Method of Ensuring the Failure Resistance of Specialized Information Technologies

Mykola Stetsyuk [a], Antonina Kashtalian[a], Andrzej Kwiecień[b], Oleg Sachenko[c]
[a] *Khmelnytskyi National University, Institutska str., 11, Khmelnytskyi, 29016, Ukraine*
[b] *Silesian University of Technology, Poland*
[c] *West Ukrainian National University, 11 Lvivska Str., 46009, Ukraine*

Abstract

An abstract model of the effects of malicious software is proposed. It allows you to identify objects in your computer system that may be affected by malicious software and computer attacks. It was used to develop a new method of ensuring the resilience of specialized IT in the face of malware and computer attacks. The resulting abstract model makes it possible to detail the objects for influences and possible consequences, which becomes the basis for the development of methods that will ensure fault tolerance, survivability of information systems and protection of information in information systems from such influences. The abstract model is the basis for the creation of specialized IT, the stable operation of which is possible under the influence of malicious software and computer attacks. Also, this model may include a feature that will be the distribution of computer system objects in the computer network and components of specialized IT.

The application of the developed method is implemented in the system. It has mechanisms for restructuring and uses redundancies. The step of the developed method has features that are parametric control of the integrity of program files. Then, it is possible to apply it to a group of program files. These files do not have a fixed checksum and, therefore, the method extends the capabilities of the known method of detecting malicious software.

The developed method provides the possibility of independent restructuring of information systems in the process of functioning with the involvement of hardware and software. In the process of restructuring information systems, the implementation of the specified functions continues. Thus, the method of ensuring the resilience of IT in the face of malicious software and computer attacks allows you to expand the capabilities of the IP in terms of its adaptability and, accordingly, the automatic change of hardware and software configuration. In addition, the steps of the developed method integrated two ways to ensure IT resilience: attracting redundancy; attracting excesses. This integration is combined with the adaptability of information systems.

Research of the method developed a method of evaluating the effectiveness of redundancies and redundancy. Experimental researches and estimation calculations are carried out. They confirmed the effectiveness of the developed method of ensuring the resilience of IT in the face of malware and computer attacks.

Keywords [1]

fault tolerance, malware, computer attacks, specialized information technologies, redundancies, redundancy, information systems

1. Introduction

The continued spread of malware and various computer attacks confirms that the problem of combating malicious software and computer attacks will remain relevant, and its relevance will only increase. Creating an information system to avoid malware threats and computer attacks is possible using scientific approaches based on threat models and methods based on them.

IntelITSIS’2022: 3rd International Workshop on Intelligent Information Technologies and Systems of Information Security, March 23–25, 2022, Khmelnytskyi, Ukraine

EMAIL: mikstt777@gmail.com (M. Stetsyuk); yantonina@ukr.net (A. Kashtalian); andrzej.kwiecien@polsl.pl (A. Kwiecień); os@wunu.edu.ua (O. Sachenko);
ORCID 0000-0003-3875-0416 (M. Stetsyuk); 0000-0002-4925-9713 (A. Kashtalian); 0000-0003-1447-3303 (A. Kwiecień); 0000-0001-9337-8341 (O. Sachenko)

When designing information systems, you need to take into account the peculiarities of the tasks assigned to them, which can be performed under the influence of malicious software and computer attacks. In this regard, the necessary scientific task is to develop specialized IT that will provide opportunities to combat malicious software and computer attacks, which will allow to develop on their basis resistant to such influences information systems.

2. Analysis Of Known Solutions

Consider the known scientific solutions to ensure fault tolerance in information technology.

The paper [1] presents a method based on an analytical study of the model for predicting productivity. Presents an overview of fault tolerance methods for high-performance computing. The method is based on checkpoint protocols and algorithms for planning, predicting, replicating, silent detection and correction of errors, as well as some program-specific methods. The emphasis is on analytical performance models. This is followed by an overview of general purpose methods, including several protocols for restoring checkpoints and rollbacks. Relevant execution scenarios are also evaluated and compared using quantitative models. An exhaustive review of the latest key studies of fault-tolerant architectures in hardware security and cryptography software is presented in [2]. The latest bug attacks on a wide range of cryptographic implementation tools are discussed, such as biased bug attacks, algebraic bug attacks, and attacks on authenticated ciphers based on encryption. Offers practical demonstrations of work with complex safe cryptographic architectures, which complements theoretical analysis. The paper uses motivating examples and real-world attack scenarios to introduce readers to the general concept of attacks with errors in cryptography. It offers insight into how the fault tolerance theories developed in the book can actually be implemented, with a particular focus on a wide range of fault models and practical fault injection techniques ranging from simple inexpensive methods to high-quality equipment. It also provides an example of a comprehensive fault-tolerant architecture based on the FPGA for AES-128, which combines a number of malfunctions. The study concludes with a discussion of how fault tolerance can be combined with the safety of side channels to achieve protection against implementation-based attacks.

