Method of Early Staged Cyber Attacks Detection in IT and Telecommunication Networks

Saulius Japertas, Tautvydas Baksys
Department of Electronics Engineering, Kaunas University of Technology, Studentu St. 50–438, LT-51368 Kaunas, Lithuania
tautvydas.baksys@ktu.lt

***Abstract*—Increasing digitization together with the benefits has also brought a lot of problems related to the challenges in cyberspace. Due to the ongoing cyberattacks yearly increase, losses in sectors that are using Telecommunication and IT services are growing. The events of the past 10 years have shown that there are particularly dangerous incidents in the cyberspace, which are pre-planned, well-prepared and carried out by terrorist groups or even by some governments. Pre-planned cyber-attacks have some stages so it is possible to distinguish the early stages where attacks do not bring significant damage to data and information.**

This article examines the features of the attacks and their characteristics and is the first part of the study's generalization. There is proposed a method for early staged detection of such attacks using a number of the logical filters. Proposed methodology provides a network analysis structure, logical filter configuration and attack detection algorithms that enable the detection of network flow parameters that characterize potential attack vectors. The results of theoretical simulation have shown that proposed method is capable of determining early-staged cyberattacks.

In the next paper, the logical mathematical model, an estimation of the sensitivity of such method and assessment of the probability of each initial stage will be presented.

***Index Terms*—Intrusion; Prediction; Response system; Hidden Markov model.**

I. INTRODUCTION

Nowadays there are no doubts about the growth of threats in cyberspace. These threats can critically effect specifically targeted economic sectors (e.g., the Ukrainian power supply system in 2015), general economy (e.g., the Estonian attack in 2007), and political system (e.g., possible US Presidential Election breach in 2016).

Although there are a lot of works for reducing and evading threats in this space but the apparent extent of threats will only increase due to the imperfection, inappropriate use of traditionally used measures, or even the fact that measures are not used at all. As shown by Trustwave's 2016 Global Security Report, even 97 % of Web applications tested appeared to be sensitive to cyberattacks [1]. According to the UK Department of Business, Innovation and Skills 2015 Security Survey, 90 % of large organizations and 74 % of small organizations have been exposed to violations in the cyber security area [2].

Today cyberattacks against the information and telecommunications systems are organized by individuals (and groups of individuals) or even governmental organizations. In 2001, there have been made attempts to categorize potential offenders and define the damage that was created due to cyberattacks [3]. In essence, this categorization remains, except for the need to supplement it with public structures that also deal with cyberattacks. It can be assumed that such structures pose a particular threat to the cyberspace, that is formed from telecommunication networks and they suffer biggest losses [4]. Cyber-attacks set up by government or organized crime structures [5], including organized terrorist groups, aim to steal or damage the information (data) of certain institutions or governmental structures and affect these institutions economically and/or politically. In order to do this, attackers create attacks that are characterized as: Harmonized, Organized, Enormous, Regimented, Scrupulously Designed, Non Spontaneous or Ad hoc, Demanding Time and Resource [6]. A classical classification of cyberattacks was proposed in [6] (Fig. 1).

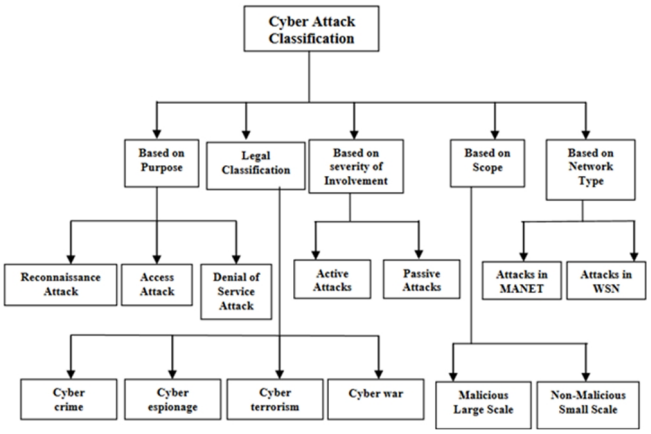

Fig. 1. Classical classification of cyber-attacks against IT and telecommunication systems [6].

For making scrupulously prepared cyberattacks (such attacks causes greatest losses), meaning they are not spontaneous, several stages are needed for an attack to form an active breach vector and become harmful. Various authors offer different stages, which may vary by number. The details of the attack stages are discussed in [7]. There are proposed 7 attack stages in this work: Reconnaissance;

Manuscript received 20 September, 2017; accepted 11 January, 2018.

Weaponization; Delivery; Exploitation; Installation; Command and Control; Action on Objectives. This model distinguishes two main stages (Fig. 2): left side of an attacks vector and right side of an attacks vector. If the stage moves to the right, it will be difficult to stop the cyberattack.

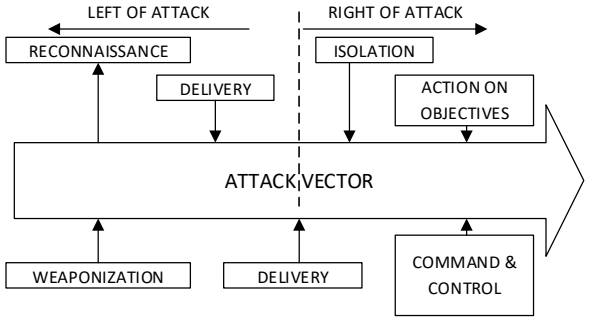

Fig. 2. Typical cyber-attack vector formed against IT and Telecommunication systems [7].

In the work [8] are proposed the 5-staged attack vector: Reconnaissance; Assessment and Strategy; Exploitation/ Invasion; Maintaining Access; Operations. Regardless of how many stages are distinguished, the importance of each stage is sufficiently large and the more thoroughly preparation is done in one stage, the more effective further stages can be realized.

It is understood that the importance of each stage is defined by those who are preparing a cyberattack [8]. In order to protect from cyber-attacks, there are known very well-organized, various hardware and software systems, incorporated into Telecommunication networks that can be divided into 3 groups [9]–[11]: Intrusion detection system (IDS); Intrusion prevention system (IPS); Intrusion Response System (IRS). Nowadays these systems acquire new features compared to traditional systems, given the use of new technologies for making the solutions more robust and effective: e.g. cloud computing for analysing data storage, GPUs and FPGAs enchasing analysis speed.

Intrusion Detection System (IDS) collects information from a computer or network (general Telecommunications or descriptive TCP/IP) and analyses the potential system or network security breaches identified for its purpose [12]. IDS are often used to spy on network packets, which helps better to understand what's happening on a particular network segment. The two main IDS priorities are: Host-based IDS (HIDS) and Network-based IDS (NIDS) [13].

The nature of IT and Telecommunications system and a network attack is different and varies [14]. E.g. system-based attacks are considered as: System Insider attack; User to root attacks; Attacks on a virtual machine (VM) or hypervisor; Network-based attacks: Flooding attack; Port scanning; Backdoor channel attacks. As it was mentioned earlier, for detecting such attacks IDSs are used; Intrusion Detection Techniques varies and are categorized as follows [15]: Signature-based Detection (SD); Anomaly-based Detection (AD); Stateful Protocol Analysis (SPA). The features of these methodologies are analysed in papers [15]–[18]. These techniques are based on various detection algorithms and models: Statistical; Data Mining Based Methods; Rule-based systems; Genetic algorithms; State transition-based; Expert-based; Petri Nets, etc., whose are investigated in papers [19]–[27]. A hybrid detection method is also used that combines several of the above methods [14], [28].

