Asian Journal of Computer and Information Systems (ISSN: 2321 – 5658)
Volume 04– Issue 05, October 2016

Early Staged Cyber Incidents Detection in Critical Infrastructures

A. Anskaitis, T. Baksys, N. Blazys, R. Rainys[*]

Kazimieras Simonavicius University
J. Basanaviciaus str. 29A, LT-03109 Vilnius, Lithuania

[*]*Corresponding author's email: rytis.rainys [AT] ksu.lt*

ABSTRACT— ***The aim of the research is to create cyber incidents early detection model based on network traffic and OS-based system analyses. Developed cyber attacks detection model is based on anomalies measurements. With the 11 selected parameters and measurement software for real-time data traffic analyze, anomalies in traffic observed during cyber-attack simulation process. For OS-based system similar approach used with 4 selected parameters and Neural-networks classification method. This measurement solution detects anomalies in parameters sets and indicates cyber incidents.***

Keywords— security, cyber-attack, incident detection, traffic anomaly.

1. INTRODUCTION

The amount and impact of cyber incidents in Internet networks are increasing. According to the Communications Regulatory Authority, in Lithuania during 2015 number of registered incidents increased to 41 583 and this is by 15% more than in 2014. Under 18 thousands vulnerable devices detected and part of them used for DoS (Denial of Service) attacks to critical infrastructures running on Internet [1]. Modern cyber attacks becoming more sophisticated with the usage of 0-day vulnerabilities. This leads to attacks that are bypassing defense systems and remains undetected. Undetected at early stages cyber incidents have a greatest negative effect for Industrial Control Systems (ICS or Supervisory Control and Data Acquisition Systems – SCADA).

Vital sectors such as energy, oil and gas, transport or chemical, rely on ICS to supervise and control their key processes. The latest demand for connectivity changed the ICS environment from proprietary, isolated systems to open architectures and IP based technologies. ICS connectivity to Internet network resulted in an increased attack surface, thus exposing the critical functions to higher cyber security risks. The Aurora vulnerability and *Stuxnet* are examples of advanced and well-prepared attacks, which were dedicated to exploit unknown vulnerabilities of particular control systems [2].

This research did the attempt to investigate real ICS situation. ICS that are connected to the Internet were scanned. Research did over 2 mln. Lithuanian range IP addresses scans. As research tool was chosen NMAP v6.40 (network discovery and security auditing instrument) installed on server with Ubuntu 14.04 operating system. Server allocation was chosen in such way that scanning results could not be affected by the firewall on server side. Lithuanian IP was scanned to search open 502 TCP port, which is used for MODBUS systems controllers to communicate with servers, reachable through Internet. As a result 509 IPs were detected as having open 502 TCP port. 98 IPs of them have other open ports: 80, 443, 8080, 8081, 8443. Results confirm hesitation that ICS are connected to the Internet and may have security vulnerabilities, which could be used for cyber security attacks, intrusions and sensitive information theft or destruction actions. For example, one of the found open MODBUS system had web administration page left with default admin password.

The aim of the research is to create cyber incidents early detection model based on data package and OS-based system analysis that enables data streams to be identified as cyber incident at initial attacks moments. Research team were tasked for this work (project number REP-15014) and financed by Research Council of Lithuania.

2. RESEARCH STRUCTURE

To achieve above mentioned goal requires performing parameters analysis at network traffic and system activity layers. Our way of data aggregation and analysis using statistical and mathematical model developed leads to measuring anomalies that in turn reflects the cyber attack detection. The structure of the research is presented at fig. 1. Data streams could be analyzed at the perimeter of ICS and if data streams are identified as forming cyber attacks, these package structure and content analysis results are captured as characterizing the set of cyber attacks parameters. Similar method used for OS-based system (for example ICS) activities analyze and anomalies detection that reflects possible cyber attack.

