Analysing of M-AHIDS with future states on DARPA and KDD99 benchmarks

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. Second generation of Multi-agent heterogeneous intrusion detection system (M-AHIDS) is a prototype proposed to detect untrusted and unusual network behaviour. The M-AHIDS is based on online traffic statistics in sFlow format acquired by network device with the sFlow agent and is able to perform a real-time surveillance of the 10 Gb networks. However, after an immense reimplementation it is capable to process also offline data set from DARPA Intrusion Detection Evaluation Data Set and KDD99 Cup data set. Offline data sets are used for the correct comparison with another IDSs. The main contribution of the system is the integration of several anomaly detection techniques, new future state prognostic and new machinery of multi-agent temporal logic with hybrid argumentation. Every detection technique is represented by featuring a specific detection autonomous agent. At 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. M-AHIDS is already partially implemented, tested and modified accordingly for more than three years.

1 Introduction

The number of users using internet and local networks is increasing every day. Consequently, 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.

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

The protection of networks is, therefore, more than useful, if it is vital for long time. This issue requires monitoring of real distributed hosts, of various events and of exchanges between these hosts. Multi agent system (MAS) is very effective approach for this kind of problems as it can integrate many different techniques to one solution.

The aim of this paper is to propose the second generation of multi-agent system for network intrusion detection M-AHIDS. The first generation was presented in [2]. This generation is based on several years of experiences with developing, improving, implementing, deploying and testing of M-AHIDS. The main contribution of the second generation of M-AHIDS is the integration of several anomaly detection techniques, new future state prognostic and new machinery of multi-agent temporal logic with hybrid negotiation based on argumentation. Every detection technique is represented by featuring a specific detection autonomous agent and every agent determines the flow trustworthiness from aggregated connection. Inspiration for our agents came from project CAMNEP [3, 4]. All CAMNEP agents are more or less separate IDS and the project CAMNEP tries to connect their results to the more trustworthy results. But we have decided to use another approach in our IDS. Our agents are as simple as possible.

We are also still improving our unique [1] Web agent. The web agent is based on our past project [5–7] about de-anonymization of an Internet user. This project has been deployed on all web pages of Comenius University for more than three years. We can detect ordinary users' behaviour from its data. We used all the collected data for deep analysis and we created Web agent which is able to detect a trustworthy host based solely on his activity on the web pages.

We have used another new approach for making decisions about intrusion from agent's knowledge base detection. For this purpose we have used specifically developed multi-agent temporal logic (M-ATL). The anomalies are used as an input for machinery of M-ATL and the new version of hybrid argumentation which are represented by a logical agent. The logical agent is one of the system advantages because it has huge capabilities for making the right decision about the intrusions from detected anomalies. All detected intrusions are the past states in M-ATL and we are using newly implemented prediction methods base on regression models of time series for the future states. The regression models are used for computation of the future states from the collection of the past and the actual connections.

The most important contributions of our research presented in this paper are THE FOLLOWING: Improving the integration of the several anomaly detection techniques in a form of an agent; Extension of machinery of the multi-agent temporal logic and hybrid negotiation about the future state; Major update of argumentation framework; Presenting new testing approach based on offline

[1] with our best knowledge

DARPA Intrusion Detection Evaluation Data Set and KDD99 Cup data set. M-AHIDS is partially implemented and tested on local network of Department of Applied Informatics. Results obtained on KDD99 are comparable to another IDS.

The organization of the paper is as follows: in **Section 2** – overview of the IDS and selected existing solutions and approaches; in **Section 3** – proposal of 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 a software, hardware or combination of both used to detect an intruder's activity. The base characteristics of IDS [8] are neutralizing illegal intrusion attempts in the real time. Consequently, it must be executed constantly in a host or in a network.