Despite the widespread use of IDS systems, they have a number of weaknesses. Major deficiencies in the NIDS include the inability to analyse encrypted traffic, late updates, time delay between attack start and warning, and the difficulty of processing data on a redundant network. HIDS deficiencies are identified as failure to recognize network scans, inefficiencies in DoS attacks [29]–[32]. Some IDSs can be relatively easily avoided (e.g., Anomaly-based or Signature based) [31], [33]. The paper [34] states that the result of using IDS is not always clear. It is also interesting that practically the same imperfections have existed for many years ([29] 2002 and [32] 2015 and [6] 2017), and even the proposed new methods (e.g., [30]) do not help to avoid them. The prevention system performs several traditional tasks, which have yielded relatively good results over a relatively long time.

However, in the current situation, the use of traditional IPS systems used in IT and Telecommunications becomes problematic for several reasons [35]: Latency: in-bound IPS requires inspection and blocking action on each network packet, which consumes cloud system resources and increases the detection latency; Resource Consumption: running the IDPS services usually consumes significant resources; Inflexible Network Reconfigurations: traditional IPS does not have network configuration features to reconfigure the virtual networking system and provide scrutinized traffic inspection and control.

The Intrusion Response System is used for responding to attackers' actions. There are two types of an IRS: Passive and Active IRS, depending on the type of response. If a system automatically takes measures leading to a response, system is called an active IRS, if it takes place in a notification or forms a response in a manual way, system is called a passive IRS [36], [37]. The Audit Expert System is currently widely used [38]. Nevertheless, despite all the advantages provided by such systems, they still have many deficiencies that are fully disclosed in works [36]–[40].

One of the bigger deficiencies noted by the experts is that such systems are susceptible to violations because they are relatively static (especially for the associative-based IRS). Other major deficiencies are the activation of such systems only when an incident is detected and a high number of false alarms, which directly depends on the quality of IDS [41]. There are more deficiencies but they are more related not to attack but to the healthy state of the system, which can be affected by the use or non-use of the IRS [41] or the use of appropriate hardware [3].

As the overview shows, tools and methods currently in place do not allow the effective control of threats in cyberspace. One of the reasons for such an ineffective fight is the fact that usually systems (IDS, IPS, and IRS) begins functioning only when the attack is already happening or even happened. Further reasons for relatively ineffective protection systems are the delay of the software updates and the ability to bypass or negatively impact protection systems functionality by exploiting their own vulnerabilities. The aim of this paper is to propose an algorithm and its method realization in order to determine the possibility of cyberattack against IT and Telecommunications systems at

its earliest stages when it can still be effectively stopped.

II. EARLY STAGED ATTACKS VECTOR

Currently, various types of cyberattacks can be distinguished (Fig. 3): targeted or random (classical) against IT and Telecommunications networks or against their services; Attacks can have different speeds – slow, lasting for months or fast, lasting up to several minutes for a quick one-time effect.

Accidental attacks are unoriented, do not have all above mentioned stages, and they are characterized by high speed and low specificity. In this type of attack, there is usually no attempt to collect information about the victim, its system and its network infrastructure. The purpose of such attacks is the rapid collapse of the infrastructure by classical force methods – DoS, DDoS, Bruteforce attacks, general-purpose computer viruses, or other malicious software a quick effect to take place.

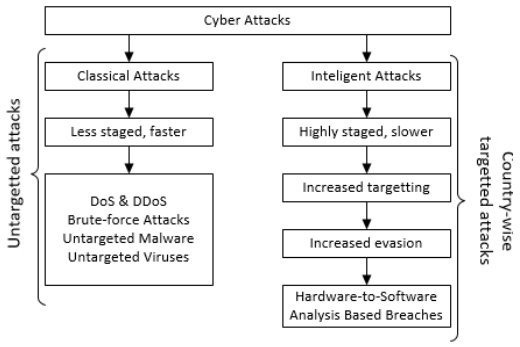

Fig. 3. Proposed cyberattacks classification.

The most dangerous are targeted attacks. They are well-organized and planned in advance. In this case, attacker can evaluate all of the victim's weaknesses and the consequences of the attack would be extremely harmful. Such attacks may take place on orders from governmental or major terrorist groups, focusing specifically on the take-over and destruction of system data or infrastructure. The main purpose of these attackers is to get the user's access to the system, so their attack vectors are directed to obtaining user rights in the system, exploiting system software vulnerabilities.

Such attacks are done remotely using a network stack. This work will highlight the nuances of network and software, allowing to detect attacks at potentially earlier times, leading to early stage detection and allow predicting cyberattacks as early as possible.

First of all, if the attack is purposefully and planned in advance, it is needed to define the stages that the attacker has to pass. The literature [1]–[6] mentions seven stages of a cyberattack, the combination of which is a fully realized attack vector. Attack stages reflect the harm done by the attacker against the victim: the early stages include processes for data collection, target tracking and attack infrastructure. In the middle stages, information from the early stages is used and actions are taken that weaken the victims' system (e.g., implementing a malicious code or process before the system, exploiting its vulnerabilities). This results in access to the system. Thereafter, the late-stage attack processes follow: direct system take-over, specific data capture, or infrastructure removal procedures. An attack vector for such a standard scenario is shown in Fig. 4.

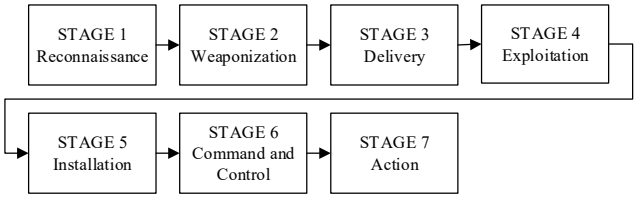

Fig. 4. Standard cyber attack stages [5].

In our opinion, the attack vector is sufficiently accurate to define the processes of the on-going attacks and bind them to the stages, but based on the experience (lessons learned), we suggest extending the attack vector to two stages: Social Engineering (STAGE 0) and the insertion of Evasion between Exploitation (STAGE 4) and Installation (STAGE5) (Fig. 5).

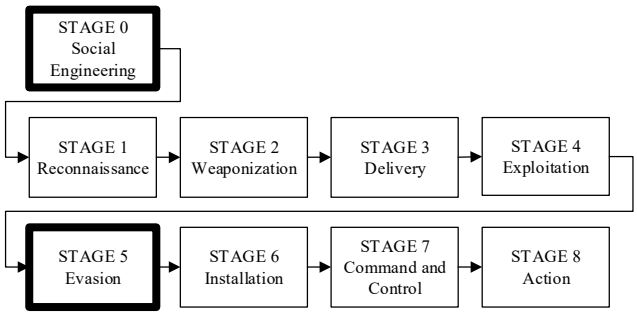

Fig. 5. Offered attack vector with additional stages.

The appearance of STAGE 0 is based on the fact that in planned and targeted cyberattacks, attackers seek to gather as much information as possible, so it is possible to collect information from the social environment or the sources close to them before the use of technical means. It is necessary to note that STAGE0 will not be further considered in this work, as this is not a direct action in the cyberspace, but an operation in the wider informational area. Attack vector is extended by the Evasion STAGE5 due to the natural attacker's need to be unnoticed or undetected. Such a fact is motivated by a more comfortable attacking position: Masked attack is less detectable, giving attacker more time to accomplish it; and Undetected infiltrating into a system to collect data (e.g., hiding as a background process to start a reverse tunnel connection for data upload). Figure 6 shows a combination of attack stages (processes) with Actions (separate Action will be discussed in Section III).

