Computer Science and Information Systems 16(2):491–514 https://doi.org/10.2298/CSIS190122008J

Logical Filter Approach for Early Stage Cyber-Attack Detection

Vacius Jusas [1], Saulius Japertas [2], Tautvydas Baksys [3], Sandeepak Bhandari [4]

1 Software Engineering Department, Kaunas University of Technology, Studentu St. 50, LT-51368 Kaunas, Lithuania
2 Department of Electronics Engineering, Kaunas University of Technology, Studentu St. 50, LT-51368 Kaunas, Lithuania
3 Department of Computer Science, Kaunas University of Technology, Studentu St. 50, LT-51368 Kaunas, Lithuania
4 Software Engineering Department, Kaunas University of Technology, Studentu St. 50, LT-51368 Kaunas, Lithuania

Abstract. The planned in advance cyber-attacks cause the most damage for the users of the information systems. Such attacks can take a very long time, require considerable financial and human resources, and therefore, they can only be organized by large interest groups. Furthermore, current intrusion detection systems, intrusion prevention systems and intrusion response systems used to protect against cyber-attacks have several shortcomings. Such systems respond only to the attack itself when it is too late to take a preventive action and they are not suitable for detecting an attack in early stages when it is possible to block the attack and minimize the losses. Early detection requires detailed monitoring of network and system parameters to be able to accurately identify the early stages of the attack when it is still possible to kill the attack chain. In this paper, we propose to consider an attack chain consisting of nine stages. The method to detect early stage cyber-attack based on the attack chain analysis using hardware implementation of logical filters is suggested. The performed experiment acknowledges the possibility to detect the attack in the early stages.

Keywords: System security; Cyber-attack; Intrusion detection; Logical circuits.

1. Introduction

The most dangerous cyber-attacks are those that are planned in advance, and they can be planned by both state structures and terrorist organizations. The planned cyber-attacks consist of a variety of different stages. Different authors describe the different number of the stages and parameters of such cyber-attacks. Symantec entitles five stages: reconnaissance, incursion, discovery, capture, and exfiltration [1]. The same number of stages, but with different names, is proposed in [2]: reconnaissance, intrusion, taking control, collecting and leaking information, eliminating traces. Meanwhile, Yadav and Rao [3] offer seven steps: reconnaissance, weaponization, delivery, exploitation, installation, command and control, act on objective. Yadav and Rao [3] clearly distinguished two groups of the stages (early stages and late stages). More research works [4]-[6] can be found, where a number of stages varies between three and eight. Different means and equipment are used to organize the detection of the attacks at the different

stages. It can be assumed that detecting and stopping the attack in the early stages can prevent the serious harmful effects [4, 5]. However, it is necessary to distinguish between the early stages and the late stages, when damage created by an attack is mainly unavoidable. Therefore, it is needful to determine the various stages of attack in order to provide the means and methods for preventing the harmful effects of the attacks.

The aim of the paper is to present a method to determine the possibility of cyber-attack against information and telecommunication systems at its earliest stages when the cyber-attack can still be effectively stopped. The method is based on the detailed monitoring of network and system parameters to accurately identify the early stages of the attack. A hardware implementation of logical filters is suggested for the method.

The rest of the paper is organized as follows. We review the related work in Section 2. We present an attack vector of early stages in Section 3. We introduce early stage cyber-attacks detection method in Section 4. We discuss the results of the experiment in Section 5. We finish with conclusions in Section 6.

2. Review of Related Work

The nature of cyber-attacks against information and telecommunication systems is different and varies [7]. The information and telecommunication networks are protected from cyber-attacks using various tools and methods. All these tools and methods can be grouped into three groups: intrusion detection system (IDS), intrusion prevention system (IPS), intrusion response system (IRS).

One of the desirable features of IDS is being a real-time system. An adaptive intrusion detection system that can detect unknown attacks in real-time network traffic is a major concern. Conventional adaptive intrusion detection systems are computationally expensive in terms of computer resources and time because these systems have to be retrained with known and unknown attacks. Rathore et al. [8] proposed a real-time intrusion detection system for ultra-high-speed big data environment using Hadoop implementation. The proposed system is based on four-layered IDS architecture that consists of the capturing layer, filtration and load balancing layer, processing or Hadoop layer, and the decision-making layer. Al-Yaseen et al. [9] suggested a method that is based on a multi-agent system to allow the intrusion detection system to adapt to unknown attacks in real-time. The detection model uses the multi-level hybrid support vector machines and extreme learning techniques. Despite the widespread use of IDS systems, they have several weaknesses. Major deficiencies in the network intrusion detection systems (NIDS) include the inability to analyze encrypted traffic, late updates, time delay between attack start and warning, and the difficulty of processing data on a redundant network. Hybrid intrusion detection systems (HIDS) deficiencies are identified as failure to recognize network scans, inefficiencies in DoS attacks [10], [11]. Some IDSs can be relatively easily avoided (e.g., anomaly-based or signature based) [12], [13]. Werlinger et al. [14] state that the result of using IDS is not always clear. It is also interesting that practically the same imperfections have existed for many years [3], [15], [16] and even the new methods [13] [17] do not help to avoid them.

An IPS is a newer approach than the IDS to fight against the cyber security threat. The IPS combines the technique of firewall with the IDS [18]. The use of traditional IPSs for information and telecommunication systems is problematic for several reasons [19]:

1. Latency: in-bound IPS requires inspection and blocking action on each network packet, which consumes cloud system resources and increases the detection latency;
2. Resource Consumption: running the intrusion detection and prevention systems (IDPS) services usually consumes significant resources;
3. 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 IRSs are 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 [20, 21]. The Audit expert system is currently widely used [21]. In support of Audit expert systems, Moon et al. [22] presented Multi-Layer Defense System (MLDS) that applies a reinforced defense system by collecting and analyzing log information and various information from network infrastructure. Heo et al. [23] suggested a system design that helps to maintain a certain level of quality of service and quality of security service in threatening environments. Nevertheless, despite all the advantages provided by such systems, they still have many deficiencies that are fully disclosed in the papers [16–21].

One of the biggest deficiencies 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 [20] and a high number of false alarms, which directly depends on the quality of IDS [21]. There are more deficiencies however they are 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 [20] or the use of appropriate hardware.

There are currently some attempts to detect cyber-attacks in the early stages. Yadav and Rao [3] suggested that the early stages include reconnaissance, weaponization, delivery, and initial part of the exploitation, in which, if an attack is detected, its effects can be eliminated. Siddique et al. [6] presented promising experimental results of the attack detection using IDS. Yan and Zhang [24] offered structured intrusion detection based on the behavioral semantics. However, it is not entirely clear how the early stage is understood and what opportunities are to process large flows of information. Vincent et al. [25] highlighted the importance of early detection and offered some solutions for detecting Trojan viruses. However, it should be noted that Vincent et al. [25] did not provide a detection algorithm. Chen et al. [26] proposed a model that integrated and correlated multiple logs to identify the early phase of targeted attacks. State-based hidden Markov model is used to detect joint attacks. However, this model is based on the IDS system, which, as noted above, has several shortcomings and, it is mostly designed to detect distributed denial-of-service (DDoS) attacks. Moreover, the idea of provided attack detection is vaguely presented. There are more investigations that are dedicated to the specific type of the attacks [27], [28]. Bhattacharya and Selvakumar [27] suggested a multi-measure multi-weight ranking approach for the identification of the network features for the detection of denial of service (DoS) and probe attacks. The approach combines the filter and wrapper feature selection methods and clustering methods to assign multiple weights to each feature. Cheng et al. [28] proposed a DDoS detection method for socially aware networking based on the time-series autoregressive integrated moving average model. The model describes a multi-protocol-fusion feature to characterize normal network flows.