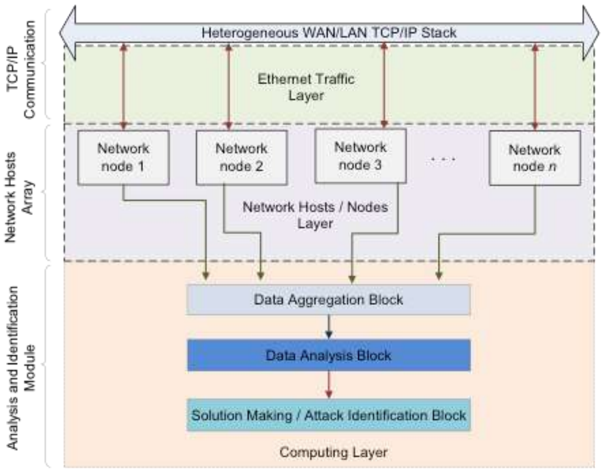

Figure 1: Structure of the experiment

Network intrusion detection systems (NIDS) are widely used in practice. For development of NIDS, anomaly or misuse detection models are used but most of commercial products so far are based on misuse detectors. Article [3] argues in details that the task of finding attacks is harder for the intrusion detection community to employ machine learning (anomaly detection) effectively. However, there are extensive academic researches still undergoing that use anomaly based intrusion detection approach as well as this article looks for its own way in cyber attacks detection by anomalies in network traffic.

3. DATA COLLECTION FOR MEASURING NETWORK TRAFFIC

Our system consists of data gathering agents at nodes of an enterprise network. Gathered network data is used for anomaly detection. Collection subsystem generates new data sample each 5 minutes. Incoming and outgoing network traffic is used as input data. Features extracted from data are listed below, in Table 1. The chose features are quite natural – they are close to raw data but at the same time some preprocessing is already done on the raw data.

Table 1: Data Types Collected

Nr.	Parameters
1	Number of unique TCP addresses.
2	Number of unique UDP addresses.
3	Number of unique ICMP addresses.
4	Number of different TCP ports (source and destination).
5	Number of different UDP ports (source and destination).
6	Average TCP throughput.
7	Average UDP throughput.
8	Average ICMP throughput.
9	Average TCP packet size.
10	Average UDP packet size.
11	Average ICMP packet size.

It can be seen that all gathered parameters are represented as numerical data. Let's denote the raw collected data as p_{ij} where *i* vary with time and takes integer values and $j \in 0 \ldots 10$ (as there are 11 parameters). In other words at time instance *i* collected raw data is a vector of length 11. If features are denoted using one index, then that index means time and value itself is a vector of length 11.

In fig. 2 below is shown a typical sequence of collected data (abscissa is marked using time values).

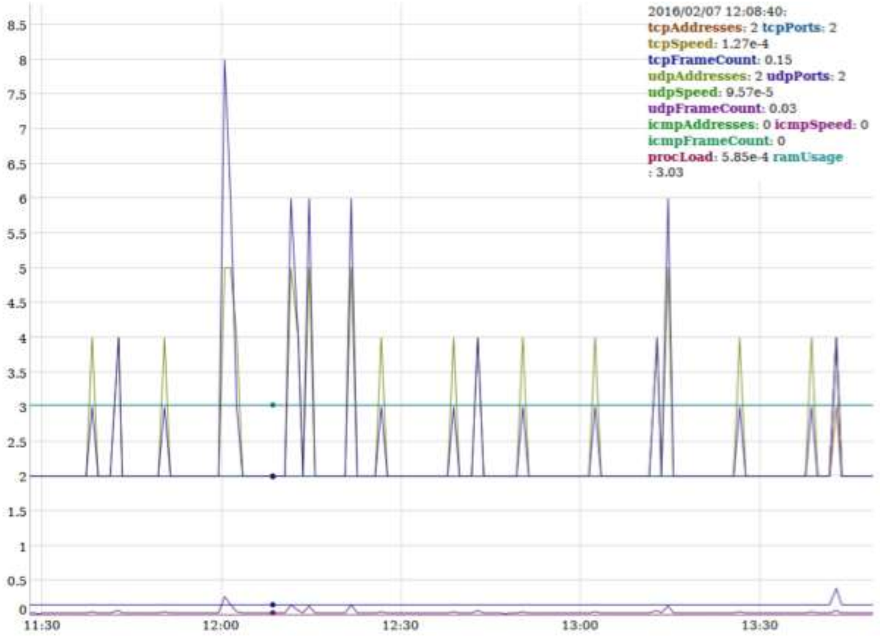

Figure 2: A typical sequence of collected data

4. APPROACH FOR ATTACKS DETECTION USING NETWORK TRAFFIC ANALYSIS

Majority of papers out there [4] uses labeled training data. Some models use data from DARPA study [5] that is somewhat outdated. Other solutions that investigated modern intrusion detection problems are reviewed in [6, 7]. One of such problems in real-time systems is data labeling – it is very time consuming process and even unrealistic in some real world situations to obtain enough “attack/non attack” data. In practice, real attacks always mutate and the nature of attack depends on many factors that necessary to measure. Because of this, labeled data may be suitable only at the sites where they were collected. Thus we propose self-learning system that uses unlabeled data for its training. The only assumption for the data we made is that anomaly is taking place only in small percentage of data. There could also be no cyber attacks in training data and the proposed algorithms should detect anomalies successfully if in traffic attacks occur.