There are many types of IDS and each of them has some advantages and disadvantages. Their strengths and weaknesses depend mostly on the way they recognize the threats. Two main approaches for detection intrusion are [1]:

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

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

In recent years, several new approaches in IDS systems have been published. Certain approaches have been identified as relevant for our project. The first, multi-agent distributed IDS(DIDS) model based on the **BP neural network** adopts the modes of distributed detection and distributed response [9]. The second, **emulation-based** network intrusion detection systems have been devised to detect the presence of shellcode in the network traffic by trying to execute (portions of) the network packet payloads in an instrumented environment and checking the execution traces for signs of shellcode activity [10]. The fourth, **multi-stage approach** to constructing hierarchical classifiers that combines process mining, feature extraction based on temporal patterns and constructing classifiers based on a decision tree [11]. The fifth, **content anomaly detection** (CAD) models the payloads of traffic instead of the higher level attributes. Zero-day attacks then appear as outliers to the properly trained CAD sensors [12]. The sixth approach is to detect TCP connection based attacks using certain **data mining algorithms**[13]. J-48 decision tree algorithm and Nave Bayes classifiers were learnt on 19 selected features from KDD 99 dataset. The selected feature had been chosen by Markov blanket and Pearson correlation. The approach could detect about 74% of novel attacks with 19 features.

3 M-AHIDS

The following section briefly proposes the foundations for the second generation network intrusion detection multi-agent system M-AHIDS. Design of the system arose from theoretical research as well as from practical experiences which have been already obtained by testing for more than three years.

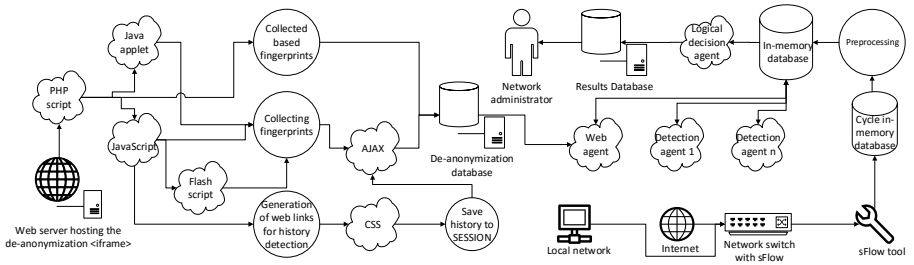

Fig. 1. Architecture of IDS

3.1 System layers

M-AHIDS network intrusion detection system consists of four layers.

The first layer contains the 10Gb network switch with the sFlow agent. This switch can be replaced by another network device with the sFlow agent. The sFlow agent sends sFlow datagram to M-AHIDSwhich functions also the sFlow collector.

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

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

The forth layer contains the new version of the logical agent, database with results and the front-end for network administrator which can be used to correct the results.

3.2 sFlow

sFlow is a multi-vendor sampling technology embedded within network 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 [14]: **Accurate, Detailed, Scalable, Low Cost** and **Timely**.

M-AHIDS saves approximately 10 minutes window of received sFlow datagrams in SQLlite in-memory database. In-memory database enables to analyse large amounts of received data very quickly. All detection agents work with this database and the database is also an input to the logical agent.

3.3 Implementation details

Diagram of the second generation M-AHIDS is shown in figure 1. 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 also be run on Linux based operating systems using mono platform. M-AHIDS is implemented as multi-thread application which uses sFlow for receiving sFlow UDP datagrams.

4 Agents

As written in [2], our agents were inspired by the project CAMNEP [3, 4]. However, there are several main differences: We have built the agents differently, we have added new type of agent - the Web agent, we have used the hybrid negotiation with argumentation and immune cell inspiration, prediction of future states and we have created a logical agent to complete the final decisions.

4.1 The pre-processing agent

The first step after IDS receives the sFlow datagram is pre-processing, as can be seen on figure 1. For the coverage of this function, a pre-processing agent is implemented. M-AHIDS is designed for a very high network traffic on 10Gb network switch. For this reason, agent needs to make quick decisions which connections are important (connection has probability of being an intrusion). Similarly to the other mentioned IDS we implemented this with several rules. The rules define which source, destination, port and protocol or their combinations are problem-free and they are not interesting for the detection agents. The administrator of the network can define and edit these rules.

