, performance will be considered to be able to receive and track data flows on the network without visibly restricting the performance of the entire computer system on which the IDS is deployed).
- Accuracy (along with the ability to work in extreme conditions without the system reducing the effectiveness of network status monitoring or intrusion detection in the monitored network).
- Portability (the ability of IDS to operate in a heterogeneous network environment).
- Completeness (in the sense that the IDS must capture all attacks that have occurred on the monitored system).

3. IDS DESIGN AND IMPLEMENTATION

The system design follows the general structure of the IDS system [3]. By dividing the system into modules that would meet the structural and functional requirements, the modules were divided into the following categories:

- *auxiliary modules* - for correct work of the IDS,
- *event generator* - a module that serves as a network monitoring module and at the same time an IDS management module. Its priority tasks are to manage IDS according to the command line settings, perform network monitoring, and generate a GIDO object for further processing,
- *database modules* - are modules mediating contact with the file system. They are categorized by existing IDS work file types,
- *analysis module* - a module for performing network intrusion detection, which uses information obtained from the *database module* and the *event generator* module during its detection. Its output is either a correct network state or a state in which a network violation has been detected,
- *countermeasures module* - IDS is ready for additional activation when IntrusionDetected interface is implemented. The interface is intrinsically defining a set of conditions to deal with a state of disruption so that the attack does not cause any or minimal damage to the system resources.

3.1. Profilation

It is possible to ensure the accuracy of the IDS by using profilation [4], that is, by creating a regular legitimate network user profile during the reporting period. System utilization varies during monitoring, users do not access resources equally throughout the process, but there are periods of increased / decreased user activity in the system. Constructing a general profile that reflects the average system utilization over the reporting period does not meet the defined efficiency requirement because, at the time of increased activity, the IDS would report more error violations, and vice versa, such system would not be able to detect system violations. The consequence of ensuring accuracy is to ensure the profilation requirement by using sampling. In this program, a sample of 10 minutes was used during which the user load of the system is considered to be even and the same profile can be used to evaluate the activity in the system. The profile created would be used as an input to detect network intrusion at the same time rounded to tens of minutes during the following days when the IDS system will perform after the initial profilation phase.

3.1.1. Conditions for creating a profile

By defining the conditions for creating a profile the basic properties that the program and network must meet during the profilation process are defined. The basic condition which is as same as for all existing IDSs based on detection of anomalous behavior is to keep the monitored network in the profilation process in a state during which the network is not disrupted. The automatically resulting condition for the program is determined as the ability of the profile to absorb the data obtained from the network so that when the intrusion detection is restarted it will not detect the intrusion over the same data source.

For the program to be able to work, it is necessary to define the monitored variables (comparative conditions) that will perform in the process of the profile creation and at the same time it is necessary to define the way of updating the monitored conditions. These conditions are defined together with their update method in chapter 3.1.2.

3.1.2. Defining monitored conditions for creating a profile

Predefined violation groups are taken into account while defining the monitored conditions for the profile creation. To define whether this is a known or unknown bitrate, it is necessary to keep in mind the information about the data streams that occurred in the profilation process. Maintaining the data streams is a complementary condition in terms of profile update, i.e. if the data stream has occurred during at least one profilation period, it will be included in the known data streams.

Due to the condition to distinguish the frequency of the data stream from the monitored quantities it is necessary to keep the monitored quantities in the database. The monitored frequency condition is from the perspective of the profile update a replacement condition.

3.2. Network Flow Analysis - Intrusion Detection

As part of the analysis strategy, using a proper statistical method is considered as a suitable way for network intrusion detection [5]. The statistical method can be easily implemented over data obtained from the network based on the number of incoming packets or the number of attempts to establish a communication. Due to the fact that to identify the attempt to establish communication and the response to the system to establish such communication it is necessary to assemble communication of two points to the highest layer, IDS will use the number of incoming packets through the monitored point both in and out of the system. The IDS will be able to detect not only the attack on the monitored system, but also the attack from the monitored system and thus increase the security of the network as a whole.

