UDC 004.77.056.57 **doi: 10.32620/reks.2022.1.11**

Kira BOBROVNIKOVA[1], Sergii LYSENKO[1], Bohdan SAVENKO[1], Piotr GAJ[2], Oleg SAVENKO[1]

[1] ***Khmelnitsky National University, Khmelnitsky, Ukraine***
[2] ***Silesian University of Technology, Poland***

TECHNIQUE FOR IOT MALWARE DETECTION BASED ON CONTROL FLOW GRAPH ANALYSIS

The Internet of Things (IoT) refers to the millions of devices around the world that are connected to the Internet. Insecure IoT devices designed without proper security features are the targets of many Internet threats. The rapid integration of the Internet into the IoT infrastructure in various areas of human activity, including vulnerable critical infrastructure, makes the detection of malware in the Internet of Things increasingly important. Annual reports from IoT infrastructure cybersecurity companies and antivirus software vendors show an increase in malware attacks targeting IoT infrastructure. This demonstrates the failure of modern methods for detecting malware on the Internet of things. This is why there is an urgent need for new approaches to IoT malware detection and to protect IoT devices from IoT malware attacks. ***The subject*** *of the research is the malware detection process on the Internet of Things.* ***This study aims*** *to develop a technique for malware detection based on the control flow graph analysis.* ***Results.*** *This paper presents a new approach for IoT malware detection based on control flow graph analysis. Control flow graphs were built for suspicious IoT applications. The control flow graph is represented as a directed graph, which contains information about the components of the suspicious program and the transitions between them. Based on the control flow graph, metrics can be extracted that describe the structure of the program. Considering that IoT applications are small due to the simplicity and limitations of the IoT operating system environment, malware detection based on control flow graph analysis seems to be possible in the IoT environment. To analyze the behavior of the IoT application for each control flow graph, the action graph is to be built. It shows an abstract graph and a description of the program. Based on the action graph for each IoT application, a sequence is formed. This allows for defining the program's behavior. Thus, with the aim of IoT malware detection, two malware detection models based on control flow graph metrics and the action sequences are used. Since the approach allows you to analyze both the overall structure and behavior of each application, it allows you to achieve high malware detection accuracy. The proposed approach allows the detection of unknown IoT malware, which are the modified versions of known IoT malware. As the mean of conclusion-making concerning the malware presence, the set of machine learning classifiers was employed. The experimental results demonstrated the high accuracy of IoT malware detection.* ***Conclusions.*** *A new technique for IoT malware detection based on control flow graph analysis has been developed. It can detect IoT malware with high efficiency.*

Keywords: *malware; IoT; IoT devices; IoT application; cybersecurity; cyberattack; control flow graph; detection of cyber threats.*

Introduction

The infrastructure building of modern cities is impossible without the usage of the Internet of Things (IoT) devices. The rapid proliferation of IoT devices has led to virtual infrastructure controlling physical objects everywhere: from CCTV cameras and smart devices in private homes, on various purposes and highways to machine tools in factories, dams and power plants. It increases the probability of cyberattacks executed by the malware on those areas that previously did not pose risks to cybersecurity [1, 2]. Critical infrastructure is particularly vulnerable to IoT cyberattacks. Extensive exploitation of the Android platform in the IoT infrastructure leads to a rapid increase in cyberattacks on Android-based devices [2]. The main factors that contribute to the unprecedented increase in the number of IoT cyberattacks [3] are:

- exponential growth of IoT devices amount;
- dangerous deployment of IoT devices with the possibility of direct access to them via the Internet;
- lack of security updates for IoT devices, that makes them vulnerable to already known attacks;
- lack of transparency regarding the security status of IoT devices.

Vulnerable components in the IoT infrastructure can be [4]:

– IoT devices, that are usually the main means of initiating attacks. Malicious users can use, in particular, unsecured default settings, legacy components and unsecured update mechanisms;

– channels connecting IoT components. Protocols used in IoT systems can have security issues that can affect the entire infrastructure;

– applications and software.

IoT vulnerabilities in web applications and software are able to compromise systems, such as stealing user credentials or sending malicious updates. In this situation, any smart IoT device, such as a coffee maker, can be a source of cyberattacks and affect critical organization systems such as Intranet and database servers, through the ability to collect data and monitor IoT systems [1].

At the same time, the motivation of criminals can be different: terrorist acts for political or other purposes, theft of information that is a trade secret, blackmail and extortion for financial gain, revenge or entertainment. The Fourth Industrial Revolution provides new possibilities for cyber-criminals that employ networks, web bots, botnets etc. [9–12]. The inability to apply security or monitoring solutions on IoT devices is a major problem to prevent malicious activity in the IoT infrastructure. Thus, the growing number of cyberattacks caused by malware on IoT infrastructure [13–16] is currently a very important issue that needs to be addressed.

The aim of this paper is to develop a technique for IoT malware detection based on control flow graph analysis.

The paper structure is the following.

Section 1 presents the state-of-art works section – a brief analysis of the very modern and the latest ideas and methods addressed to solve the problem of IoT malware detection with its advantages and disadvantages.

Sections 2 discusses the stages of the proposed technique for IoT malware detection based on control flowgraph analysis.

Section 3 describes the experiments. In addition, conclusions present further work concerning the IoT malware detection.

1. Related works

There are a large number of various works devoted to the problem of preventing, detecting cyberattacks on the IoT infrastructure.

In [17] several methods to mitigate the attacks on IoT infrastructure have been developed. One of the methods is based on opcodes analysis to classify IoT malware. These opcodes are retrieved from disassembled IoT malware programs. To evaluate the effectiveness of the proposed approach supervised machine learning classifiers such as Support Vector Machine, Decision Tree, Random Forest, Naïve Bayes and K-Nearest Neighbor and data set of 512 IoT malware samples and benign application have been used. Results of experiments showed that the Random Forest classifier performed better than other classifiers.

In [18] a wrapper technique-based feature selection algorithm to filter the features accurately and to select effective features for the machine learning algorithms was developed.

In [19] the prospects of using machine learning classification algorithms for securing IoT infrastructure against attacks were investigated.

In [20] the security issues inherent to federated learning paradigm and possibilities enabled by this paradigm concerning IoT malware detection. A framework for IoT malware detection based on federated learning was presented. To evaluate the proposed framework the dataset N-BaIoT which modeling network traffic of real IoT devices infected by malware, has been used. Two federated models - supervised and unsupervised (multi-layer perceptron and autoencoder) - were evaluated. The performance of both models has been compared with using two traditional approaches. The first approach allowed each participant train a model locally with only its own data. The second approach used to share data by the participants with a central entity which is responsible of training a global model. It was concluded that the use of more diverse data has a considerable positive impact on the performance of model. However, the federated models showed similar results as the centralized, and wherein they allowed preserving the participant’s privacy. At the same time, the aggregation averaging step of baseline model which was used in most of federated learning algorithms was highly vulnerable to different attacks.

The paper [21] focuses on capturing the attacks in IoT networks by using Cowrie honeypot.

In paper [22] deep learning model DBN-IDS (Deep Belief Network intrusion detection system) was proposed. For experiments conduction the dataset was utilized which contains the examples of attacks caused by malware.

In [23] a web attack detection system which is based on URLs analysis and distributed deep learning was developed. It employs several concurrent deep models which deploys in different servers and trains separately. The overall detection results are obtained from these concurrent deep models to enhance the stability of the system and the convenience in updating.

In paper [24] deep learning models were proposed and evaluated using datasets for attack detection. Also, aspects of using deep learning algorithm for IoT cyberattack detection were reviewed.

In [25] an intelligent ADE-based (the averaged dependence estimator) framework for IoT attack detection was presented.

In [26] Security Information and Event Management-based system which detects and blocks IoT malicious traffic by monitoring of the specific packet types including TCP SYN, ICMP and DNS packets was proposed.