4.2 Detection agents DA

Six types of innovated intrusion detection agents have been tested. Two of these agents have the arguments suitable for specification. Using this, we get **15** intruder detection agents. Every detection agent evaluates every connection from the pre-processing agent. The output of this evaluation is an integer. Higher number indicates behaviour that is more unusual.

The count agent is the first scalable type of agent which is counting number of connections with the same property ('dscIP', 'srcIP', 'dscPort', 'srcPort'). Higher number of connections with particular property means that connections are more suspicious. The exact mathematical formula is:

$$R_{CO}(V) = \{r_v | r_v = |C_v| : \forall v \in V\} \tag{1}$$

where R_{CO} is the set of results of the count agent, r_v is the result for all connections with particular property [2] $v \in V$. C_v is the set of the connections with property v and V is the set of all properties. M-AHIDS has a separate agent for every connection's property which is running in its own thread.

The average agent is the second scalable type of agent and it computes average number of connections with the same property ('dscIP', 'srcIP', 'dscPort', 'srcPort'). Higher difference between the number of connections and the average number of connections with particular property means more suspicious connections. The exact mathematical formula is:

$$R_{AVG}(V) = \{r_v | r_v = ||C_v| - \texttt{Avg}(R_{CO}(V))| : \forall v \in V\} \tag{2}$$

where R_{AVG} is the set of the results of the average agent.

The volume agent counts the number of connections which have the same value in linked properties. Specifically, agent links srcIP to dstIP, dstIP to srcIP,

[2] e.g. dscPort 45

srcIP to dstPort and dstIP to srcPort. All of these links are provided by separate agents, which are running in parallel. The exact mathematical formula is:

$$R_{VOL}(V) = \{r_v | r_v = |C_v| : \forall v \in V\} \tag{3}$$

where V is the set of ordered pairs $\{(v_{1,1}, v_{1,2}), (v_{2,1}, v_{2,2}), \ldots, (v_{n,1}, v_{n,2})\}$ and C_v is the set of all connections which have property $v_1 \in v \in V$ and they are linked with connections with property $v_2 \in v \in V$.

The cluster agent is the most computationally complex agent. This agent computes normalization distance between each of the connections. Agent uses dscIP, srcIP, dscPort, srcPort, dstMac and srcMac for distance computations.

$$R_{CLU} = \left\{ r | r = \frac{\sum_{c' \in C} |c, c'|}{n} : \forall c \in C \right\} \tag{4}$$

where C is the set of connections, $|c, c'|$ is the distance between two connections from C, $c, c' \in C$ and n is the capacity of set C.

The Web agent is one of our contributions in this area of research. The web agent uses the database from de-anonymization system shown on left side of the fig. 1. It compares the IP from the sFlow database with IP address of all the visitors of all web pages. If the IP address is in both databases, agent calculates if there is a higher probability of a system or a real user behind a connection and then agent determines the intrusion score for the connection using the analysis of the visited pages. If the web pages are systematically visited page by page, then there is a high probability that the visitor is a system. If the same page is visited more than once in short time, then there is a high probability that the visitor is a real user. The database of the university serving as web page visitors database was created using Internet users anonymity research [5–7].

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

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

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

4.3 The logical agent (LA)

The logical agent makes the final decision about every connection and if this agent evaluates that this connection is an intrusion, then the agent inserts this connection to the permanent database and can be used to alert server administrator. The LA is based on Multi-Agent Temporal Logic M-ATL which was presented in [2] with argumentation upgrades described below. The M-ATL and also the argumentation was developed specifically for the needs of M-AHIDS.