Fault-tolerant control is aimed at gradual shutdown in automated systems in case of malfunctions analyzed in the work [3]. It satisfies industrial demand for increased affordability and safety, as opposed to traditional fault responses that lead to sudden outages and loss of availability. The book presents effective methods of analysis and design based on the model for the diagnosis of malfunctions and malfunctions. Architectural and structural models are used to analyze the spread of faults in the process, to check the possibility of detecting a malfunction and to find excesses in the process that can be used to ensure resistance to failure. It also presents appropriate design methods for diagnostic systems and fault-tolerant controllers for continuous processes described by analytical models of discrete event systems represented by automata. The framework of methods of monitoring and control of processes based on data is analyzed [4]. The development of fault diagnostic systems and fault-tolerant data-driven control systems present the main statistical methods for monitoring the process, troubleshooting and control and implements advanced data-driven circuits for the design of fault diagnostics and fault-tolerant control systems. With ever-increasing requirements for reliability, availability and safety of technical processes and assets, process monitoring and fault tolerance have become important issues related to the design of automatic control systems. The works [5]-[7] analyze the impact of malware on the functioning of information systems, in particular by ensuring this impact with metamorphic computer viruses. The study of such influences was carried out by analyzing the API of calls. This made it possible to form behavioral signatures. The process of researching this class of viruses is associated with the topic of ensuring the stability of information technology in the context of malware exposure through a joint study of the mechanisms of influence.

The work [8] confirmed the advanced diagnosis of malfunctions and fault-tolerant circuits with the help of real aerospace systems. The possibility of setting up and checking system for detecting and diagnosing malfunctions based on the model is shown. Fault diagnostics and fault-tolerant control and guidance for aerospace technology demonstrates the attractive potential of recent management developments to address challenges such as flight performance, self-defense and extended lifespan designs. The problem of design and testing of safety and fault tolerance is devoted to the work [9]. It shows various issues related to security, privacy and fault tolerance. Various aspects of security, confidentiality and reliability in the development, analysis and testing of the Internet of Things and cyber-physical systems are proposed. In particular, various well-established theories and practices of both scientific and industrially oriented models are presented and properly organized. The paper [10] describes the comprehensive application of training methods based on models, based on data and statistics. This helps to understand advanced and integrated design, as well as online methods for optimizing fault diagnostics and fault-tolerant control in complex systems. Describes engineering tools to solve diagnostic and fault tolerance problems, control problems. Advanced design methods and online optimizations for troubleshooting and fault-tolerant control on various aspects are presented. Given the types of systems, fault diagnostics and fault tolerance problems are considered for linear unchanging in time and time-changing systems, as well as for nonlinear and distributed (including network) systems. From a methodological point of view, diagrams are investigated both on the basis of the model and on the basis of data. The work [11]-[12] shows that Honeypots play an important role in network security because they receive information about attackers, their goals, methods, and tools. This is important to ensure it is fault-tolerant in computer attack impact languages. In addition, in the next era of Industry 4.0, interpersonal messages will become increasingly important. The predominant online messaging service has always been emailed. Therefore, the properties of e-mail and its main competitors are considered to understand the current problems of e-mail. Therefore, ensuring the fault tolerance of IT is shown in the work is an important and promising area of research.

The work [13] presents the diagnosis of malfunctions and fault-tolerant control, which is solved with the help of differential algebra and fractional calculus tools, contains simulations and results of experiments in real time. Algebraic and differential methods are presented, as well as fractional calculus, which are used to diagnose and reject malfunctions in nonlineural systems of the whole or fractional order. This is an extension of a very important and widely studied problem in management theory, namely the diagnosis and deviation of malfunctions (using differential algebraic approaches), to systems representing fractional dynamics, that is, systems whose dynamics are represented by derivatives and integrals of an untethered order. The authors offer a thorough overview on fault-tolerant management applied to dynamic fractional and whole-order systems, and they are introducing new management and surveillance methodologies described by fractional and integer models, along with successful real-time modeling and applications. The work [14] presents the selected diagnostics of malfunctions and fault-tolerant control strategies for nonlinear systems in a single structure. To implement the proposed framework for fuzzy systems described using the well-known Takagi-Sugeno models, selected fault diagnostics and fault-tolerant control strategies for nonlinear systems in a single structure are presented. This book [15] focuses on the development of an observer to diagnose faults for a dial-up system. Model-based fault diagnostics and fault control are one of the most popular areas of research in recent decades. With the spread of the Internet of Things through wireless sensor networks, a huge amount of sensor data is generated at unprecedented speeds, leading to a very large amount of explicit or implicit information [16]. Resistance to disruptions in cloud computing is becoming massive due to the unprecedented growth of the cloud [17]. To improve the reliability of cloud service, various methods of fault tolerance were proposed in the literature. Many problems with the reliability and availability of cloud services are caused by the failure of virtual machines. Thus, this study contributes to a new approach of adaptive fault tolerance based on thresholds in the cloud by combining proactive and reactive approaches.

The fault tolerance of the multi-agent system [18] has attracted the research community over the past decade. With the development of consensus theory in multi-agent systems, many researchers have paid attention to the need to manage the complexity of the system, considering system errors in various processes of agent software development. The works [19], [20] present means of countering computer attacks based on multi-agent technologies, which in turn improve the fault tolerance of such systems. The work [21] describes the use of integrated sliding modes within the framework of fault-tolerant control tasks. New integrated circuits of sliding mode for linear parameters are presented. The results in the work [22] focus on the system of fault-tolerant control from the point of view of the theory of the behavioral system. As the importance of [23] public cloud services for sharing various files, including copyrighted content, increases, a survey has been organized among Czech university students about their use of such services to identify their habit of using such services. The results of more than 750 respondents showed that there are certain typical patterns of use. But such systems must be provided with fault-tolerant means. The high number of components and subsequent impact on volume and reliability were the main problems for the practical use of multilevel inverters [24]. In this paper, some of the newly proposed topology has been reviewed and analyzed in light of the possibility of fault tolerance in the event of a malfunction of the open switch of any of the power switches. The results obtained in this way are experimentally confirmed to prove the feasibility of the proposed fault-tolerant topology. The work [25] provides advanced methods of fault tolerance in the control computer of the spacecraft, with an emphasis on practical engineering knowledge. The study of the sustainability of the property of the system is devoted to the work [26]. This article contributes to current sustainability studies by focusing on the different definitions given for this system property. The work [27] presents the effects of malware and computer attacks in internet of things systems.