The proposed approach for anomalies detection consists of three processing parts. Each part is essential for good system performance. These parts may be interleaved in time. Of course, data collection phase must precede the other parts. Phase one is data collection $j \in 0 \ldots 10$ phase. During this phase raw data is recorded using *libpcap (www.tcpdump.org)* library. The recorded data is also processed to obtain the above-described features. All data is also stored in SQL database. In our system new data sample (vector of length 11) is obtained each minute.

The second phase is attack detection system training phase. During this phase representation of normal behavior is created using data from phase one. The method used for model construction is classical k-means algorithm [10]. This is a very well known algorithm that is successful in many applications. We do not improve on this algorithm and the only particularity of the algorithm is Manhattan distance function used. Human operator who analyses similar graphs as shown in Fig, 2, selects the data for the training phase of algorithm. Operator must choose time interval that is anomaly free. The result of this training stage is a set of centroids that represents normal system behavior. Let us denote the centroids c_i, where $i \in 1 \ldots N$ and *N* is selected number of centroids (system operator is also responsible for this, but default value of 100 may be used).

The third phase, which may be called operational phase, is responsible for anomaly detection during everyday system operation. This phase operated in conjunction with data collection phase. When new feature vector *p* arrives, minimal distance to centroids is calculated. Mathematically this is expressed as:

$$d = \min_{i}(dist(c_i, p)).$$

For more precise description of our proposed early staged cyber attacks detection method we present model algorithm scheme in Fig. 4. Important to mention that each of these modes collects networks traffic data and extracts certain parameters for anomalies detection. In training mode operator selects time interval when no attack is present. Then *k*-means model out of the selected data is constructed. The constructed data is further used in attack detection during normal system operation mode.

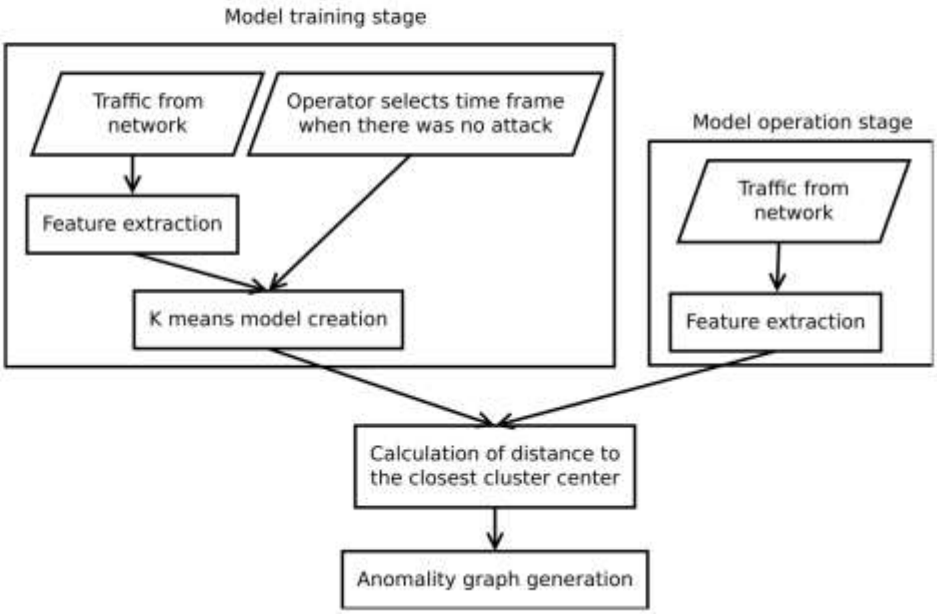

Figure 3: Algorithm of processes of anomalies detection in traffic

This distance is a measure of anomaly. Small value of distance signals means no anomaly and high value signals means that system anomaly behavior is suspected. We draw a graph that shows how *d* varies over time (we get one point on a graph each minute). An example of such graphs is shown in Fig. 5 and 7. It can be seen that some points have much higher values than others do. By suitable selection of the threshold, cyber attack can be successfully detected and we proved that by cyber attacks simulation exercises described in next section.

