Centralized Distributed System for Cyberattack Detection in Corporate Network Based on Multifractal Analysis

Anastasiia Nicheporuk[a], Serhii Tabenskyi[b], Pavlo Rehida[a], Oleg Savenko[a], Andrii Nicheporuk[a]

[a] *Khmelnytskyi National University, Instytutska str., 11, Khmelnytskyi, 29016, Ukraine*
[b] *National Academy of the State Border Guard Service of Ukraine named after Bogdan Khmelnitsky, Shevchenko str., 46, Khmelnytskyi, 29000, Ukraine*

Abstract

The paper presents a centralized distributed system for cyberattack detection in corporate network based on multifractal analysis. According to the study the concept of a system was developed, which combines the requirements of centralization, distribution and self-organization. The proposed distributed system for cyberattack detection in corporate network is implemented in the form of software. The functioning of the detection module is achieved by involving the method of detecting cyberattacks based on multifractal traffic analysis. The KDD Cup data set what used information about normal traffic and known network attacks was used. Experimental studies with a centralized distributed network attack detection system in computer networks have demonstrated detection overall accuracy at the level of 91% with an average data processing time of 8 seconds.

Keywords

Network traffic, centralized distributed systems, anomaly detection, multifractal analysis, cyberattack

1. Introduction

With the growing of information transmitted over the network the number of hardware and software to interfere with the data transmission process increases. Every day, more and more Internet users fall victim to malicious software [1, 2]. Particularly dangerous are attacks on corporate networks that exploit new vulnerabilities or methods of attack, which leads to various kinds of damage to users of such networks. Therefore, researchers are focused on finding new approaches that would minimize human interference in the business logic of cyberattack detection and prevention systems, as well as on the development of distributed attack detection systems on corporate networks that will satisfying the principles of centralization and self-organization and will involve the computing power of many components in the network.

The paper proposes the architecture and concept of centralized distributed system for cyberattack detection in corporate network based on multifractal analysis. The proposed distributed system for cyberattack detection in corporate network is implemented in the form of software.

2. Related works

Today, the problem of detecting malicious software and cyberattacks based on them is given considerable attention. Modern methods of detecting malicious software and cyberattacks at the

IntelITSIS'2022: 3rd International Workshop on Intelligent Information Technologies and Systems of Information Security, March 23–25, 2022, Khmelnytskyi, Ukraine
EMAIL: eldess06@gmail.com (A. Nicheporuk); tlkkaf16@gmail.com(S. Tabenskyi); pavlo.rehida@gmail.com (P. Rehida); savenko_oleg_st@ukr.net (O. Savenko); andrey.nicheporuk@gmail.com(A. Nicheporuk)
ORCID: 0000-0001-5366-5792 (A. Nicheporuk); 0000-0002-4104-745X(O. Savenko) 0000-0002-7230-9475(A. Nicheporuk)

core include the use of signature analysis and behavioral analysis [3-18] (these methods are usually based on machine learning methods that include classification and clusterization algorithms [18, 19]). Signature methods are to some extent limited, as they involve matching patterns with a set of defined templates. In the case of application of obfuscations (for example, in metamorphic viruses [5-6]) to piece of the code or the choice of alternative branches of execution, the signature method is not able to fully identify the threat.

Therefore, one of the main areas of detection of threats is observe the features of malicious activities and recognize of abnormal behavior. In general, the process of detecting abnormal activity involves tracking historical changes in the study environment, forming a normal "trajectory" of the object and subsequent monitoring and analysis of features that deviate from the data of this trajectory for a certain period operation.

In [8] authors proposed a model based on the behavior of dendritic cells, as well as their interaction with the human immune system. The basis of the presented approach is the dendritic cell algorithm, which was combined with Multiresolution Analysis Maximal Overlap Discrete Wavelet Transform. In order to make decision and distinction between normal and abnormal behavior the authors presented a binary classifier that aims to analyze a time-frequency representation of time-series data.

In [9] authors proposed a method for detecting bots in real time, which allowed to make decisions based on the binary classifier for streams of Web server requests. As a machine learning algorithm, a neural network was used, which after training was used to classify of each incoming HTTP request, followed by sequential probabilistic analysis to assess the relationship with subsequent HTTP requests within each session.

The authors in [10] try to solve the problem of detecting malicious network activity by finding solutions for scalability in all stages of cyberattacks detection: network traffic monitoring, data collection, feature extraction and deployment of different algorithms that provide statistical analysis or technics of machine learning. In theirs framework Big Data Platform Hadoop for storage data is used.

Some approaches are focused on finding anomalies based on DNS [11-13], in particular on the basis of active and passive data collection. In [11] proposed approach of abnormal source IPs detection based on Local Outlier Factor algorithm. Authors in [13] examine the anomaly of DNS network traffic to extract significant enriched features. Two machine learning algorithms were used to analyze the obtained features. As result a novel hybrid rule detection model that utilizes the output of two algorithms was presented.