JARVIS is a research and development project [28] developed jointly by industrial partners from small and medium-sized businesses and the University of Florence aimed at developing a hardware and software system that supports integration between physical IoT devices, data analysis software agents and human operators involved in operation and maintenance. Sustainable Industry Systems 4.0. At the heart of the JARVIS architecture is a set of software digital twins deployed in a Java EE environment that supports monitoring during execution and control of the hierarchy of system hardware configuration elements, capturing their composition and presenting their failure modes using an architectural display template that allows flexible adaptation to the evolution of configurations. In addition, analytical modules can be deployed as microservices, using both the knowledge base provided by digital twins and data coming from the level of reception. This allows you to quickly develop advanced monitoring and control services that maintain maintainability and sustainability. In growing information systems, systems [29] are becoming a new research frontier. They are formed as a set of constituent systems that live on their own with well-established functions and requirements, and, under certain circumstances, they must cooperate to achieve a common mission. In this scenario, security is one of the decisive properties that must be taken into account in the early stages of the life cycle. The article [31] is devoted to a detailed description of the distributed system for collecting data from workstations and servers based on Windows. The study presented at the outset demonstrates that neither solutions for collecting data on attacks on Windows-based computers are currently available, nor other security tools and additional programs can be combined to achieve the necessary collection of attack data from Windows computers.

Given that the scientific task of ensuring the resilience of specialized information technology in the context of malicious software and computer attacks is not solved, so it is relevant.

3. Abstract Malware Impact Model on Objects of Computer Systems

In order to ensure the stability of information systems to the effects of malicious software and computer attacks in the process of their functioning, we synthesize in IT information systems with specialized functionality for its appointment, as well as components, the purpose of which will be to maintain the performance of information systems to perform specialized functionality for performing the main task in the face of malware and computer attacks. We will set the components of specialized IT M_{IT}:

$$M_{IT} = \{F_0, F_1, F_2, \dots, F_{N_{IT}}, A_{IT}\}, \quad (1)$$

where F_0– functionality of the main IT task and is necessarily present in M_{IT}; i – the same component element in IT, which provides additional functionality; $i = 1,2, \dots, N_{IT}$; N_{IT} – number of additional components $F_1, F_2, \dots, F_{N_{IT}}$ in IT that activates elements F_0 in IT for occurrence of certain events or requests from the element F_0 and it does not contain additional functionality to perform other actions.

Since modern information systems can have different architectures that will affect the design of IT, and they are mainly distributed, its constituent elements presented in formula 1 will be considered to combine, respectively, in their elements all the components that are located in different computer stations, but have a component related to a component of a certain type. In particular, with this representation, we will have the following ratios:

$$M_{IT} = \left\{ \begin{matrix} F_0 |\, F_0 = \bigcup_{i=1}^{N_{IT}} F_{0,i} \\ F_1 |\, F_1 = \bigcup_{i=1}^{N_{IT}} F_{1,i} \\ \dots \\ F_{N_{IT}} |\, F_{N_{IT}} = \bigcup_{i=1}^{N_{IT}} F_{N_{IT},i} \\ A_{IT} |\, A_{IT} = \bigcup_{i=1}^{N_{IT}} A_{IT,i} \end{matrix} \right\}, \quad (2)$$

where $F_{0,i}$ – functionality of the main IT task in the i - th computer station and is necessarily present in M_{IT}; $i = 1,2,\dots,N_{ks}$; N_{ks} — multiple information systems of computer stations in which the components of the information systems are installed; $F_{j,i}$ - j is the component element in IT in the i - th computer station, which provides additional functionality; $j = 1,2,\dots,N_{IT}$; N_{IT}, i – number of additional components in IT; $A_{IT,i}$ – an integral element in IT in the i- computer station that activates elements, $F_{1,i}, F_{2,i}, \dots, F_{N_{IT},i}$ in IT for occurrence of certain events or requests from the element $F_{0,i}$ and it does not system additional functionality to perform other actions.

Not all components of the information systems, which are located in computer stations, can be placed all the constituent elements, $F_{j,i}$, where $i = 1,2,\dots,N_{ks}$; N_{ks}– multiple information systems of computer stations in which the components of the information systems are installed; $j = 1,2,\dots,N_{IT}$. Also, in different computer stations there may be different constituent elements that belong to the same type. In particular, these components may differ for the server and the components in the computer station. But more informational systems of components of the same type in different components of computer stations may be the same. This, in addition to simplifying the development of specialized IT, allows you to synchronize, as well as, the means of maintaining the stability of information systems when exposed to malware or computer attacks through coordination and interaction between them directly or with the involvement of the server part of the information systems. This demonstrates the possible information systems to scale the communities of specialized IT between different computer stations in networks and possible information systems to perform the task within the same computer station.