5. CYBER-ATTACKS SIMULATION EXERCISE

In order to test our systems functions and ability to identify network threats (to prove hypothesis that k-means algorithm (with Manhattan distance function) is suitable for cyber-attacks identification) was decided to install agent to a node of an enterprise network. A node has been chosen server with running apache service (open default http 80 port) WEB server. Doesn't matter which port has been chosen, because data could be collected through all open ports. Firstly, system was collecting raw data shown in Fig. 4 for several days.

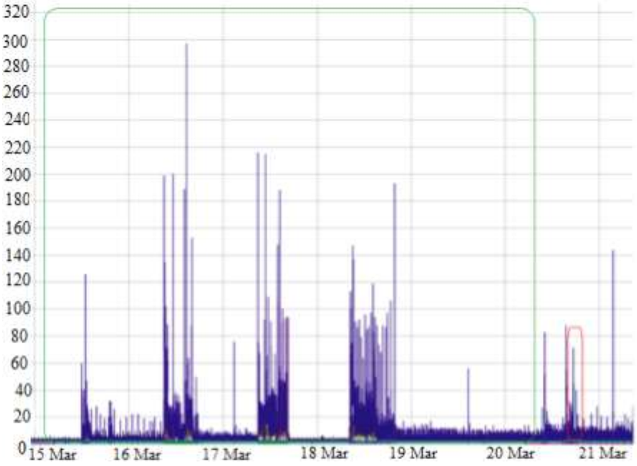

Figure 4: Snapshot of collected raw data, red oval – nmap attack

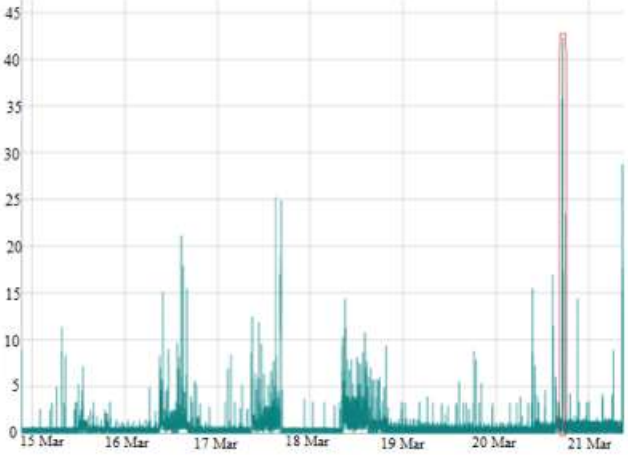

Figure 5: Anomaly score variation over time, red oval - is nmap attack

Screenshot of raw data shows stochastic activity on measured network. Then was planned to make a cyber-attack against the node. Cyber-attack instrument were chosen NMAPv6.40 tool (https://nmap.org) that suppose to use for networking scanning. Command requesting status of port 80 and operating system version were initiated. Collected raw data of this activity is shown in Fig. 4. Screenshot of raw data visually shows no signs of suspicious activity.

It is important to mention the system-training phase we did for fixing common network behaviors. Training time in Fig. 4 market by green line. For system training in this case, 100 centroids were used.

It can be seen that anomaly detecting graph which is plotted using the above presented algorithm (Fig. 5), shows clear signs of anomaly.

Similar experiment was performed using DDOS attack imitation on the system. In this case raw data graph (Fig. 6) didn't show clear anomaly indicators but our proposed algorithm indicates visible suspicious activity (Fig. 7).

By suitably selecting detection threshold, we were able to detect DDOS attacks 99.2% of the time. These exercises prove that k-means algorithm is suitable for network data analysis and anomaly detection.