Another approach of anomalies detection in computer networks is use of honeypots. In [14] authors present a concept of the virtual environment in which malicious samples manifest own activities which cannot break work of the host system is offered. Such approach allowed to accumulate information about vulnerabilities and attacks. They focus on detecting attacks on low-interaction server honeypots

Detection of malicious activity is possible by means of use of a blockchain technology. In [15] authors present dynamic botnet detection framework PAutoBotCatcher that utilize concept of blockchain. Based on the results of the study, the authors proposed a number of optimization techniques, including caching of detection's output and pre-processing of shared network traffic, which allowed to increase the reliability of detection.

In addition to host-based detection methods, much of the approaches are devoted to involving several network components for the joint detection of harmful activity. It can be distributed decentralized system or traditional client-server architecture [16, 17].

A review of previous research has shown that modern approaches use different tools and methods to detecting abnormal behavior and are characterized by a fairly high efficiency of detecting threats. However, these systems are not able to detect new threats that lead to a breach of the decision-making center, thus putting the whole system on a par with the victim.

3. Concept of a centralized distributed system for cyberattack detection in corporate network based on multifractal analysis

During the design of a proposed centralized distributed attack detection system in corporate networks was taken as a basis on a service-oriented approach that allows to create flexible systems with the possibility of further improvement and dynamic expansion. Each object in such system we call a component. Combining components is done by remotely calling procedures. In the case of a distributed system, the called objects do not have to be executed on the same machine where the call was made. An object-oriented architecture is attractive in that it encapsulates data (object states) and operations (methods) into a single entity. The interface provided by the object hides the details of the implementation, making it independent of the environment. That is why the object can be considered as a separate entity, which may also include other services (representation of the application as a set of different services [20, 21]).

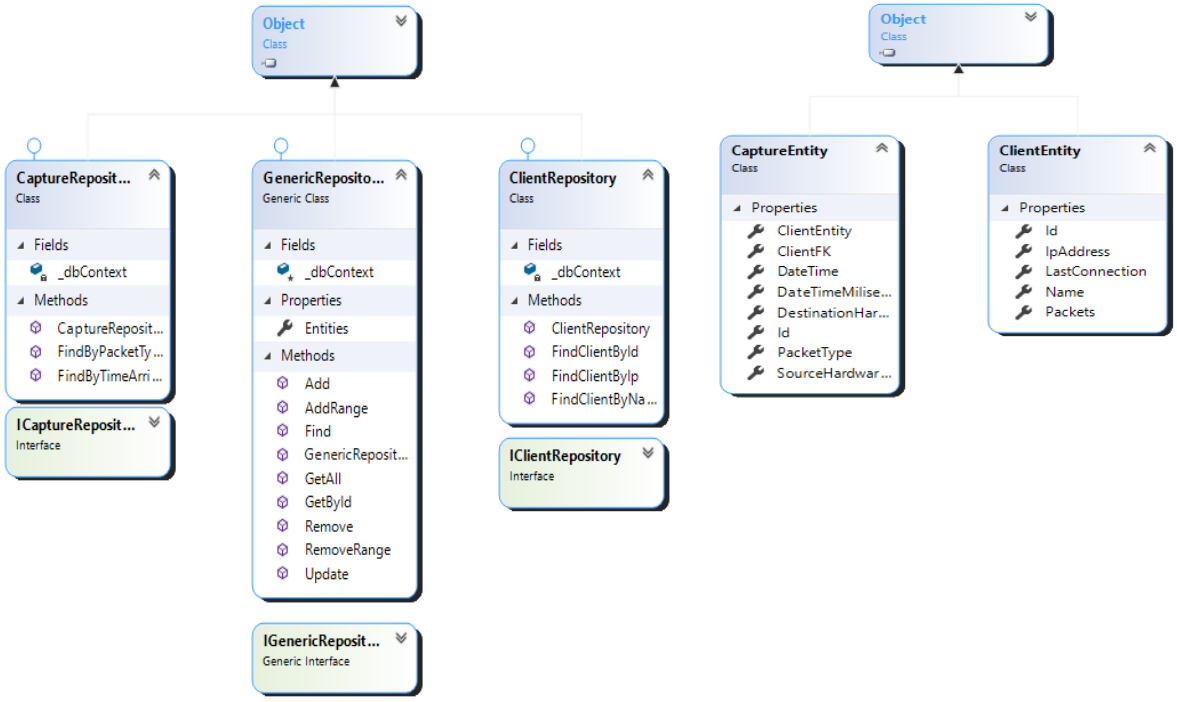
In the proposed distributed system, the server acts as a data warehouse and decision center. The client part collects network information and transfers data to the server. The components are connected through two-way communication between the client and the server in order to make a timely decision in the event of a network attack. Maintaining the integrity and resilience of the system is based on the dynamic choice of the leader, in case of division of the cluster into parts or failure of the current control node. With such an organization, in case of detection of suspicious activity of a node in the network, a second poll is conducted, which withdraws the results of the first, resulting in communication with this node is blocked throughout the network. The main condition for maintaining integrity is the presence of nodes that can take over the management role, which will ensure the fault tolerance of the system in the event of a successful network attack. The number of such nodes should be more than half, to ensure a quorum - the minimum number of control units and the smooth operation of the cluster. For example, if the network consists of 9 nodes, including the control center, the network attack resulted in the separation of nodes, after which the control center manages only 4 nodes out of 8, while others do not have a leader. The control center selection process begins when the regular node does not receive instructions from the manager for a long time. Then this node receives the status of a candidate. Other nodes vote for the candidate from whom they received the first request. After the configuration change we have two independent networks. The operation of the system continues, as the decision center was chosen new for the second group of nodes. After reconnection, the clusters merge to reset the system components and restore the default decision center.

