Multi-agent heterogeneous intrusion detection system*

Mikuláš Pataky and Damas P. Gruska

Department of Applied Informatics, Faculty of Mathematics, Physics and Informatics, Comenius University in Bratislava, Slovak Republic
{pataky,gruska}@fmph.uniba.sk

Abstract. Multi-agent heterogeneous intrusion detection system (M-AHIDS) is a prototype proposed to detect untrusted and unusual network behaviour. The main contribution of the system is the integration of several anomaly detection techniques and machinery of multi-agent temporal logic with hybrid argumentation. Every detection technique is represented by featuring a specific detection autonomous agent. In this stage, every agent determines the flow trustfulness from aggregated connection. The anomalies are used as an input for machinery of multi-agent temporal logic which is represented by the logical agent. The logical agent is one of the system's advantages because it has huge capabilities for making a right decision about intrusions from detected anomalies. Another significant advantage of M-AHIDS is a new innovative agent – Web agent. The Web agent is capable to detect trusted host from his activity on web pages. The system M-AHIDS is based on traffic statistics in sFlow format acquired by network device with sFlow agent and is able to perform a real-time surveillance of the 10 Gb networks.

1 Introduction

The number of users using internet and local networks is increasing every day. As a consequence, there are many threats of trying to have an access to private password, to data or to injure users by other ways. Fortunately, current generation of network devices allows a real-time scraping of structured snapshots of a traffic on the networks. This information is provided by various technologies. Two the mostly used technologies are the NetFlow format introduced by CISCO and the sFlow format. These technologies allow us to observe the individual flows on the network. A flow is an unidirectional component of TCP connection (or UDP/ICMP equivalent), defined as a set of packets with identical source and destination IP addresses, ports and protocol, packed size, MAC addresses, switch ports, flags and more.

An information provided by NetFlow or sFlow can be used to detect a network attack. The most frequent attacks on networks can be divided to three main classes [1]: **Breaks privacy rules**, compromising the information confidentiality; **Alters information**, compromising the data integrity; **Denial of**

* This work was supported by the grants VEGA 1/1333/12 and UK/241/2014.

service attacks (DOS or DDOS attacks), which make a network infrastructure unavailable or unreliable, compromising the availability of a resource.

The protection of networks is, therefore, more than useful, if it is vital for long time. This problem requires the monitoring of real distributed hosts, the various events and exchanges between these hosts. It is necessary to use MAS due to the complexity of this problems.

The aim of this paper is to propose a multi-agent system for network intrusion detection M-AHIDS. The main contribution of the M-AHIDS is the integration of several anomaly detection techniques and machinery of multi-agent temporal logic with hybrid negotiation. Every detection technique is represented by featuring a specific detection autonomous agent and every agent determines the flow trustworthiness from aggregated connection. We took an inspiration for our agents in project CAMNEP [2, 3]. All CAMNEP agents are more less separate IDS and the project CAMNEP tries to connect their results to more trustworthy result. But we have decided to use another approach in our IDS. Our agents are as simple as possible. In addition to that, we have a developed new innovative agent – Web agent which is a significant advantage of our system. The Web agent is able to detect a trustworthy host from his activity on the web pages and this is based on our past project [4–6] about de-anonymization of an Internet user. This project is still deployed on all web pages of Comenius University and we can detect ordinary users' behaviour from its data.

We have used another new approach for making decisions about intrusion from detection agent's knowledge base. For this propose we have used specifically developed multi-agent temporal logic (MTL). The anomalies are used as an input for machinery of MTL which is represented by a logical agent. The logical agent is one of the system advantages because it has huge capabilities for making a right decision about the intrusions from detected anomalies. MTL allows us to collect knowledge from every detection agent from past to future. All detected intrusions are our past states in MTL and for the future states we will use the prediction methods from past and actual connections collection.

The most important contributions of our research presented in this paper are: Integration of the several anomaly detection techniques in a form of agent; Machinery of the multi-agent temporal logic; Hybrid negotiation with argumentation and immune cell inspiration; New innovative detection agent – Web agent which is able to detect a trustworthy host from his activity on the web pages. M-AHIDS is partially implemented and tested on our Department of Applied Informatics. Obtained results of M-AHIDS are comparable to another IDS.