The components in IT will synthesize the following: fault-tolerant information systems, living information systems, information protection. These components will be implemented as separate completed modules, but with possible information activation systems in the conditions of signaling about the impacts and needs that the functionality of the main task will require. That is, at $N_{IT} = 3$, then the generalized structure of specialized IT will have a representation shown in Fig. 1.

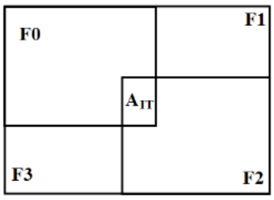


Figure 1: Generalized structure of specialized IT with elements

Consider presenting the possible impacts of malware and computer attacks on information systems in computer networks. The study of such influences is important at all stages of the functioning of the information systems and in all computer stations as a whole and separately. The effects of malware and computer attacks on computer stations on the network can be directed to their various objects, both hardware, hardware and software. Moreover, these influences can be multi-stage and one-stage, distant, directly directed to the object and indirectly. They can be implemented by various means, which can be ordinary means for working in the network and computer systems, as well as specially created tools. All this diversification of malicious influences not only complicates the process of detecting malware and computer attacks by special antivirus tools, but also complicates the classification of malicious actions. Today, to carry out an accurate classification of malicious influences, there is a small sign field of characteristics, so special antivirus tools do not provide full detection. There remains a set of malware and computer attacks that penetrate through these special antivirus tools. Therefore, despite the various information systems of malware and computer attacks, without examining their features and displaying them in the databases of malicious programs and attacks carried out in the implementation of special anti-virus tools, it is advisable to study their possible variants of malicious influences on specific objects of computer systems. That is, the construction of possible influences precisely through the formation of a set of such influences in relation to specific objects in computer systems will allow to form a limited set of malicious influences, each of the elements of which will be associated with a certain object of the computer system, as well as, for example, with certain elements of the information systems, the stages of its functioning, including the beginning of work and completion, including its various components. In connection with this correlation of influences to objects of computer systems, taking into account their time of functioning, we obtain connections of specific objects over time with possible influences on them. In addition, exposure to malware and computer attacks can be destructive and non-destructive. Non-destructive influences can be divided into those that have not achieved their goal and, therefore, the processes created by them function sum informing systems or in parallel with the processes created by a given user or computer system and aimed at other objects in the computer system, that is, objects from a given information systems and resources necessary for its functioning. Some of the non-destructive effects at certain points in the future may move to the category of destroyers. Destructive influences can be directed to information systems, in which countermeasures mechanisms will be implemented, or to other objects of the computer system that are not related to the information systems and resources for its functioning. In addition, such influences can achieve both a partial goal at a certain point in time, and in the subsequent resulting goal of disabling a computer station, a network node, suspending or destroying processes, destroying information on a hard disk, etc.

Set the impacts of malware and computer attacks plural, M_{VP}:

$$M_{VP} = M_{VP,r} \cup M_{VP,nr}, \tag{3}$$

where $M_{VP,r}$, – a set of destructive influences; $M_{VP,nr}$ – a set of non-destructive influences. Attribution of influences to subsets of set M_{VP} depends on the current time and may change. Let's set a subset of influences by listing their elements as follows:

$$M_{VP,r} = \{m_{VP,r,1}, \dots, m_{VP,r,n_{VP,r}}\}, M_{VP,nr} = \{m_{VP,nr,1}, \dots, m_{VP,nr,n_{VP,nr}}\}, \tag{4}$$

where $m_{VP,r,i}$ – the element of the set $M_{VP,r}$, which means i -that destructive effect at a certain point in time; $i = 1,2, \dots, n_{VP,r}$; $n_{VP,r}$ – general multiple information systems of destructive effects; $m_{VP,r,j}$ – the element of the set $M_{VP,nr}$, which means j -th non-destructive effect at a certain point in time; $j = 1,2, \dots, n_{VP,nr}$; $n_{VP,nr}$ – total number of non-destructive influences.

Part of the non-destructive effects of the set $M_{VP,nr}$ in the process of its implementation may not harm the objects of the computer station. This can happen due to the changes in the formation of system information systems of functions specified in them, and through imperfect information systems of functionality in a certain environment of a computer station. The rest of the non-destructive effects at some point in time can go to the destroying. Such consideration of influences in dynamics is necessary for the behavior of the model of influences in relation to computer station objects that are dynamically changing.

The direction of influences of malware and computer attacks can be carried out on information systems, for which mechanisms for ensuring stability during impacts are designed and which is specified by the set M_{IT}, and the resources that ensure its functioning. Also, the direction of influences can be carried out on the objects of a computer station that are not related to the information systems and the impact on them will not affect the functioning of the information systems. Therefore, we consider, as possible, two types of such influences. Since influences can dynamically change from non-destructive to destructive, we will set the set M_{VP} list of its elements as follows:

$$M_{VP} = \{m_{VP,1}, \dots, m_{VP,n_{VP}}\}, \tag{5}$$

where $m_{VP,i}$ – the element of the set M_{VP}, which means i -th influence at a certain point in time; $i = 1,2, \dots, n_{VP}$; n_{VP} – general leveling system of influences. The result of the impacts of malware and computer attacks on computer systems will be the consequences, the set of which will be set as follows:

$$M_r = \{m_{r,1}, \dots, m_{r,n_r}\}, \tag{6}$$

where $m_{r,i}$ – the element of the set M_r, which means i is the result of exposure at a certain point in time; $i = 1,2, \dots, n_r$; n_r – general leveling system of consequences of influences. If the effects of malware and computer attacks are associated with the objects of the computer systems to which they are aimed, and the result of such interactions will be consequences, then these consequences will be as follows:

$$M_r = \begin{pmatrix} m_{r,1,1} & \cdots & m_{r,1,N_{VP}} \\ \vdots & \ddots & \vdots \\ m_{r,N_{IT},1} & \cdots & m_{r,N_{IT},N_{VP}} \end{pmatrix}, \tag{7}$$

where $m_{r,i,j}$ – an element of a set of consequences of influences on objects of computer systems; $i = 1,2,\dots,N_{VP}$; $j = 1,2,\dots,N_{IT}$.

We introduce for a set of objects of the computer system and the influences of malware and computer attacks algebraic structure as follows:

$$\Omega = \langle\, \Omega_{ks},\ \ \Omega_{VP},\ \ \Omega_{RVP} \rangle, \tag{8}$$

where Ω_{ks} – a set of computer system objects that can be influenced by malware and computer attacks; Ω_{VP} – a set of functions that implement the effects of malware and computer attacks; Ω_{RVP} – a set of predicates given on the set Ω_{ks}, which reflect successful information systems / unsuccessful information systems in the implementation of functions from the set Ω_{VP}; $\alpha = 1,\ \beta = 1$ is the arousal of operations, so the system type is $\tau = (1, 1)$.

As elements of the set Ω_{ks} computer system objects will consider all objects of the file system, the boot sector of the disk, RAM, network packages, which can be the object of influences of malware and computer attacks. Elements of the set Ω_{VP} there are isolated elements that inform the unified functionality, the implementation of which allows you to carry out malicious influence of malware and computer attacks on a specific single object of the computer system and their combinations. To achieve the result of the effect of a single element from the set Ω_{VP} may involve some of the other objects of the computer system, that is, to exert indirect influence, but the impact is still directed at one object. Then, a combination of such elements will form the remaining elements of this set Ω_{VP}. Such elements of the set Ω_{VP} are generating for the rest of the different elements of this set. Plural functions Ω_{VP} influences will not always be successfully implemented, so to present the impacts of malware and computer attacks with a set of predicates, Ω_{RVP}. It will reflect the result of the successful/unsuccessful impact of malware and computer attacks on computer systems. Predicates belonging to the set Ω_{RVP} determine that they will be information systems if the result of malicious exposure to malware and computer attacks on an object or objects of a computer system is hastily, that is, a function from a set of Ω_{VP} will be executed. Otherwise, the result of the predicate will be false. Then, let's move from the formula (8) to the abstract model, which we will set as follows:

$$\mathfrak{R} = \langle\, \Omega_{ks},\ \Omega_{RVP} \rangle, \tag{9}$$

where Ω_{ks} – a set of computer system objects that can be influenced by malware and computer attacks; Ω_{RVP}– a set of predicates given on the set Ω_{ks}, which reflect the success / failure in the implementation of functions from the set Ω_{VP}; $\alpha = 1,\ \beta = 1$ is the arousal of operations, so the system type is $\tau = (1,\ 1)$.

If the effects of malware and computer attacks are successful, then they will have consequences, that is, refer to the set M_r, which is given by the formula (6). As a result, the function of displaying elements of the set of influences M_{VP} in the set of consequences M_r:

$$\Omega_{RVP}: M_{VP} \xrightarrow{\Omega_{VP}} M_{RVP}. \tag{10}$$

The result of this representation is an abstract model and a set of functions that provide possible information systems to represent the processes that are carried out in computer systems during the functioning of the information systems and the possible impacts of malware and computer attacks on computer systems. It combines components such as computer system objects, including components and elements of information systems, impacts on objects and impact traces. Thus, the resulting abstract model makes it possible to detail objects for influences and possible consequences, which becomes the basis for the development of methods that will provide fault-tolerant information systems, living information systems of information systems and information protection in information systems from such influences. The abstract model is the basis for the creation of specialized IT, the sustainable functioning of which is possible in the face of malware and computer attacks. Also, this model may include special information systems, which will consist in the distribution of computer systems in a computer network and components of specialized IT.

4. Method of Ensuring Fault Tolerance of Specialized IT

When providing fault tolerance with specialized IT mechanisms that will make it impossible to influence malware and computer attacks, we will consider IT components as being divided into server and client. If there are no server parts of IT, then the results will be used for clients, which will be considered as having part of the capabilities of server parts. The components of specialized IT information systems the software part and require certain hardware and software for their functioning, so the consideration of the effects of malware and computer attacks should be taken into account in these two components. According to data from the conjugation matrix (formula (7) of malware exposure and computer attacks with computer system objects, Then, it is necessary to develop a method of ensuring the fault tolerance of IT, which would make it impossible to successfully perform the display according to the formula (10), that is, the existing information systems of elements in the conjugation matrix (formula (7)), or would reduce their multiple information systems or reliable information system of appearance. Thus, there would be a provision of fault-tolerant IT systems in the face of malware exposure and computer attacks. Taking into account the need to integrate anti-malware mechanisms and computer attacks that can be applied equally to the reforming system elements from the conjuring matrix (formula (7), we will present the method of ensuring it fault tolerance with the main steps that will apply to both the client part and the server parts.