Other methods of network intrusion detection require either access to the system audit resources or require the complete construction of network communication up to the application layer [6]. By using statistics over the monitored network, the following system intrusion groups are created within the IDS:

- disruption of the system from a known source,
- increased activity of the known source,
- system disruption from an unknown source,
- increased activity of the unknown source.

The design of individual groups of intrusion is based on two criteria, where the first criterion is the knowledge, ie the credibility of the data stream source. As for the known data stream source, the acceptance of the monitored stream is higher than the unknown data stream. Because of the fact that IDS would be deployed over an open system it is necessary to assume the occurrence of data streams from the unknown sources as well as taking into the account the fact that the disruption occurs from an unknown data stream source rather than a known data stream source, since the network state in the profile creation is assumed to be a normal network state and hence the known data flow is considered more secure.

The second criterion in the design of groups is the frequency of the data stream, ie the statistical occurrence of packets in the profile. By defining two basic statistical characteristics of the data stream, namely the maximum number of packets of a given data stream and the average number of packets of the monitored data stream, it is possible to recognize three levels of data stream splitting.

- Below-average bitrate - a bitrate in this category is considered a secure bitrate, it won't be considered as a violation.
- Data stream whose frequency is between average and maximum - data stream belonging to this category is considered as an increased activity, so it is advisable to monitor the given data stream and record the activity of the monitored data stream for more detailed examination.
- Data stream whose frequency is above the maximum - data stream belonging to this category is considered to be a violation, so it is necessary to record the activity of the data stream for later reconstruction of the activity and repair or minimization of damage caused to the system.

3.2.1. Normality test

A normality test is used to test whether the number of packets of the stream under review meets the selected statistical distribution. In this case, the monitored variable is X - the number of packets of this data stream in a period of 10 seconds. Formulation of mathematical hypothesis is as follows:

H_0: Random variable X ζ Poisson distribution at the 99% significance level

Above this variable the data are collected from a small network and display its results in the following histogram (see Fig. 1).

$$\sum_{i=0}^{n}\left(\frac{n_i - n.p_i}{n}\right)^2 = \left(\frac{n_0 - n.p_0}{n}\right)^2 + \left(\frac{n_1 - n.p_1}{n}\right)^2 + \ldots + \left(\frac{n_8 - n.p_8}{n}\right)^2 =$$

$$= \left(\frac{252 - 758.0{,}02247}{758}\right)^2 + \left(\frac{265 - 758.0{,}19243}{758}\right)^2 + \ldots + \left(\frac{47 - 758.0{,}00029}{758}\right)^2 = 0{,}243316101$$

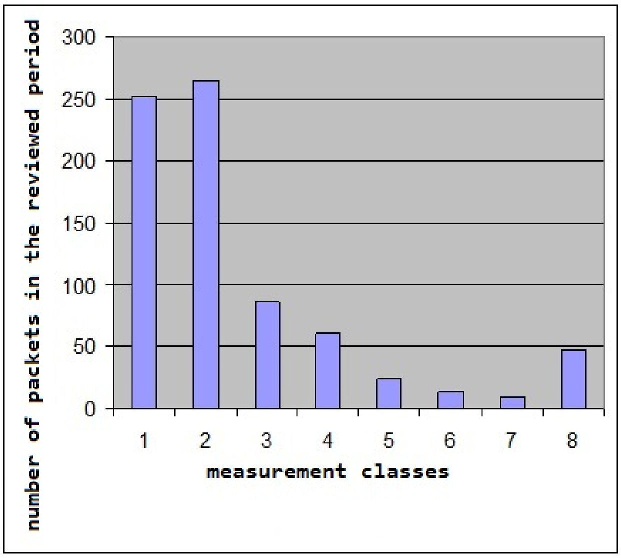

Fig. 1 Histogram of the data collected divided into measurement classes

The mathematical formula for the probability of occurrence of a phenomenon at Poisson distribution is as follows:

$$P(X = x) = \frac{\alpha^x . e^{-\alpha}}{x!} \tag{1}$$

where α = E(X). And since it is a discrete mathematical quantity:

$$E(x) = \frac{1}{n}\sum x_i \tag{2}$$

Based on the measurement: α = 8,916551.