The organization of the paper is as follows: in **section 2** – overview of the existing solutions and approaches which we use; in **section 3** – proposal of a detection system architecture; in **section 4** – detailed description of all agents in M-AHIDS; in **section 5** – overview of case study, tests and results.

2 Intrusion detection systems

Intrusion Detection System or IDS is software, hardware or combination of both used to detect an intruder's activity. The base characteristics of IDS [7] are

neutralizing illegal intrusion attempts in real time. For this reason it must be executed constantly in a host or in a network.

There are many IDS. Each of them has some advantage and disadvantage. Their strengths or weaknesses depend mostly on how they recognizes the threats. Two main approaches for detection intrusion are [1]:

Behavior-based intrusion detection approach, which discovers intrusive activity by a comparing a user's or a system's behaviour with a normal behaviour profile;

Knowledge-based (signature-based) intrusion detection approach, which detects intrusions upon a comparison between the parameters of the users' session and the known pattern attacks stored in a database.

An advantage of behaviour-based IDS is an ability to detect new form of intrusion, but their disadvantage is a possibility of un-detection of small intrusion or intrusion hidden in normal behaviour. On the another side knowledge-based IDS has an advantage in low false-positive alert for well known intrusion and high success rate for this intrusion. Their disadvantage is a low probability of detection of new intrusion.

One of the best known **knowledge-base** IDS is Snort [8]. Snort is an open source IDS available to general public. Architecture of Snort is logically divided into multiple components. These components work together to detect particular attacks and to generate output in a required format from the detection system. A Snort-based IDS consists of the following major components: *Packet Decoder*, *Preprocessors*, *Detection Engine*, *Logging and Alerting System* and *Output Modules*. Snort uses rules stored in text the files that can be modified by a text editor. Finding signatures and using them in rules is a tricky job, since more rules you use, more processing power is required to process captured data in real time.

There are several **behaviour-based** IDS. One of the most complex solution is CAMNEP[2, 3]. This project is based on trust models of network flows which is built from trustfulness values of individual flows from all agents. CAMNEP uses five type of detection agent. Each of these agent has different methodology of intrusion detection and all these agents are in core separate IDS. Authors of CAMNEP named this agents as: **Lakhina Entropy** agent [9], **Lakhina Volume** agent [10], **MINDS** agent [11], **TAPS** agent [12] and **XU** agent [13]. All of these agents use the same NetFlow protocol and all agents have capability to decide if a connection is intrusion or not. These agents are more less separate IDS and project CAMNEP tries to connect their result to more trustworthy result. We have decided to use another approach in our IDS. Our agents are as simple as can be.

One agent covers only one intrusion detection method and every agent separately evaluates every connection. Evaluating of connection means that agent compute score for the connection. Higher score indicates more suspicion behaviour. We have achieved more effective structure with this approach, because we don't have redundant computation. Another positive effect of this approach is that we know exactly how well which agent evaluates every connection.

Different interesting IDS for our research is the Multi-Agents Immune System for Network Intrusions detection (**MAISId**) [7]. Biological inspiration is very useful for many scientific departments. Inspiration in this case is biological immune cell. Immune cells have membrane receivers, who allow them to recognize specifically an epitope of an antigen [7]. The immune system is mainly founded on three elements: gene database of genes, negative selection and the clonal selection. The gene database makes it possible to generate antibodies. The negative selection makes it possible to remove the inappropriate antibodies, and the clonal selection makes it possible to keep the best antibodies to make cells memories of them. These three processes are independent; they are subjected to no central body to manage them.

MAISId is a system that performs frames analyses by a group of immune agents collaboration. These agents are distributed on the network to achieve simultaneous treatments, and are auto-adaptable to the evolution of the environment and have also the property of communication and coordination in order to ensure a good detection of intrusions in a distributed network.

An advantage of this approach is that MAISId can generate many different patterns to recognise intrusion in network flow. A disadvantage is a possibility that the system throws away a pattern which can be useful in the future.