The paper [27] proposes a decentralized security architecture that relies on the three core technologies in order to detect the IoT attacks: Software Defined Networking (SDN) to continuously monitor and analyze the traffic data; Blockchain which supports decentralized attack detection to overcome the single point of failure problem; Fog and mobile edge computing which contributes to lesser storage constraints, cheaper computation, and shortening the time taken to detect and mitigate attack.

In [28] a security scheme named learning-driven detection mitigation (LEDEM) that detects and mitigates attacks was proposed. LEDEM scheme is based on semi-supervised machine-learning and leverages the cloud and software-defined network (SDN) paradigm to mitigate the attacks.

Majority of machine learning approaches for malware detection require a significant prior knowledge about malware features. To solve the problem of limited availability of malware samples in [29] a framework for generating and detecting new IoT malware samples was proposed. For this purpose, the raw byte code at the edge layer of the IoT networks was utilized. Convolutional Neural Network was used for high-level features extraction. To generate new IoT malware samples technique of boundary-seeking Generative Adversarial Network was used. To capture the long-term and short-term features dependency an attention-based model, a combination of CNN and LSTM (Long Short-Term Memory) were applied. The attention mechanism allows improving the performance of model by decreasing or increasing attention to certain parts of the malware features.

In [30] two defense approaches against attack to a IoT malware detection system for mobile multimedia applications were proposed. They are CNN and 1- nearest neighbors (C4N) combination and Robust-NN. These approaches allow to modify training data set that has been poisoned by an attack. Thus, the trained machine learning model accuracy is improved, and if the malicious IoT program is executing on any IoT device, the model generates the alerts.

In [31] a dynamic analysis-based approach for new and well-known IoT malware detection was proposed. The proposed approach used the convolution neural network model for dynamic analysis IoT malware in cloud environment. It extracted behaviors related to process, system call, memory, virtual system and network. Extracted and analyzed behavior data are converted into the IoT malware behavior images. These images are classified with using the Convolution Neural Network.

In the paper [32] thorough survey of approaches based on static IoT malware detection was conducted. The definition, security threats and evolution of IoT malware were introduced. The latest IoT malware detection approaches were analyzed and compared. IoT malware detection approaches were divided into two groups: graph-based and non-graph-based methods. The non-graph based approaches can be useful when detecting known malware without obfuscation or customization. But these methods lose accuracy for unknown malware detection. The graph-based approaches are based on analysis of the control flow and therefore characterized by certain complexity. Wherein these approaches show advantages to accurately detect complicated or unknown IoT malware. Thereafter these approaches were tested and evaluated with using the same experimental configuration and IoT malware dataset, consisting of 11200 samples. The advantages and limitations of these approaches summarized that they can be used to improve the efficiency of IoT malware detection.

In [33] MQTTset – the legitimate dataset related to the MQTT protocol, which can be used to train machine learning models to implement the detection systems ability to protect IoT networks was described.

Despite the large number of different methods for detecting and mitigating cyberattacks caused by malware on the IoT infrastructure, the steady increase in their number confirms that this problem is not solved today.

2. Technique for IoT malware detection based on control flowgraph analysis

The proposed method is based on the analysis of control flow graphs built for IoT applications. It allows detecting unknown IoT threats caused by malicious software, which are the modified versions of known IoT threats. The control flow graph is represented as a directed graph, which contains information about the components of the program and the transitions between them, taking into account unconditional transitions, branches, cycles, function calls and exclusions. The control flow graph describes the control flows that may occur during the program execution.

Typically, methods based on CFG analysis deal s with the NP-complete problem when are used for large codes analysis. However, IoT applications are small in size due to the simplicity and limitations of the IoT operating systems environment. This makes it possible to effectively detect IoT threats based on CFG analysis.

The proposed method consists of the following stages:

– creation of training/testing data;

– system training;

– system testing and evaluation of detection accuracy;

– IoT malware detection.

Let’s consider the steps of the stages of the proposed method.

2.1. Stage of Training / Testing Data Creation

The steps of the stage of training/testing data creation are shown in Fig. 1.

In order to create training /test data, a set of malware and benign IoT applications are disassembled.

For each IoT application the control flow graph from source codes is constructed.

Let’s denote the control flow graph for IoT application υ as:

$$G_{CFG}(F,R) = \langle F,R \rangle , \qquad (1)$$

where $F=\{f_i\}_{i=1}^{O}$ – the nodes (vertices) of the CFG, which present functions of the IoT application υ ;

$R=\{r_i\}_{i=1}^{S}$ – the edges of the CFG, which present the call relationship between f_i in υ ;

O – is the order of the CFG;

S – is the size of the CFG.

At the next step the obtained control flows graphs are to be analyzed in order to evaluate the set of metrics (Table 1), that characterize the structure of the CFG and thus describe the structure of the IoT application code.

In order to process the control flow graph $G_{CFG}(F,R)$ it can be presented as the adjacency matrix $A_{CFG} = \left(a_{i,j}\right)_{i=1,j=1}^{OO}$. Then let us denote the CFG metric extraction and CFG metrics feature vectors $\bar{M}$ formation function φ_1 as: $\varphi_1\left(A_{CFG}\right) \to \bar{M}$, where $\bar{M}$ – is feature vector formed from the metrics obtained for each IoT application υ, $\bar{M}=\left(O,S,m_1..m_8\right)$.

The set of feature vectors $\bar{M}$ are further marked as one of the IoT malware families class or a benign IoT application and are added to the database of training / testing data.

Also, in order to analyze the behavior of the IoT application for each of the control flow graph $G_{CFG}(F,R)$ the actions graph $G_B(B,E)$ is to be build. It is an abstract graph description of the program.

Such description consists of IoT application’s actions (for example, API calls). Let denote the actions graph $G_B(B,E)=\langle B,E \rangle$, where $B=\{b_i\}_{i=1}^{L}$ – the nodes of the $G_B(B,E)$, each of which presented the action made by IoT application υ ; L – the number of nodes; $E=\{e_i\}_{i=1}^{P}$ – the edges of the $G_B(B,E)$, which presented transitions between b_i ; P – the number of edges.

On the basis of the obtained actions graph $G_B(B,E)$ for each IoT application υ the set of sequences of actions B_{Act} is formed.

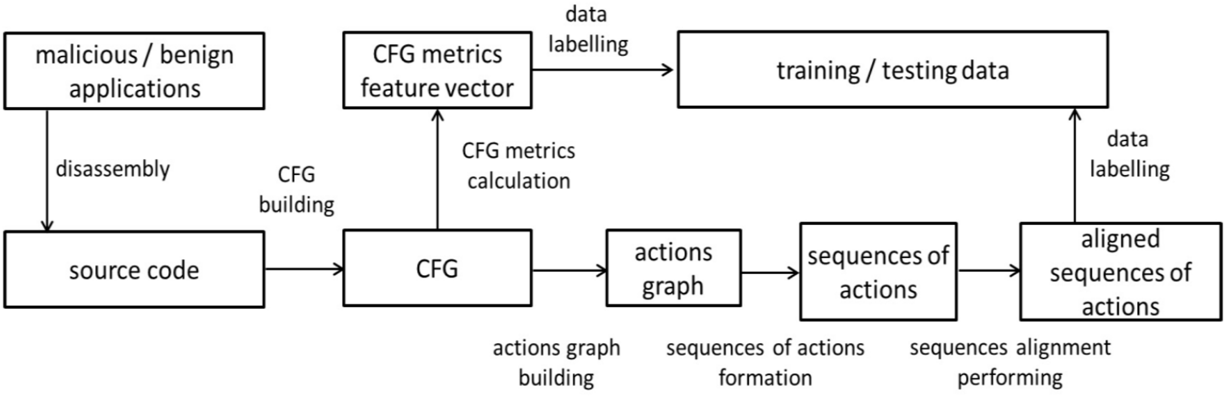

Figure 1. Stage of training / testing data creation

Table 1

The set of CFG metrics that characterize the structure of the IoT application

Metric	The metrics value
O	the order of graph (the number of nodes)
S	the size of graph (the number of edges)
m_1	the radius – the minimum distances among all the maximum distances (the edges number in a shortest path) between a node to all other nodes
m_2	the diameter – the maximal distance (the edges number in a shortest path) between the pair of nodes
m_3	The graph density – the ratio of the number of graph edges with respect to the maximum possible number of edges
m_4	the shortest path in the graph
m_5	the centrality degree is defined as the number of links incident upon a node
m_6	the normalized closeness centrality – the reciprocal of the sum of the shortest path‘s length between the graph node and all other nodes
m_7	the normalized betweenness centrality – the number of shortest paths that pass through the node
m_8	the number of components – number of subgraphs in which any two nodes are connected to each other by paths and is not connected to other nodes in the rest of the graph

Let us denote the formation function φ_2 for actions sequences B_{Act} as: $\varphi_2=(G_B(B,E))\to B_{Act}$, where $B_{Act}=\{B_i\}_{i=1}^{3}$.