The review shows that the 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) begin 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.

3. Early Stages of the Cyber-Attack

As already mentioned, the most dangerous cyber-attacks are those that are planned. The preparation of such attacks and their initial stages can last quite long – for months or even years. The attacker can assess all the victim's weaknesses and the consequences of the attack would be extremely harmful. The main purpose of these attackers is to get the user's access to the system, so their attack vectors are directed to obtain user rights in the system, exploiting system software vulnerabilities. At their late stages, such attacks normally cannot be terminated without causing losses. Such prepared cyber-attacks are difficult to detect because of their well-planned steps, but if they occur and enter the late stage sector, their consequences are the greatest comparing to other types of attacks. It would be advisable to distinguish two types of attacks: a classic attack and an intelligent attack.

The first type of attack is characterized by the fact that it has practically no individual stages, or in some cases, it is possible to distinguish one or two stages: exploration and attack. Such attacks are relatively fast, often without a well-defined target, they are poorly organized and coordinated. The tools used in such attacks are for creating rugged effects, i.e. launching DoS and DDoS attacks, various viruses (untargeted), malware, and the similar ones. Such attacks cause losses, but usually these losses relate to a single entity or individual object, the effects of such attacks are relatively easy recoverable, and the attackers are easily detectable. Attackers creating such attacks are normally represented in relatively low-impact output groups, i.e. hackers, crackers, phreakers or vandals. Normally, these groups are formed from a small number of members and in the most cases, just one member forms a group.

The second type of attacks (intelligent attacks) has the following characteristics: detailed planning, many stages, and slow progress. Attackers have a well-defined target and a well-defined goal. Attacks use malware specifically designed for the target and deep self-disguise. The effects of such attacks are extremely damaging, requiring a lot of effort to eliminate the consequences of the attacks. Such attackers are in large groups and well-organized, with sufficient financial resources from criminal groups, terrorist organizations or state structures.

For intelligent attacks, it is necessary firstly define their possible stages. As stated in the introduction, the elaboration of those stages is an important factor in enabling the most accurate estimation of the initial stages of the attack, which have not yet done any harm, and which can still be described as "chain killing". Yadav and Rao [3] proposed a vector for intelligent attacks that is formed out of seven stages. Although Yadav and Rao [3] clearly distinguished early stages and late stages, our offer is to extend the number of groups into three:

1. Early stages;
2. Transitional stages;

3. Late stages.

The early stages include processes for data collection, target tracking and attack infrastructure. In the transitional stages, information from the early stages is used and actions are taken to weaken the victims' system (e.g., implementing a malicious code or process against the system, exploiting its vulnerabilities). This enables the access to the system. Thereafter, the late stage attack processes follow direct system take-over, specific data capture, or infrastructure removal procedures.

Table 1. General classification of cyber-attacks

Attacks parameters	Classic	Intelligent
Number of Stages	1–2	> 3
Speed of Attack	Fast	Slow
Attack types	DoS, DDoS, malware, virus	Classic and custom-made software tools purposefully delivered to a certain target and specifically adapted to victim's network and system configuration
Attacker types	Individual person or small groups	Criminal and terrorist groups, state structures
Target of an attack	Separate object or subject	Object groups, state institutions, economical branches, wide social groups
Attackers financial resources	Relatively small	Wide financial resources, in some cases, unlimited
Consequences	Relatively small, easily recovered	Hard losses, hardly recovered

Based on the literature analysis and our own experience, we suggest an extension of the number of stages proposed in [3] to nine by adding the stage of social engineering and incorporating the stage of evasion (Fig. 1).

In the social engineering phase (Stage 0), there is an attempt to extract certain information about future cyber-attack target (entity or object) with some information that would facilitate a cyber-attack using the psychological effects of human beings. Experts say that the impact of social engineering is almost impossible to avoid. Therefore, this is a good way to extract certain primary data. In this paper, this stage will not be discussed further because it is an information collection step that involves various social and psychological manipulation techniques. However, to the extent that it aims to obtain data for planning a cyber-attack, social engineering should be considered as the initial stage of a cyber-attack.

We suppose that an attack can be withheld if it was detected in the preliminary stages 1 – 3, i.e. reconnaissance, weaponization and delivery, since an attack detection during these stages allows killing or blocking the attack. Because of the continuing attack, noticeable damage starts directly interfering with system and network work. It is necessary to detect these processes until they reach the 4-th stage (stage 4 – exploitation).

The first stage of the attack (reconnaissance) consists of three actions: port scan, host scan and system version scan. Port scan is a scanning of the network ports using a SYN request. Host scan is a scanning of the nodes in the system and obtaining their IP addresses. Version scan is an obtaining of the version of the system.

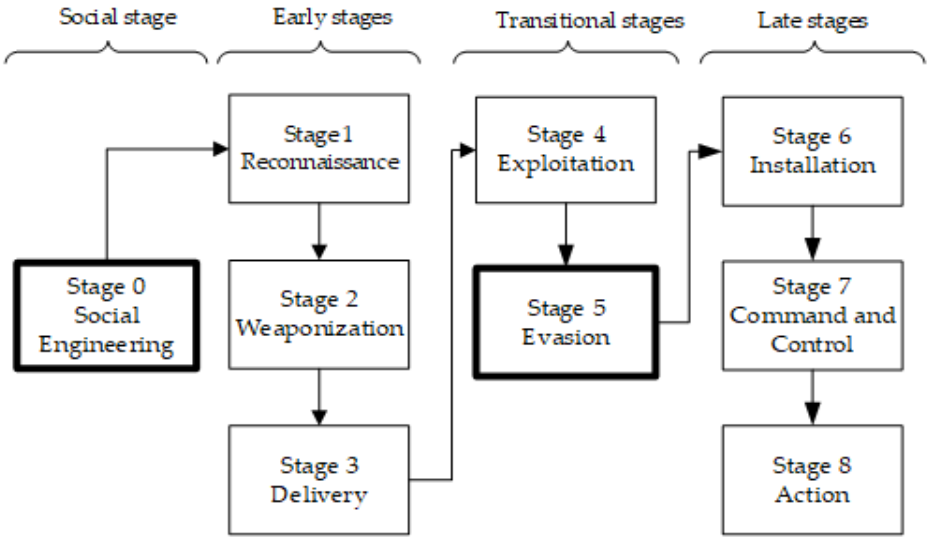

Fig. 1. Elaborated attack stages

The second stage (weaponization) includes two factors: system & services version scan and service stress test. The third stage (delivery) is a stage when the first part of the malicious code is delivered to victim's infrastructure to be executed at a certain time and it starts damaging processes against the targeted system. The third stage consists of two actions: version check and spoofing. Each of these actions has its own activities. For example, the spoofing action includes activities such as modifying network packets and programs with malicious code infiltration; performing stress tests over system processes remotely.

Processes and actions, which are executed by an adversary, can be registered by monitoring the network stack and system behavior. Results of the monitoring enable distinguishing the features inherent in these ongoing processes and application of them for detection of system anomalies and recognition of an attack to begin.

Attacks in the different stages have several characteristics that identify attack process. In this paper, we distinguish three characteristic groups that allow to characterize the ongoing processes: physical network stack parameters, logical parameters of the system being attacked, network stack flags.