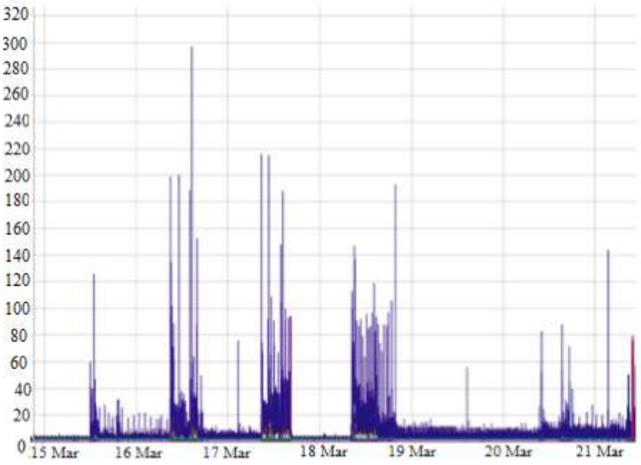

Figure 6: Snapshot of collected raw data, red oval – DDOS attack

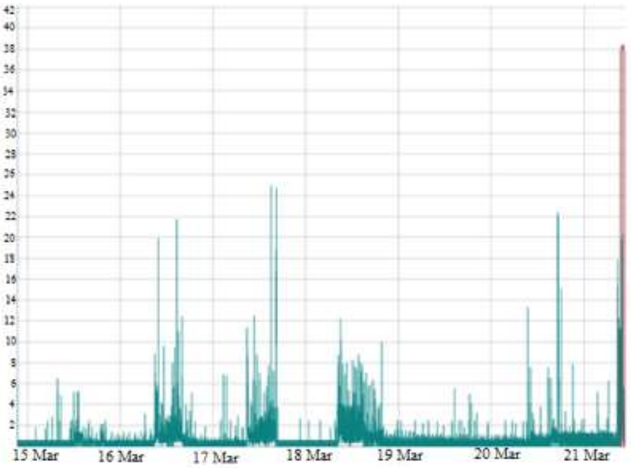

Figure 7: Anomaly score variation over time, red oval - DDOS attack

6. NEURO-NETWORK ALGORITHM BASED DETECTION

In recent years, deep artificial neural networks (including recurrent ones) have won numerous contests in pattern recognition and machine learning.

Shallow and deep learners are distinguished by the depth of their credit assignment paths, which are chains of possibly learnable, causal links between actions and effects. A standard neural network (NN) consists of many simple, connected processors called neurons, each producing a sequence of real-valued activations. Input neurons get activated through sensors perceiving the environment, other neurons get activated through weighted connections from previously active neurons. Some neurons may influence the environment by triggering actions.

Learning or credit assignment is about finding weights that make the NN exhibit desired behavior, such as driving a car. Depending on the problem and how the neurons are connected, such behavior may require long causal chains of computational stages, where each stage transforms (often in a non-linear way) the aggregate activation of the network. Deep Learning is about accurately assigning credit across many such stages [8].

7. APPROACH OF ATTACKS DETECTION IN OS-BASED SYSTEMS

In many cases, network-based anomaly detection is a good choice for undergoing attacks (like DoS, DDoS, Network based Bruteforce, Mitm) that are mainly categorized as classical method. Therefore, attacks are getting dedicated for data manipulation (stealth, encryption or destruction) that requires deeper analysis of processes that are happening in the system. In this chapter we did the approach of detecting attacks in their early stages using data from Operating System based network nodes (computers).

For this purpose, Neural-networks classification method was chosen and simulation was done on Linux based system. For simulation of the system testing structure shown in Fig. 8 were created. The structure contains of 3 basic elements: client host, encrypted network stack and analysis server. All the clients contain a set of data collection software that pushes the information through SSL Encrypted network stack. All of the data is gathered in the analysis server. A block diagram of mentioned software sets is shown in Figure 9.

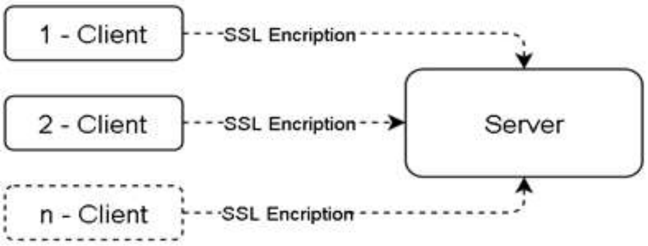

Figure 8: Structure of experimental environment

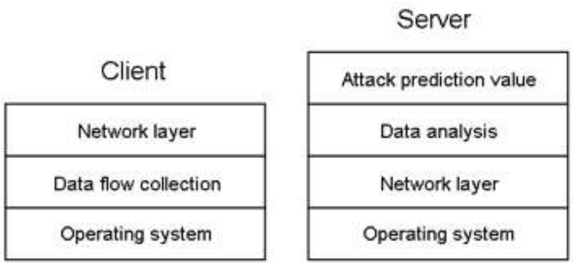

Figure 9: A block diagram of client and server hosts

Client side contains three blocks: Operating system; Data flow collection; Network layer. Server side contains additional analytical blocks: Data analysis Blok and Attack prediction block. Client side performs data collection (b), data output to server (a); Attack detection server consists of orientated Attack detection and Data analysis blocks. The algorithm schemes are shown in Fig. 10.