Consider the first step of the method of ensuring the fault tolerance of IT, the essence of which will be the use of block tags. client part of information systems during implementation. As for the application software, which includes client automated working information systems (ARM), the critical errors that can occur during the operation of the working information systems of the information systems are recorded together with their parameters in the system registry in an automatic way and, subsequently, are used for analysis in order to eliminate the causes that caused them. This became possible thanks to the strategy, which is based on the introduction of some redundancy in the software ARM information systems, similar to the methods of ensuring the fault tolerance of the hardware information systems. To this end, all settlement procedures that hypothetically can system critical for the functioning of ARM errors are developed in compliance with a certain similar template for constructing algorithms for their implementation.

In the structure of the stages of the first step of the method, the algorithm for performing any non-trivial procedure is divided into two interacting blocks. In the first block, the function of the information systems procedure is implemented, and in the second error handler. In the process of performing some procedure that implements one of the functions of the ARM information systems, both blocks interact with each other, transferring control of the computing process to each other until the executable function is completed. The essence of the first step of the method according to block labels is that the algorithm that implements the information systems function is divided by markers (label 1, ..., label n) into fragments on the principle of functional completeness.

Before starting the execution of the current fragment of the algorithm, information about a hypothetically possible error is entered into the register of fatal errors (ARM instance code, function code, label number, time, etc.). In the future, the following options for the development of events are possible: 1. A fragment of the function algorithm was successfully executed. In this case, the information in the registry about the error that did not happen is destroyed, and the computing process proceeds to the execution of the next fragment. 2. An error occurred during the execution of the fragment, but it was successfully localized by the error handler. In this scenario, the error information may also be removed from the registry. 3. In the process of executing the fragment, an error occurred that was not localized by the error handler. In this case, information about a possible error will remain in the registry.

Thus, the proposed first step of the method according to block labels allows you to by default solve the problem of ensuring fault tolerance for the entire set of functions of the client part of the information systems. The second step of the method of ensuring the fault tolerance of IT is to use functional redundancy. The next of the significant internal factors that adversely affect the information system is overloading the hardware platform of the client PC with tasks, which can dramatically worsen the time parameters of the tasks performed by the ARM, or even make it impossible to work, due to the exhaustion of technical resources. In order to neutralize the effect of this factor on information systems, in the development of software, namely the part of it that is responsible for the implementation of "business logic", functional redundancy is used. The existing information systems of the functional reserve of "heavy" design functions allow maneuvering the computing power of the hardware platform information systems, in case of overloading of some of its links, thus increasing the fault-tolerant information systems of information systems. Since the procedure that is functionally reserved (for example, Funk1) is developed in two versions according to the same algorithm, but in different software environments, then for execution on various technical means this fact can be used to neutralize such a negative factor as the existing information systems errors in the application software ARM, in the case when an error appears in one of the options of the procedure. This shows a positive multiplicative information system of the effect of functional redundancy, which increases the overall fault-tolerant IT systems

The third step of the method of ensuring IT fault tolerance is cross-booking. The solution to the problem, as always in such cases, is to create some reserve. Analysis of the work of ARM information systems showed that some of them have a time reserve and over information systems of performance. Therefore, it was natural to decide to use this reserve at critical moments in the work of the client part of the information systems. As a reserve here serves any other, client PC, which, according to the plan to overcome the critical situation, can take over the maintenance of ARM, whose PC failed. This approach allows you not to keep a separate PC as a reserve, as well as to have stocks of components, which reduces operating costs, without losing the fault tolerance of the system as a whole. As a rule, software modules, in customized form, are stored in the information systems software repository and on those client PCs where they are planned to be used at critical moments according to the reservation plan. In case of failure of critical computer equipment, which will make it impossible to perform ARM of its functions, it is transferred to a suitable other computer. The time spent on reconfiguration of the client part is calculated in minutes, which is an acceptable value for ensuring the survivability of information systems that perform information support, for example, in such a subject applied area as the financial and economic activities of a higher education institution.

This reconfiguration of the client part became possible due to the fact that absolutely no data is stored on client computers on which the ARM software is executed. At the same time, the ARM software module itself, for convenience, is compiled into one file and does not require an installation procedure. It is enough to copy it to another computer. Information systems of which it will be ready to work. This approach allows even information systems to fail several PCs, which in itself has low probability information systems, to preserve the full functional information systems of information systems. There is only one limitation – each instance of the ARM software must first be registered with the information systems. Otherwise, an attempt to run such a program will be considered as an attempt to unauthorized access to the system, even with the correct user registration data. Control of information systems for all instances of their ARM allows you to block the attempts of attackers who managed to master the user's account data to gain access to the system. At the same time, the program that the attacker mastered does not receive access to the data of the information systems, and the fact of an attempt by such a program to connect to the system is recorded in the register of fatal errors with the relevant data, which allows them to take organizational measures against the attacker with their use. The fourth step of the method of ensuring IT fault tolerance is focused on application in the server part and, therefore, in the client part it will mostly not be applicable, except in cases of a combination of tasks and problems of both parts of the information systems.