4. Method to Detect the Early Stages of the Cyber-Attack

The determination provided in Section 3 of early stages of the cyber-attack enables the exploration of the ways to recognize the presence of such stages. The essence of the

proposed method is to use the appropriate logical filters to classify the certain parameters of the traffic. For this purpose, the total analysed data flow is considered to consist of two parts: the normal flow (i.e., the flow that is not harmful) and the attacker's flow (malicious flow). The generic filter consists of two blocks: a packet analysis block and a parameter processor. The traffic input into the filter is analyzed on the packet level, which results in a packet parameter (e.g., DST IP). The obtained parameter is passed to the internal parameter processor that forms an indicator value according to the conditions provided.

The detection method consists of three parts: filter part, evaluation block and action block (Fig. 2). The filters are implemented in two blocks: consolidated network filtering and system monitoring (CNFSM) and parameter preprocessing (PP). In the CNFSM block, the filters are grouped into three groups: filtering of network parameters (NF), filtering of system parameters (SF), filtering of network stack flags (LF). The evaluation block consists of three logical circuits that are connected at the outputs of the corresponding filter groups. The purpose of the filters is to register parameters and, if their values exceed predefined values, indicate the malicious activity. The purpose of the evaluation block is to collect the binary parameters and process them for the indication of the possible attack action. The purpose of the action block is to decide which stage of the attack is observed.

Using this principle, it is possible to analyze network traffic and system behavior adaptively by adjusting filters for analysis according to the need (available resources, depth of analysis, speed and tolerances of created system or network delays). To ensure early detection, different types of filters are used: network parameters NF (shown in circle); system parameters SF (depicted in rectangular); network stack flags LF (shown in hexagon). The three filter groups in total include 31 different filters: 12 filters belong to the NF group; 6 filters belong to the SF group and the remaining 13 filters belong to the LF group. These filters are consolidated, i.e. they perform the collection of the parameters and their analysis.

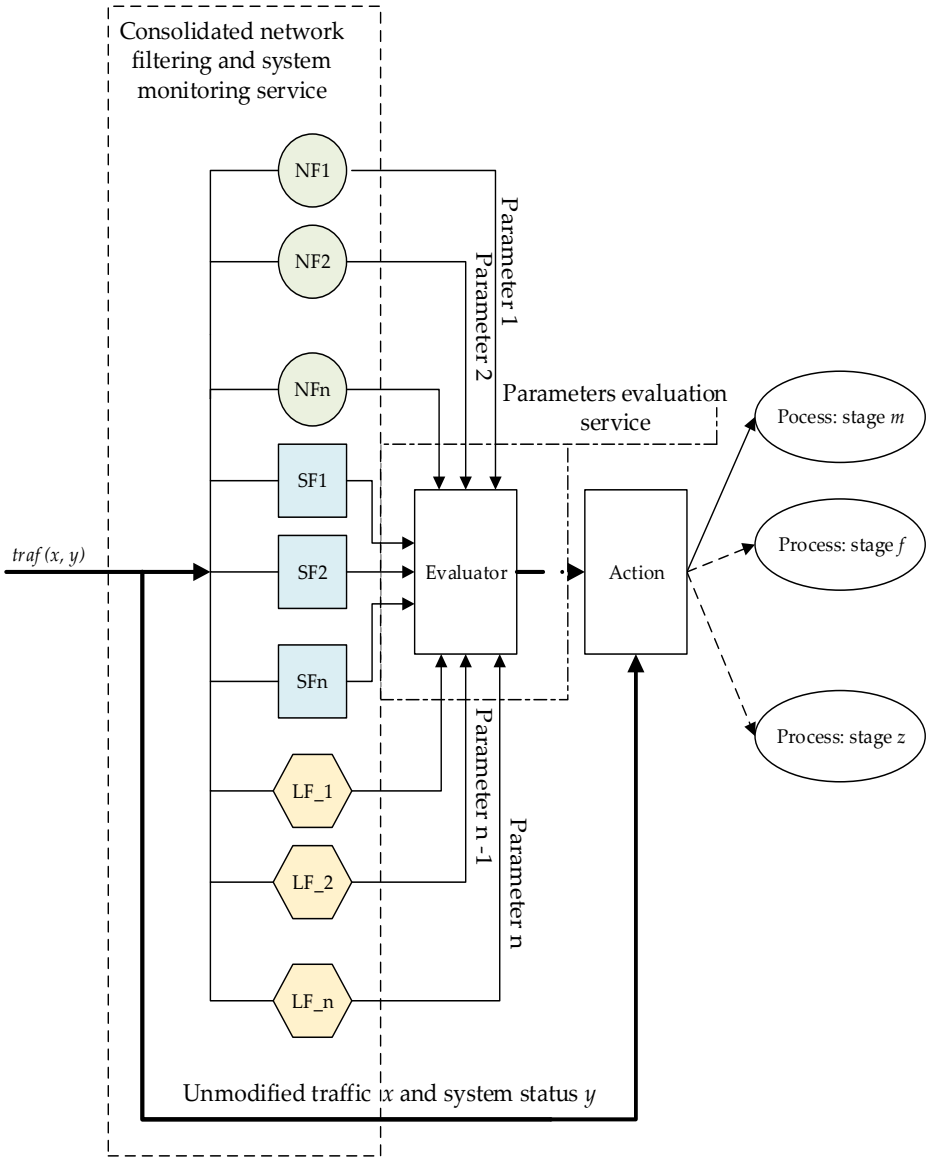

Fig. 2. Detailed schematic view of the filters

The filters of network parameters are numerated from 1 to 12 (NF1 ... NF12 and the outputs of the filters, which form inputs to the logical circuit, are labeled as x1 ... x12). The filters of system behavior are labeled from SF1 to SF6. The outputs of the filters are labeled as x13 ... x18. The filters of network stack flags are labeled from LF1 to LF13. The outputs of the filters are labeled as x19 ... x31. All the filters and their functions are enumerated in Table 2.

Table 2. Functions of attack monitoring filters

No.	Filter name	Filtering parameter	Filter description
1	NF1	IP	Attacker's IP address
2	NF2	IP COUNT	IP address repetition
3	NF3	PORT NUMBER	Port number to which the information is sent
4	NF4	PORT DISTRIBUTION	Distribution of ports according to the token information
5	NF5	PACKET COUNT	The number of packets in the network tract
6	NF6	STACK BYTES	Amount of data transferred in the session
7	NF7	PACKETS A-> B	Number of packets sent from the attacker to the victim
8	NF8	PACKETS B->A	Number of packets sent from the victim to the attacker
9	NF9	BYTES A->B	Amount of data transferred from the attacker to the victim
10	NF10	BYTES B->A	Amount of data transmitted from the victim to the attacker
11	NF11	DURATION	Duration of the active single session between the attacker and the victim
12	NF12	ABSOLUTE TIME	Absolute start time for the session
13	SF1	PERIPHERAL STATUS	Whether the peripheral device has changed
14	SF2	UNLISTED PROCESS	What processes in the system are in the list
15	SF3	FLAWLESS USER LOGIN	Whether an unexpected user connection was attempted or a password or unconnected

			connection was attempted
16	SF4	SUSPICIUOS TIME	System clock times which average is significantly deviating from standard user connection time
17	SF5	DISK ACTIVITY	Is the increased activity of the disk array detected by comparing with an average value
18	SF6	PORT BINDING	Whether the port is bound to port
19	LF_FIN	FIN FLAG	Packet's FIN flag
20	LF_SYN	SYN FLAG	Packet's SYN flag
21	LF_TCP_CONN()	TCP_CONN() FLAG	TCP Connection request
22	LF_NULL	NULL FLAG	NULL flag
23	LF_PING	ICMP FLAG	ICMP request
24	LF_VERSION_DETECTION	VER FLAG	VERSION flag
25	LF_UDP_SCAN	UDP FLAG	UDP request
26	LF_BULK_SCAN	BULK FLAG	Random request
27	LF_WINDOWS_SCAN	WIN_SCAN FLAG	Versions of Windows query
28	LF_RPC_SCAN	RPC FLAG	Identify the RPC protocol
29	LF_LIST_SCAN	LST FLAG	A query that results a list of the previous query vector
30	LF_IDLE_SCAN	IDL FLAG	An IDLE process request
31	LF_FTP_BOUNCE	BOUNCE FLAG	FTP service request