A biological inspiration from MAISId was useful also in our M-AHIDS. We have used the idea of the biological immune cells in two cases. The first case of application is in the middle between the evaluation score from detection agent and the multi-agent temporal logic in logical agent. The second case of application is during negotiation among agents. The negotiation approaches are described bellow in this section. M-AHIDS has not created new agents for intrusion detection yet, but we are rating successfulness of our agent. This rating influences weights in logical agent, which finally makes decision about the connection.

There are two major inconveniences of the existing IDS [14]. The first one is their **difficulties to adapt oneself** to the changes of the network architecture and especially how to integrate these modifications in the detection methods. The second one is their **high rate of false-positives** (false alert).

On the another side the intrusion detection system is effective if it has the following characteristics [15, 1]: **Distribution** – to ensure the monitoring in various nodes of the network the analysis task must be distributed. **Autonomy** – for a fast analysis, distributed entities must be autonomous at the host level. **Delegation** – each autonomous entity must be able to carry out its new tasks in a dynamical way. **Communication and cooperation** – complexity of the coordinated attacks requires a correlation of several analyses carried out in network nodes. **Reactivity** – intrusion detection major goal is to react quickly to an intrusion. **Adaptability** – an intrusions detection system must be open to all network architecture changes.

The negotiation is essential in settings where autonomous agents have conflicting interests and a desire to cooperate. For this reason, a mechanisms in which the agents exchange the potential agreements according to the various rules of interaction which have become very popular in recent years as evident,

for example, in the auction and mechanism design community[16]. We use negotiation for finally deciding in M-AHIDS which connection is intrusion and which is normal.

There are basically 3 type of negotiation: Heuristic, Game-theoretic and Argumentation.

The heuristic-base approach can be a model for multi-issue negotiation under time constraints in an incomplete information setting. An important property of this model is the existence of a unique equilibrium [17]. Another solution [18] uses *approximating the rational choice of negotiation strategies with the use of decision functions*. PhD thesis [19] describes lot of heuristic-base approaches and other approaches used for negotiation.

The game-theoretic approach for negotiation can be used in an auction [20], where the seller wants to sell the items and to get the highest possible payments for them while every bidder wants to acquire the items at the lowest possible price. Authors of paper [21] use *mathematical model of the network security domain*. This concrete method is used for IDS and provides the mathematical formulation for the two persons security game between the defender and the attacker. Another similar approach is *trust-based solution for robust self-configuration* of distributed intrusion detection systems from [22, 23] is defined as a game-theoretical frame-work suitable for the collaboration of multiple heterogeneous IDS systems and it introduces a simple effective game solution concept ϵ-FIRE.

The argumentation as negotiation is the most interesting approach for our M-AHIDS. Argumentation works by constructing series of logical steps (arguments) for and against propositions of interest and as such may be seen as an extension of classical logic [24]. In classical logic, an argument is a sequence of inferences leading to a true conclusion. In argumentation system arguments can be not only a proof that propositions are true or false, but also a suggestion that propositions might be true or false. The strength of such suggestion is ascertained by examining the propositions used in the relevant arguments. This form of argumentation may be seen as a formalisation of work on informal logic and argumentation in philosophy, though it should be stressed that it was developed independently.

A formal mental model of the agents based on minimal-structure of possible worlds (time lines) has been developed using modal operators for beliefs, desires, intentions and goals having an appropriate set of properties in [25]. This approach was an inspiration for our argumentation and for a logical machinery implemented in the logical agent. Our solution is describe in the next section 4.3.

3 M-AHIDS

Diagram of M-AHIDS is shown in figure 2. M-AHIDS is based on Microsoft .net 4.5 framework and multi-vendor sampling technology sFlow. It originally runs on Microsoft server 2012. However, it can run also on Linux base operation system

with mono project. M-AHIDS is implemented as multi-thread application which uses sFlow for receiving sFlow UDP datagrams.

3.1 sFlow

sFlow is a multi-vendor sampling technology embedded within switches and routers. It provides the ability to continuously monitor application level traffic flows at wire speed on all interfaces simultaneously. sFlow monitoring of high-speed, routed and switched networks has the following properties [26]: **Accurate, Detailed, Scalable, Low Cost** and **Timely**

M-AHIDS save approximately 10 minute window of received sFlow datagrams in SQLlite in-memory database. This technology of in-memory database enables to analyse a lot of received data very quickly. All detection agents work with this database and it is also an input to logical agent.