The proposed distributed system for cyberattack detection in corporate network is implemented in the form of software and consists of the following components:

- Web client application to display statistical information;
- Configuring database entities;
- Setting up database tables and relationships from them;
- Repository – a repository of methods for remote procedure calling;
- Common settings of all components;
- Client part – a software application for intercepting and analyzing traffic «on-the-fly».

The components are structurally related and interdependent. Thus, a multilevel structure was formed. To solve the problem of a distributed system, software components must interact and at the same time be autonomous. This is achieved by calling the functions of remote components, which is implemented in the repository. Data exchange takes place through a relational database. The client application consists of the following components: Worker background run module, Capture Handler packet event handler, CaptureTask algorithm, PcapDevice network adapter detection and connection interface, client information collection interface, and database connection method. The structure of the client application and entities of database are shown in Fig. 1a and Fig. 1a, respectively.

The relationships in the tables are formed so that for each unique client a sample of network packets is formed. The method of data connection is common to all components of the solution. This component is used to quickly modify the connection string in the event of a change in the database server address.



a) b)

Figure 1: The structure of the client application (a); entities of database(b)

4. Detection of cyberattacks and decision-making in a centralized distributed system based on multifiractal analysis

The primary functions of the proposed centralized distributed system are to detect cyberattacks in corporate networks. The proposed system is based on the method of detecting cyberattacks based on multifiractal traffic analysis. Let's consider in more detail proposed method.

The use of a multifractal approach means that some objects can be divided into parts that have their own similarity characteristics that are different from others. Network traffic is self-similar at some time intervals [22]. Therefore, for its analysis we use the method of maxima of wavelet transform modulus [23], which allows to determine the features of the signal. Wavelet analysis applies to construct the coefficients used in the distribution of the output signal to the basic functions. The signal can be the intensity of network traffic or the correlation data of the final IP addresses. Wavelet transform allows to convert the most important data into a signal that corresponds to the specified oscillation amplitude and discard less useful information with a small amplitude, classifying it as noise.

Let's present the process steps of analysis of the parameters of the multifractal spectrum in the form of the following algorithm:

1) Decomposition of the output signal $f(t)$ by the coefficients of the parent wavelet $\psi(t)$:

$$W_f(u,j) = \left(f(t), \psi_{u,s}(t)\right) = 2^{-j/2} \int_{-\infty}^{\infty} \frac{t-u}{2^j} dt, \quad (1)$$

where u a scale parameter, j is a spatial coordinate (time);

2) In the array of coefficients find the positions of local maxima $\{u_p(j)\}_{p \in Z}$ and find their absolute value and form an array of maxima:

$$\left|W_f(u_p, j)\right|. \quad (2)$$

3) Define the partition function:

$$S(q,j) = \sum_p \left|W_f(u_p, j)\right|^q. \quad (3)$$

4) For each $q \in R$ calculate the scale coefficient:

$$\tau(q,j) = lim \inf_{j \to 0} \frac{\ln S(q,j)}{\ln 2^j}. \quad (4)$$

5) Calculate the multifractal spectrum using the Legendre transformation:

$$f_L(\alpha) = \min_{q \in R}[q(\alpha + 1/2) - \tau(q)]. \quad (5)$$

6) For each interval *j* calculate multifractal dimensions of order *q:*

$$D_{q,j} = \frac{1}{q-1}[q(\alpha(q,j) - f(\alpha(q),j)]. \quad (6)$$

In the proposed algorithm for analyzing network traffic, its intensity is selected, i.e. the number of sent and received packets per unit time.

Based on the algorithm of multifractal spectrum analysis, the process of detecting intrusions will be presented as follows: Let X be the time series of normal traffic, Y be the time series of malicious traffic, and Z be the time series of anomalies, hence Y = X + Z. Regardless of the presence of self-similarity properties in the time series of anomalies, Y will still be a self-similar process if X is a stationary self-similar process. However, the degree of self-similarity may change. Let $s_X s_Y s_Z$ autocorrelation functions for X, Y and Z respectively. Then during the cyberattack we focus on $\|s_Y - s_X\|$, with $s_Y = s_X + s_Z$. For each $H \in (0.5, 1)$ there is only one autocorrelation function with self-similarity. Therefore it is considered $\|H_Y - H_X\|$, where H_Y and H_X – the average value of Hurst X and Y, respectively. The Hurst index is introduced to increase the accuracy of estimating the self-similarity of the system. The disadvantage of the approach is the need to restart the definition of the self-similarity of traffic for each scale. Therefore, a signal of a change in self-similarity will be given regardless of whether it exists for another scale. After determining the network cyberattack, the traffic is divided into several parts. The intensity of the cyberattack can be determined by analyzing the Hurst index and the rate of change, i.e. the difference between the Hurst index before and after the cyberattack. General algorithm of involving multifractal analysis for network traffic and anomaly detection process are shown in Fig. 2.