Data collected from all types of the logical filters is sent to the parameter preprocessing block, in which, according to filtered parameters, sets of attack parameters are further processed. If the value of the filtered parameter exceeds the predefined value, then this parameter is assigned a binary value 1 (anomaly value), otherwise – the binary value 0 (normal value). According to the result of this process, we can distinguish seven attack actions that fall into early three attack stages. The actions are as follows:

1. HS – Host Scan;
2. PS – Port Scan;
3. SSV – System and Services Version;
4. SST – Services Stress Tests;
5. SP – Spoofing;

6. LA – Login Attempt;
7. SE – Service Exploitation.

Processed session parameter sets are sent to the evaluation block of logical circuits. The evaluation block consists of three independent logic circuits: the first logical circuit performs analysis of evaluated parameters of NF, the second logical circuit evaluates the analyzed SF parameters, the third logical circuit performs analysis of filtered LF parameters. The configuration of these filters allows to create a setup of the detection, the result of which is determined by the logical circuits.

The logical circuits operate on the sets of binary parameters. Seven types of the possible attack actions were determined; therefore, we have designed the logical circuits having seven primary outputs. In such a way, every primary output indicates the presence of the different attack action. A logical circuit of NF analysis uses primary inputs x1 ... x12 and produces seven primary outputs labeled as F21 ... F27. A logical circuit of SF analysis uses primary inputs x13 ... x18 and produces seven primary outputs labeled as F35 ... F41. A logical circuit of LF analysis uses primary inputs x19 ... x31 and produces seven primary outputs labeled as F42 ... F48. Subsequently, the primary outputs of all the logical circuits are combined at the final point in the evaluator block.

The bit stream, which arrives at the inputs of the evaluation block, is divided into three parts and supplied into three independent logical circuits. Every circuit is dedicated and produces seven bits. The values at the primary outputs of all three logical circuits are joint into single vector. Only the values of the combined vector can implicate the presence of the attack action. The presence of the values of the final vector in the lookup table indicates the early stage of the cyber-attack.

The analytical form of logical circuit of NF analysis is shown in (1). A member xA, where $A \in \{1 \ ... \ 12\}$, corresponds to the binary 1, and a member $\overline{xA}$, where $A \in \{1 \ ... \ 12\}$, corresponds to the binary 0. This form contains output logical functions, which consist of inputs $x1$ ... $x12$ and Output 1 is a vector of F21 ... F27 values.

$$OUTPUT1 = \begin{Bmatrix} F21 \\ F22 \\ F23 \\ F24 \\ F25 \\ F26 \\ F27 \end{Bmatrix} = \cdot \begin{Bmatrix} x1 \cdot x2 \cdot \overline{x3} \cdot \overline{x4} \cdot \overline{x5} \cdot \overline{x6} \cdot \overline{x7} \cdot \overline{x8} \cdot \overline{x9} \cdot \overline{x10} \cdot \overline{x11} \cdot \overline{x12} \\ x1 \cdot x2 \cdot x3 \cdot x4 \cdot \overline{x5} \cdot \overline{x6} \cdot \overline{x7} \cdot \overline{x8} \cdot x9 \cdot x10 \cdot \overline{x11} \cdot \overline{x12} \\ x1 \cdot x2 \cdot \overline{x3} \cdot \overline{x4} \cdot x5 \cdot \overline{x6} \cdot \overline{x7} \cdot \overline{x8} \cdot \overline{x9} \cdot \overline{x10} \cdot \overline{x11} \cdot \overline{x12} \\ x1 \cdot x2 \cdot \overline{x3} \cdot \overline{x4} \cdot \overline{x5} \cdot x6 \cdot x7 \cdot x8 \cdot \overline{x9} \cdot \overline{x10} \cdot \overline{x11} \cdot \overline{x12} \\ x1 \cdot x2 \cdot \overline{x3} \cdot \overline{x4} \cdot x5 \cdot \overline{x6} \cdot \overline{x7} \cdot \overline{x8} \cdot \overline{x9} \cdot \overline{x10} \cdot x11 \cdot x12 \\ x1 \cdot x2 \cdot \overline{x3} \cdot \overline{x4} \cdot x5 \cdot \overline{x6} \cdot \overline{x7} \cdot \overline{x8} \cdot \overline{x9} \cdot \overline{x10} \cdot x11 \cdot x12 \\ x1 \cdot x2 \cdot \overline{x3} \cdot \overline{x4} \cdot x5 \cdot x6 \cdot \overline{x7} \cdot \overline{x8} \cdot \overline{x9} \cdot \overline{x10} \cdot \overline{x11} \cdot \overline{x12} \end{Bmatrix} \quad (1)$$

Table 3 shows the attack actions that make up the attack vector. The values in the column under name “Action” have the attribute “part” because the single circuit on its own cannot define fully the action of the attack. The action of the attack can be defined only when the results of the all three circuits are combined.

Table 3. Lookup table of *x*1…*x*12 parameter set and NF output values

No.	ACTION	INPUTS												OUTPUTS						
		x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	F21	F22	F23	F24	F25	F26	F27
1	HS part	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
2	PS part	1	1	1	1	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0
3	SSV part	1	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
4	STT part	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0
5	SP part	1	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	1	0	0
6	LA part	1	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	1	0
7	SE part	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1

The logical circuit of NF analysis is presented in Fig. 3. On the left side of the picture, it is marked binary processed NF parameter inputs, these parameters are obtained directly from the network driver.

The logical circuit uses 23 logical gates. The primary outputs F25 and F26 are identical due to the identity of the parameter values analyzed (the parameters of the SP Part and LA part analyzed are identical in this analysis, so the generated response is the same, but the outputs are different).