Let us denote the actions sequences as following:

$B_1=\{b_i\}_{i=1}^{N}$ – a sequences of unique actions, that determines what unique actions are made by the IoT application, where N is the number of unique actions;

$B_2=\{b_i\}_{i=1}^{K}$ – a sequences of actions, which determines the order of actions in time, where K is the total number of all actions;

$B_3=\{\langle b,n\rangle_i\}_{i=1}^{N}$ – a sequences of actions, which determines how many times each action is uses, where n is the number of times the action was executed.

At the next step, the Multiple Sequence Alignment algorithm (MSA) is applied to the resulting actions sequences, then they are also marked and added to the database of training / testing data.

Further, the resulting database D is divided into two parts: training data T and test data E samples: $D=T\cup E$.

Thus, each of training and test data samples consists of two subsets: $T=T_M\cup T_B$ and $E=E_M\cup E_B$, where T_M and E_M are CFG metric-based data, which are built from the feature vectors $\bar{M}$; T_B and E_B are data, which are built from actions sequences B_{Act}.

2.2. Stages of System Testing and Evaluation of Detection Accuracy

The steps of the stage of system testing and evaluation of detection accuracy are shown in Fig. 2.

At this stage the system is trained. For this purpose, we use the training data T.

After that we use the test data E for two malware detection models based on CFG metric and on sequences of actions.

Mentioned models make it possible to obtain its accuracy score that indicates that the IoT application belongs to a particular malware class or is a benign application.

These accuracy score obtained by each of models is further applied as weight and will be used to evaluate the overall result via the soft voting method.

2.3. IoT Malware Detection Stage

The steps of the IoT malware detection stage are shown in Fig. 3. In order to detect IoT malware for each suspicious IoT application the control flow graph is built. On the basis of each constructed graph the feature vectors $\bar{M}$ of CFG metrics that characterize the structure of the IoT application is formed. Also, on the basis of each constructed graph the sequences of actions set B_{Act} are formed which describe behavior of IoT application. After that, a decision on the presence or absence of IoT malware using both attack detection models is made.

The classifier in each of two malware detection models provides the prediction that a suspicious application belongs to a malicious or benign class. The prediction of each malware detection models is weighted according to the previously obtained detection accuracy and are summed up.

Let’s define an IoT malware detection function φ_3 as:

$$\varphi_3:\{\upsilon|(\gamma_1(\bar{M}_\nu)\times\Phi_{\gamma 1})+(\gamma_2(B_\upsilon)\times\Phi_{\gamma 2})\}\to D, \quad (2)$$

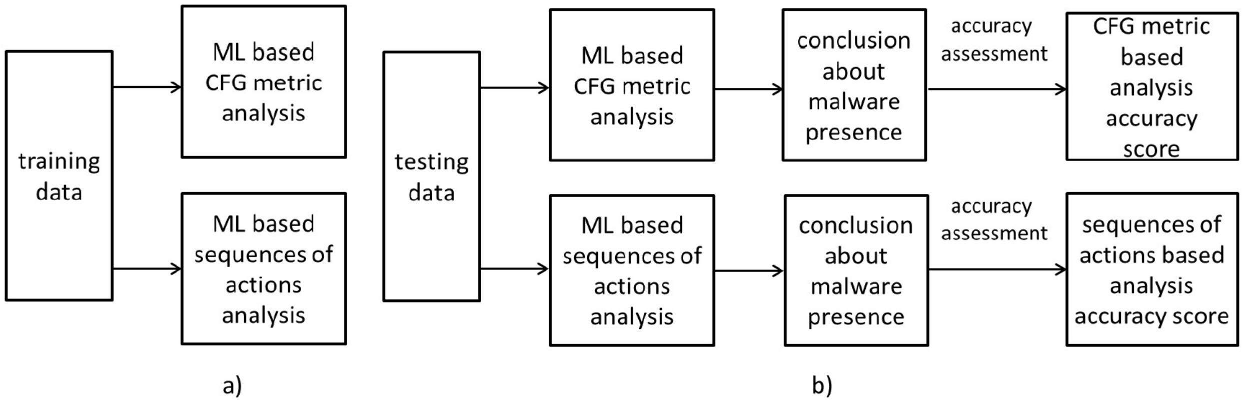
training data
ML based CFG metric analysis
ML based sequences of actions analysis
a)
testing data
ML based CFG metric analysis
conclusion about malware presence
accuracy assessment
CFG metric based analysis accuracy score
ML based sequences of actions analysis
conclusion about malware presence
accuracy assessment
sequences of actions based analysis accuracy score
b)

Figure 2. The training and testing of the detection system:
a) system training stage;
b) system testing and evaluation of detection accuracy

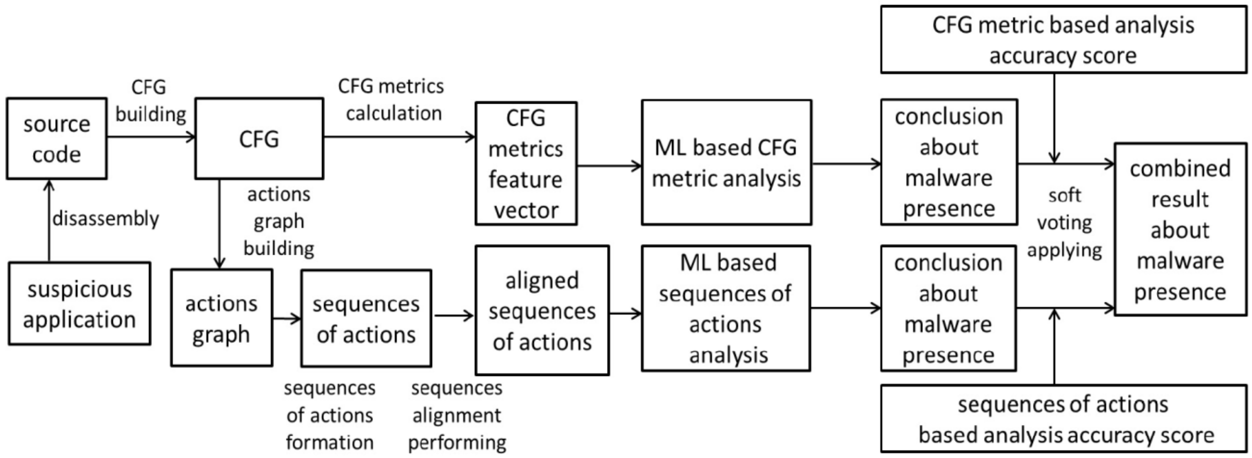
source code
CFG building
CFG
CFG metrics calculation
CFG metrics feature vector
ML based CFG metric analysis
conclusion about malware presence
CFG metric based analysis accuracy score
disassembly
suspicious application
actions graph building
actions graph
sequences of actions
sequences of actions formation
aligned sequences of actions
sequences alignment performing
ML based sequences of actions analysis
conclusion about malware presence
soft voting applying
combined result about malware presence
sequences of actions based analysis accuracy score

Figure 3. The IoT malware detection stage

where γ_1 – the Matching Learning function based on CFG metric analysis;

γ_2 – the Matching Learning function based on sequences of actions analysis;

$\bar{M}_v$ – CFG metrics feature vector formed for υ;

B_υ – set of sequences of actions formed for υ;

$\Phi_{\gamma 1}$ – CFG metric-based analysis accuracy score;

$\Phi_{\gamma 2}$ – sequences of actions-based analysis accuracy score;

D – combined result about an IoT malware presence.