The new (upgraded) version of LA also contains computation of the future states. The past states in M-ATL come from previous results which are saved in the permanent database. The future states are computed based on regression models of time series [15]. This approach was chosen as it is one of the fastest prediction technique. Low computational complexity is very important for real time IDS. The inputs are the counts of the same connection during each 10 seconds from 10 minutes time window. That means that we have 60 counts for each connection which makes the time series. From the time series six future

states are computed for the following one minute. The trend part of the time series is chosen based on MAPE [16] rating from linear trend $T = a_0 + a_1 t$, parabolic trend $T = a_0 + a_1 t + a_2 t^2$ or exponential trend $T = a_0 a_1^t$; where T is trend function, a_i is parameter of the function and t is time.

The LA has three important tasks. The first one is to build a knowledge base from the results of the DA. At this stage, LA normalizes the results to interval $\langle 0, 1\rangle$. The normalization uses the network administrator's corrections and the immune inspiration for updating the DA's trust weights. The trust weights are also real numbers from interval $\langle 0, 1\rangle$. Higher number means more trust for the agent. LA converts the results of DA to boolean value. This conversion is based on agents' trust and the mathematical formula is:

$$C_A = \{c_{Ai} | \exists r_{Ai} \in R_A : r_{Ai} > (1 - W_A)\} \tag{5}$$

where $C_A = \{c_{A1}, c_{A2}, \ldots, c_{An}\}$ is the set of intrusions detected by DA A, $R_A = \{r_{A1}, r_{A2}, \ldots, r_{An}\}$ is the set of normalized results from agent DA A and W_A is the trust weight of the agent A.

After normalization, LA uses the new argumentation framework to negotiate the final decision – which connections are intrusions. We describe our argumentation framework below. The last task for LA is to save the results to the permanent database.

The argumentation framework (FA) is one of the approaches for negotiation amongst agents. The implemented FA can evaluate all used logical clauses but is not complete as the intrusion detection is computationally hard and M-AHIDS must work in parallel with network operation.

However, as the logic machinery of our M-ATL runs after all our DA agents in M-AHIDS have finished evaluation of all connections, we do not have to think about incomplete knowledge in our argumentation framework. This fact simplifies the proposal of argumentation framework.

The new version of argumentation framework is based on work of Dung [17]

An argumentation framework AF is ordered pair $AF = \langle AR, attacks\rangle$ where AR is set of arguments and $attacks$ is binary relation based on AR: $attacks \subseteq AR \times AR$

A conflict-free set of arguments S is if there are no arguments $A, B \in S$ such that $(A, B) \in attacks$

An acceptable argument $A \in AR$ with respect to a set S iff for each argument $B \in AR$: if B $attacks$ A then B is $attacked$ by S.

An admissible set of arguments is a conflict-free set of arguments S iff each argument in S is acceptable with respect to S.

A preferred extension of an argumentation framework AF is a maximal (with respect to set inclusion) admissible set of AF, which defines the (credulous) semantics of an argumentation framework.

Important provable conclusion [17] is, that every argumentation framework possesses at least one preferred extension.

A stable extension is a conflict-free set of arguments$S$ iff S $attacks$ each argument which does not belong to S.

S is a stable extension iff $S = \{A | A$ is not attacked by $S\}$

Another important conclusion that every **stable extension** is also preferred extension, but not vice versa, is proved in [17]. This determination of the argumentation framework is sufficient for our proposes.

The base of **our new version of argumentation** is also the binary relation of preferences (*attaks*) $\longmapsto$. $\varphi \longmapsto \varphi'$ means that φ is stronger than φ'. The logical formulas φ and φ' belong to $\longmapsto$ iff both contain the same atomic formula p with an opposite value. That means that the two DAs have contradict results about trust of the same connection. For building relation of preferences we use rules:

$$X_I\varphi : w_I \longmapsto X_J\varphi : w_J \text{ iff } \sum_{i\in I} w_i > \sum_{j\in J} w_j, \tag{6}$$