Formula to calculate the normality test:

$$\sum_{i=0}^{n}\left(\frac{n_i - n.p_i}{n}\right)^2 < \chi^2_{1-\alpha}(k - g - 1) \tag{3}$$

where:

- n_i – frequency of class elements in measurement,
- p_i – percentage of class based on mathematical calculation,
- n – total number of elements,
- k – number of classes in measurement (k = 8),
- g – degree of freedom (g = 1),
- α – significance level (α = 0,99).

Calculation:

Mathematical quantity $\chi^2_{1-\alpha}(k - g - 1)$ in this case is $\chi^2_{0{,}01}(6) = 0{,}87209$

Based on the outcome, the Hypothesis is not to be refutted, and thus the number of incoming packets can be controlled by Poisson distribution. To calculate the profile, the value of the variable X must be determined via the following formula:

$$P(X) < 1 - \sum_{i=0}^{x-1} P(A_i) \tag{4}$$

3.2.2. Problem of mathematical calculation in relation to a program efficiency

Mathematically, the problem in relation to the effectiveness of the program lies in the time consuming calculation of the formula (4) used to determine the statistical variables in the network monitoring. The computational time is reflected by the exponential function of time, where the computation time exponentially increases with increasing α and a specified percentage of the need for observation.

The time required for the calculation is so heavy on the system that when it is deployed, the system uses system resources to the extent that it is unable to perform any network intrusion detection. However, the solution arises from predefining values that will be calculated and stored in the program so that the system does not have to perform the calculation. The time-consuming problem of the program has translated into the problem of storing all the values that the program needs for its calculations.

However, since the system should be usable in a heterogeneous network environment, there is a presumption that the value of α varies in different environments and may vary several times. The need for a table where we would have stored several thousand calculations in memory loses efficiency. However, the solution gives us the assumption of using the Poisson distribution, where the average number of occurrences of the phenomenon A is directly proportional to the length of the time period.

By setting an appropriate length of time, the IDS affects the value of α so that it falls within a range of predetermined values. For effective work of the program it is also suitable that the length of time period during which the network analysis is performed varies according to the network traffic density, expressed by the value α.

The IDS defines a range of values to which the value α must belong <1; 10>. In the first measurement, if the value of α does not fall within the value range, the program changes the length of the analysis run so that the α value falls within the specified range of values. The length of the analysis period, along with the α value, is a hallmark of network traffic in the reviewed period.

4. EXPERIMENTAL VERIFICATIONS

Experimental verifications are aimed to verify the correct operation of the program over a network communication sample. The first verification will test whether the created profile has actually received the request that shouldn't be detected as a violation over the same network (information) sample.

The goal of the next experiment is to confirm the assumption of a legitimate user's network profile that cannot detect network intrusion over a sample of the same information it was built from. In other words, the legitimate user's network profile created must fully accept all the information on which the network profile was created.

The result of the experiment will be graphed to show the maximum number of packets in the network during the reviewed period. If during the experiment, the IDS detects a violation above the same data stream sample, it means that the created profile does not fully accept all the information obtained from the data stream on which it was compiled. Failure in this experiment would result in a malfunction of profile creation, or inappropriateness of using the information obtained above the data stream sample, or inability to obtain this information. The result of the experiment will be considered successful if and only if the analytics module did not detect a violation during the tracking and the profile created fully accepts the network information.

Fig. 14 shows the maximum number of packets from network traffic during the profilation phase. Fig. 15 is a network traffic sample obtained during the profilation period, showing that it is the same network traffic sample in both of the figures.