3.2 System layers

M-AHIDS network intrusion detection system is made as four layer system.

The first layer contains in our case network 10Gb switch with sFlow agent. Switch can be replaced with another network device with sFlow agent. sFlow agent sends sFlow datagram to our IDS, which is also the sFlow collector.

The second layer contains sFlowTool and pre-processing agent. sFlowTool receives sFlow UDP datagrams. M-AHIDS reads encoded result from sFlowTool and important data saves to in-memory database. Nowadays we use these information from sFlow: 'srcIP', 'dstIP', 'srcMAC', 'dstMAC', 'srcPort', 'dstPort', 'IPProtocol', 'sampledPacketSize', 'UDPBytes', 'TCPFlags', 'inPort', 'outPort' and 'time'.

The third layer contains the detection agents. Every agent is implemented as an autonomy thread. The number of the actually active agents depends on the number of the cores in computer processor.

The forth layer contains logical agent, database with results and front-end for network administrator, which admin can use to correct the results.

4 Agents

As we mentioned in section 2, we have taken an inspiration for our agent in the project CAMNEP [2, 3]. However, there are two main differences: We have built the agents differently and we have a logical agent to complete the final decisions.

4.1 Pre-processing agent

The first step after IDS receive sFlow datagram is pre-processing as can be seen on figure 2. For covering this function we implement a pre-processing agent. Our IDS is designed for a huge network traffic on 10Gb switch. For this reason, we must do some quick decisions, which connections are interesting (connection has

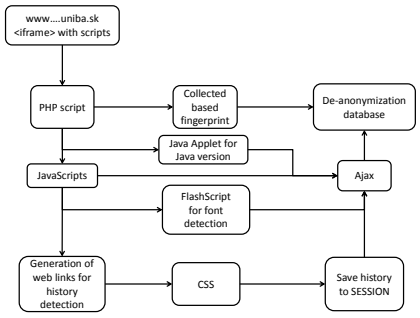

Fig. 1. Architecture of de-anonyzation system [6]

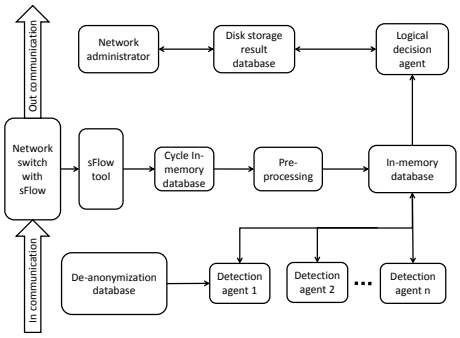

Fig. 2. Architecture of IDS

probability of being a intrusion). Like the other mentioned IDS we do this with several rules. The rules define which source, destination, port and protocol or they combination are OK and they are not interesting for the detection agents. Administrator of network can define and edit these rules.

4.2 Detection agents DA

Nowadays we have tested 5 types of intrusion detection agents. Two of these agents have arguments suitable for specification. Using this, we get 11 intruder detection agents. Every detection agent evaluate every connection from pre-processing agent. This evaluation is a integer number. Higher number means more unusual behaviour.

Average agent computes average number of connections with same property (dscIP, srcIP, dscPort, srcPort).

Volume agent counts number of the connections which have a same property and which are connected to the connections which have another same property. Concretely, we map with this method srcIP to dstIP, dstIP to srcIP, srcIP to dstPort and dstIP to srcPort. All of these mappings are provided by separate agents, which are running parallelly.

Cluster agent is the most computationally hard agent. This agent computes normalization distance between each of the connections. Agents use dscIP, srcIP, dscPort, srcPort, dstMac, srcMac for distance computations.