Thus, soft voting is applied and the target label with the highest weighted probability is used as a resulted vote.

3. Experiments

In order to evaluate the efficiency of the proposed approach the experiments were held. They were based on the usage of the BotGRABBER – cyberattacks' detection tool – was used [34–38].

In order to conduct the experiments, 5762 samples of IoT malicious applications such as Mirai, Ares, IPStorm, ADB.miner, Fbot, Matryosh, Trinity, Guerrilla, Hiddad, Hajime, Gafgyt and other from [39–41] and 1824 samples of IoT benign applications collected from Google Play Store [42] and [43] were used.

With purpose of the IoT applications disassembling as well as the CFG's and the actions graphs construction, a framework for reverse-engineering and binaries analyzing Radare2 [44] was used.

For CFG metrics to obtained a library for studying graphs and networks NetworkX [45] was employed. For each sample of the IoT application from the obtained CFG metrics the feature vectors $\bar{M}$ were formed. In addition, the actions series B_{Act} from the action graphs were extracted.

The obtained feature vectors and actions series were labelled as malicious or benign IoT class, respectively. Further they were split into three parts in the ratio of 2:3:5 to create:

1. The training data.
2. The test data E that is used to obtain the accuracy score of each of two malware detection models based on CFG metric and on sequences of actions are used.
3. Data for reliability assessment of proposed method, respectively.

For the experiments conducting the following machine learning algorithms for each detection model were employed: C-means, Support Vector Machine, Naïve Bayes, K-Nearest Neighbors, Random Forest and Rotation Forest [46-49].

The prediction of each detection models is weighted according to its detection accuracy and are summed up.

In order to evaluate the efficiency of the proposed approach the following quantity measures were used:

$$SN=TP/(TP+FN)\,, \tag{3}$$

$$SP=TN/(TN+FP)\,, \tag{4}$$

$$Q=(TP+TN)/(TP+TN+FP+FN)\,, \tag{5}$$

where TP – True Positives;
TN – True Negatives;
FP – False Positives;
FN – False Negatives;
SN – Sensitivity;
SP – Specificity;
Q – Overall Accuracy.

The obtained results are presented in the Table 2, and they showed that detection efficiency is up to 98 %, while the false positives is about 3-5 %.

Conclusion

The research presents the new technique for IoT malware detection based on control flow graph analysis.

It is based on the analysis of control flow graphs built for IoT applications.

Proposed approach allows detecting unknown IoT threats caused by malicious software, which are the modified versions of known IoT threats.

The main core of the approach is the usage of the control flow graph that is represented as a directed graph and contains the information about the components of the program. Developed approach is based on CFG analysis.

As the mean of conclusion making concerning the malware presence a set of machine learning classifiers were employed.

Experimental results demonstrated that the proposed technique's detection efficiency is up to 98 %, while the false positives is about 3-5 %.

Further work will be devoted to techniques for IoT malware detection that may be further improved by choosing a more refined set of malicious samples, as well as the involving different machine learning algorithms.

Contribution of authors: developed the conceptual provisions of the technique for IoT malware detection based on control flowgraph analysis – **K. Bobrovnikova**; performed the review and analysis of references devoted to the problem of preventing, detecting cyberattacks on the IoT infrastructure – **S. Lysenko**; held the experimental research and provided main results – **B. Savenko** and **P. Gaj**; made the introduction section of the paper – **O. Savenko**. All authors have read and agreed to the published version of the manuscript.

Table 2

The detection results

Classifier	Training set (malicious and benign samples)	Number of evaluation samples				Results		
		Malicious		Benign		Sensitivity, %	Specificity, %	Overall accuracy, %
		TP	FN	TN	FP			
c-means	7586	44	2	7207	333	95,65	95,58	95,58
SVM	7586	98	8	7357	123	92,45	98,36	98,27
Naïve Bayes	7586	19	2	7388	177	90,48	97,66	97,64
K-Nearest Neighbors	7586	78	2	7333	173	97,50	97,70	97,69
Random Forest	7586	91	5	7236	254	94,79	96,61	96,59
Rotation Forest	7586	85	7	7316	178	92,39	97,62	97,56

References (GOST 7.1:2006)

1. Trend Micro. The IoT Attack Surface: Threats and Security Solutions [Electronic resource]. – Access mode: https://www.trendmicro.com. – 11.01.2022.

2. Check point software cyber security report 2022 [Electronic resource]. – Access mode: https://www.ntsc.org. – 11.01.2022.

3. Nozomi Networks Labs. What IT Needs to Know about OT/IoT Security Threats in 2022 [Electronic resource]. – Access mode: https://www.nozominetworks.com. – 11.01.2022.

4. OWASP Internet of Things [Electronic resource]. – Access mode: https://owasp.org. – 11.01.2022.

5. Computer systems resilience in the presence of cyber threats: taxonomy and ontology [Text] / S. Lysenko, V. Kharchenko, K. Bobrovnikova, R. Shchuka // Radioelectronic and computer systems. – 2020. – Vol. 1. – P. 17-28. DOI: 10.32620/reks.2020.1.02.

6. A hierarchical fuzzy quality assessment of complex security information systems [Text] / V. Shelekhov, N. Barchenko, V. Kalchenko, V. Obodniak // Radioelectronic and computer systems. – 2020. – Vol .4. – P. 106-115. DOI: 10.32620/reks.2020.4.10.

7. Kolisnyk, M. Vulnerability analysis and method of selection of communication protocols for information transfer in Internet of Things systems [Text] / M. Kolisnyk // Radioelectronic and computer systems. – 2021. – Vol. 1. – P. 133-149. DOI: 10.32620/reks.2021.1.12.

8. Methods and technologies for ensuring cybersecurity of industrial and web-oriented systems and networks [Text] / O. Morozova, A. Nicheporuk, A. Tetskyi, V. Tkachov // Radioelectronic and computer systems. – 2021. – Vol. 4. – P. 145-156. DOI: 10.32620/reks.2021.4.12.

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

10. Detection DNS Tunneling Botnets [Text] / B. Savenko, S. Lysenko, K. Bobrovnikova, O. Savenko, G. Markowsky // 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). – 2021. – P. 64-69. DOI: 10.1109/IDAACS53288.2021.9661022.

11. Online Web Bot Detection Using a Sequential Classification Approach [Text] / A. Cabri, G. Suchacka, S. Rovetta, F. Masulli // 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS). – Exeter. – United Kingdom. – 2018. – P. 1536-1540. DOI: 10.1109/HPCC/SmartCity/DSS.2018.00252.

12. Scanzio, S. Heterogeneous and dependable networks in industry – A survey [Text] / S. Scanzio, L. Wisniewski, P. Gaj // Computers in Industry. – 2021. – Vol. 125. – Article No. 103388. DOI: 10.1016/j.compind.2020.103388.

13. Vijayakumaran, C. A reliable next generation cyber security architecture for industrial internet of things environment [Text] / C. Vijayakumaran, B. Muthusenthil, B. Manickavasagam, // International Journal of Electrical and Computer Engineering. – 2020. – vol. 10, no. 1. – P. 387-395. DOI: 10.11591/ijece.v10i1.pp387-395.

14. Wani, A. Ransomware protection in IoT using software defined networking [Text] / A. Wani, S. Revathi // International Journal of Electrical and Computer Engineering. – 2020. – Vol. 10, no. 3. – P. 3166-3175. DOI: 10.11591/ijece.v10i3.pp3166-3175.