$$p_I : w_I \longmapsto p_J : w_J \text{ iff } \sum_{i\in I} w_i > \sum_{j\in J} w_j \tag{7}$$

$$X_i p_i \longmapsto p_j \tag{8}$$

$$H_i\varphi \longmapsto P_j\varphi \tag{9}$$

$$G_i\varphi \longmapsto F_j\varphi \tag{10}$$

$$X_A\varphi \text{ iff } \varphi \tag{11}$$

where $X \in \{F, G, P, H\}$, p_i is the evaluation of connection by agent a_i, I, J are same sets of labelling of agents, $i \in I$, $j \in J$ and w_i is the weight of agents' a_i trustfulness. The connectors F (some future state), G (all future states), P (some past state), H (all past states) and logical formula φ are defined in our previous paper [2]. Rules 6 - 11 should be interpreted as: 6 and 7 - the agents with higher collective trust beat the agents with lower trust; 8 - complex knowledge beats simple knowledge; 9 - all past states beat one past state; 10 - all future states beat one future state; 11 - formula is true iff all agents have the same evaluation.

The LA computes preferred extensions of AF and that is a solution for the problem with evaluation of the connection represented by one atomic variable p. If this extension is also stable extension and it contains arguments which claim that the connection is part of the intrusion, LA will write this connection to the permanent database of results.

5 Results

We have implemented M-AHIDS bottom up using several iterations, because the most important requirement on IDS is the real time detection. After each iteration performance test and optimization were performed.

Table 1. False negative (FN) rate of DA and LA

		Agents							
Attack	**#**	**Count**	**Average**	**Volume**	**Cluster**	**Web**	**Entropy**	***Logical***	**FP**
DOS	100	96	98	99	99	95	97	99	1,00%
DDOS	100	94	95	60	97	99	99	96	4,00%
Port Scans	100	96	97	95	96	98	95	98	2,00%
BitTorrents	100	70	73	98	95	23	96	97	3,00%
Malwares	100	62	59	99	97	56	94	97	3,00%
ALL	500	418	422	451	484	371	481	487	2,60%
FN		**16,40%**	**15,60%**	**9,80%**	**3,20%**	**25,80%**	**3,80%**	**2,60%**	

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

		Agents							
Attack	**#**	**Count**	**Average**	**Volume**	**Cluster**	**Web**	**Entropy**	*Logical*	**FP**
DOS	100	177	183	80	125	137	145	127	27,00%
DDOS	100	165	170	58	128	165	150	129	29,00%
Port Scans	100	139	144	122	123	135	128	132	32,00%
BitTorrents	100	69	75	146	124	33	134	143	43,00%
Malwares	100	70	59	161	138	68	126	157	57,00%
ALL	500	620	631	567	638	538	683	688	**37,60%**
FP		24,00%	26,20%	13,40%	27,60%	7,60%	36,60%	**37,60%**	

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

During the tests, the system was supervised and it learnt the usual network behaviour. After three days of learning we tested system for attacks like DoS, DDoS, Port Scanning, BitTorrent (usually unwanted in commercial networks) and Malware attacks.

The Table 2 shows a false positive rate of the agents and the Table 1 shows a false negative rate of the agents.M-AHIDS was tested during usual week network operation. Every attack was sent 100 times and with these attacks we sent the same number of connections with similar properties as the sent attacks.

This test scenario was repeated two times. Once with the simpler LA with smaller AF and afterwards with the new AL with more complex AF. The new LA had better FN about $0,4$ percentage points which is $13,33$ **percentage progress** and it had the worst FP about $1,2$ percentage points which is only $3,3$ percentage retrogression.