In this work early stages of an attack vector are analysed, i.e. from Stage 1 to Stage 3 inclusive (Fig. 5). The first stage of the attack (RECCONNAISANCE) is formed by three Actions: PORT SCAN, HOST SCAN and SYSTEM VERSION SCAN. PORT SCAN is a scanning of network port ports using a SYN request. HOST SCAN is a scan of nodes in the system and obtaining their IP addresses. VERSION SCAN is a service version of the system. Actions, described in Fig. 7, may be time-based, parallel, occur periodically or non-periodically, but all the needed actions are required for from a stage.

It is considered, that these factors create the RECCONAISANCE process (STAGE 1). This attack stage is characterized by poor systemic intervention and used for

collecting general data on the victim's system. WEAPONIZATION (STAGE 2) is formed from a repetitive System and Service Version Action, along with Services Stress Tests Actions. Services Stress Tests Action performs over-loaded system processes remotely. This is done in order to: obtain information about an increase in the response times of potentially vulnerable processes; by loading the system with an atypical flow, it is expected to summon the destabilization of the processes that are in process, to exploit it and create a new vulnerability that could potentially be exploited to deepen the attack vector.

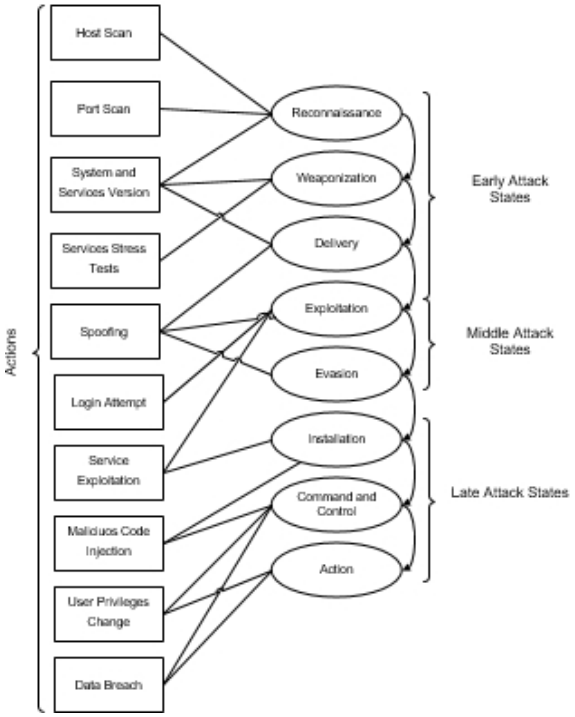

Fig. 6. Actions impact of the attack stages.

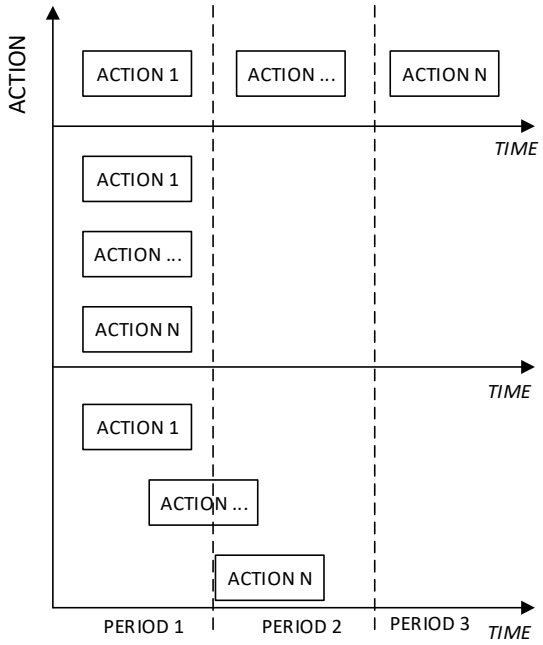

Fig. 7. Actions configuration in a time-lapse.

Delivery (STAGE 3) is the stage where the first parts of the malicious code/software or queries are sent from the victim's information about the victim's infrastructure. It is realized in several ways: Sending a victim malicious code incorporated to an ordinary file (e.g. a document); Sending it as an application; Sending through vulnerable software. The whole purpose of all this is to install a malicious code or integrate attackers structure into the victims' system (or a network). That allows attacker to directly affecting and interfere with victims provided services. This stage consists of 2 Actions: Version Check and Spoofing. Spoofing Action consists of several factors: Making a good program with hazardous code into a damaging one; Modification of the network packets in order to take information from victims' network stack or combine network traffic so that the victim can potentially easily install malicious software.

Early staged detection essentially ends with this step in our proposed model. Otherwise, because of continuing attack vector, tangible damage starts directly interfering with system and network work. It is necessary to detect these processes until they reach the 4th stage (STAGE 4 – Exploitation).

Processes and ACTIONS can be registered by monitoring the network stack and system behaviour. Observation results can exclude the features inherent in these ongoing processes and apply them flexibly to detect system anomalies and recognize ACTIONS in their beginning.

Attacks in the different stages consist of characteristics and features of the attack process or processes. Each of the processes mentioned in the vector has its own technical characteristics. In this work, three characteristic groups are distinguished which allow to characterize the ongoing process: Physical network stack parameters; Logical parameters of the system being attacked; Complex Network stack flags parameters.

An analysis of these parameters is presented in the next section.

III. PROPOSED DETECTION STRUCTURE

In this section general structure of attack detection (Fig. 8), attacks parameters detection and analysis is discussed. The analysed flow consists of two streams: normal (non-harmful) flow and from the attack stream. Normal flow is the traffic generated by users between their infrastructure and service nodes. Attack traffic is generated by the attacker's infrastructure services and their activity. The IT and Telecommunication services traffic flow body is formed on an external network that is connected to the internal network.

An internal network with an external one is connected using a router. The internal network stack comprises: Network Monitoring Units (Hardware Network Probe); Computer hosts; Data Analysis Server.

Our approach offers to install additional package analysis software that realizes the function of the filters in a real time (Fig. 12) to a router, connecting external network with internal, i.e. allowing to analyse packets that passes certain filters. Filters allow to exclude the characteristics of the attack from the flow and system parameters with the corresponding characteristics. Basically, these are logical filters that analyse the relevant network traffic parameters (such as DST IP, SRC IP, ABS TIME).

Logical filter structure is show in Fig. 9 (it will be detailed below). The results of network analysis are sent to the Data Analysis Server inside the network, where a

general analysis of these parameters is performed and the probability of a possible attack stage is determined.

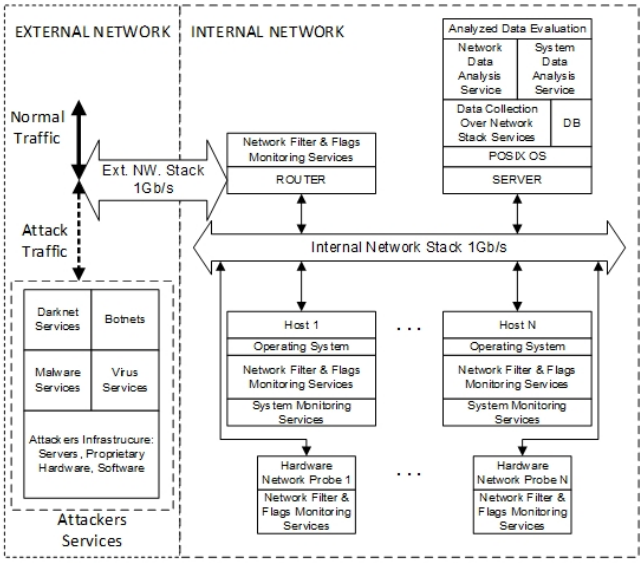

Fig. 8. Structure of the attacks detection system.