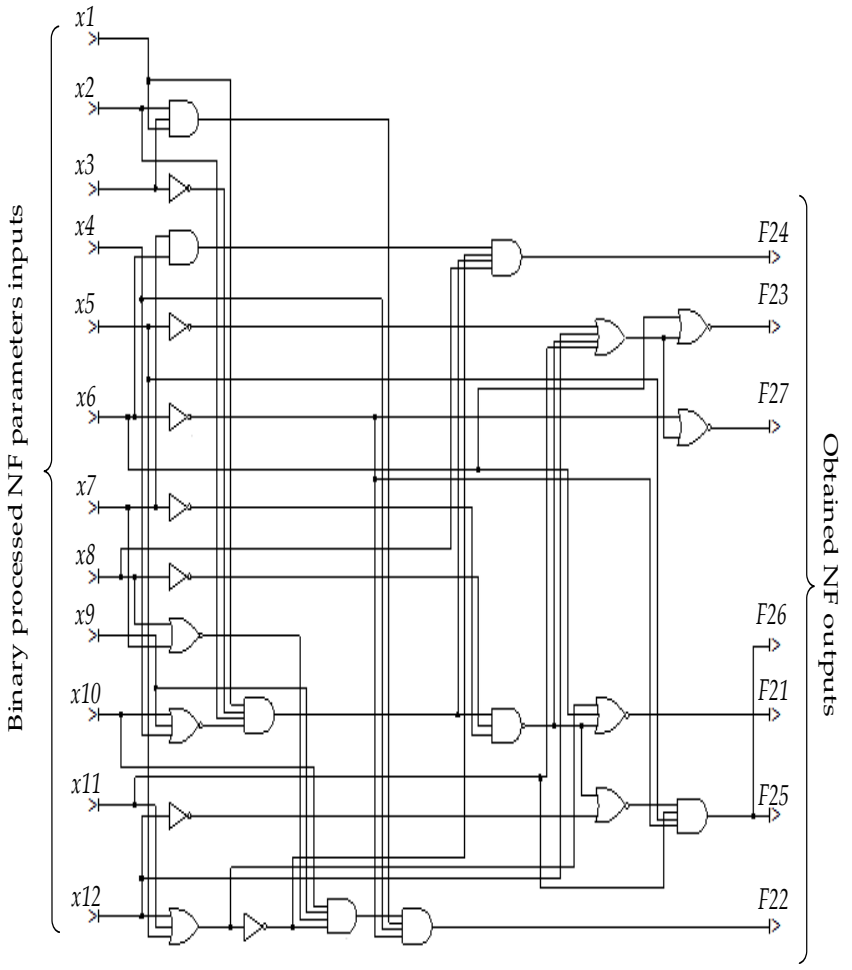

Fig. 3. Logic circuit of x1…x12 bit stream parameters

For example, the values on the primary inputs indicating the HS action part of the attack are as follows: $x1 = 1$, $x2 = 1$, $x3 = 0$, $x4 = 0$, $x5$ ... $x12 = 0$, and the primary output F21 = 1, the remaining primary outputs F22 ... F27 = 0. In this case, the HS action part of the attack will be detected when the parameter $x1$ "IP address" and the parameter $x2$ "IP repetition" exceed their predefined values, meanwhile, the predefined values of the remaining NF filter parameters $x3$ ... $x12$ will not be exceeded. In this case, the entire output vector will have a value of 1000000. Such an assessment is only part of the overall assessment of the HS action process, and other parts of the assessment are performed at SF and LF logical circuits, respectively.Output 1 is the first part of the logical analysis results, further results are obtained from SF and LF analysis, named as Output 2 (SF) and Output 3 (LF), respectively.

Table 4. Lookup table of x13…x18 parameter and SF output

No.	ACTION	INPUTS						OUTPUTS						
		x13	x14	x15	x16	x17	x18	F35	F36	F37	F38	F39	F40	F41
1	HS part	0	0	0	0	0	0	1	0	0	0	0	0	0
2	PS part	0	0	0	0	0	0	0	1	0	0	0	0	0
3	SSV part	0	0	0	0	0	1	0	0	1	0	0	0	0
4	STT part	0	1	1	1	1	1	0	0	0	1	0	0	0
5	SP part	0	0	0	0	0	1	0	0	0	0	1	0	0
6	LA part	0	0	1	1	0	1	0	0	0	0	0	1	0
7	SE part	0	1	1	1	1	1	0	0	0	0	0	0	1

$$OUTPUT\,2 = \begin{Bmatrix} F35 \\ F36 \\ F37 \\ F38 \\ F39 \\ F40 \\ F41 \end{Bmatrix} = \begin{Bmatrix} \overline{x13} \cdot \overline{x14} \cdot \overline{x15} \cdot \overline{x16} \cdot \overline{x17} \cdot \overline{x18} \\ \overline{x13} \cdot \overline{x14} \cdot \overline{x15} \cdot \overline{x16} \cdot \overline{x17} \cdot \overline{x18} \\ \overline{x13} \cdot \overline{x14} \cdot \overline{x15} \cdot \overline{x16} \cdot \overline{x17} \cdot x18 \\ \overline{x13} \cdot x14 \cdot x15 \cdot x16 \cdot x17 \cdot x18 \\ \overline{x13} \cdot \overline{x14} \cdot \overline{x15} \cdot \overline{x16} \cdot \overline{x17} \cdot x18 \\ \overline{x13} \cdot \overline{x14} \cdot x15 \cdot x16 \cdot \overline{x17} \cdot x18 \\ \overline{x13} \cdot x14 \cdot x15 \cdot x16 \cdot x17 \cdot x18 \end{Bmatrix} \quad (2)$$

Table 4 shows the attack actions that make up the attack vector of the SF analysis. In this case, the attack action HS part and the attack action PS part describe the values on the primary inputs *x*13 ... *x*18 as zeroes. This is because the SF analysis parameters *x*13 ... *x*18 do not take part in forming HS and PS actions. However, this result is important, the outputs F35 and F36 are assigned the appropriate values. The logical circuit used for SF analysis is shown in Fig. 4. The analysis is based on six criteria, so there are six primary inputs and, as previously mentioned, seven primary outputs to identify action of the attack.

In the logical circuit of SF analysis, 12 logical gates are used. As in the case of NF circuit, there are input sequences that are identical, therefore, the primary outputs F35 and F36, the primary outputs F37 and F39, the primary outputs F38 and F41 are connected in parallel. Even though some input vectors for the SF circuit are the same, their combination with input vectors of the NF circuit makes the unique input vector and produces a different final output result.

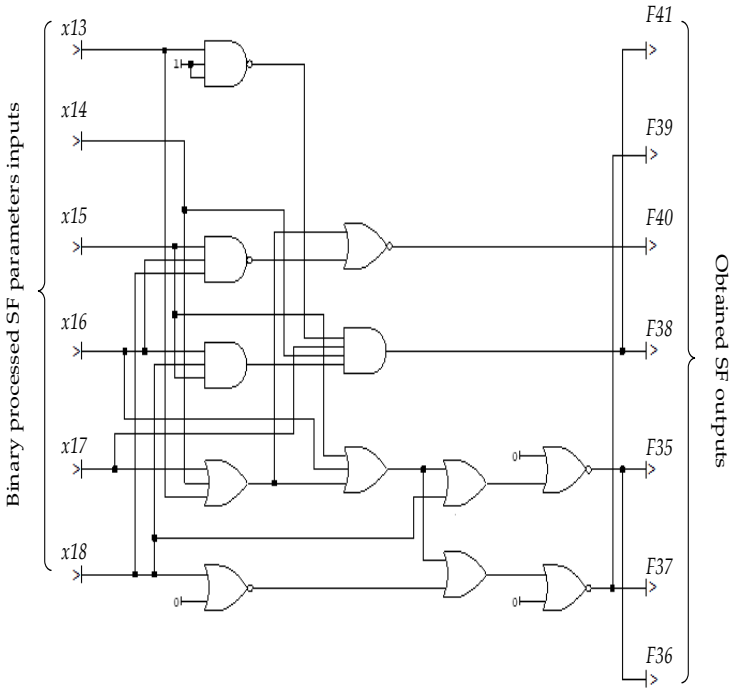

Fig. 4. Logic circuit of x13…x18 bit stream parameters

Analytical approach to the LF Analysis Output 3 result is shown in the (3). Analogically to the Output 1 and Output 2 forms, a member xA, $A \in \{19 \ldots 31\}$, corresponds to the binary 1, and a member $\overline{xA}$, $A \in \{19 \ldots 31\}$, corresponds to the binary 0. Output 3 is the last component of the logical gate analysis, characterizing the flag states in the network stack. The values on the primary outputs of the logical circuits are combined and the attack factors are determined according to the obtained result.