15. Karande, J. DEDA: An algorithm for early detection of topology attacks in the internet of things [Text] / J. Karande, S. Joshi // International Journal of Electrical & Computer Engineering. – 2021. – Vol. 11, no. 2. – P. 1761-1770. DOI: 10.11591/ijece.v11i2.pp1761-1770.

16. Chetan, R. A comprehensive survey on exiting solution approaches towards security and privacy requirements of IoT [Text] / R. Chetan, R. Shahabadkar // International Journal of Electrical & Computer Engineering. – 2018. – vol. 8, no. 4. – P. 2319-2326. DOI: 10.11591/ijece.v8i4.pp2319-2326.

17. Anidu, A. Evaluation of machine learning algorithms on Internet of Things (IoT) malware opcodes [Text] / A. Anidu, Z. Obuzor // Handbook of Big Data Analytics and Forensics. – Springer, Cham. – 2022. – P. 177-191. DOI: 10.1007/978-3-030-74753-4_12.

18. CorrAUC: a malicious bot-IoT traffic detection method in IoT network using machine learning techniques [Text] / M. Shafiq, Z. Tian, A. Bashir, K. Du, X., M. Guizani // IEEE Internet of Things Journal. – 2021. – Vol. 8, no. 5. – P. 3242-3254. DOI: 10.1109/JIOT.2020.3002255.

19. Verma, A. Machine learning based intrusion detection systems for IoT applications [Text] / A. Verma, V. Ranga // Wireless Personal Communications. – 2020. – Vol. 111, no. 4. – P. 2287-2310. DOI: 10.1007/s11277-019-06986-8.

20. Federated learning for malware detection in IoT devices [Text] / V. Rey, P. M. S. Sánchez, A. H. Celdrán, G. Bovet // Computer Networks. – 2022. – Vol. 204. – Article No. 108693. DOI: 10.1016/j.comnet.2021.108693.

21. Shrivastava, R. K. Attack detection and forensics using honeypot in IoT environment [Text] / R. K. Shrivastava, B. Bashir, C. Hota // International Conference on Distributed Computing and Internet Technology. – Springer, Cham, 2019. – P. 402-409. DOI: 10.1007/978-3-030-05366-6_33.

22. Effective attack detection in internet of medical things smart environment using a deep belief neural network [Text] / S. Manimurugan, S. Al-Mutairi, M. M. Aborokbah, N. Chilamkurti, S. Ganesan, R. Patan // IEEE Access. – 2020. – Vol. 8. – P. 77396-77404. DOI: 10.1109/ACCESS.2020.2986013.

23. A distributed deep learning system for web attack detection on edge devices [Text] / Z. Tian, C. Luo, J. Qiu, X. Du, M. Guizani // IEEE Transactions on Industrial Informatics. – 2020. – Vol. 16, no. 3. – P. 1963-1971. DOI: 10.1109/TII.2019.2938778.

24. Roopak, M. Deep learning models for cyber security in IoT networks [Text] / M. Roopak, G. Y. Tian, J. Chambers // IEEE 9th annual computing and communication workshop and conference (CCWC). – 2019. – P. 0452-0457. DOI: 10.1109/CCWC.2019.8666588.

25. Averaged dependence estimators for DoS attack detection in IoT networks [Text] / Z. A. Baig, S. Sanguanpong, S. N. Firdous, T. G. Nguyen, C. So-In // Future Generation Computer Systems. – 2020. – Vol. 102. – P. 198-209. DOI: 10.1016/j.future.2019.08.007.

26. SIEM-based detection and mitigation of IoT-botnet DDoS attacks [Text] / B. Al-Duwairi et al. // International Journal of Electrical & Computer Engineering. – 2020. – Vol. 10, no. 2. – P. 2182-2191. DOI: 10.11591/ijece.v10i2.pp2182-2191.

27. Rathore, S. BlockSecIoTNet: Blockchain-based decentralized security architecture for IoT network [Text] / S. Rathore, B. W. Kwon, J. H. Park // Journal of Network and Computer Applications. – 2019. – Vol. 143. – P. 167-177. DOI: 10.1016/j.jnca.2019.06.019.

28. Ravi, N. Learning-driven detection and mitigation of DDoS attack in IoT via SDN-cloud architecture [Text] / N. Ravi, S. M. Shalinie // IEEE Internet of Things Journal. – 2020. – Vol. 7, no. 4. – P. 3559-3570. DOI: 10.1109/JIOT.2020.2973176.

29. Generative adversarial network to detect unseen internet of things malware [Text] / Z. Moti, S. Hashemi, H. Karimipour, A. Dehghantanha, A. N. Jahromi, L. Abdi, F.Alavi // Ad Hoc Networks. – 2021. – vol. 122, no. 2. – Article No. 102591. DOI: 10.1016/j.adhoc.2021.102591.

30. Taheri, R. Adversarial android malware detection for mobile multimedia applications in IoT environments [Text] / R. Taheri, R. Javidan, Z. Pooranian // Multimedia Tools and Applications. – 2021. – Vol. 80, no. 3. – P. 16713-16729. DOI: 10.1007/s11042-020-08804-x.

31. Jeon, J. Dynamic analysis for IoT malware detection with convolution neural network model [Text] / J. Jeon, J. H. Park, Y. S. Jeong // IEEE Access. – 2020. – Vol. 8. – P. 96899-96911. DOI: 10.1109/ACCESS. 2020.2995887.

32. A survey of IoT malware and detection methods based on static features [Text] / Q. D. Ngo, H. T. Nguyen, V. H. Le, D. H. Nguyen // ICT Express. – 2020. – Vol. 6, no. 4. – P. 280-286. DOI: 10.1016/j.icte. 2020.04.005.

33. MQTTset, a New Dataset for Machine Learning Techniques on MQTT / I. Vaccari, G. Chiola, M. Aiello, M. Mongelli, E. Cambiaso // Sensors. – 2020. – Vol. 20, no. 22. – Article No. 6578. DOI: 10.3390/s20226578.

34. A Cyberattacks Detection Technique Based on Evolutionary Algorithms [Text] / S. Lysenko, K. Bobrovnikova, R. Shchuka, O. Savenko // In 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT). – 2020. – P. 127-132. DOI: 10.1109/DESSERT50317. 2020.9125016.

35. Lysenko, S. DDoS Botnet Detection Technique Based on the Use of the Semi-Supervised Fuzzy c-Means Clustering [Text] / S. Lysenko, O. Savenko, K. Bobrovnikova // CEUR-WS. – 2018. – Vol. 2104. – Paper No. 251. – P. 688-695.

36. Detection of the botnets' low-rate DDoS attacks based on self-similarity [Text] / S. Lysenko, K. Bobrovnikova, S. Matiukh, I. Hurman, O. Savenko // International Journal of Electrical and Computer Engineering, ISSN 2088-8708. – 2020. – Vol. 10, no. 4. – P. 3651-3659. DOI: 10.11591/ijece.v10i4.pp3651-3659.

37. Technique for IoT Cyberattacks Detection Based on DNS Traffic Analysis [Text] / S. Lysenko, K. Bobrovnikova, O. Savenko, R. Shchuka // CEUR-WS. – 2020. – Vol. 2623. – Paper No. 19. – P. 208-218.

38. BotGRABBER: SVM-Based Self-Adaptive System for the Network Resilience Against the Botnets' Cyberattacks [Text] / S. Lysenko, K. Bobrovnikova, O. Savenko, A. Kryshchuk // Communications in Computer and Information Science. – 2019. – P. 127-143.

39. IoT dataset [Electronic resource]. – Access mode: https://github.com/thieu1995/iot_dataset. – 11.01.2022.

40. IoTPOT [Electronic resource]. – Access mode: https://sec.ynu.codes/iot/ – 11.01.2022.

41. IoTPOT: A novel honeypot for revealing current IoT threats [Text] / Y. M. P. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, C. Rossow // Journal of Information Processing. – 2016. – Vol. 24, no. 3. – P. 522-533. DOI: 10.2197/ipsjjip.24.522.