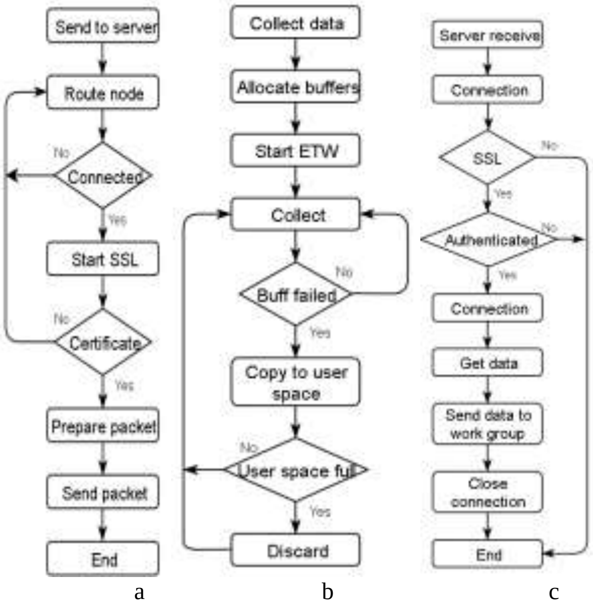

Figure 10: System workflow algorithms: (a) – Client side, network block (b) – Client side, data collection, (c) –Server side, data reception

Client side collects data from active system functional blocks: RAM, CPU, HDD I/O, Network and User activity. Starting the collection of data, buffers are allocated and ETW process is started, real-time collection procedures are launched. The information of active systematic blocks is stored in a buffer, copied to user space, when the user space is full, the data is discarded and collection continued.

Client side network block sends collected data to the analysis server. Firstly, the route node detection procedures are started, the connection is established and encrypted SSL connection is done. Data is transmitted via TCP stack, using additional CRC checks.

Server side, data reception block authenticates the session of SSL, receives data and sends it to the analysis and detection block. The algorithm of analysis block is shown in Fig. 11.

The analysis is stared after the data is received from the Client side node. Firstly, the data is written into an SQL DB and updated. All the data is transmitted through a Neuro-network classificatory, module, that is being trained recursively with the data taken from the database module.

For this simulation, classificator is being used for a Matlab Neuro-networks simulation package with reduced error rate. After the analyzed data is parsed, threat detection block is activated – the parameters of suspicious activity is compared to the real-time learning data based model results. If analysis shows that results are similar to the proprietary attack specifications, a control signal is sent to the attacks elimination logical block. In both cases, attacks data is sent to the Classificator for further training of the complex detection system.

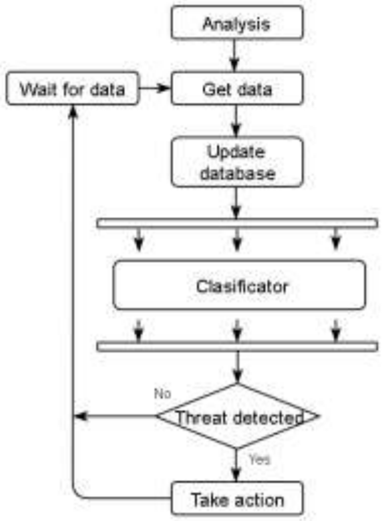

Figure 11: Algorithm of attack detection

The workflow of Calssificator is shown in Figure 12. The data array of the system analysis (x_0... x_n) is pushed into the cache buffer for temporary saving. After that, the training of the detection algorithm begins – algorithm is formed from three sections: Neural network training, Rule based training, Self trained data.

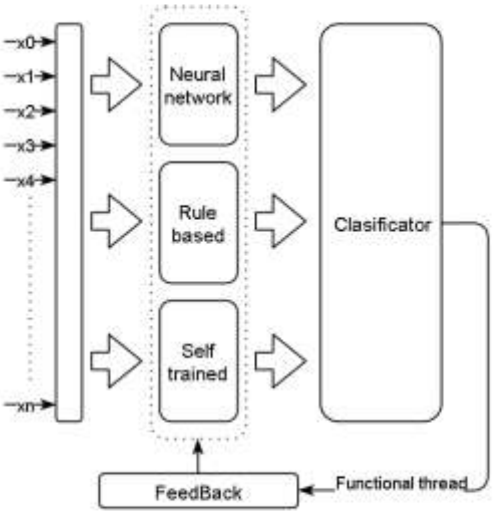

Figure 12: Workflow realisation of attacks detection using SVM

The data is transmitted into the Classificator, where the classification is parameters is done. In our structure, the Classificator logical block aggregates the trained data and indicates the anomalies with the highest probability. After the identification of the anomaly, the data is pushed again into the input vector for further training and system learning. Using this method, a disk – encryption attack was detected. Malware, containing disk encryption and network upload function was installed into a client host. It was selected to analyze 4 parameters – RAM activity, HDD I/O activity, CPU activity. The results of the detection are posted below.