Table 5 shows the attack actions that make up the attack vector of the LF analysis. In this case, the primary inputs x19 ... x31 and the primary outputs F42 ... F48 are used. Differently from NF and SF analysis, there are no duplicate output cases. All the primary outputs are activated with unique combinations on the primary inputs

$$OUTPUT3 = \begin{Bmatrix} F42 \\ F43 \\ F44 \\ F45 \\ F46 \\ F47 \\ F48 \end{Bmatrix} = \begin{Bmatrix} \overline{x19}\cdot\overline{x20}\cdot\overline{x21}\cdot\overline{x22}\cdot x23\cdot\overline{x24}\cdot\overline{x25}\cdot\overline{x26}\cdot\overline{x27} \\ x19\cdot x20\cdot\overline{x21}\cdot\overline{x22}\cdot\overline{x23}\cdot\overline{x24}\cdot\overline{x25}\cdot\overline{x26}\cdot\overline{x27} \\ \overline{x19}\cdot\overline{x20}\cdot x21\cdot x22\cdot\overline{x23}\cdot x24\cdot x25\cdot x26\cdot x27 \\ \overline{x19}\cdot\overline{x20}\cdot x21\cdot\overline{x22}\cdot x23\cdot\overline{x24}\cdot\overline{x25}\cdot\overline{x26}\cdot\overline{x27} \\ \overline{x19}\cdot\overline{x20}\cdot x21\cdot x22\cdot\overline{x23}\cdot\overline{x24}\cdot\overline{x25}\cdot\overline{x26}\cdot\overline{x27} \\ \overline{x19}\cdot\overline{x20}\cdot x21\cdot\overline{x22}\cdot x23\cdot\overline{x24}\cdot\overline{x25}\cdot\overline{x26}\cdot\overline{x27} \\ \overline{x19}\cdot\overline{x20}\cdot x21\cdot x22\cdot x23\cdot\overline{x24}\cdot\overline{x25}\cdot\overline{x26}\cdot\overline{x27} \end{Bmatrix} \cdot \begin{Bmatrix} \overline{x28}\cdot\overline{x29}\cdot x30\cdot\overline{x31} \\ \overline{x28}\cdot\overline{x29}\cdot\overline{x30}\cdot\overline{x31} \\ x28\cdot x29\cdot\overline{x30}\cdot\overline{x31} \\ \overline{x28}\cdot\overline{x29}\cdot\overline{x30}\cdot x31 \\ \overline{x28}\cdot\overline{x29}\cdot\overline{x30}\cdot\overline{x31} \\ \overline{x28}\cdot\overline{x29}\cdot\overline{x30}\cdot\overline{x31} \\ \overline{x28}\cdot\overline{x29}\cdot x30\cdot x31 \end{Bmatrix}.$$

Table 5. Lookup table of *x*18…*x*31 parameter set and LF output values

No.	ACTIONS	INPUTS													OUTPUTSS					
		x19	x20	x21	x22	x23	x24	x25	x26	x27	x28	x29	x30	x31	F42	F43	F44	F45	F46	F47	F48
1	HS part	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
2	PS part	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
3	SSV part	**0**	0	1	1	0	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0
4	STT part	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
5	SP part	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
6	LA part	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
7	SE part	0	0	1	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1

The logical circuit used for LF analysis is shown in Fig. 5. For the analysis, 13 criteria are used and 13 primary inputs x19 ... x31 correspond to them. The seven primary outputs F42 ... F48 for identifying attack actions are used. The logical circuit consists of 26 logical gates. In this circuit, unlike NF and SF cases, there are no identical value combinations on the primary inputs.

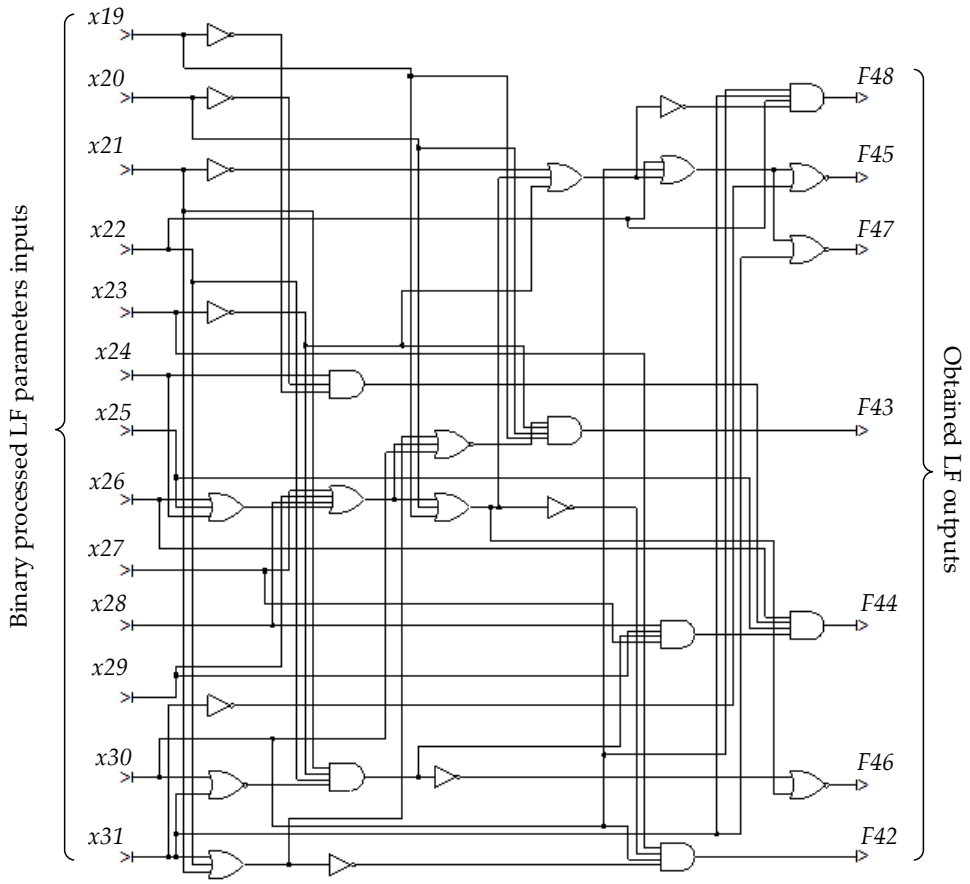

Figure 5. Logic circuit of *x19*…x31 bit stream parameters

The analytical aggregated expression combining the previously presented separate results is presented in (4). Formula (4) combines output vectors of Output 1, Output 2, and Output 3 into single vector for the cyber-attack detection. For example, HS denotes that attack action Host Scan is fully characterized by the code HS = 100000010000001000000 consisting of a set of NF = {F21 ... 27}, SF = {F35 ... 41} and LF = {F42 ... F48} filters. We can determine the early stage of the attack according to the values in Table 6.

$$OUTPUT\ ACTION\ FULL\ (OAF) = OUTPUT1 + OUTPUT2 + OUTPUT3 \quad (4)$$

Table 6. Lookup table for definition of attack actions

No.	ACTIONS	NF OUTPUTS							SF OUTPUTS							LF OUTPUTS						
		F21	F22	F23	F24	F25	F26	F27	F35	F36	F37	F38	F39	F40	F41	F42	F43	F44	F45	F46	F47	F48
1	HS	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
2	PS	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
3	SSV	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0
4	STT	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0
5	SP	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0
6	LA	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0
7	SE	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1

5. Experiment

We carried out the experiments on the synthesized data to evaluate the capabilities of the proposed method. The experimental set consists of two parts:

1. Attack vector: values from the binary set {0, 1} are filled in deterministically using the determined attack parameters.
2. Random vector sequence: values from the binary set {0, 1} are generated randomly.