The developed method provides for possible information systems for the independent restructuring of information systems in the process of functioning with the involvement of hardware and software. In the process of restructuring, information systems for performing given functions continue. Thus, the method of ensuring the fault tolerance of IT in the conditions of exposure to malware and computer attacks allows you to expand the capabilities of the information systems in terms of its adaptability and, accordingly, automatic change of hardware and software configuration. In addition, in the steps of the developed method, two ways to ensure the fault tolerance of IT are integrated: attracting reservations; involvement of excesses. This integration is combined with adaptive information systems information systems.

5. Experimental Research and Evaluation of the Effectiveness of the Method Ensuring Fault Tolerance of Specialized IT

Establishing the possibility of using the method of ensuring the fault tolerance of specialized IT in the conditions of exposure to malware and computer attacks will be carried out by conducting appropriate experimental studies and evaluating its effectiveness. Evaluation of the effectiveness of the method of ensuring the fault tolerance of specialized IT in the conditions of exposure to malware and computer attacks will be carried out according to the criteria that will meet the indicators involved in it and, accordingly, the functional capabilities. In particular, the following indicators are as follows: redundancy; automatic change of hardware and software configuration of information systems. We will assess the impact of various excesses on ensuring the fault tolerance of IT by the developed method. Let's set the set of excesses as follows:

$$M_{nd} = \{m_{nd,1}, \dots, m_{nd,p}\}, \qquad (11)$$

where $m_{nd,i}$ – i - and over-information systems in specialized IT; p – the number of the considered excesses that can be implemented in IT.

We believe that in the structure of specialized IT, taking into account the peculiarities of its use in the conditions of exposure to malware and computer attacks, there will be the following excesses: $m_{nd,1}$ - struktural; $m_{nd,2}$ - time; $m_{nd,3}$ - informational; $m_{nd,4}$ - functional; $m_{nd,5}$ - algorithmic; $m_{nd,6}$ - software; $m_{nd,7}$ - hardware; $m_{nd,8}$ – multilevel. We will make their contribution to specialized IT depending on the weighting factors:

$$O_{nd} = \alpha_i \cdot m_{nd,i}, \qquad (12)$$

where α_i – the weight factor of the contribution of redundancy to ensuring the fault tolerance of specialized IT; $m_{nd,i}$ $i.$ - i-th redundancy; $i = 1, \dots, p$; p – number of excesses.

Then, let's normalize the contribution of frills O_{nd} in ensuring the fault tolerance of specialized IT to establish its relationship with the effects of malware and computer attacks on computer systems and the consequences as follows:

$$Q_{nd} = \frac{\sum_{i=1}^{p} \alpha_i \cdot m_{nd,i}}{\sum_{i=1}^{p} m_{nd,i}}, \qquad (13)$$

where $\sum_{i=1}^{p} m_{nd,i} = p$, $\sum_{i=1}^{p} \alpha_i = 1$.

It follows from the formula (13) that not all excesses may be used for certain effects of malware and computer attacks. And this will be reflected by the corresponding values of the coefficients. Multiple information systems successfully performed functions from the set of influences Ω_{ks} will decrease with the use of excesses and will depend on the number of excesses involved, which will be expressed by the magnitude of their application assessment Q_{nd}. Therefore, the set of predicates Ω_{RVP}, given on the set Ω_{VP}, which will be information systems will decrease. In this regard, it is necessary to evaluate the element of the conjuring matrix (formula (7)) in the context of the application of the method in which redundancies are used. To do this, each element of the conjuring matrix will be considered separately and so that a method is applied to it. And, also, the case when the method is applied to several elements of the conjuring matrix at the same time. In this case, it is necessary to establish possible information systems that lose its effectiveness. For the case of applying the step of the method using excesses to one element of the conjugation matrix, we introduce an efficiency function and set it depending on the factors of influence and counteraction as follows:

$$Q_{m_{r,i}} = \frac{1}{Q_{nd}} \sum_{j=1}^{N_{VP}} Q_{m_{VP,j}}, \qquad (14)$$

where $m_{r,i}$ – the element of the set, M-r., which means i is the result of influences at a certain point in time; $i = 1,2, \dots, n_r$; n_r – general leveling system of consequences of influences; $m_{VP,j}$ – the element of the set M_{VP}, which means i -th influence at a certain point in time; $i = 1,2, \dots, N_{VP}$; N_{VP} – general leveling system of influences; Q_{nd} - the regularized amount of the contribution of excesses to counteract the influences; $Q_{m_{r,i}}$ – a value that reflects the consequence of the influences of information systems for counteracting the effects of excesses; $Q_{m_{VP,j}}$ – normalized value of influences implemented by functions and expressed by their corresponding estimates compared to all functions of influences.

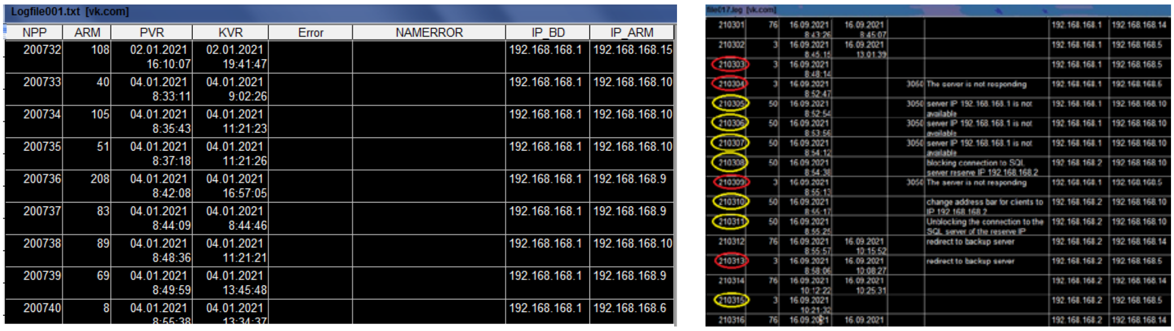
There may be several or one influences, or all available. Therefore, counteraction to them by means of fault tolerance may decrease with the simultaneous implementation of a wide range of different influences. This is reflected in the formula (14). But all these influences or one influence are focused on one object of the computer system in the formula (14). The result of this formula (14) will be a consequence of influences different from the effect that would be obtained without the involvement of frills from the first step of the method of ensuring fault tolerance.