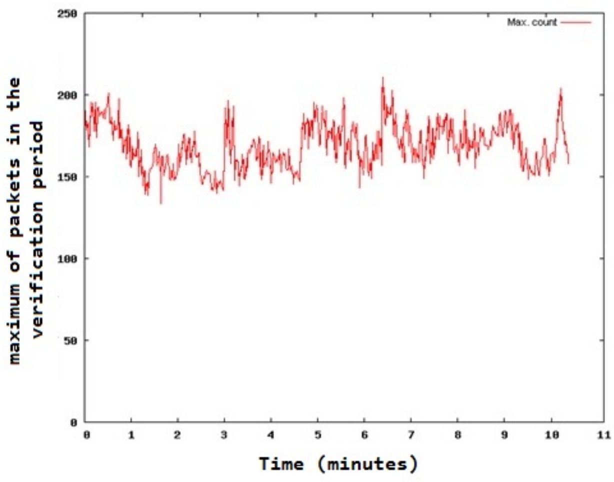

Fig. 2 Maximum number of packets on the network during the profilation phase

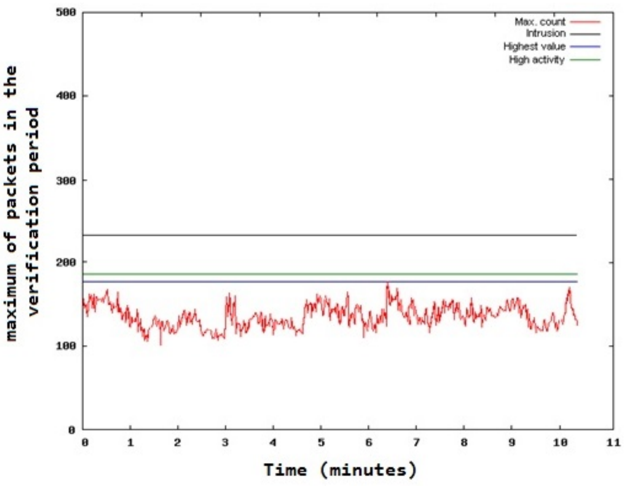

Fig. 3 A profiled network sample obtained from the data stream

This confirmed the program's assumption regarding the acceptance of a profiled network traffic sample, since in the second graph all maximum data flow values at the analyzed intervals fall below the values shown in the graph as network violation values from a known data stream and increased activity from a known data stream intrusion groups.

The aim of the next experiment is to confirm the accuracy of the analytical module, which was built on statistical assumptions. The result of the experiment is rated positively if the readings from the graph are close to what we expected. If the predicted values are lower than the measured values, this may indicate an increased system sensitivity, and the mathematical model for determination of the upper and lower detection limits will need to be reconsidered to correct the analysis module error. If the predicted values are significantly higher than the measured values, this may indicate a reduced system sensitivity. In this case, there may be a profile creation error where the profile may have created a profile that contains a system violation, so it is a good idea to run the experiment over a new profile sample in this case. If the results of the experiment show the same results after re-creating the profile, the mathematical model of the analytics module must be reconsidered.

Fig. 4 shows a graph of the traffic frequency and the number of packets marked as a violation in all four defined intrusion groups. According to the data from the figure - the most of the values were close to the mathematically created assumptions. Based on this measurement of network traffic, the derived mathematical model can be considered suitable for the IDS analysis module.

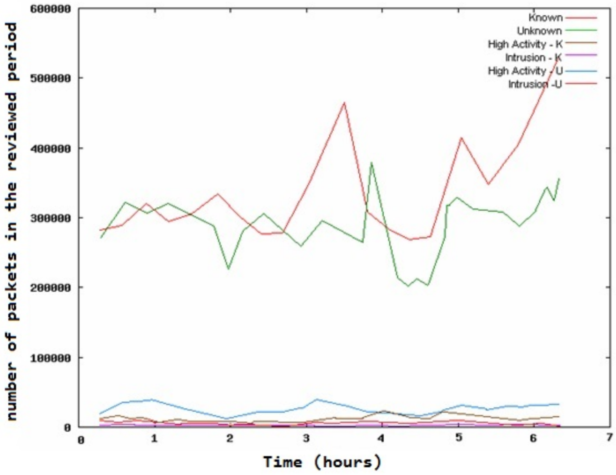

Fig. 4 Total number of detected attacks

Fig. 5 shows one selected sample of monitored maximum data stream values in the network communication during the analysis execution time periods and network intrusion values derived from the generated profile. The graph shows the functionality of IDS in real network communication with differentiated known and unknown data flows.