Web agent is one of our new contribution for this area of research. Web agent uses the database of university web page's visitors and it compares IP address of web page visitor and IP address form sFlow. If IP address is in both databases, we can decide if behind connection there is some system or a real user and then we can determine intrusion score for the connection. To determine the connection, the visited pages are analysed. If web pages are systematically visited page by page, then this is done with high probability by some system. If same page is visited more than once in short time, then the visitor was with

high probability a real human user. We have database of university web page visitors from our Internet users anonymity research [4–6].

Entropy agent captures degree of diffusion or gathering of distribution of connection properties. This detection method is based on equation:

$$H(X) = -\sum_{i=1}^{N}(\frac{n_i}{S})log_2(\frac{n_i}{S})$$

where $S = \sum_{i=1}^{N} n_i$ and X is set of connection properties $X = \{n_1, ..., n_N\}$.

4.3 Logical agent LA

Logical agent makes final decision about every connection and if this agent decides that this connection is intrusion, then agent inserts this connection to result disk storage database. Our logical agent is based on Multi-agent Temporal Logic MTL which we mentioned in section 2 and which we describe in subsubsection **MTL in M-AHIDS** below. This logic is developed especially for needs of M-AHIDS. The past states in MTL are from previous results, which are saved in permanent database. The future states will be computed by time series and Fourier transform. These future states are not implemented yet.

Logical agent has 3 important tasks. The first is to build knowledge base from results of detection agent. In this stage, LA normalizes the results to real numbers from interval $\langle 0, 1\rangle$. Normalization uses network administrator's corrections and immune inspiration for updating DA trust weights. Trust weights are also real numbers from interval $\langle 0, 1\rangle$. Higher number means more trust for the agent.

After normalization, LA uses argumentation framework to negotiate final decision – which connections are intrusions. We describe our argumentation framework in subsubsection **Argumentation framework** below. The last task for LA is to save results to permanently database.

MTL in M-AHIDS is one of the modal logics. Naturally, there are many approaches of how to build logical agents but we have decided for the multi-agent temporal logic (MTL). We have chosen this logic, because it allows as to compare every detection agent in time. This property of the MTL we use to decide, which connections are finally the intrusion.

We define simple logic syntax because nowadays we use only small subset of possible power of MTL. There are many reasons for this choice. One of the most significant is real time running of computationally hard problems in IDS. However, it is strength enough for making correct final decisions. Syntax of logic where ϕ is logic formula and $p \in prop$ is:

$$\varphi ::= \top \mid \bot$$
$$\varphi ::= p \mid \neg\varphi$$
$$\phi ::= F_i\varphi \mid G_i\varphi \mid P_i\varphi \mid H_i\varphi$$
$$\phi ::= F_A\varphi \mid G_A\varphi \mid P_A\varphi \mid H_A\varphi$$

Connectors F_i, G_i, P_i and H_i are temporal connectors for one agent $a_i \in A$ and F_A, G_A, P_A and H_A are connectors for all agents. For every judge connection there is one atomic formula p which acts in M-AHIDS as a connection with normal behaviour.

Table 1. Semantic rules of MTL

$\langle \mathcal{M}, s, i \rangle \models \top$ allways true
$\langle \mathcal{M}, s, i \rangle \not\models \bot$ never true
$\langle \mathcal{M}, s, i \rangle \models p$ iff $p \in V(s)$
$\langle \mathcal{M}, s, i \rangle \models \neg\varphi$ iff $\langle \mathcal{M}, s \rangle \not\models \varphi$
$\langle \mathcal{M}, s, i \rangle \models F_i\varphi$ iff $\exists s'(s \prec_i s') : \langle \mathcal{M}, s', i \rangle \models \varphi$
$\langle \mathcal{M}, s, i \rangle \models G_i\varphi$ iff $\forall s'(s \prec_i s') : \langle \mathcal{M}, s', i \rangle \models \varphi$
$\langle \mathcal{M}, s, i \rangle \models P_i\varphi$ iff $\exists s'(s' \prec_i s) : \langle \mathcal{M}, s', i \rangle \models \varphi$
$\langle \mathcal{M}, s, i \rangle \models H_i\varphi$ iff $\forall s'(s' \prec_i s) : \langle \mathcal{M}, s', i \rangle \models \varphi$
$\langle \mathcal{M}, s \rangle \models F_A\varphi$ iff $\forall i(a_i \in A) : \langle \mathcal{M}, s', i \rangle \models F_i\varphi$
$\langle \mathcal{M}, s \rangle \models G_A\varphi$ iff $\forall i(a_i \in A) : \langle \mathcal{M}, s', i \rangle \models G_i\varphi$
$\langle \mathcal{M}, s \rangle \models P_A\varphi$ iff $\forall i(a_i \in A) : \langle \mathcal{M}, s', i \rangle \models P_i\varphi$
$\langle \mathcal{M}, s \rangle \models H_A\varphi$ iff $\forall i(a_i \in A) : \langle \mathcal{M}, s', i \rangle \models H_i\varphi$