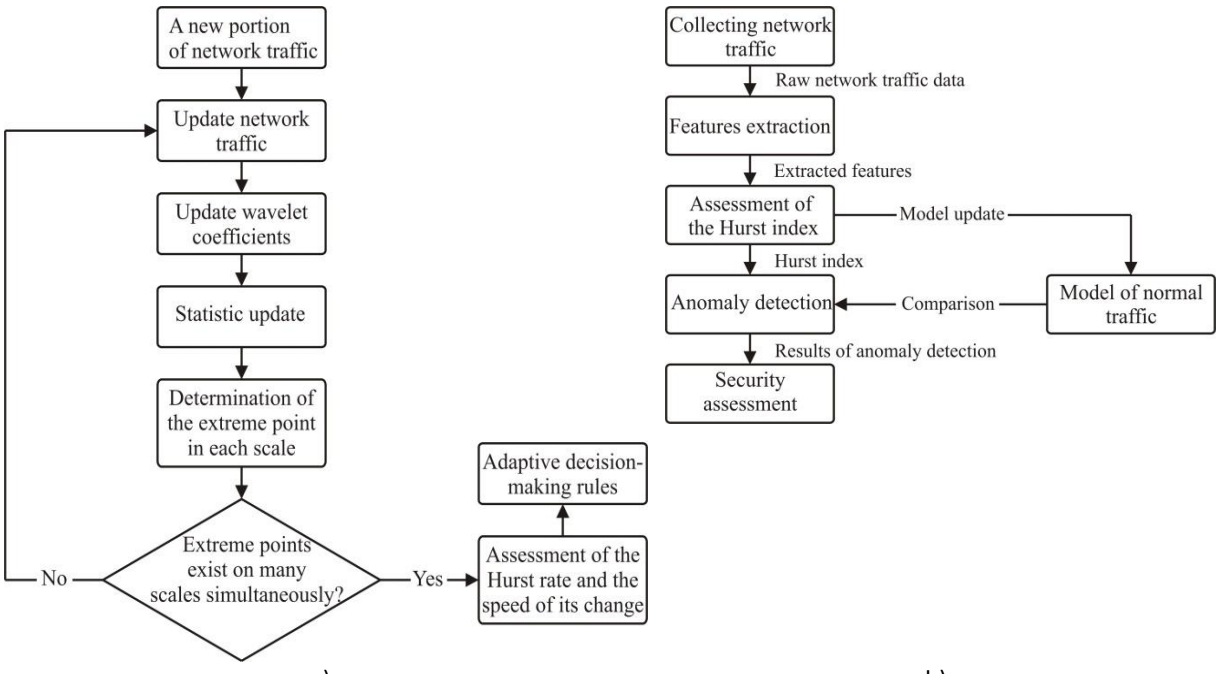


Figure 2: The process of involving multifractal analysis for network traffic (a); anomaly detection process (b)

Thus general algorithm of anomaly detection process caused by a cyberattack can be presented as follow (Fig. 2b):

1) Traffic collection;
2) Statistical analysis;
3) Estimation of the Hearst index;
4) Determination of anomalies;
5) Security assessment.

To reduce the impact on network performance, traffic is duplicated on the server that collects network information from each of the connected clients.

The presented method of cyberattacks detection based on the use of multifractal analysis is a part of a distributed system, which is characterized by the following features: recommended network speed of 1 Mbit/s; the need for data storage for the server component in the amount of 1TB; the need for separate network interfaces for each individual component of the distributed system; availability of installed NPcap and libpcap libraries for components running Windows and Linux, respectively. Testing of the developed distributed centralized system took place in a configuration of 3 hosts. Running client software is resource-intensive and invisible to the user as it runs as a background service. An example of software implementation of a centralized distributed system based on multifiractal analysis is presented on the fig.3 and fig.4.