To determine the detection capabilities of the proposed method, we have generated an array of 100352 events that were analysed by the proposed logical circuits. As described in Section 4, the proposed early detection method consists of three filters: network filter (NF), system filter (SF) and network flags filter (LF). The generated array was analysed

by these filters separately and the obtained results were combined to determine the attack action formed out of events. The generated 100352 events consist of 100053 randomly generated events and 299 events that have the parameters of known attacks. NF, SF and LF filters identification values are shown in Table 7. Eight parameters are shown in the table of the filters: seven parameters that indicate a detection of an attack: HS, LA, PS, SE, SP, SST, SSV, and a parameter 0 that corresponds to non-malicious traffic.

Table 7. NF, SF and LF filters attack identification value

Actions	Filter NF			Filter SF			Filter LF		
	No. of events	Randomly generated events	Deterministically generated events	No. of events	Randomly generated events	Deterministically generated events	No. of events	Randomly generated events	Deterministically generated events
HS	6242	95%	5%	7632	96%	4%	3728	92%	8%
LA	1790	83%	17%	2133	86%	14%	4068	93%	7%
PS	438	32%	68%	7632	96%	4%	2979	90%	10%
SE	944	68%	32%	888	66%	34%	6251	52%	48%
SP	1790	83%	17%	6997	96%	4%	3535	92%	8%
SST	616	51%	49%	888	66%	34%	2162	86%	14%
SSV	1802	83%	17%	6997	96%	4%	3429	14%	86%
0	8610	10%	0%	6710	10%	0%	8210	10%	0%

As it was possible to predict, the largest values are for non-malicious traffic, which are shown in the last row of Table 7. The values 0% in this row show the very important obtained result that all the deterministically generated events were detected as malicious. The biggest number of detected events in the filter NF was HS and the lowest – PS. HS actions also had a high detection ratio in the filter SF. The mostly noticeable difference between the filters SF and NF was that PS action in the filter SF had a high detection ratio of randomly generated events. The filter LF showed much smaller number of events in comparison with the filters NF and SF for action HS. Such differences in action detection ratios among filters show methods specificity. In the Table 8, an aggregated form out of three filters for actions HS, LA, and PS is shown, which indicates the detection of the first stage – Reconnaissance. The obtained result confirms that the proposed method is able to

detect all the deterministically on purpose generated events. The proposed method has also detected a number of randomly generated events, which varies depending on the action.

For the second part of the experimental set, we have generated an array of randomly selected 100352 events. The objective of this part of the experiment is to evaluate the possibility to create an attack randomly. The results of three filters NF, SF and LF and accumulation results $A = NF$ & SF& LF are shown in Table 9.

Table 8. Aggregated form of three filters

Name	HS	LA	PS
Detected events	1154	305	316
Detected randomly generated events	855	6	17
Detected deterministically generated events	299	299	299
Deterministically/TOTAL %	26%	98%	95%

Table 9. The results of filters NF, SF, LF and accumulated detection

Action	Filter *NF*		Filter *SF*		Filter *LF*		Accumulated detection	
	No of detected events	% of total events	No of detected events	% of total events	No of detected events	% of total events	No of detected events	% of total events
HSL	6424	6%	7616	8%	3790	4%	1181	1%
A	1784	2%	2165	2%	4107	4%	301	0,3%
PS	466	0%	7616	8%	2922	3%	312	0,3%
SE	967	1%	874	1%	609	1%	289	0,3%
SP	1784	2%	6653	7%	3495	3%	310	0,3%
SST	621	1%	874	1%	2148	2%	270	0,3%
SSV	1803	2%	6653	7%	320	0%	2	0%
"0"	86503	86%	67901	68%	82961	83 %	97687	97,6 %

The main part of the traffic is generated randomly for the both parts of the experiment. Therefore, we can compare a detection of malicious actions in random traffic in Table 7 and in Table 9. The main stream of the randomly generated traffic is non-malicious (see the last rows of Table 7 and Table 9). Moreover, the obtained numbers of the non-malicious traffic are quite similar in both tables, e.g. filter NF showed 86730 events for action HS in Table 7 and filter NF showed 86503 events for action HS in Table 9. The same is true for detection of malicious events, as well. For example, filter SF showed 2133 events for action LA in Table 7 and filter SF showed 2165 events for action LA in Table 9.

We carried out the experiment on the real attack data that were taken from open databases. The experiment was carried out on the network of virtual machines. The results of the experiment are presented in Table 10.

Table 10. The results of detection of real attacks

No.	Simple or complex attack	Type of attack	Number of attacks	Detection (%)
1.	Simple targeted attack	Syn Flood	37	92 %
		Ack Flood	22	91 %
		IP fragment attack	20	80 %
		Xmas scan	16	81 %
		Password Bruteforce	70	87 %
2.	Complex attack	Cryptolocker	20	90 %
		Wannacry	15	80 %

We can observe (see Table 10) that our proposed method can detect the real simple and complex cyber-attacks at their early stages. Not all the cyber-attacks are detected. The least percent of the detection is 80. Not all the cyber-attacks follow our introduced rules for the attack detection at the early stages.

Our experiments confirmed that the proposed method is capable to detect the early stages of the cyber-attacks in the network traffic. The method showed that the randomly generated traffic consists of 1,6% events indicating reconnaissance stage (the first stage), 0,3% events indicating weaponization stage (the second stage) and 0,3 % events indicating delivery stage (the third stage). The proposed method is a part of a larger work that is oriented to a near real-time cyber-attack detection.

6. Conclusions

Scientific and technical literature analysis and good practice show that the current system of response to cyber threats using IDS, IPS and IRS systems has a number of shortcomings, the main problem is that they start up only when a cyber-attack is taking place, i.e. such a system does not play a preventive role.

Intelligent cyber-attacks are characterized using certain stages. To determine the precautionary stage, when preventive measures can "kill the chain", the identification of those stages must be complete as possible. In this paper, we suggested to consider an attack chain of nine steps to describe a cyber-attack vector.

The paper proposes a method to detect an intelligent cyber-attack, which takes several preparation steps, and which is the most dangerous one, in the early stages of the cyber-attack. The method is based on the use of several logical filters. We have built the analytical aggregated expressions for the detection of threats caused by the early stages of the cyber-attacks.

The mode to detect the early stages of the cyber-attack may be appropriate for both standard information systems and small-sized mobile devices, since the suggested method is suitable for processing data on devices with a limited memory and computing power.

The experiments to test the ideas implemented in the proposed method were carried out. The essence of the experiments was to evaluate the reliability of the suggested method. All the values, which were generated deterministically for the attack, were identified as the malicious ones. The proposed method was able to detect many real simple and complex cyber-attacks at their early stages. In our opinion, such a result shows a good base for further work in increasing the sensitivity of the method to other forms of the cyber-attacks.