8. SYSTEM ANALYSIS RESULTS

From our investigation we noticed that the neural networks properly classified the network traffic similar to the one presented during the learning phase. That's why they could be a good solution for detection of the attacks that were modified by an aggressor in order to cheat intrusion detection systems.

Unfortunately, the new attacks and the new normal traffic that is significantly different from the one presented in the training phase cannot be classified with sufficiently good accuracy. It is important to check if adding a new vector influences negatively on the classification accuracy.

For example in a situation when we add a new normal traffic to the learning data set and the number of not detected attacks increased significantly, the reason may be that the new vectors can be too much similar to an attack representation.

Even though SVMs are limited to making binary classifications, their superior properties of fast-training, scalability and generalization capability give them an advantage in the intrusion detection application. Finding cost-efficient ways to speed up or parallel the multiple runs of SVMs (to make multi-class identification) is also under investigation [9].

Analysis were done in four sections: analyzing RAM activity with Transferred data over network (shown in Fig. 13), Disk IO activity with Tranferred data (shown in Fig. 14), CPU activity with Transferred data (shown in Fig. 15) and Account activity with Tranferred data (shown in Fig. 16).

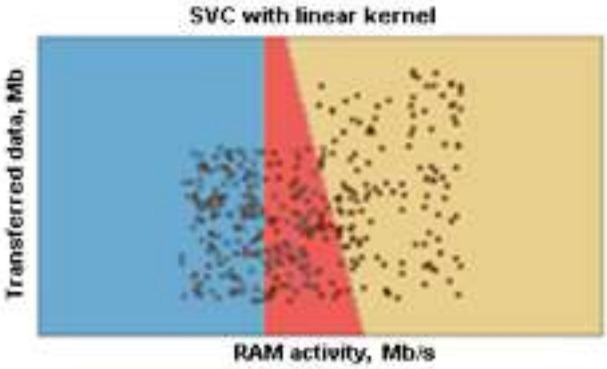

Figure 13: RAM activity with Transferred data

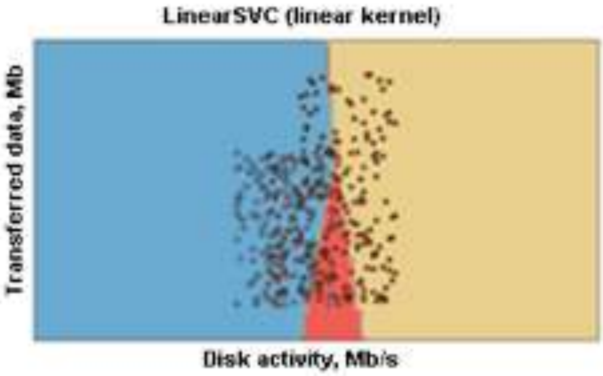

Figure 14: Disk I/O activity with transferred data

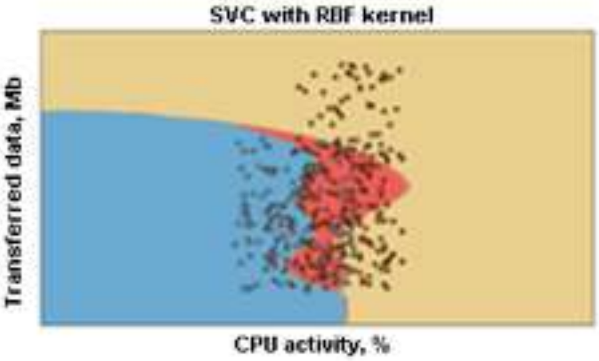

Figure 15: CPU activity with transferred data

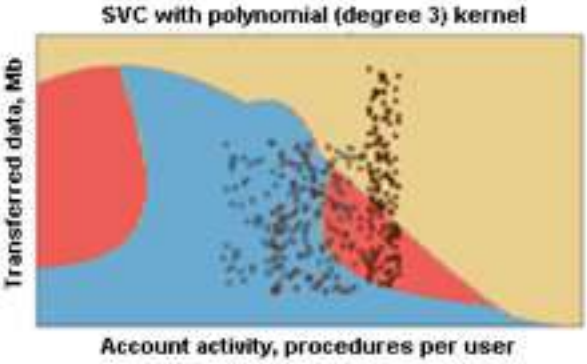

Figure 16: Account activity with transferred data

As can be seen in the neural simulation graph, anomaly is detected RAM activity with Transferred data (red segment). Furthermore, there are just small amount of points in the segment of both activities (Transferred data and Disk I/O), that concludes, that a big amount of data is needed to process an anomalies detection using Neural networks calculations. SVC with linear kernel method was used for detecting anomalies.