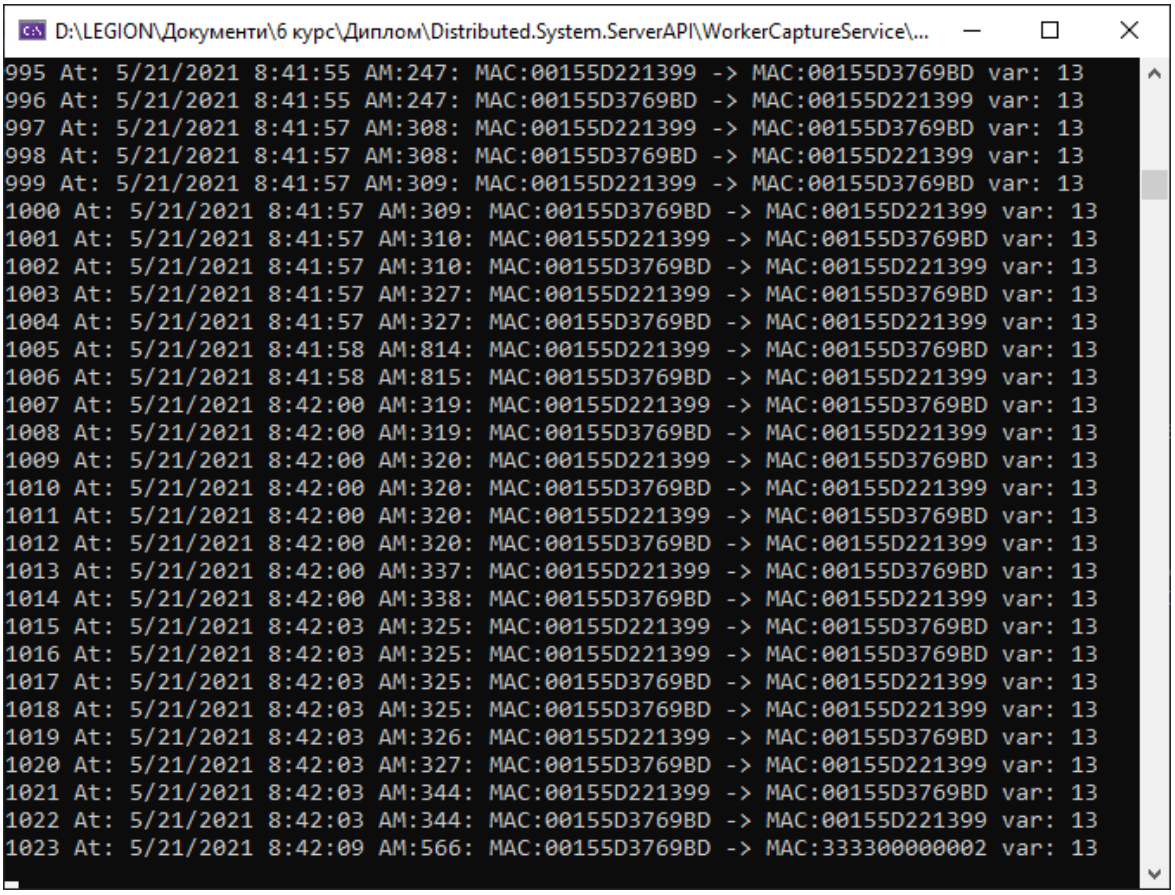


Figure 3: Client activities (client sends network packet data to the database server for further processing and storage of traffic history)

4.1. Detectors of system

The detectors in the proposed distributed centralized system are modules of the endpoints (hosts) of the network, which register network packets and perform their processing. The following fields are used as features from network traffic packets: total number of packets, number of TCP, UDP and ARP packets per unit time for a specific network node.

The KDD Cup data set [24] is used as the reference traffic models with which the received traffic is compared, which contains information about normal traffic and known network attacks, the list of which is given in Table 1.

Id	DateTime	DateTimeMiliseconds	DestinationHardwareAddress	SourceHardwareAddress	PacketType	ClientFK	ClientEntityId
2084	2021-05-21 08:41:57.3089430 +00:00	308	00155D221399	00155D3769BD	IPv4	0	13
2085	2021-05-21 08:41:57.3094060 +00:00	309	00155D3769BD	00155D221399	IPv4	0	13
2086	2021-05-21 08:41:57.3094830 +00:00	309	00155D221399	00155D3769BD	IPv4	0	13
2087	2021-05-21 08:41:57.3102680 +00:00	310	00155D3769BD	00155D221399	IPv4	0	13
2088	2021-05-21 08:41:57.3103830 +00:00	310	00155D221399	00155D3769BD	IPv4	0	13
2089	2021-05-21 08:41:57.3274730 +00:00	327	00155D3769BD	00155D221399	IPv4	0	13
2090	2021-05-21 08:41:57.3276130 +00:00	327	00155D221399	00155D3769BD	IPv4	0	13
2091	2021-05-21 08:41:58.8146470 +00:00	814	00155D3769BD	00155D221399	Arp	0	13
2092	2021-05-21 08:41:58.8151470 +00:00	815	00155D221399	00155D3769BD	Arp	0	13
2093	2021-05-21 08:42:00.3190200 +00:00	319	00155D3769BD	00155D221399	IPv4	0	13
2094	2021-05-21 08:42:00.3191700 +00:00	319	00155D221399	00155D3769BD	IPv4	0	13
2095	2021-05-21 08:42:00.3200490 +00:00	320	00155D3769BD	00155D221399	IPv4	0	13
2096	2021-05-21 08:42:00.3201390 +00:00	320	00155D221399	00155D3769BD	IPv4	0	13
2097	2021-05-21 08:42:00.3208970 +00:00	320	00155D3769BD	00155D221399	IPv4	0	13
2098	2021-05-21 08:42:00.3209840 +00:00	320	00155D221399	00155D3769BD	IPv4	0	13
2099	2021-05-21 08:42:00.3379560 +00:00	337	00155D3769BD	00155D221399	IPv4	0	13