42. Google Play [Electronic resource]. – Access mode: https://play.google.com/store/apps?hl=ru&gl=US – 11.01.2022.

43. OpenWrt [Electronic resource]. – Access mode: https://openwrt.org/ – 11.01.2022.

44. Radare2 [Electronic resource]. – Access mode: https://www.radare.org/n/ – 11.01.2022

45. NetworkX [Electronic resource]. – Access mode: https://networkx.org/ – 11.01.2022.

46. Choudhary, S. Malware Detection & Classification using Machine Learning [Text] / S. Choudhary, A. Sharma // International Conference on Emerging Trends in Communication, Control and Computing (ICONC3). – 2020. – P. 1-4. DOI: 10.1109/ICONC345789.2020.9117547.

47. Tirandasu, R. K. A Review on Malicious Software Detection using Machine Learning Algorithms [Text] / R. K. Tirandasu, Y. Prasanth // Second International Conference on Electronics and Sustainable Communication Systems (ICESC). – 2021. – P. 1945-1948. DOI: 10.1109/ICESC51422.2021.9532700.

48. Malware Detection Using Machine Learning [Text] / P. Singh, S. Kaur, S. Sharma, G. Sharma, S. Vashisht, V. Kumar // 2021 International Conference on Technological Advancements and Innovations

(ICTAI). – 2021. – P. 11-14. DOI: 10.1109/ ICTAI53825.2021.9673465.

49. Köse, Ü. Detection of Malware with Deep Learning Method [Text] / Ü. Köse, R. Samet // 6th International Conference on Computer Science and Engineering (UBMK). – 2021. – P. 665-669. DOI: 10.1109/UBMK52708.2021.9559020.

References (BSI)

1. *Trend Micro. The IoT Attack Surface: Threats and Security Solutions*. Available at: https://www.trendmicro.com (accessed 11.01.2022).

2. *Check point software cyber security report 2022*. Available at: https://www.ntsc.org (accessed 11.01.2022).

3. *Nozomi Networks Labs. What IT Needs to Know about OT/IoT Security Threats in 2022*. Available at: https://www.nozominetworks.com (accessed 11.01.2022).

4. *OWASP Internet of Things*. Available at: https://owasp.org (accessed 11.01.2022).

5. Lysenko, S., Kharchenko, V., Bobrovnikova, K., Shchuka, R. Computer systems resilience in the presence of cyber threats: taxonomy and ontology. *Radioelectronic and computer systems*, 2020, vol. 1, pp. 17-28, 10.32620/reks.2020.1.02.

6. Shelekhov, V., Barchenko, N., Kalchenko, V., Obodniak, V. A hierarchical fuzzy quality assessment of complex security information systems. *Radioelectronic and computer systems*, 2020, vol. 4, pp. 106-115. DOI: 10.32620/reks.2020.4.10.

7. Kolisnyk, M. Vulnerability analysis and method of selection of communication protocols for information transfer in Internet of Things systems. *Radioelectronic and computer systems*, 2021, vol. 1, pp. 133-149. DOI: 10.32620/reks.2021.1.12.

8. Morozova, O., Nicheporuk, A., Tetskyi, A., Tkachov, V. Methods and technologies for ensuring cybersecurity of industrial and web-oriented systems and networks. *Radioelectronic and computer systems*, 2021, vol. 4, pp. 145-156. DOI: 10.32620/reks.2021.4.12.

9. Sochor, Tomas., Chalupova, Nadezda. Interpersonal Internet Messaging Prospects in Industry 4.0 Era. *Recent Advances in Soft Computing and Cybernetics*. Springer, Cham, 2021, pp. 285-295. DOI: 10.1007/978-3-030-61659-5_24.

10. Savenko, B., Lysenko, S., Bobrovnikova, K., Savenko, O. and Markowsky, G. Detection DNS Tunneling Botnets, *11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications* (IDAACS), 2021, pp. 64-69. DOI: 10.1109/IDAACS53288.2021.9661022.

11. Cabri, A., Suchacka, G., Rovetta, S., Masulli, F. Online Web Bot Detection Using a Sequential Classification Approach. *IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, Exeter, United Kingdom, 2018, pp. 1536-1540. DOI: 10.1109/HPCC/ SmartCity/DSS.2018.00252.

12. Scanzio, S., Wisniewski, L., Gaj, P. Heterogeneous and dependable networks in industry – A survey. *Computers in Industry*, 2021, vol 125, article no. 103388. DOI: 10.1016/j.com-pind.2020.103388.

13. Vijayakumaran, C., Muthusenthil, B., & Manickavasagam, B. A reliable next generation cyber security architecture for industrial internet of things environment. *International Journal of Electrical and Computer Engineering*, 2020, vol. 10, no. 1, pp. 387-395. DOI: 10.11591/ijece.v10i1.pp387-395.

14. Wani, A., & Revathi, S. Ransomware protection in loT using software defined networking. International *Journal of Electrical and Computer Engineering*, 2020, vol. 10, no. 3, pp. 3166-3175. DOI: 10.11591/ijece.v10i3.pp3166-3175.

15. Karande, J., Joshi, S. DEDA: An algorithm for early detection of topology attacks in the internet of things. *International Journal of Electrical & Computer Engineering*, 2021, vol. 11, no. 2, pp. 1761-1770. DOI: 10.11591/ijece.v11i2.pp1761-1770.

16. Chetan, R., Shahabadkar, R. A comprehensive survey on exiting solution approaches towards security and privacy requirements of IoT. *International Journal of Electrical & Computer Engineering*, 2018, vol. 8, no. 4, pp. 2319-2326. DOI: 10.11591/ijece.v8i4.pp2319-2326.

17. Anidu, A., Obuzor, Z. Evaluation of machine learning algorithms on Internet of Things (IoT) malware opcodes. *Handbook of Big Data Analytics and Forensics* Springer, Cham, 2022, pp. 177-191. DOI: 10.1007/978-3-030-74753-4_12.

18. Shafiq, M., Tian, Z., Bashir, A. K., Du, X., Guizani, M. CorrAUC: a malicious bot-IoT traffic detection method in IoT network using machine learning techniques. *IEEE Internet of Things Journal*, 2021, vol. 8, no. 5, pp. 3242-3254. DOI: 10.1109/JIOT.2020. 3002255.

19. Verma, A., & Ranga, V. Machine learning based intrusion detection systems for IoT applications. *Wireless Personal Communications*, 2020, vol. 111, no. 4, pp. 2287-2310. DOI: 10.1007/s11277-019-06986-8.

20. Rey, V., Sánchez, P. M. S., Celdrán, A. H., & Bovet, G. Federated learning for malware detection in iot devices. *Computer Networks*, 2022, vol. 204, article no. 108693. DOI: 10.1016/j.comnet.2021.108693.

21. Shrivastava, R. K., Bashir, B., & Hota, C. Attack detection and forensics using honeypot in IoT environment. *International Conference on Distributed Computing and Internet Technology*, Springer, Cham, 2019, pp. 402-409. DOI: 10.1007/978-3-030-05366-6_33.

22. Manimurugan, S., Al-Mutairi, S., Aborokbah, M. M., Chilamkurti, N., Ganesan, S., & Patan, R. Effective attack detection in internet of medical things smart environment using a deep belief neural network. *IEEE Access*, 2020, vol. 8, pp.77396-77404. DOI: 10.1109/ACCESS.2020.2986013.

23. Tian, Z., Luo, C., Qiu, J., Du, X., & Guizani, M. A distributed deep learning system for web attack detection on edge devices. *IEEE Transactions on Industrial Informatics*, 2019, vol. 16, no. 3, pp. 1963-1971. DOI: 10.1109/TII.2019.2938778.