We define the model of MTL logic as triple $\mathcal{M} = \langle S \times A, \{\prec_i : a_i \in A\}, V \rangle$, where:

- $S = \{s_1, s_2, ...\}$ is non-empty set of states
- $A = \{a_1, a_2, ...\}$ is non-empty set of agents
- $\prec_i \subseteq S \times S$ is binary relation of pair (s, s'), which specifies from which state s can agent a_i go to state s'.
- $V : S \times A \to \wp(prop)$ is evaluating function. Function sets for every pair $(s, a) \in S \times A$, which atomic formula $p \in prop$ is true. This function reflects result of the DA and it uses value weight of the DA for encoding agent's normalise result in real number to boolean.

Semantic of connectors is shown in table 1.

The argumentation framework is one of the approaches for negotiation amongst agents. Nowadays, we use only very tiny framework which is definitely not complete because the intrusion detection is very computationally hard and M-AHIDS must work parallel with network operation. But we are still optimizing it and we will also extend this argumentation framework.

The base of our argumentation is the binary relation $\longmapsto$. $\phi \longmapsto \phi'$ means that ϕ is stronger than ϕ'. The logical formulas ϕ and ϕ' belong to $\longmapsto$ iff both contain same the atomic formula p with a opposite value. That means that the two DAs have contradictorily results about trust of same the connection. For solving this contradiction we use this rules: $X_i\varphi : w \longmapsto X_j\varphi : w'$, $H_i \longmapsto P_j$, $G_i \longmapsto F_j$ and if $X_A\varphi$ then φ where $X \in \{F, G, P, H\}$ and agent weights $w > w'$.

5 Case study

We have implemented M-AHIDS button up using several iterations, because the most important requirement on IDS is real time detection. After each iteration we did performance test and optimization. Nowadays we have the proposed intrusion detection system M-AHIDS partially implemented .

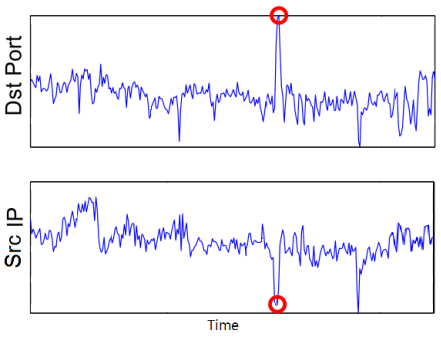

Fig. 3. Port scan anomaly

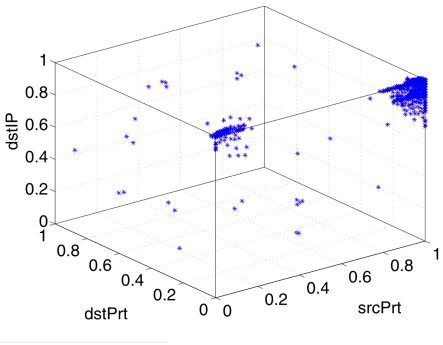

Fig. 4. Exploit cluster profile

M-AHIDS is now running on sever based on Intel i7-4770S, 2x8GB 1600MHz DDR3 CL10 DIMM RAM, 1TB HDD and OS Windows 2012 server. sFlow agent is runnig on switch Zyxel GS1910-24.

We did not make a long time test, because the M-AHIDS is still in implementing and developing stage. However, we did some tests. During these tests, the system was supervised and it learnt usual network behaviour. After three day of learning we tested system for some attack as DOS, DDOS, Port Scans, BitTorrents (there are usually unwanted in department network) and Malwares.