Figure 4: A snippet of the user's network activity history with ID 13, which is related to the data of the table of connected users

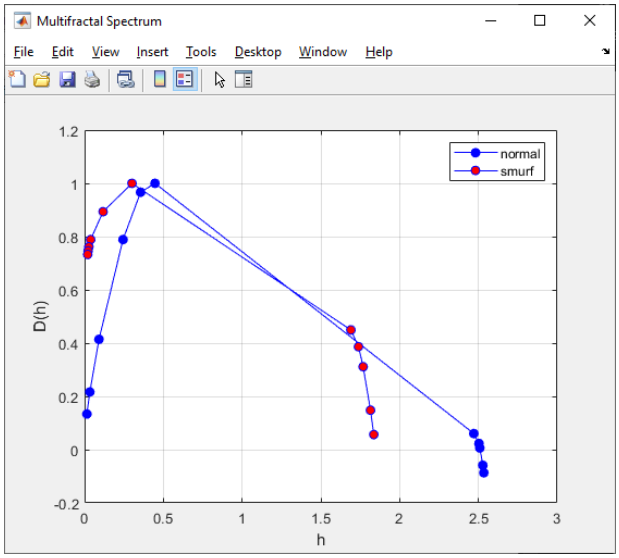
Table 1
Type of network cyberattacks used as data model for proposed system

Type of network cyberattacks	Subtype
DOS	Back, Land, Neptune, Pod, Smurf, Teardrop.
Probe	Ipsweep, Nmap, Portsweep, Satan
U2R	Buffer_overflow, Loadmodule, Perl, Rootkit,
R2L	Ftp_write, Guess_password, Imap, Multihop, Phf, Spy, Warezclient, Warezmaster

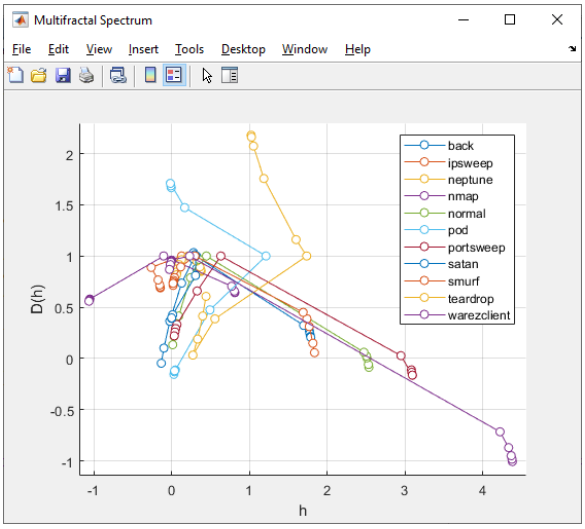
Studies [22] show that network traffic has signs of self-similarity, so there is no need to process all the data obtained. Therefore, in order to speed up the system and make timely decisions in case of intrusion, only part of the network traffic is analyzed. To convert the data collected by the user component into a user-friendly format, you need to select the scale and calculate the number of network packets in each time period. Next, using the developed method to construct a multifractal spectrum and compare observed data with normal traffic. In case of significant discrepancies, the system analyzes the similarity with known cyberattacks attacks to determine the type of network cyberattack.

We formulated decision making algorithm as follows: if all points have deviations from the values of normal traffic less than threshold value Δ, than we consider the value of the coordinate as normal points; if significant deviations are present in no more than ε than network traffic is suspicious; if more than ε points have deviations above Δ - there is a network attack. Based on experiments values Δ and ε were chosen as 0.15 and 0.29 respectively.

Fig. 5 demonstrates the comparison of network traffic: normal traffic and activity during a network attack (fig. 5a) and difference between multifractal spectra of all considerable malicious type of network traffic and normal traffic (fig. 5b).



a)



b)

Figure 5: Comparison of network traffic: normal traffic and activity during a network cyberattack (a) and difference between multifractal spectra of all considerable malicious type of network traffic and normal traffic (b)

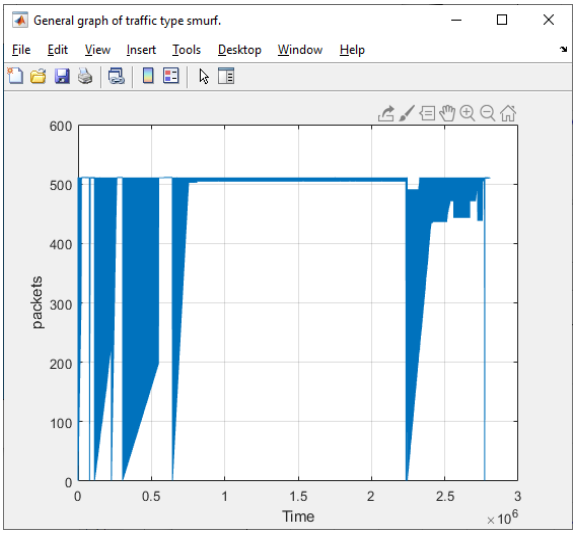
5. Experiments

In order to evaluate proposed detection method that is a part of centralized system, a package of Matlab analysis and programming applications was used [25-27]. In order to visualize the multifractal spectrum, the program code presented in Appendix A was implemented.