24. Roopak, M., Tian, G. Y., & Chambers, J. Deep learning models for cyber security in IoT networks. *IEEE 9th annual computing and communication workshop and conference* (CCWC), 2019, pp. 0452-0457. DOI: 10.1109/CCWC.2019.8666588.

25. Baig, Z. A., Sanguanpong, S., Firdous, S. N., Nguyen, T. G., & So-In, C. Averaged dependence estimators for DoS attack detection in IoT networks. *Future Generation Computer Systems*, 2020, vol. 102, pp. 198-209. DOI: 10.1016/j.future.2019.08.007.

26. Al-Duwairi, B. et al. SIEM-based detection and mitigation of IoT-botnet DDoS attacks. *International Journal of Electrical & Computer Engineering*, 2020, vol. 10, no. 2, pp. 2182-2191. DOI: 10.11591/ijece.v10i2.pp2182-2191.

27. Rathore, S., Kwon, B. W., & Park, J. H. BlockSecIoTNet: Blockchain-based decentralized security architecture for IoT network. *Journal of Network and Computer Applications*, 2019, vol. 143, pp. 167-177. DOI: 10.1016/j.jnca.2019.06.019.

28. Ravi, N., & Shalinie, S. M. Learning-driven detection and mitigation of DDoS attack in IoT via SDNcloud architecture. *IEEE Internet of Things Journal*, 2020, vol. 7, no. 4, pp. 3559-3570. DOI: 10.1109/JIOT.2020.2973176.

29. Moti, Z., Hashemi, S., Karimipour, H., Dehghantanha, A., Jahromi, A. N., Abdi, L., & Alavi, F. Generative adversarial network to detect unseen internet of things malware. *Ad Hoc Networks*, 2021, vol. 122, no. 2, article no. 102591. DOI: 10.1016/j.adhoc.2021.102591.

30. Taheri, R., Javidan, R., Pooranian, Z. Adversarial android malware detection for mobile multimedia applications in IoT environments. *Multimedia Tools and Applications*, 2021, vol. 80, no. 3, pp. 16713-16729. DOI: 10.1007/s11042-020-08804-x.

31. Jeon, J., Park, J. H., & Jeong, Y. S. Dynamic analysis for IoT malware detection with convolution neural network model. *IEEE Access*, 2020, vol. 8, pp. 96899-96911. DOI: 10.1109/ACCESS.2020.2995887.

32. Ngo, Q. D., Nguyen, H. T., Le, V. H., & Nguyen, D. H. A survey of IoT malware and detection methods based on static features. *ICT Express*, 2020, vol. 6, no. 4, pp. 280-286. DOI: 10.1016/j.icte.2020.04.005.

33. Vaccari, I., Chiola, G., Aiello, M., Mongelli, M., Cambiaso, E. MQTTset, a New Dataset for Machine Learning Techniques on MQTT. *Sensors*, 2020, vol. 20, no. 22, article no. 6578. DOI: 10.3390/s20226578.

34. Lysenko, S., Bobrovnikova, K., Shchuka, R. Savenko, O. A Cyberattacks Detection Technique Based on Evolutionary Algorithms. *IEEE 11th International Conference on Dependable Systems, Services and Technologies*, 2020, pp. 127-132. DOI: 10.1109/DESSERT50317.2020.9125016.

35. Lysenko, S., Savenko, O., Bobrovnikova, K. DDoS Botnet Detection Technique Based on the Use of the Semi-Supervised Fuzzy c-Means Clustering. *CEUR-WS*, 2018, vol. 2104, paper no. 251, pp. 688-695.

36. Lysenko, S., Bobrovnikova, K., Matiukh, S., Hurman, I., Savenko, O.Detection of the botnets' low-rate DDoS attacks based on self-similarity. International *Journal of Electrical and Computer Engineering*, ISSN 2088-8708, 2020, vol. 10, no. 4, pp. 3651-3659. DOI: 10.11591/ijece.v10i4.pp3651-3659.

37. Lysenko, S., Bobrovnikova, K., Savenko, O., Shchuka, R. Technique for Cyberattacks Detection Based on DNS Traffic Analysis, *CEUR-WS*, 2020, vol. 2623, paper no. 19. pp. 208-218.

38. Lysenko, S., Bobrovnikova, K., Savenko, O., Kryshchuk, A. BotGRABBER: SVM-Based Self-Adaptive System for the Network Resilience Against the Botnets' Cyberattacks. *Communications in Computer and Information Science*, 2019, pp. 127-143.

39. *IoT dataset*. Available at: https://github.com/thieu1995/iot_dataset (accessed 11.01.2022).

40. *IoTPOT*. Available at: https://sec.ynu.codes/iot/ (accessed 11.01.2022).

41. Pa, Y. M. P., Suzuki, S., Yoshioka, K., Matsumoto, T., Kasama, T., Rossow, C. IoTPOT: A novel honeypot for revealing current IoT threats. *Journal of Information Processing*, 2016, vol. 24, no. 3, pp. 522-533. DOI: 10.2197/ipsjjip.24.522.

42. *Google Play*. Available at: https://play.google.com/store/apps?hl=ru&gl=US (accessed 11.01.2022).

43. *OpenWrt*. Available at: https://openwrt.org/ (accessed 11.01.2022).

44. *Radare2*. Available at: https://www.radare.org/n/ (accessed 11.01.2022).

45. *NetworkX*. Available at: https://networkx.org/ (accessed 11.01.2022).

46. Choudhary, S., Sharma, A. Malware Detection & Classification using Machine Learning, *International Conference on Emerging Trends in Communication, Control and Computing*, 2020, pp. 1-4. DOI: 10.1109/ICONC345789.2020.9117547.

47. Tirandasu, R. K., Prasanth, Y. A Review on Malicious Software Detection using Machine Learning Algorithms, *Second International Conference on Electronics and Sustainable Communication Systems (ICESC),* 2021, pp. 1945-1948. DOI: 10.1109/ICESC51422.2021.9532700.

48. Singh, P., Kaur, S., Sharma, S., Sharma, G., Vashisht S., Kumar, V. Malware Detection Using Machine Learning, *International Conference on Technological Advancements and Innovations* (ICTAI), 2021, pp. 11-14. DOI: 10.1109/ICTAI53825.2021.9673465.

49. Köse, Ü., Samet, R. Detection of Malware with Deep Learning Method, *International Conference on Computer Science and Engineering* (UBMK), 2021, pp. pp. 665-669. DOI: 10.1109/UBMK52708.2021.9559020.

Надійшла до редакції 15.01.2022, розглянута на редколегії 16.02.2022

МЕТОД ВИЯВЛЕННЯ ЗЛОВМИСНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕРНЕТУ РЕЧЕЙ НА ОСНОВІ АНАЛІЗУ ГРАФОВ ПОТОКІВ КЕРУВАННЯ