5.1 Benchmark KDD99 Cup data set

Furthermore the second generation M-AHIDS was adapted for KDD99 Cup data set. KDD99 Cup data set was adopted for this study because it is widely used intrusion detection data set. The paper [18] compared 125 intrusion detection systems using KDD99 Cup data set between 2010 and 2015. This indicates that although KDD99 dataset is more than 15 years old, it is still widely used in the academic research. By this way the comparison between M-AHIDS and other similar studies is achieved. KDD99 Cup data set is created by extracting some features (IP number, port number, initial date) from DARPA 98 and it has about 4 900 000 data vectors. This data set is prepared by Stolfo et al. [19] and is built based on the data captured in DARPA98 IDS evaluation program [20]. KDD99 Cup data set includes 80% of attack and 20% of normal data. The package with the cup data set also contains training data set (labelled) and testing data set (without label). Each connection vector has 41 features and is labelled either normal or attack.

However M-AHIDS was not compatible with KDD99 Cup data set, because it was designed for recognizing intrusion directly from sFLOW. So it had to be adapted for comparison. The adaptation processes was done in two major steps:

1. The first step was about the changing of M-AHIDS to process offline data. The raw data from DARPA Intrusion Detection Evaluation Data Set [21] was used. This was similar to data set which had been provided by sFlow.
2. The second step was the modification of M-AHIDS to process extracted features provided by KDD99 Cup data set. Difference between KDD99 data set and data set provided by sFLOW is significant. For that reason all agents had to be updated.

The detailed description about those adaptation processes are out of range of this paper. Only results achieved after each step are presented.

The biggest disadvantage of using the DARPA Intrusion Detection Evaluation Data Set and KDD99 Cup data set is that the important Web Agent feature of M-AHIDS cannot be used because there are no data for it. Despite this fact we compared our approach with both data sets.

In addition, the measures were used to evaluate the performance of MAS-IDS: accuracy, detection rate, false alarm rate:

$$DedectionRate = \frac{NumberOfDetectedAttacks}{NumberOfAttacks} \cdot 100\% \tag{12}$$

$$FalsePositive = \frac{MisclassifiedConnections}{NumberOfNormalConnection} \cdot 100\% \tag{13}$$

$$Accuracy = \frac{CorrectClassifiedConnections}{NumberOfConnections} \cdot 100\% \tag{14}$$

The detailed results can be seen in Tables 3 and 4

Table 3. Darpa success rates

	Agents					
Measures	**Count**	**Average**	**Volume**	**Cluster**	**Entropy**	***Logical***
Detection Rate	78,4	81,4	79,4	85,3	83,9	**93,1**
False Positive	10,1	9,8	10	9,2	9,3	**8,8**
Accurancy	78,2	80,9	79,1	84,8	83,1	**92,3**

Table 4. KDD99 success rates

	Agents					
Measures	**Count**	**Average**	**Volume**	**Cluster**	**Entropy**	***Logical***
Detection Rate	77,9	81,1	78,8	84,9	82,4	**91,9**
False Positive	10,6	10,1	10,1	9,5	9,7	**9,1**
Accurancy	77,7	80,5	78,2	84,1	82,6	**91,5**

6 Conclusion

In this paper we presented the proposal for the second generation of the system for detection intrusions in a network. The most important system features of the developed and partially implemented M-AHIDS are integration of several innovated anomaly detection techniques in a form of agent, machinery of a multi-agent temporal logic, hybrid negotiation with new version of argumentation and immune cell inspiration, newly implemented computation of future

states and last but not least the 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 three years.

When the system passed $2,6\%$ false negative in the normal connections, the system achieved $37,6\%$ false positive in the malicious connections. That is a satisfactory result as project CAMNEP [4] achieved with 1% false negative in the normal connections only 40% false positive in the malicious connections.

Satisfactory results were achieved on DARPA Intrusion Detection Evaluation Data Set (Detection Rate = 93,1; False Positive = 8,8; Accuracy = 92,3) and KDD99 Cup data set (Detection Rate = 91,9; False Positive = 9,1; Accuracy = 91,5) which are comparable witch other IDS systems. However we have to consider that one of our major feature (Web agent) can not be used due to the lack of data. There is a reasonable belief that the results in the online testing with Web agent will yield better results.