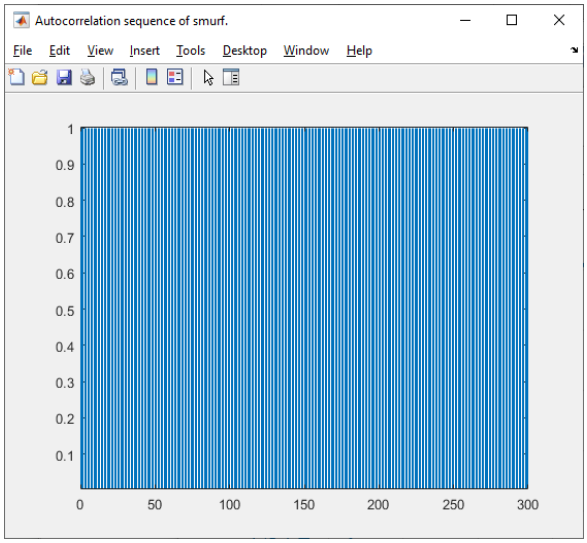
The KDD Cup data set containing information about normal traffic and known network attacks was used as a training traffic model with which to compare the observed traffic.

This set was converted to a table data with means of Matlab. In this data set, each row can be considered as activity statistics per unit time.

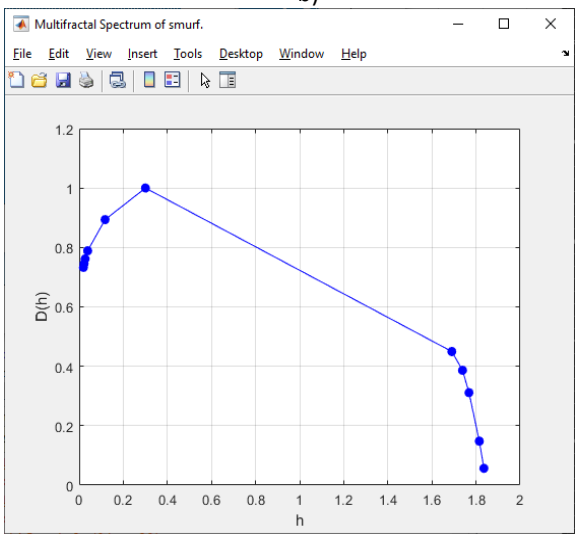
For example, for a smurf network cyberattack, the network traffic graph, autocorrelation sequence graph, and multifractal spectrum are shown in Fig. 6.



a)



b)



c)

Figure 6: Smurf network cyberattack detection results: activities of network traffic (a); autocorrelation sequence (b); multifractal spectrum (c)

In order to test the accuracy rate of multifractal analysis to detect network attacks in the corporate network, a number of experiments were conducted.

Experimental studies evaluating network attack detection based on multifractal analysis yielded the following results: 91% attack detection accuracy (generalized value, Table 2), total data processing time is about 8 seconds (total amount of data is 4.9 million rows, size of data more than 700 megabytes).

Table 2

Accuracy and false positive alarms of detecting network attacks using multifractal analysis

Attack type/metric	Accuracy, %	False Positive,%
DOS	91	5.1
Probe	92	2.4
U2R	87	2.7
R2L	94	6.4

Table 2 shows the generalized results of the experiments, namely the values of the detection accuracy and the 1-st type errors. These values are considered as the average detection results at different time intervals of the analysis of network traffic. It should be noted that in the case of insufficient values in the autocorrelation sequence it is impossible to form a multifractal spectrum. This is influenced by the number of cases of a certain type of attack on the timeline.

6. Conclusions

As a result of the study, the architecture of a distributed network attack detection system as well as its software implementation were developed, which combines the requirements of centralization, distribution and self-organization. The proposed system is grounded on the method of detecting attacks based on multifractal traffic analysis.

Experimental studies with a centralized distributed network attack detection system in computer networks have demonstrated detection accuracy at the level of 91% with an average data processing time of 8 seconds. However, it should be noted that the proposed method has limitations, if there are insufficient values in the autocorrelation sequence, it is impossible to form a multifractal spectrum.

7. References

[1] What Is a DDoS Attack and How to Stay Safe from Malicious Traffic Schemes, 2021. URL: https://www.mcafee.com/blogs/tips-tricks/ddos-attack-work/

[2] T. Sochor, N. Chalupova, Interpersonal Internet Messaging Prospects in Industry 4.0 Era, Recent Advances in Soft Computing and Cybernetics, Springer, Cham, 2021, 285-295. doi: 10.1007/978-3-030-61659-5_24

[3] O. Savenko, A. Nicheporuk, I. Hurman, S. Lysenko, Dynamic signature-based malware detection technique based on API call tracing, CEUR Workshop Proceedings 2393 (2019) 633–643

[4] S. N. Thanh, M. Stege, P.I. El-Habr, J. Bang, N. Dragoni, Survey on Botnets: Incentives, Evolution, Detection and Current Trends, Future Internet, 13(8) (2021) 198. doi: 10.3390/fi13080198