Router performs a detection of the parameters of the incoming and outgoing traffic, and only certain internal traffic parameters are detected in the network monitoring units. Network analytics devices (filters) are hardware nodes that use ARM processor architecture, POSIX OS, and our analytics software. The set of filter settings in the router and network analytics nodes are the same.

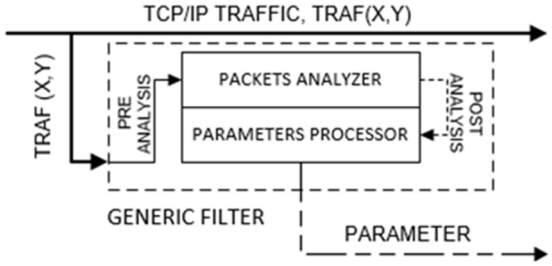

Fig. 9. Structure of a generic filter.

Separation of network parameters and system behaviour parameters is carried out at service nodes. Service nodes are standard computerized equipment, which are additionally equipped with a process analysis and key parameter detection module and a network parameter detection block. Network process analysis is done at level RING0. An analogous structure is also applied to the POSIX system; the detection of parameters is done at the kernel level.

Network nodes send processed parameters to the Data Analysis Server, which includes blocks for data collection, system processes analysis and data analysis; also this block evaluates the collected parameters. Data Analysis Server manages the monitoring software and hardware management with the RPC protocol using the GRPC framework. The processed network and system parameters are sent to the Data Analysis Server, which defines the attack stage.

The server performs a three-step analysis: Analysis of TRAF (XB_{RX}, XB_{TX}) data sent by the router; Analysis of the information is sent by TRAF (X_{RX}, X_{TX}) from the network monitoring nodes; Analysis of information is sent from hosts by TRAF (X, Y).

This analysis is carried out quantitatively and qualitatively. Based on the analysis results an attack stage with an appropriate probability is determined. Figure 10 provides a general block diagram of the system operation.

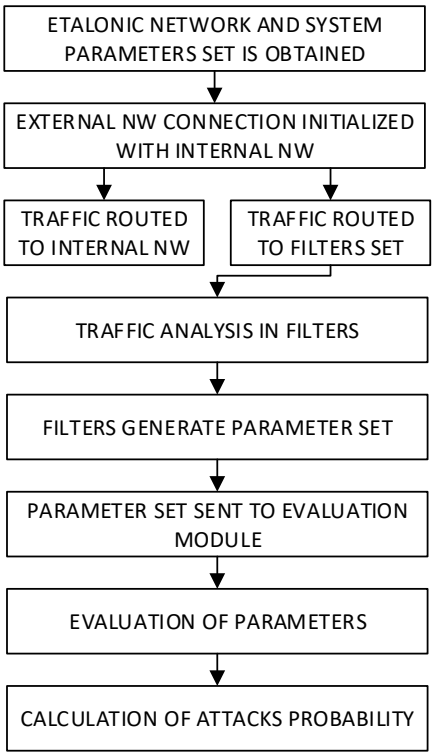

Fig. 10. Proposed detection workflow.

Each of the monitoring blocks has parameter exclusion filters that separate the network flow or the system parameters are sent to the estimation block.

An Element Filter (Fig. 9) consists of two blocks: a packet analysis block and a parameter processor. The TRAF (X, Y) input into the filter is analysed on the packet level, which results in a packet parameter (e.g., DST IP). An excluded parameter is passed to the internal parameter processor, which, according to the conditions provided, forms the final parameter.

Proposed detection structure is based on the network resource economy: the desire to create as little as possible a minimum of service traffic between nodes without overloading the network overhead information. These parameters are sent to the evaluation block.

The schematic diagram of block containing filters is shown in Fig. 11. Parameters are collected in the EVALUATOR block, which performs the analysis and potentially determines the potential attacks that form the attack processes in the described stages. In the proposed model, TRAF (X, Y) traffic is unmodified because there is a need to maintain the traffic of the system without affecting the system services reliability. It is shown in the Fig. 11 as a separate line (UNMODIFIED TRAFFIC X AND SYSTEM STATUS Y). To ensure early detection, different types of filters are used: Network parameters, NF (shown in circle); System parameters, SF (in the graph depicted in square brackets); Complex Network stack flags, LF filters (shown in hexagon graph). Proposed parameter analysis filters are presented in Table I.

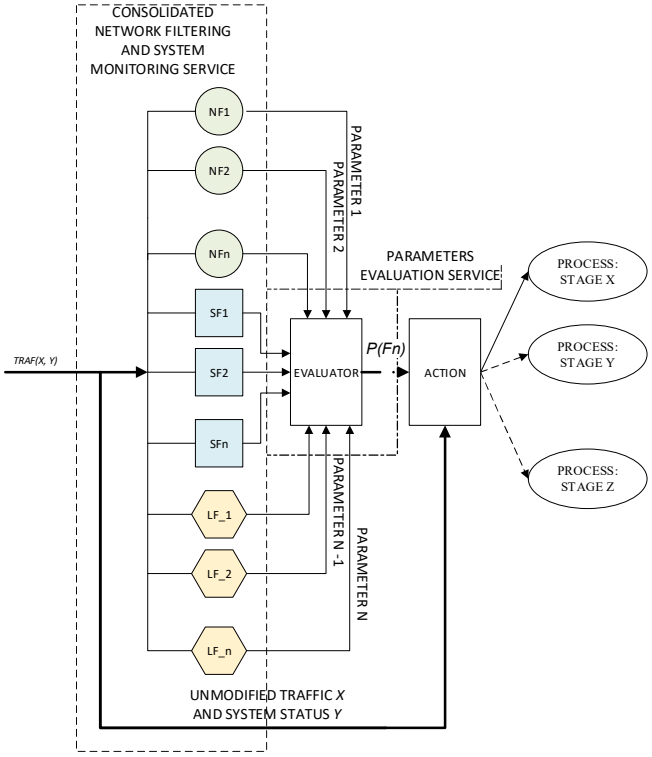

Fig. 11. Filters and actions chain.

31 filters were created, which are used to analyse the processes in the network and in the system. The collected information identifies features of ongoing attacks. In this work the composed sample contains 12 network analysing parameters, denoted by the prefix NF, numbered from 1 to 12, i.e. NF1 ... NF12.

Selected filter flow parameters are IP, IP COUNT, PORT NUMBER, PORT DISTRIBUTION, PACKET COUNT, STACK BYTES, PACKETS A-> B, PACKETS B-> A, BYTES A-> B, BYTES B-> A, DURATION, ABSOLUTE TIME. I.e., IP (NF 1) defines the attacker's IP address; IP Count (NF 2) defines IP address repetition; PORT NUMBER (NF 3) defines the port number to which the information is sent; PORT DISTRIBUTION (NF 4) defines the distribution of ports according to the token information, PACKET COUNT (NF 5) generically analyses the number of packets in the network tract; STACK BYTES (NF 6) determines the amount of data transferred in the session; PACKETS A-> B (NF 7) determines the number of packets sent from the attacker to the victim; PACKETS B-> A (BYFTES A-> B (NF 9)) defines the amount of data (bytes) transmitted from the attacker to the victim; BYTES B-> A (NF 10) determines the amount of data transferred from the BYTES A-> B (NF 9) to the attacker the victim's attacker; DURATION (NF 11) sets the duration of the active single session between the attacker and the victim; ABSOLUTE TIME (NF 12) sets the absolute start time for the session. These parameters are assigned numbers from 1 to 12. The given numbers correspond to the numbering of the filters which are shown in Table II.