К. Ю. Бобровнікова, С. М. Лисенко, Б. О. Савенко, П. Гай, О. С. Савенко

Під Інтернетом речей розуміють мільйони пристроїв у всьому світі, підключені до Інтернету. Незахищені пристрої Інтернету речей, розроблені без належних функцій безпеки, є метою багатьох Інтернет-загроз. Швидка інтеграція Інтернету в інфраструктуру Інтернету речей у різних сферах людської діяльності, включаючи вразливу критичну інфраструктуру, робить виявлення зловмисних програм в Інтернеті речей все більш важливим. Щорічні звіти компаній, що займаються кібербезпекою інфраструктури Інтернету речей, та виробників антивірусного програмного забезпечення показують зростання кількості атак зловмисного програмного забезпечення, спрямованих на інфраструктуру Інтернету речей. Це свідчить про неспроможність сучасних методів виявлення зловмисних програм в Інтернеті речей. Ось чому існує нагальна потреба в розробленні нових підходів до виявлення зловмисних програм в Інтернеті речей і захисту пристроїв Інтернету речей від атак зловмисного програмного забезпечення Інтернету речей. **Предметом** дослідження є процес виявлення зловмисних програм в Інтернеті речей. **Метою** роботи є розробка методу виявлення зловмисного програмного забезпечення в Інтернеті речей на основі аналізу графу потоку управління. **Результати.** У статті представлено новий підхід до виявлення зловмисного програмного забезпечення в Інтернеті речей на основі аналізу графа потоку управління. Графи потоку управління будуються для підозрілих додатків Інтернету речей. Граф потоку управління представлений у вигляді орієнтованого графа, який представляє компоненти підозрілої програми та переходи між ними. На основі графа потоку управління можна вилучити метрики, які описують структуру програми. Враховуючи, що програми Інтернету речей мають невеликі розміри через простоту та обмеження середовища операційних систем Інтернету речей, виявлення зловмисного програмного забезпечення на основі аналізу графу потоку управління є можливим у середовищі Інтернету речей. Щоб проаналізувати поведінку програми Інтернету речей для кожного графа потоку управління потрібно побудувати граф дій. Це абстрактний опис програми у вигляді графа. На основі отриманого графа дій для кожного додатка Інтернету речей формується набір послідовностей дій. Це дозволяє визначити поведінку програми. Таким чином, з метою виявлення зловмисного програмного забезпечення Інтернету речей використовуються дві моделі виявлення зловмисних програм: заснована на метриках, вилучених з графу потоку управління, та на послідовностях дій. Завдяки тому, що підхід дозволяє аналізувати як загальну структуру програми, так і поведінку кожного додатка, він дозволяє досягти високої точності виявлення зловмисних програм. Запропонований підхід дозволяє виявляти невідомі зловмисні програми Інтернету речей, які є модифікованими версіями відомих зловмисних програм Інтернету речей. Як засіб для висновку щодо наявності зловмисного програмного забезпечення використовувався набір класифікаторів на основі машинного навчання. Результати експериментів продемонстрували високу точність виявлення зловмисних програм Інтернету речей. **Висновки.** Розроблено новий метод виявлення зловмисних програм Інтернету речей на основі аналізу графа потоку управління. Метод здатний з високою ефективністю виявляти зловмисне програмне забезпечення Інтернету речей.

Ключові слова: зловмисне програмне забезпечення; Інтернет речей; пристрої Інтернету речей; додаток Інтернету речей; кібербезпека; кібератака; граф потоку управління; виявлення кіберзагрози.

МЕТОД ВЫЯВЛЕНИЯ ВРЕДОНОСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНТЕРНЕТА ВЕЩЕЙ НА ОСНОВЕ АНАЛИЗА ГРАФОВ ПОТОКОВ УПРАВЛЕНИЯ

К. Ю. Бобровникова, С. Н. Лысенко, Б. О. Савенко, П. Гай, О. С. Савенко

Под Интернетом вещей понимают миллионы устройств во всем мире, подключенные к Интернету. Незащищенные устройства Интернета вещей, разработанные без надлежащих функций безопасности, являются целью многих Интернет-угроз. Быстрая интеграция Интернета в инфраструктуру Интернета вещей в различных сферах человеческой деятельности, включая уязвимую критическую инфраструктуру, делает обнаружение вредоносных программ в Интернете вещей все более важным. Ежегодные отчеты компаний, занимающихся кибербезопасностью инфраструктуры Интернета вещей, и производителей антивирусного программного обеспечения показывают рост количества атак вредоносного программного обеспечения, направленных на инфраструктуру Интернета вещей. Это свидетельствует о несостоятельности современных методов обнаружения вредоносных программ в Интернете вещей. Вот почему существует острая потребность в разработке новых подходов к выявлению вредоносных программ в Интернете вещей и защиты устройств Интернета вещей от атак вредоносного программного обеспечения Интернета вещей. **Предметом** исследования является процесс обнаружения вредоносных программ в Интернете вещей. **Целью** работы является разработка метода обнаружения вредоносного программного обеспечения в Интернете вещей на основе анализа графа потока управления. **Результаты.** В статье представлен новый подход к выявлению вредоносного программного обеспечения в Интернете вещей на основе анализа графа потока управления. Графы потока управления строятся для подозрительных приложений Интернета вещей. Граф потока управления представлен в виде ориентированного графа, представляющего компоненты подозрительной программы и переходы между ними. На основе графа потока управления можно извлечь метрики, описывающие структуру программы. Учитывая, что программы Интернета вещей имеют небольшие размеры из-за простоты и ограничения среды операционных систем Интернета вещей, обнаружение вредоносного программного обеспечения на основе анализа графа потока управления возможно в среде Интернета вещей. Чтобы проанализировать поведение программы Интернета вещей, для каждого графа потока управления нужно построить граф действий. Это абстрактное описание программы в виде графа. На основе полученного графа действий для каждого приложения Интернета вещей формируется набор последовательностей действий. Это позволяет определить поведение программы. Таким образом, для выявления вредоносного программного обеспечения Интернета вещей используются две модели обнаружения вредоносных программ: основанная на метриках, изъятых из графа потока управления, и на последовательностях действий. Благодаря тому, что подход позволяет анализировать как общую структуру программы, так и поведение каждого приложения, он позволяет добиться высокой точности обнаружения вредоносных программ. Предлагаемый подход позволяет выявлять неизвестные вредоносные программы Интернета вещей, которые являются модифицированными версиями известных вредоносных программ Интернета вещей. В качестве средства вывода о наличии вредоносного программного обеспечения использовался набор классификаторов на основе машинного обучения. Результаты экспериментов продемонстрировали высокую точность обнаружения вредоносных программ Интернета вещей. **Выводы.** Разработан новый метод обнаружения вредоносных программ Интернета вещей на основе анализа графа потока управления. Метод способен с высокой эффективностью выявлять вредоносное программное обеспечение в Интернете вещей.

Ключевые слова: вредоносное программное обеспечение; Интернет вещей; устройства Интернета вещей; приложение Интернета вещей; кибербезопасность; кибератака; граф потока управления; обнаружение киберугрозы.

Бобровнікова Кіра Юліївна – канд. техн. наук, доц. каф. комп’ютерної інженерії та інформаційних систем, Хмельницький національний університет, Хмельницький, Україна.

Лисенко Сергій Миколайович – д-р техн. наук, проф., проф. каф. комп’ютерної інженерії та інформаційний систем, Хмельницький національний університет, Хмельницький, Україна

Савенко Богдан Олегович – асп. каф. комп’ютерної інженерії та інформаційний систем, Хмельницький національний університет, Хмельницький, Україна.

Гай Пьотр – д-р техн. наук, інститут комп’ютерних наук, Сілезький технологічний університет, Глівіце, Польща.

Савенко Олег Станіславович – д-р техн. наук, проф., декан факультету інформаційних технологій, Хмельницький національний університет, Хмельницький, Україна.

Kira Bobrovnikova – PhD, Associate Professor of Computer Engineering & Information Systems Department, Khmelnytskyi National University, Khmelnytskyi, Ukraine,
e-mail: bobrovnikova.kira@gmail.com, ORCID: 0000-0002-1046-893X.

Sergii Lysenko – Doctor of Science, Full Professor, Professor of Computer Engineering & Information Systems Department, Khmelnytskyi National University, Khmelnytskyi, Ukraine,
e-mail: sirogyk@ukr.net, ORCID: 0000-0001-7243-8747.

Bohdan Savenko – PhD student of Computer Engineering & Information Systems Department, Khmelnytskyi National University, Khmelnytskyi, Ukraine,
e-mail: savenko_bohdan@ukr.net, ORCID: 0000-0001-5647-9979.

Piotr Gaj – Doctor of Science, Institute of Computer Science, Silesian University of Technology, Gliwice, Poland,
e-mail: piotr.gaj@polsl.pl, ORCID: 0000-0002-2291-7341.

Oleg Savenko – Doctor of Science, Full Professor, the Dean of the Information Technologies Faculty of National University, Khmelnytskyi, Ukraine,
e-mail: savenko_oleg_st@ukr.net, ORCID: 0000-0002-4104-745X.