[5] O. Savenko, S. Lysenko, A. Nicheporuk, B. Savenko, Approach for the Unknown Metamorphic Virus Detection, Proceedings of the 2017 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS 2017), Bucharest, Romania, September 21-23 (2017) 71-76. doi: 10.1109/IDAACS.2017.8095052

[6] O. Savenko, S. Lysenko, A. Nicheporuk, B. Savenko Metamorphic Viruses' Detection Technique Based on the Equivalent Functional Block Search CEUR Workshop Proceedings 1844 (2017) 555–569.
[7] I. Krak, O. Barmak, and E. Manziuk, Using visual analytics to develop human and machine-centric models: A review of approaches and proposed information technology, Comput. Intell., vol. 1, pp. 1–26, Feb. 2020, doi:10.1111/coin.12289
[8] D. Limon-Cantu & V. Alarcon-Aquino, Multiresolution dendritic cell algorithm for network anomaly detection. PeerJ. Computer science, 7, e749, (2021). doi:10.7717/peerj-cs.749
[9] G. Suchacka, A. Cabrib, S. Rovettab, F. Masulli, Efficient on-the-fly Web bot detection, Knowledge-Based Systems, 223 (2021). doi: 10.1016/j.knosys.2021.107074
[10] S. H. Mousavi, M. Khansari, R. Rahmani, A fully scalable big data framework for Botnet detection based on network traffic analysis, Information Sciences, 512 (2020). doi: 10.1016/j.ins.2019.10.018
[11] H. Qin, J. Yang, X. Li et al, Research on DNS anomaly detection technology based on multiple features, Journal of Shenzhen University Science and Engineering, (2020) 36-43. doi:10.3724/SP.J.1249.2020.99036
[12] G. Xuanzhen, P. Zulie & C. Yuanchao, Application of Passive DNS in Cyber Security, 2020 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS), Shenyang, China, 2020, 257-259, doi: 10.1109/ICPICS50287.2020.9202344.
[13] S. Al-mashhadi, M. Anbar, I. Hasbullah, TA. Alamiedy, Hybrid rule-based botnet detection approach using machine learning for analysing DNS traffic. PeerJ Computer Science, 2021 7:e640. doi:10.7717/peerj-cs.640
[14] P. Sokol, M. Zuzčák, T. Sochor, Definition of Attack in the Context of Low-Level Interaction Server Honeypots, Computer Science and its Applications. Lecture Notes in Electrical Engineering, 2015, 330, Springer, Berlin, Heidelberg. doi:10.1007/978-3-662-45402-2_74
[15] A. Lekssays, L. Landa, B. Carminati, E. Ferrari, PAutoBotCatcher: A blockchain-based privacy-preserving botnet detector for Internet of Things, Computer Networks, 200 (2021). doi: 10.1016/j.comnet.2021.108512
[16] M. Zuzcak, T. Sochor, M. Zenka, Intrusion Detection System for Home Windows based Computers. KSII Transactions on Internet and Information Systems. 13(9) (2019) 4706-4726. doi:10.3837/tiis.2019.09.021.
[17] Snort - Network Intrusion Detection & Prevention System, URL: https://www.snort.org/
[18] O. Olowoyo, & P.A. Owolawi, Malware Classification using Deep Learning Technique, 2020 2nd International Multidisciplinary Information Technology and Engineering Conference (IMITEC), (2020) 1-6
[19] S. Pai, F. Di Troia, C. Visaggio, et. al, Clustering for malware classification. Journal of Computer Virology and Hacking Techniques, 13(2) (2017). 13. doi:10.1007/s11416-016-0265-3.
[20] F. Jammes et al.,Technologies for SOA-based distributed large scale process monitoring and control systems, In 38th Annual Conference on IEEE Industrial Electronics Society, 2012, pp. 5799-5804. doi: 10.1109/IECON.2012.6389589.
[21] A. Rotem-Gal-Oz, E. Bruno, U. Dahan, SOA Patterns, Shelter Island: Manning, 2012, 296 p.
[22] E. F. Melo and H. M. de Oliveira, An Overview of Self-Similar Traffic: Its Implications in the Network Design, arXiv:2005.02858v1, 2020.
[23] Z. Wang, J. Chang, S. Zhang, S. Luo, et al. Application of wavelet transform modulus maxima in raman distributed temperature sensors. Photonic Sensors. 4, 2014. doi:10.1007/s13320-014-0179-y.
[24] KDD Cup. URL: https://www.kdd.org/
[25] Matlab. URL: https://www.mathworks.com/products/matlab.html.
[26] B. Savenko, S. Lysenko, K. Bobrovnikova, O. Savenko, G. Markowsky. Detection DNS Tunneling Botnets. Proceedings of the 2021 IEEE 11th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), IDAACS'2021, Cracow, Poland, September 22-25, 2021.
[27] Elike Hodo, Xavier Bellekens, Andrew Hamilton, Christos Tachtatzis, Robert Atkinson. Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey. 2017. https://doi.org/10.48550/arXiv.1701.02145.