The logical flow parameters of the system are denoted as SF*xx*. The PERIPHERAL STATUS (SF1) indicates whether the status of the peripheral device has changed; UNLISTED PROCESS (SF2) indicates what processes in the system are in the list; FLAWLESS USER LOGIN (SF3) indicates whether an unexpected user connection was attempted or a password or unconnected connection was attempted; SUSPICIUOS TIME (SF4) specifies system clock times which average is significantly deviating from the standard user connection time; DISK ACTIVITY (SF5) indicates whether the increased activity of the disk array is detected comparing with an average value; PORT BINDING (SF6) indicates whether the port is bound to port. Parameters are numbered from 13 to 18.

TABLE I. ATTACKS PROPERTIES FILTER.

No.	Filter Name	Filtering Parameter
1	NF1	IP
2	NF2	IP COUNT
3	NF3	PORT NUMBER
4	NF4	PORT DISTRIBUTION
5	NF5	PACKET COUNT
6	NF6	STACK BYTES
7	NF7	PACKETS A-> B
8	NF8	PACKETS B->A
9	NF9	BYTES A->B
10	NF10	BYTES B->A
11	NF11	DURATION
12	NF12	ABSOLUTE TIME
13	SF1	PERIPHERAL STATUS
14	SF2	UNLISTED PROCESS
15	SF3	FLAWLESS USER LOGIN
16	SF4	SUSPICIUOS TIME
17	SF5	DISK ACTIVITY
18	SF6	PORT BINDING
19	LF_FIN	FIN FLAG
20	LF_SYN	SYN FLAG
21	LF_TCP_CONN()	TCP_CONN() FLAG
22	LF_NULL	NULL FLAG
23	LF_PING	ICMP FLAG
24	LF_VERSION_DETECTION	VER FLAG
25	LF_UDP_SCAN	UDP FLAG
26	LF_BULK_SCAN	BULK FLAG
27	LF_WINDOWS_SCAN	WIN_SCAN FLAG
28	LF_RPC_SCAN	RPC FLAG
29	LF_LIST_SCAN	LST FLAG
30	LF_IDLE_SCAN	IDL FLAG
31	LF_FTP_BOUNCE	BOUNCE FLAG

The logical network setup consists of 13 parameters denoted LF*xx*. LF_FIN refers to packet's FIN flag; LF_SYN refers to packet's SYN flag; LF_TCP_CONN () refers to TCP Connection request; LF_NULL indicates NULL flag; LF_PING refers to ICMP request; LF_VERSION_DETECTION refers to VERSION flag; LF_UDP_SCAN refers to UDP request; LF_BULK_SCAN refers to random request; LF_WINDOWS_SCAN refers to versions of Windows query; LF_RPC_SCAN specifies a request to identify the RPC protocol; LF_LIST_SCAN specifies a query that gives a list from the previous query vector; LF_IDLE_SCAN specifies an IDLE process request; LF_FTP_BOUNCE specifies an FTP service request. These parameters are numbered from 19 to 31.

In this work the combinations of these parameters to determine the traffic anomalies is analysed. These parameters are analysed by filters and sent to the Data Analysis Server to evaluate the potential attacks stage and calculate the probability of attack to advance. Network traffic parameters and system status parameters from service nodes are consolidated using the System Data Analysis Service, located at Data Analysis Server. The consolidation of 31 parameters allows to approach a valid set of different type parameters to perform an early-staged attacks detection.

Table II gives 7 Actions, which form the 3 stages, which are considered early. Each action setup uses the appropriate set of filters. Filters and their parameters are shown in the Table I. Active filter number in Table II matches the filter number in Table I.

TABLE II. FILTER PARAMETERS CLASSIFICATION IN EARLY STAGES.

Active filter No.	ACTION						
	HOST SCAN, HS	PORT SCAN, PS	SYSTEM AND SERVICES VERSION, SSV	SERVICES STRESS TESTS, STT	SPOOFING, SP	LOGIN ATTEMPT, LA	SERVICE EXPLOITATION, SE
1	+	+	+	+	+	+	+
2	+	+	+	+	+	+	+
3		+					
4		+					
5			+		+	+	+
6				+			+
7				+			
8				+			
9		+					
10		+					
11					+	+	
12					+	+	
13							
14				+			+
15				+		+	+
16				+		+	+
17				+			+
18			+	+	+	+	+
19		+					
20		+					
21			+	+	+	+	+
22			+		+		+
23	+			+		+	+
24			+				
25			+				
26			+				
27			+				
28			+				
29			+				
30	+						+
31				+			+

In this work there are 7 factors that form the attacks in the early stages: Host Scan, Port Scan, System and Services Version, Services Stress Tests, Spoofing, Login attempt, Service Exploitation.

The table of data and factors consists of the following parameters:

$$F_{HS} \in \{1,2,23,30\}, \quad (1)$$

$$F_{PS} \in \{1,2,3,4,9,10,19,20\}, \quad (2)$$

$$F_{SSV} \in \{1,2,5,18,21,22,24,25,26,27,28,29\}, \quad (3)$$

$$F_{STT} \in \{1,2,6,7,8,14,15,16,17,18,21,23,31\}, \quad (4)$$

$$F_{SP} \in \{1,2,5,11,12,18,21,22\}, \quad (5)$$

$$F_{LA} \in \{1,2,5,11,12,15,16,18,21,23\}, \quad (6)$$

$$F_{SE} \in \{1,2,5,6,14,15,16,17,18,21,22,23,30,31\}, \quad (7)$$

$$F_{Early\ Detection} = \sum\{F_{HS}, F_{PS}, F_{SSV}, F_{STT}, F_{SP}, F_{LA}, F_{SE}\}. \quad (8)$$

It is seen that the parameters quantity of the actions, which form next stages of the attacks, increases.

The configuration of these filters allows to create a setup of the detection, the result of which is determined by the logical circuits. Theoretical simulation results are presented in Section IV.

IV. THEORETICAL SIMULATION RESULTS

In order to demonstrate performance of the method developed for attack detection in early stages the model of the logical circuit representing part of the algorithm was created and the simulation was carried out. A logical circuit simulation was done to illustrate the adequacy of our proposed method. The logical circuits operate on the sets of binary parameters. Seven types of the possible attack actions were determined, therefore, there were designed logical circuits for detecting each action (HS, PS, SSV, SST, SP, LA and SE) of the attack. As supposed, aggregated actions create a cyberattack. Therefore, a logical circuit is presented, that aggregates cyberattack actions, analyses the collected parameters and obtains a binary code that resembles attacks stage. Test simulations are also done to demonstrate the proposed method capabilities for detecting attacks in their early stages.

In such a way, every primary output indicates the presence of the described attack action. A logical circuit of Host Scan (HS) analysis uses four primary inputs NF1, NF2, LF_PING, LF_IDLE_SCAN and produces a primary output labelled as HS. The analytical form of logical circuit of Host Scan analysis is shown in (9)

$$HS = \left(NF1 \cdot NF2 \cdot LF\_PING \cdot LF\_IDLE\_SCAN\right). \quad (9)$$

Logical circuit schematic for Host Scan analysis is shown in Fig. 12.

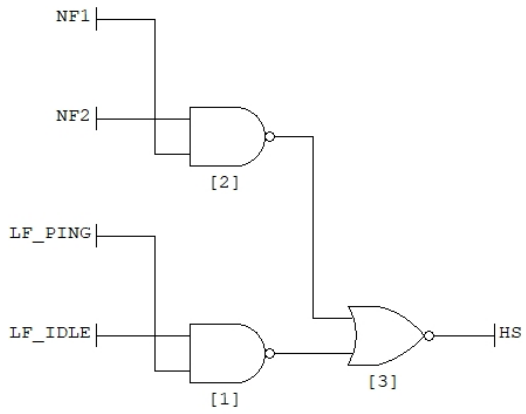

Fig. 12. Logic circuit for Host Scan detection.