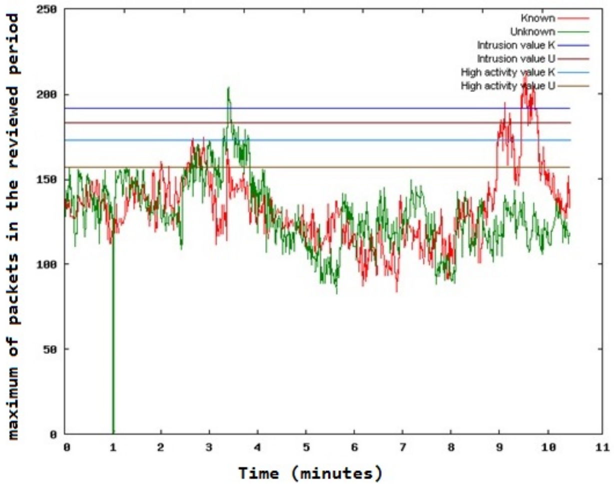

Fig. 5 The result of the network communication analysis in the reviewed period

5. CONCLUSIONS

According to the set of conditions for the IDS system, we managed to construct a functional program that based on an analysis strategy and the detection of anomalous behavior can distinguish network communication which has the characteristics of probable system disruption or violation. By adhering to and implementing the set of conditions, we managed to create an IDS system with prerequisites for high efficiency, but the program efficiency testing must be carried out after the construction of complete network communication with real IDS systems. The task of the program is then to provide information suitable for additional analysis of the problem by using a log file containing all necessary data for the construction of the communication and its control not only on the network, but up to the application layer of the ISO / OSI model.

Using the statistical values defined in the paper, the limit values are determined, at which the program detects a violation or increased user activity in the system.

However, the program itself already meets the requirements of passive IDS, but its main task is to separate suspicious network traffic from the rest of the network communication and prepare data for in-depth analysis. The in-depth analysis process is resource intensive, so suspicious network traffic needs to be filtered out prior to the in-depth analysis.

REFERENCES

[1] HUNG-JEN, L. – CHUN-HUNG, R. L. – YING-CHIH, L. – KUANG-YUANG, T.: Intrusion detection system: A comprehensive review, *Journal of Network and Computer Applications*, vol. 36, no. 1, 2013, pp. 16–24.

[2] YUE, W. T. – ÇAKANYILDIRIM, M.: A cost-based analysis of intrusion detection system configuration under active or passive response, *Decision Support Systems*, vol. 50, no. 1, 2010, pp. 21–31.

[3] CROTHERS, T.: Implementing Intrusion Detection Systems. Willey Publishing, 1st Edition. 2003.

[4] PENG, J. – KIM-KWANG, R. C. – ASHMAN, H.: User profiling in intrusion detection: A review, *Journal of Network and Computer Applications*, vol. 72, 2016, pp. 14–27.

[5] GARCÍA-TEODOROA, P. – DÍAZ-VERDEJOA, J., MACIÁ-FERNÁNDEZA, G. – VÁZQUEZB, E.: Anomaly-based network intrusion detection: Techniques, systems and challenges, *Computers & Security*, vol. 28, no. 1-2, 2009, pp. 18–28.

[6] VOKOROKOS, L. – BALÁŽ, A. – MADOŠ, B.: Application Security through Sandbox Virtualization, *ACTA POLYTECHNICA HUNGARICA*, vol. 12, no. 1, 2015, pp. 83–101.

Received December 9, 2019, accepted December 18, 2019

BIOGRAPHIES

Martin Štancel was born in 1991. In 2016 he graduated (MSc) with distinction at the Department of Computers and Informatics of the Faculty of Electrical Engineering and Informatics at Technical University in Košice, Slovakia. Since 2016 he has been a PhD student at the same department and his scientific research is focusing on computer security and object detection.

Martin Chovanec received his engineering degree in informatics in 2005 from 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.

Eva Chovancová graduated (Ing.) at the Department of Computers and Informatics at the Faculty of Electrical Engineering and Informatics of the Technical University of Košice in 2009. She defended her Ph.D. thesis in the field of computers and computer systems in 2012; her thesis itle was "Specialized Processor for Computing Acceleration in the Field of Computer Vision". Since 2012 she has been working as Assistant Professor at the Department of computers and Informatics. Her scientific research is focused on the multicore computer architectures.