In the figure 3 detection of port scan anomaly can be seen . The SrcIP figure shows the relation between the number of unique source IP address and the number of all source IP address in time. The DstPort figure shows the relation between the number of unique destination ports and the number of all destination ports. Red point highlights time when anomaly was executed. In the next figure 4 exploit cluster profile can be seen, because the most of the connections are located in two clusters with the small diameter. This figure shows partial (just 3 dimension space) result from cluster agent.

The table 2 shows a false positive rate of the agents. We tested M-AHIDS during usual week network operation. Every anomaly was sent 100 times and with these anomalies we sent same number of connections with similar properties as sent anomalies. During these tests we got 3 percent false negative detections.

6 Conclusion

In this paper we have presented a proposal of a system for detection intrusions in a network. The most important system features of developed and partially implemented M-AHIDS are integration of the several anomaly detection techniques in a form of agent, machinery of a multi-agent temporal logic, hybrid negotiation with argumentation and immune cell inspiration and last but not least new innovative Web agent which is able to detect trustworthy host from his activity on web pages. This agent is based on our previous research which is deployed on all web pages of Comenius University for one and half year.

When we set the system to pass about 3 percent false negatives in the normal connections then we got 36 percent false positives in malicious connections,

Table 2. False positive (FP) rate of DA and LA

Anomaly	**#**	**Average**	**Volume**	**Cluster**	**Web**	**Entropy**	***Logical***	**FP**
DOS	100	185	76	129	145	138	125	25,00%
DDOS	100	170	60	131	168	153	123	23,00%
Port Scans	100	140	126	120	145	127	132	32,00%
BitTorrents	100	73	144	124	23	134	144	44,00%
Malwares	100	59	158	140	56	126	158	58,00%
ALL	500	627	564	644	537	678	682	**36,40%**
FP		25,40%	12,80%	28,80%	7,40%	35,60%	**36,40%**	

what is satisfaction result because project CAMNEP [3] has with 1 percent false negatives in the normal connections 40 percent false positives in malicious.

M-AHIDS is still in developing state. However, we have implemented the most of the presented features of M-AHIDS. Only one important feature we have not implemented yet – prediction of a normal network behaviour from the collected data.

As a next step we would like to implement the rest of the features to M-AHIDS, to optimize the already implemented features and to provide more and longer tests.

References

1. Boudaoud, K., Labiod, H., Guessoum, Z., Boutaba, R.: Network security management with intelligent agents. In: NOMS 2000, IEEE/IFIP Network Operations and Management Symposium, 08-14 avril 2000, Honolulu, Hawaii, Honolulu, UNITED STATES (04 2000)
2. Rehak, M., Pechoucek, M., Bartos, K., Grill, M., Celeda, P., Krmicek, V.: Camnep: An intrusion detection system for high-speed networks. Progress in Informatics **5**(5) (March 2008) 65–74
3. Rehák, M., Pechoucek, M., Grill, M., Stiborek, J., Bartoš, K., Celeda, P.: Adaptive multiagent system for network traffic monitoring. IEEE Intelligent Systems **24**(3) (2009) 16–25
4. Pataky, M.: The anonymity of the internet user. In: Proceedings of the Scientific Conference of Technology and Innovation Processes 2013, Hradec Králové, CZ, MAGNANIMITAS (2013) 35–41
5. Pataky, M.: Anonymita používateľa v internete. In: ITAT 2013: Information TechnologiesApplications and Theory Proceedings, CreateSpace Independent Publishing Platform (2013) 18–23
6. Pataky, M.: De-anonymization of an internet user based on his web browser. In: CER Comparative European Research 2014 Proceedings, London, Sciemcee Publishing (2014) 125–128
7. Benyettou, N., Benyettou, A., Rodin, V., Berrouiguet, S.Y.: The multi-agents immune system for network intrusions detection (MAISID). Oriental Journal Of Computer Science & Technology **6**(4) (December 2013) 383–390
8. Rehman, R.U.: Intrusion Detection Systems with Snort: Advanced IDS Techniques Using Snort, Apache, MySQL, PHP, and ACID. Prentice Hall PTR, Upper Saddle River, New Jersey 07458, USA (2003)
9. Lakhina, A., Crovella, M., Diot, C.: Mining anomalies using traffic feature distributions. SIGCOMM Comput. Commun. Rev. **35** (August 2005) 217–228
10. Lakhina, A., Crovella, M., Diot, C.: Diagnosing network-wide traffic anomalies. SIGCOMM Comput. Commun. Rev. **34** (August 2004) 219–230