Logical circuit for Host Scan detection contains a set of three logical gates, and values on the primary outputs of the logical circuits are combined and the attack factors are determined according to the obtained result. Logical circuits are built from logical conditions and if primary bit stream meets the requirements of the described conditions, primary output is assigned to a binary value of "1" (action detected), otherwise the binary value is "0" (action is not detected).

Algorithm of action detection using logical circuit is shown in Fig. 13.

The obtained values are aggregated in the decision

making logic circuit and output is obtained, showing which stage is ongoing, according to analysed parameters.

All of the logic detection circuits are designed in the analogical way and their principles of functioning are the same. Although that we have obtained satisfying results for all our logic circuits, for the space limiting issues, it was decided to publish two sets of logical circuits: a logical circuit for analysing SSV action and a logical circuit for all actions aggregation.

SSV logical circuit consist of 12 primary inputs: *NF1, NF2, NF5, SF6, LF_TCP_C, LF_NULL, LF_VER_D, LF_UDP_S, LF_BULK, LF_WIN, LF_RPC_S, LF_LIST* and a primary output SSV that obtains the result. These inputs are selected purposefully to resemble realistic network user stack. Analytical forms of SSV simulation vectors are shown in (10) and (11), where a primary input (e.g. NF1, SF6, LF_LIST) is described as member A where $A \in \{1\ldots12\}$, corresponds to the binary „1“, and a member A, where $A \in \{1\ldots12\}$, corresponds to the binary „0“. In the simulation, ***SSV_A*** (shown in side (a)) contains a bit stream of “11111111111”, that resembles the SSV actions described parameters and ***SSV_B*** (shown in side (b)) contains a randomized bit stream of “11001110011”. The vectors SSV_A and SSV_B are streamed to the designed SSV action detection logical circuit.

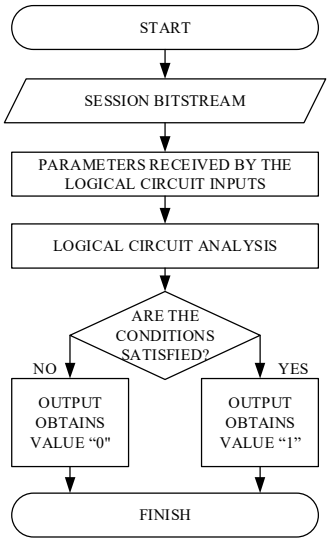

Fig. 13. Algorithm of logic circuit detection.

$$\begin{aligned} \textbf{\textit{SSV\_A}} = (&NF1 \cdot NF2 \cdot NF5 \cdot SF6 \cdot LF\_TCP\_C \cdot \\ &\cdot LF\_NULL \cdot LF\_VER\_D \cdot \\ &\cdot LF\_UDP\_S \cdot LF\_BULK \cdot LF\_WIN \cdot \\ &LF\_RPC\_SCAN \cdot LF\_LIST) = (11111111111), \end{aligned} \quad (10)$$

$$\begin{aligned} \textbf{\textit{SSV\_B}} = &NF1 \cdot NF2 \cdot \overline{NF5} \cdot SF6 \cdot \overline{LF\_TCP\_C} \cdot \\ &\cdot \overline{LF\_NULL} \cdot LF\_VER\_D \cdot \overline{LF\_UDP\_S} \cdot \\ &\cdot \overline{LF\_BULK} \cdot \overline{LF\_WIN} \cdot \overline{LF\_RPC\_SCAN} \cdot \\ &\cdot LF\_LIST) = (110100100001). \end{aligned} \quad (11)$$

The simulation results in a side by side comparison are shown in Fig. 14. SSV logic circuit consists of 15 logical gates. A red line shows, which part of the logic circuit is active and blue line shows the inactive part. As shown in the figure, side (b) has a blue primary output that means, the value “0” is obtained in a primary output and action “SSV” is not considered as active. Primary output obtained a value of zero because the primary input parameters, received from the testing vector SSV_B did not match the full criteria set (described in Table 2) for logical SSV action detection. In side (a) primary output results is shown in a red colour, output obtains value “1”, which describes the input bitstream as SSV action is positive. Different and correct evaluation of the results highlights the selectivity of our proposed method, showing user session parameters (e.g. IP, IP Count) can be used for attack action detection. Therefore, it is supposed, that aggregation of attack actions would lead to determine cyber attacks in early stages. Algorithm for the early staged attack detection, is shown in Fig. 15.

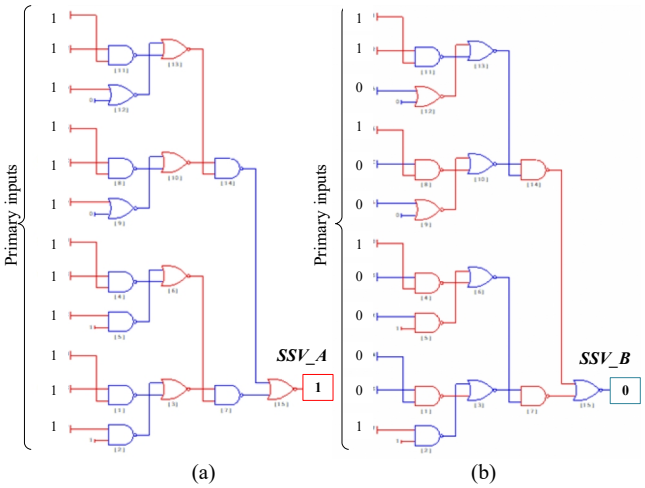

Fig. 14. Simulation results for System and Services Version logic circuit.

As algorithm describes, session parameters are delivered to three independent logic circuits: HS, PS and SSV for attack action detection. If all primary outputs of these circuits produce a value of “1”, Stage 1 (Recognissance) is obtained.

The session parameters are sent for further analysis to two independent logic circuits SSV and SST. Even though SSV action circuit exists in the first analysis step, we propose the second evaluation to minimize the possibility of faulty evaluation. If both, SSV and SST, logic circuit primary outputs produce values of “1”, Stage 2 (Weaponization) is obtained. If Stage 2 is obtained, last analysis step is performed and session parameters are delivered to two circuits: SSV and SP analysis. If all of the primary outputs produce value of “1”, Stage 3 (Delivery) is obtained and ongoing attack is determined. Further session parameter analysis would allow detecting later attack stages.

For the evaluation of the proposed algorithm, a logical circuit was synthesized and tested. The logical circuit used for aggregated analysis is shown in Fig. 16. The analysis is based on seven criteria, so there are seven primary inputs and three primary outputs to identify value of the attack. The logical circuit consists of 26 logical gates.