References

1. Symantec.: Preparing of a cyber-attack. Available online: http://symc.ly/1PHHl3n, accessed on 02 June 2018.
2. Kearney, A. T.: Information security: preparing for the next hack attack. Available online: http://bit.ly/2vtIwkM, accessed on 30 June 2018.
3. Yadav, T., Rao, A. M.: Technical aspects of cyber kill chain. Proc. of Security in Computing and Communications: Third International Symposium on Security in Computing and Communication, Kochi, India, 10-13 August 2015, pp. 438-452, Springer, Switzerland.
4. Husak, M.: Early detection and mitigation of multi-stage network attacks. PhD thesis, Masarykova Univerzita Fakulta Informatiky, Brno, Czech. Available online: https://is.muni.cz/th/ccz8a/thesis_proposal.pdf, accessed on 12 July 2018.
5. Morinaga, M., Nomura, Y., Furukawa, K., Temma, S.: Cyber-attack countermeasure technologies using analysis of communication and logs in internal network. Fujitsu Scientific and Technical Journal, 52 (3), 2016, 66-71.
6. Siddique, K., Akhtar Z., Lee H., Kim, W., Kim, Y.: Toward bulk synchronous parallel-based machine learning techniques for anomaly detection in high-speed big data networks. Symmetry, 9(9), 2017, 197.
7. Sharifi, A. A., Noorollahi, B. A., Farokhmanesh, F.: Intrusion detection and prevention systems (IDPS) and security issues. International Journal of Computer Science and Network Security. 14 (11),2014, 80-84.
8. Rathore, M.M., Ahmad, A., Paul, A.: Real time intrusion detection system for ultra-high-speed big data environments. J Supercomput, 72, 2016, 3489-3510.
9. Al-Yaseen W. L., Othman A. L., Nazri M. Z. A.: Real-time multi-agent system for an adaptive intrusion detection system. Pattern Recognition Letters, 85,2017, 56–64.
10. SANS Institute InfoSec Reading Room.: IDS - today and tomorrow. Available online: https://www.sans.org/reading-room/whitepapers/detection/ids-today-tomorrow-351, accessed on 22 August 2018.
11. Rajan, S. S., Cherukuri, V. K.: An overview of intrusion detection systems. Proc. IDT Workshop on Interesting Results in Computer Science and Engineering (IRCSE). Available online:https://pdfs.semanticscholar.org/012c/3afab09d34bad3d62cb499f2f57c40675062.pdf, accessed on 22 August 2018.
12. Guevara, C., Santos, M., López V.: Data leakage detection algorithm based on task sequences and probabilities. Knowledge-Based Systems, 120, 2017, 236-246.

13. Kashyap, S., Agrawal, P., Pandey, V. C., Keshri, S. P.: Importance of intrusion detection system with its different approaches. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 2 (5), 2013, 1902-1908.
14. Werlinger, R., Hawkey, K., Muldner, K., Jaferian, P., Beznosov, K.: The challenges of using an intrusion detection system: is it worth the effort? Proc. of ACM Symposium on Usable Privacy and Security (SOUPS), Pittsburgh, Pennsylvania, USA, 23-25 July,2008, pp. 107-118, ACM, New York.
15. Lo, C.H., Ansari, N.: Consumer: a novel hybrid intrusion detection system for distribution networks in smart grid. IEEE Transactions on Emerging Topics in Computing, 1 (1),2013, 33-44.
16. Singh, K., Tamrakar, S.: A review of intrusion-detection system- clustering and classification using RBF and SOM networks. International Journal of Emerging Technology and Advanced Engineering, 5 (7), 2015, 502-505.
17. Ghazi Z., Doustmohammadi A.: Intrusion detection cyber-physical systems based on Petri net. Information Technology and Control, 47 (2), 2018, 220-235.
18. Chi, Y., Jiang, T., Li, X., Gao C.: Design and implementation of cloud platform intrusion prevention system based on SDN. Proc. 2nd IEEE International Conference on Big Data Analysis (ICBDA), Beijing, Peoples R China, March 10-12, 2017, pp. 847-852.
19. Xing, T., Huang, D., Xiong, Z., Medhi, D.: SDNIPS: enabling software-defined networking-based intrusion prevention system in clouds. Proc. of International Conference on Network and Service Management (CNSM), Rio de Janeiro, Brazil, 17-21 November,2014, pp. 308-311.
20. Ragsdale, D. J., Carver, C. A., Humphries, J. W., Pooch, U. W.: Adaptation techniques for intrusion detection and intrusion response systems. Proc. of the IEEE International Conference on Systems, Man, and Cybernetics, Nashville, TN, USA, 8-11 October,2000, pp. 2344–2349.
21. Shameli-Sendi, A., Cheriet, M., Hamou-Lhadj, A.: Taxonomy of intrusion risk assessment and response system. Computers and Security. 45, 2014, 1–16.
22. Moon, D., Im, H., Lee, J. D., Park, J. H.: MLDS: Multi-layer defense system for preventing advanced persistent threats. Symmetry. 6, 2014, 997-1010.
23. Heo, S., Lee, S., Doo, S., Yoon, H.: Design of a secure system considering quality of service. Symmetry, 6,2014 938-953.
24. Yan, X., Zhang, J. Y.: Early detection of cyber security threats using structured behavior modeling. ACM Transactions on Information and System Security. Available online:http://www.cs.cmu.edu/~xiaohuay/papers/draft_TISSEC.pdf, accessed on 12 July 2018.
25. Vincent, H., Wells, L., Tarazaga, P., Camelio, J.: Trojan detection and side-channel analyses for cyber-security in cyber-physical manufacturing systems. Procedia Manufacturing, 1, 2015,77-85.
26. Chen, M. C., Yang, P. Y., Ou, Y. H., Hsiao, H. W.:Targeted attack prevention at early stage. Proc. 28th International Conference on Advanced Information Networking and Applications Workshops. Victoria, BC, Canada, 13-16 May, pp. 866-870, 2014.
27. Bhattacharya, S., Selvakumar, S.: Multi-measure multi-weight ranking approach for the identification of the network features for the detection of DoS and probe attacks. *The Computer Journal*, 59 (6),2016, 923–943.
28. Cheng, J., Zhou, J., Liu, Q., Tang, X., Guo, Y.: A DDoS detection method for socially aware networking based on forecasting fusion feature sequence. The Computer Journal, 61 (7),2018, 959–970.

Vacius Jusas graduated from Kaunas Polytechnic Institute (Lithuania) in 1982. He received the D.Sc. degree from Kaunas Polytechnic Institute in 1988. Since 2006, he is professor at Department of Software Engineering, Kaunas University of Technology, Lithuania. He is author and co-author of more than 100 papers. He is Editor of journal "Information Technology and Control". His research interests include brain-computer interface, adaptive signal processing, forensics investigation, cybercrime.

Saulius Japertas graduated Vilnius University in 1982. He received his Ph.D. degree from the Lithuanian Energy Institute in 1991. Since 1992 he joined to Lithuania Army Air forces as a head of communication department and from 2003 to 2009 (retired) was a head of CIS Service under MoD. Since 1999 to 2018, he was an assoc. prof. in Electrical and Electronics faculty and since 2018 he is the assoc. prof in Mechanical engineering and design faculty, Kaunas University of Technology. He is author and co-author of more than 40 papers. His research interests include wireless networks, security and protection of electronics and IT&T networks.

Tautvydas Baksys graduated Kaunas University of Technology (Lithuania) in 2012. He received Master degree in Electronics. Now, he is a PhD student in Electronics and Electrical Engineering at Kaunas University of Technology. He is author and co-author of 3 papers. His research interests include wireless networks, security and protection of electronics and IT&T networks.

Sandeepak Bhandari graduated from I.K. Gujral Punjab Technical University (India) in 2013. He received Master of Technology in Computer Science and Engineering from I.K. Gujral Punjab Technical University. Now, he is a PhD student in Electronics and Informatics Engineering at Kaunas University of Technology. He is author and co-author of 14 papers. His research interests include wireless networks and security, forensics investigations and Data mining.

Received: January 22, 2019; Accepted: May 20, 2019