M-AHIDS is still in the development phase, but parts of the system are deployed for more than three years on the department network. Here, we have implemented the most of the presented features of M-AHIDS.

As the 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 lasting tests. Here we also consider more sophisticated approach for data clustering as in [22].

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 April 2000, Honolulu, Hawaii, Honolulu, UNITED STATES (04 2000)
2. Pataky, M., Gruska, D.P.: Multi-agent heterogeneous intrusion detection system. In: Proceedings of the 23th International Workshop on Concurrency, Specification and Programming, Chemnitz, Germany, September 29 - October 1, 2014. (2014) 184–195
3. 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
4. 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
5. 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
6. Pataky, M.: Anonymita používateĺa v internete. In: ITAT 2013: Information Technologies - Applications and Theory Proceedings, CreateSpace Independent Publishing Platform (2013) 18–23
7. 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

8. 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
9. Zhai, S., Hu, C., Weiming, Z.: Multiagent distributed intrusion detection system model based on bp neural network. International Journal of Information and Network Security (IJINS) **3**(3) (2014)
10. Abbasi, A., Wetzels, J., Bokslag, W., Zambon, E., Etalle, S.: On emulation-based network intrusion detection systems. In Stavrou, A., Bos, H., Portokalidis, G., eds.: Research in Attacks, Intrusions and Defenses. Volume 8688 of Lecture Notes in Computer Science. Springer International Publishing (2014) 384–404
11. Bazan, J.G., Szpyrka, M., Szczur, A., Dydo, L., Wojtowicz, H.: Classifiers for behavioral patterns identification induced from huge temporal data. In: Proceedings of the 23th International Workshop on Concurrency, Specification and Programming, Chemnitz, Germany, September 29 - October 1, 2014. (2014) 22–33
12. Whalen, S., Boggs, N., Stolfo, S.J.: Model aggregation for distributed content anomaly detection. In: Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop. AISec '14, New York, NY, USA, ACM (2014) 61–71
13. Ugtakhbayar, N., Usukhbayar, B., Nyamjav, J.: An approach to detect tcp/ip based attack. International Journal of Computer Science and Network Security **16**(4) (April 2016) 37–40
14. sFlow.org: Traffic monitoring using sflow (2003)
15. OSTERTAGOV, E.: Modelovanie asovch radov. The 13th International Scientific Conference: Trends and Innovative Approaches in Business Processes 2010 (2010)
16. Tofallis, C.: A better measure of relative prediction accuracy for model selection and model estimation. JORS **66**(8) (2015) 1352–1362
17. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artif. Intell. **77**(2) (September 1995) 321–357
18. Özgür, A., Erdem, H.: A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015. PeerJ PrePrints **4** (2016) e1954
19. Stolfo, S.J., Fan, W., Lee, W., Prodromidis, A., , Chan, P.K.: Cost-based modeling for fraud and intrusion detection: Results from the jam project. discex **2** (2000) 1130
20. Lippmann, R.P., Fried, D.J., Graf, I., Haines, J.W., Kendall, K.R., McClung, D., Weber, D., Webster, S.E., Wyschogrod, D., Cunningham, R.K., Zissman, M.A.: Evaluating intrusion detection systems: The 1998 darpa off-line intrusion detection evaluation. discex **2** (2000) 1012
21. Lippmann, R., Haines, J.W., Fried, D.J., Korba, J., Das, K.: The 1999 darpa off-line intrusion detection evaluation. Comput. Netw. **34**(4) (October 2000) 579–595
22. Lasek, P., Lasek, K.: Relative constraints as features. In Popova-Zeugmann, L., ed.: Proceedings of the 23th International Workshop on Concurrency, Specification and Programming, Chemnitz, Germany, September 29 - October 1, 2014. Volume 1269 of CEUR Workshop Proceedings., CEUR-WS.org (2014) 121–125