As in the SSV logic circuit analysis case, primary input described as A, where $A \in \{1\ldots7\}$, corresponds to the binary „1“, and a member A, where $A \in \{1\ldots7\}$, corresponds to the binary „0“. This form contains output logical functions, which consist of inputs, resembling attack actions: HS, PS, SSV, SST, SP, LA and SE. Primary output

S is a vector of "F21 ... F23" values

$$S=\begin{Bmatrix}F21\\F22\\F23\end{Bmatrix}=\left\{\overline{HS}\cdot\overline{PS}\cdot\overline{SSV}\cdot SP\cdot LA\cdot SE\right\}+$$
$$\begin{Bmatrix}0\\\overline{HS}\cdot\overline{PS}\cdot SSV\cdot\overline{SST}\cdot SP\cdot\overline{LA}\cdot SE\\\overline{HS}\cdot\overline{PS}\cdot SSV\cdot\overline{SST}\cdot SP\cdot\overline{LA}\cdot\overline{SE}\end{Bmatrix}. \quad (12)$$

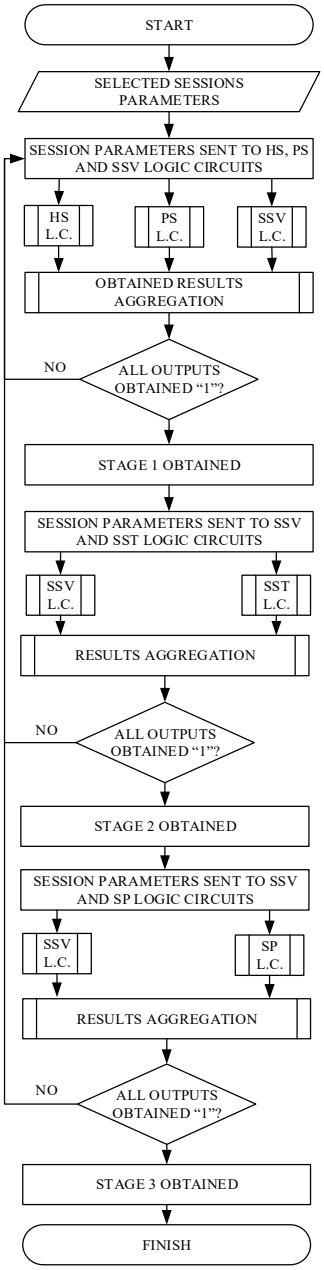

Fig. 15. Algorithm of early-staged cyber-attack detection using aggregative logic circuit.

A simulation of third stage was done (aggregating SSV and SP attack actions). Simulation vector is given in (13).

$$S=\begin{Bmatrix}F21\\F22\\F23\end{Bmatrix}=\left\{\overline{HS}\cdot\overline{PS}\cdot SSV\cdot\overline{SST}\cdot SP\cdot\overline{LA}\cdot\overline{SE}\right\}. \quad (13)$$

Generated bitstream of "0010100" was sent to the logic circuit. As it can be seen in Fig. 16, primary outputs of F23 and F22 became active (shown in a red line), F21 left inactive (shown in a blue line). The primary output codes in binary and attack stages are described in Table III.

As shown in simulated logic circuit, primary outputs obtain values: F21 = 0, F22 = 1 and F23 = 1. That corresponds to a primary output code of S = 011, resembling a third stage number, that is named as "Delivery".

According to the synthesized logic circuits and their simulated test results we are able to determine the early stage of the attack. This approach is a part of a large work, that is orientated to a near real-time cyberattack detection.

TABLE III. PRIMARY OUTPUT CODES AND ATTACK STAGES.

Primary output code in binary			Stage number	Stage name
F21	F22	F23		
0	0	0	-	-
0	0	1	1	Recognissance
0	1	0	2	Weaponization
0	1	1	3	Delivery
1	0	0	4	Exploitation
1	0	1	5	Evasion

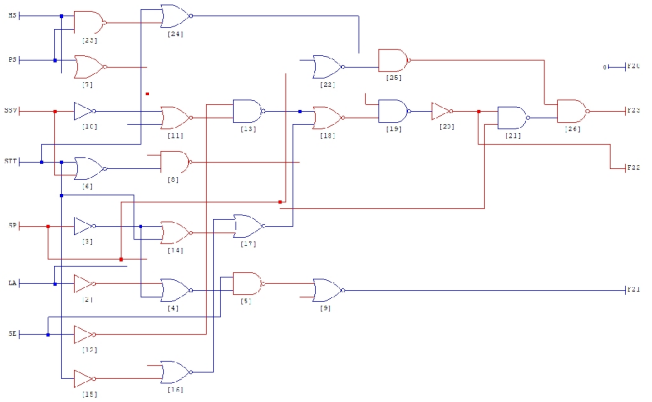

Fig. 16. Simulation results. Detecting third stage – primary outputs generate a binary code of "011".

V. CONCLUSIONS

This work proposes an early warning method for a possible cyber-attack in IT and Telecommunication networks. This method is based on the use of a set of 31 logical filters. The collected information identifies features of ongoing attacks.

There were proposed analytical aggregated expressions for the detection of threats caused by the early staged cyberattacks.

The set of parameters are analysed in a proposed early-staged attacks detection system. This system collects network and system nodes information and evaluates the possible attacks stage. For early-staged detection there were provided software implementations and hardware recommendations of the active nodes and a full Data Analysis Server logic setup;

The results of theoretical simulation have shown, that proposed method is capable of determining early-staged cyberattacks and the approach illustrates the possibility for practical method implementation;

In future works will be provided mathematical approach of the proposed method based on real computerized network data. The essence of this mathematical method would be to propose the probability of identifying the potential risk of each initial stage and evaluate the sensitivity of this model.

REFERENCES

[1] *Trustwave Global Security Report 2016*, Trustwave, 2016. [Online]. Available: http://bit.ly/2zLCcaz

[2] *2015 Information Security Breaches Survey,* UK Department of Business, Innovation and Skills, 2015. [Online]. Available: https://pwc.to/2AQVpHX

[3] S. M. Furnell, "The Problem of categorising cybercrime and cybercriminals", in *2nd Australian Information Warfare and Security Conf.*, 2001, pp. 29–36. [Online]. Available: http://bit.ly/2zKDe8S

[4] S. Starr, D. Kuehl, T. Pudas, "Perspectives on building a cyber force structure", in *Proc. Conf. on Cyber Conflict*, 2010, pp. 163–181. [Online]. Available: http://bit.ly/2hu476K

[5] R. M. Clark, S. Hakim, R. M. Clark, S. Hakim, "Protecting critical infrastructure at the state, provincial, and local level: issues in cyber-physical security", *Cyber-Physical Security*, pp. 1–17, 2017. DOI: 10.1007/978-3-319-32824-9_1

[6] M. Uma, G. Padmavathi, "A survey on various cyber attacks and their classification", *International Journal of Network Security*, vol. 15, no. 4, pp. 390–396, 2013. [Online]. Available: http://bit.ly/2hDNR70

[7] H. Al-Mohannadi, Q. Mirza, A. Namanya, I. Awan, A. Cullen, J. Disso, "Cyber-attack modeling analysis techniques: An overview", in *IEEE Int. Conf. Future Internet of Things and Cloud Workshops (FiCloudW 2016)*, 2016, pp. 69–76. DOI: 10.1109/W-FiCloud.2016.29.

[8] R. Singh, S. Lavania, P. Chaturvedi, N. Dhanda, "Intrusion prevention system using unique application indentification", *International Journal of Scientific & Engineering Research*, vol. 5, no. 8, pp. 610–613, 2014. [Online]. Available: http://bit.ly/2zN4IZw

[9] K. Gai, M. Qui, L. Tao, Y. Zhu, "Intrusion detection techniques for mobile cloud computing in heterogeneous 5G", *Security and Communication Networks*, vol. 9, no. 16, pp. 3049–3058, 2016. DOI: 10.1002/sec.1224.

[10] C. Kolias, G. Kambourakis, A. Stavrou, S. Gritzalis, "Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset", *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 184–208, 2016. DOI: 10.1109/COMST.2015.2402161.