In Disk I/O activity with transferred graph can be seen, that anomalies are being detected in the red sector. *LinearSVC* method was used for detecting anomalies.

For identifying attack in CPU activity with transferred data was a SVC with RBF kernel used. It can be seen, that the form of red segment is different and non-linear (as in the Fig. 14 and Fig. 15). This method gives a slightly better error correction rate, but it rises the calculation time up to 15%. That concludes, that detecting real-time anomalies using SVC with RBF kernel is highly costly and hardly usable in practical applications and attack detection.

For account activity with transferred data analysis is used SVC with polynomial kernel. It can be seen, that there are two red segments of attack detection. That raises the error probability. In all of the graphs there are correlative points that indicate complex anomalies in a single system. Further researches should be taken to select the best fitting model for attacks detection.

9. CONCLUSIONS AND FURTHER WORKS

We presented anomaly detection system using k-means clustering algorithm and Neural-networks classification method. Our anomalies detection technique adopted for use at network perimeter measuring traffic data flows and internal system (like ICS infrastructure) measuring system behavior. Described in this article solutions implemented via developed software. Cyber attacks simulation exercises proved the concept and measured anomalies in traffic flows and system behavior.

Network traffic system does not need labeled training data for its functioning. System operator must assert to the system that some data was gathered using “calm” periods. Such data is used to create model of a normal behavior.

It was demonstrated that the system is capable of detecting cyber incidents like port scanning or DDOS attacks.

Further work will concentrate at extending the system functionality to increase cyber incidents detection accuracy. More testing and attacks simulation planned to be done. In next research work both anomalies detection mechanisms (network traffic and OS-based system) will be combined to perform data correlation. This should increase the precision of cyber incidents detection.

10. REFERENCES

1. Communications Regulatory Authority of the Republic of Lithuania. National incidents response team CERT-LT report for 2015. Source: https://www.cert.lt/doc/2015.pdf.
2. ENISA. Analysis of ICS-SCADA Cyber Security Maturity Levels in Critical Sectors. 2015. Source: https://www.enisa.europa.eu/activities/Resilience-and-CIIP/critical-infrastructure-and-services/scada-industrial-control-systems/maturity-levels/at_download/fullReport.
3. R. Sommer, V. Paxson. Outside the Closed World: On Using Machine Learning For Network Intrusion Detection. 2010. Publisher: IEEE.
4. V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. 2009. ACM Comput. Surv. 41, 3, Article 15 (July 2009), 58 pages. DOI = 10.1145/1541880.1541882 http://doi.acm.org/10.1145/1541880.1541882
5. C. F. Tsai, Y. F. Hsu, C. Lin, W. Lin.; Intrusion detection by machine learning: a review. Experts Systems with Applications, 36(10): 11994-12000, 2009
6. Cheng-Yuan Ho, Ying-Dar Lin, Yuan-Cheng Lai, I-Wei Chen, Fu-Yu Wang, and Wei-Hsuan Tai, „False Positives and Negatives from Real Traffic with Intrusion Detection/Prevention Systems“, International Journal of Future Computer and Communication, vol. 1, No. 2, August 2012, pp. 87 – 90.
7. Natesan, P., P. Balasubramanie, G. Gowrison, „Improving the Attack Detection Rate in Network Intrusion Detection using Adaboost Algorithm“, Journal of Computer Science 8 (7): 1041-1048, 2012 ISSN 1549-3636 2012 Science Publications.
8. James Cannady, Artificial Neural Networks for Misuse Detection.
9. Srinivas Mukkamala, Guadalupe Janoski, Andrew Sung, “Intrusion Detection: Support Vector Machines and Neural Networks”, 2002.
10. Tapas Kanungo, An Efficient k-Means Clustering Algorithm: Analysis and Implementation., IEEE transactions on pattern analysis and machine intelligence, VOL. 24, NO. 7, 2002.