11. Ertöz, L., Eilertson, E., Lazarevic, A., Tan, P.N., Kumar, V., Srivastava, J., Dokas, P.: 3. In: MINDS - Minnesota Intrusion Detection System. MIT Press (2004) 21
12. Sridharan, A., Ye, T.: Tracking port scanners on the ip backbone. In: Proceedings of the 2007 workshop on Large scale attack defense. LSAD '07, New York, NY, USA, ACM (2007) 137–144
13. Xu, K., Zhang, Z.L., Bhattacharyya, S.: Reducing unwanted traffic in a backbone network. In: Proceedings of the Steps to Reducing Unwanted Traffic on the Internet on Steps to Reducing Unwanted Traffic on the Internet Workshop, Berkeley, CA, USA, USENIX Association (2005) 2–2
14. Majorczyk, F., Totel, E., Me, L.: Experiments on cots diversity as an intrusion detection and tolerance mechanism. In: Workshop on Recent Advances on Intrusion-Tolerant Systems (WRAITS). (March 2007)
15. Boudaoud, K., Guessoum, Z.: A multi-agents system for network security management. In: SMARTNET 2000, 6th IFIP Conference on Intelligence in Networks, September 18-22, 2000, Vienna, Austria, Vienna, AUSTRIA (09 2000)
16. Rahwan, I., Ramchurn, S., Jennings, N.R., McBurney, P., Parsons, S., Sonenberg, L.: Argumentation-based negotiation. The Knowledge Engineering Review **18**(4) (2003) 343–375
17. Fatima, S.S., Wooldridge, M., Jennings, N.R.: Multi-issue negotiation under time constraints. In: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 1. AAMAS '02, New York, NY, USA, ACM (2002) 143–150
18. Braun, P., Brzostowski, J., Kersten, G., Kim, J., Kowalczyk, R., Strecker, S., Vahidov, R.: e-negotiation systems and software agents: Methods, models, and applications. In: Intelligent Decision-making Support Systems. Decision Engineering. Springer London (2006) 271–300
19. Faratin, P.: Automated Service Negotiation Between Autonomous Computational Agents. PhD thesis, University of London, Queen Mary and Westfield College, Department of Electronic Engineering (2000)
20. Sandholm, T.: Algorithm for optimal winner determination in combinatorial auctions. Artificial Intelligence **135**(12) (2002) 1 – 54
21. Vaněk, O., Yin, Z., Jain, M., Bošanský, B., Tambe, M., Pěchouček, M.: Game-theoretic resource allocation for malicious packet detection in computer networks. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 2. AAMAS '12, Richland, SC, International Foundation for Autonomous Agents and Multiagent Systems (2012) 905–912
22. Bartos, K., Rehak, M.: Trust-based solution for robust self-configuration of distributed intrusion detection systems. In: In Proceedings of the 20th European Conference on Artificial Intelligence (ECAI), IOS Press (2012) 121–126
23. Bartos, K., Rehak, M.: Distributed self-organized collaboration of autonomous ids sensors. In: Dependable Networks and Services, Heidelberg, Springer (2012) 113–117
24. Parsons, S., Giorgini, P.: An approach to using degrees of belief in bdi agents. In Bouchon-Meunier, B., Yager, R., Zadeh, L., eds.: Information, Uncertainty and Fusion. Volume 516 of The Springer International Series in Engineering and Computer Science. Springer US (2000) 81–92
25. Kraus, S., Sycara, K., Evenchik, A.: Reaching agreements through argumentation: a logical model and implementation. Artificial Intelligence **104**(12) (1998) 1 – 69
26. sFlow.org: Traffic monitoring using sflow (2003)