[11] R. Shanker, A. K. Luhach, A. Sardar, "To enhance the security in wireless nodes using centralized and synchronized IDS technique", *Indian Journal of Science and Technology*, vol. 9, no. 32, pp. 1–5, 2016. DOI: 10.17485/ijst/2016/v9i32/100196.

[12] C. Thomas, B. Narayanaswamy, *Sensor Fusion and Its Applications*, Croatia: Sciyo, 2010, ch. 10. DOI: 10.5772/3302.

[13] R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas, Y.-L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system", *Information Sciences*, vol. 378, pp. 484–497, 2017. DOI: 10.1016/j.ins.2016.04.019.

[14] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, M. Rajarajan, "A survey of intrusion detection techniques in cloud", *Journal of Network and Computer Applications*, vol. 36, pp. 42–57, 2013. DOI: 10.1016/j.jnca.2012.05.003.

[15] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, K.-Y. Tung, "Intrusion detection system: A comprehensive review", *Journal of Network and Computer Applications*, vol. 36, pp. 16–24, 2013. DOI: 10.1016/j.jnca.2012.09.004.

[16] M. Naik, N. Geethanjali, "Multi-fusion pattern matching algorithm for signature-based network intrusion detection system", *Preprints*, pp. 1–8, 2013. DOI: 10.20944/preprints201608.0197.v1.

[17] M. H. Bhuyan, D. K. Bhattachryya, J. K. Kalita, "Network anomaly detection: methods, systems and tools", *IEEE communications surveys & tutorials*, vol. 16, no. 1, pp. 303–336, 2013. DOI: 10.1109/SURV.2013.052213.00046.

[18] B. J. Kang, K. Mclaughlin, S. Sezer, "Towards a stateful analysis framework for smart grid network intrusion detection", in *Proc. 4th Int. Symposium for ICS & SCADA Cyber Security Research*, 2016, pp. 1–8. DOI: 10.14236/ewic/ICS2016.14.

[19] U. Bashir, M. Chachoo, "Intrusion detection and prevention system: Challenges & opportunities", in *Proc. IEEE Int. Conf. Computing for Sustainable Global Development (INDIACom 2014)*, 2014, pp. 806–809. DOI: 10.1109/IndiaCom.2014.6828073.

[20] A. Girma, M. Garuba, J. Li, C. Liu, "Analysis of DDoS attacks and an introduction of a hybrid statistical model to detect DDoS attacks on cloud computing environment", in *Proc. IEEE 12th Int. Conf. Information Technology-New Generations (ITNG 2015)*, 2015, pp. 212–217. DOI: 10.1109/ITNG.2015.40.

[21] Y. Bhavsar, K. C. Waghmare, "Intrusion detection system using data mining technique: Support vector machine", *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no. 3, pp. 581–586, 2013.

[22] G. V. Nadiammai, M. Hemalatha, "Effective approach toward Intrusion Detection System using data mining techniques", *Egyptian Informatics Journal*, vol. 15, no. 1, pp. 37–50, 2014. DOI: 10.1016/j.eij.2013.10.003.

[23] Y. Yang, K. McLaughlin, T. Littler, S. Sezer, H. F. Wang, "Rule-based intrusion detection system for SCADA networks", in *2nd IET Renewable Power Generation Conf. (RPG 2013)*, 2013, pp. 1–4. DOI: 10.1049/cp.2013.1729.

[24] P. G. Majeed, S. Kumar, "Genetic algorithms in intrusion detection systems: A survey", *International Journal of Innovation and Applied Studies*, vol. 5, no. 3, pp. 233–240, 2014.

[25] S. S. Kumar, T. R. Prasad, "Network intrusion detection systems using genetic algorithm", *IJSEAT*, vol. 2, no. 3, pp. 107–111, 2014.

[26] M. N. Ahmed, A. H. Abdullah, O. Kaiwartya, "FSM-F: finite state machine based framework for denial of service and intrusion detection in MANET", *MANET. PLoS ONE*, vol. 11, pp. 1–17, 2016. DOI: 10.1371/journal.pone.0156885.

[27] I. Butun, S. D. Morgera, R. Sankar, "A survey of intrusion detection systems in wireless sensor networks", *IEEE communications surveys & tutorials*, vol. 16. pp. 266–282, 2014. DOI: 10.1109/SURV.2013.050113.00191.

[28] C.-H. Lo, N. Ansari, "Consumer: A novel hybrid intrusion detection system for distribution networks in smart grid", *IEEE Trans. Emerging Topics in Computing*, vol. 1, no. 1, pp. 33–44, 2013. DOI: 10.1109/TETC.2013.2274043.

[29] SANS Institute InfoSec Reading Room, "IDS - Today and Tomorrow", 2002.

[30] S. S. Rajan, V. K. Cherukuri, "An overview of intrusion detection systems", in *Proc. IDT Workshop on Interesting Results in Computer Science and Engineering (IRCSE 2009)*, 2009, pp. 1–8.

[31] S. M. Cho, "Intrusion detection systems vs. Intrusion Prevention Systems", Technical Report ACC 626, 2010.

[32] K. Singh, S. Tamrakar, "A review of intrusion-detection system-clustering and classification using RBF and SOM networks", *International Journal of Emerging Technology and Advanced Engineering,* vol. 5, no. 7, pp. 502–505, 2015.

[33] S. Kashyap, P. Agrawal, V. C. Pandey, S. P. Keshri, "Importance of intrusion detection system with its different approaches", *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 2, no. 5, pp. 1902–1908, 2013.

[34] R. Werlinger, K. Hawkey, K. Muldner, P. Jaferian, K. Beznosov, "The challenges of using an intrusion detection system: is it worth the effort?", in *Proc. of ACM Symposium on Usable Privacy and Security (SOUPS 2008)*, 2008, pp. 107–118. DOI: 10.1145/1408664.1408679.

[35] T. Xing, D. Huang, Z. Xiong, D. Medhi, "SDNIPS: Enabling Software-Defined Networking based intrusion prevention system in clouds", in *Proc. Int. Conf. Network and Service Management (CNSM 2014)*, 2014, pp. 308–311. DOI: 10.1109/CNSM.2014.7014181.

[36] D. J. Ragsdale, C. A. Carver, J. W. Humphries, U. W. Pooch, "Adaptation techniques for intrusion detection and intrusion response systems", in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, 2000, pp. 2344–2349. DOI: 10.1109/ICSMC.2000.884341.

[37] A. Shameli-Sendi, M. Cheriet, A. Hamou-Lhadj, "Taxonomy of intrusion risk assessment and response system", *Computers and Security*, vol. 45, pp. 1–16, 2014. DOI: 10.1016/j.cose.2014.04.009.

[38] S. Anwar, J. M. Zain, M. F. Zolkipli, Z. Inayat, S. Khan, *et al*., "From intrusion detection to an intrusion response system: fundamentals, requirements, and future directions", *Algorithms*, vol. 10, no. 39, pp. 1–24, 2017. DOI: 10.3390/a10020039.

[39] A. C. J. Carver, "Adaptive agent-based intrusion response", Ph.D. dissertation, College Station, Texas A&M University, TX, USA, 2001.

[40] S. Anwar, J. M. Zain, Z. Inayat, R. U. Haq, A. Karim, A. N. Jabir, "A static approach towards mobile botnet detection", in *Proc. 3rd Int. Conf. Electronic Design (ICED 2016)*, 2016, pp. 563–567. DOI: 10.1109/ICED.2016.7804708.

[41] A. Shameli-Sendi, M. Cheriet, A. Hamou-Lhadj, "Taxonomy of intrusion risk assessment and response system", *Computers & Security*, vol. 45, pp. 1–16, 2014. DOI: 10.1016/j.cose.2014.04.009.