If there are several objects of the computer system and influences will be concentrated on them, then this will also lower the result of resistance to influences, because the means of ensuring fault tolerance will additionally spend the resources of the computer system. Then, the result of the impacts will be estimated as follows:

$$\sum_{i=1}^{N_r} Q_{m_{r,i}} = \frac{N_r}{Q_{nd}} \sum_{j=1}^{N_{VP}} Q_{m_{VP,j}}. \tag{15}$$

The right side of the equality reflects that the overall fault tolerance assessment in this case reflects a decrease in the possibility of changing the effects of impacts. This assessment is scaled within a distributed system and receives a result for the server and computer stations in which the components of the information systems are installed. Thus, the obtained formulas (14), (15) make it possible to assess the impact of excesses on the consequences of influences to ensure the fault tolerance of IT. Reservation in specialized IT, which affects the provision of its fault tolerance, is part of measures for the dynamic restructuring of the system and can be assessed based on the time of use of server components, the time of their use. Experimental studies on checking the effectiveness of the developed method of ensuring it fault tolerance are carried out in two stages. First, we examine the information systems without the method implemented in it. Information systems of this at the second stage are investigating information systems with the developed method implemented in it. When setting such an experiment, the sources of influence are an essential aspect. There may be options when the information systems will work for a long time, so that in a long time with a certain probable information system it is possible to obtain influences that will lead to the activation of means of ensuring fault tolerance or if they are not implemented in the information systems, then fixing such influences. But then the impact on the information systems for experiments for two such cases will not be the same, because it will be real and random. To carry out requires long-term information systems of the experiment for a very long time, for example, a year. This is due to the fact that the analysis of reports of computer attacks within Ukraine gives statistics of massive attacks for about three to two years over the past 6-8 years. For two experiments, it is possible to solve the issue of conducting them sequentially, then two years, or in parallel to operate two identical information systems, in one of which there are no imposing means of ensuring fault tolerance, and in the other there are no. In addition, long-term information systems of experiments can be reduced by creating cyber polygons in a closed environment and establishing artificial sources of influence in it.

Information systems record the results of experiments in its internal format, which, if necessary, the information systems of the experiments conducted can be investigated. Figure 2 (fragment of the event retention file) shows fragments from the results of the work of two information systems. In one information systems, fault tolerance tools were not implemented and, therefore, the result was the statistics of the failure of information systems components in the process of functioning under the influence of malware and computer attacks. In the second information systems, in which the means of ensuring fault tolerance were implemented, the result was the time of influence, the time of involvement of fault tolerance and long-term information systems and effective information systems. In both cases, impact studies were recorded specifically on the need to ensure events caused by internal non-regulatory work.



Logfile001.txt [vk.com]

NPP	ARM	PVR	KVR	Error	NAMERROR	IP_BD	IP_ARM
200732	108	02.01.2021 16:10:07	02.01.2021 19:41:47			192.168.168.1	192.168.168.15
200733	40	04.01.2021 8:33:11	04.01.2021 9:02:26			192.168.168.1	192.168.168.10
200734	105	04.01.2021 8:35:43	04.01.2021 11:21:23			192.168.168.1	192.168.168.10
200735	51	04.01.2021 8:37:18	04.01.2021 11:21:26			192.168.168.1	192.168.168.10
200736	208	04.01.2021 8:42:08	04.01.2021 16:57:05			192.168.168.1	192.168.168.9
200737	83	04.01.2021 8:44:09	04.01.2021 8:44:46			192.168.168.1	192.168.168.9
200738	89	04.01.2021 8:48:36	04.01.2021 11:21:21			192.168.168.1	192.168.168.10
200739	69	04.01.2021 8:49:59	04.01.2021 13:45:48			192.168.168.1	192.168.168.9
200740	8	04.01.2021 8:55:38	04.01.2021 13:34:37			192.168.168.1	192.168.168.6

file017.log [vk.com]

210301	76	16.09.2021 8:43:26	16.09.2021 8:45:07			192.168.168.1	192.168.168.14
210302	3	16.09.2021 8:45:15	16.09.2021 13:01:39			192.168.168.1	192.168.168.5
210303	3	16.09.2021 8:48:14				192.168.168.1	192.168.168.5
210304	3	16.09.2021 8:52:47		3060	The server is not responding	192.168.168.1	192.168.168.6
210305	50	16.09.2021 8:52:54		3050	server IP 192.168.168.1 is not available	192.168.168.1	192.168.168.10
210306	50	16.09.2021 8:53:56		3050	server IP 192.168.168.1 is not available	192.168.168.1	192.168.168.10
210307	50	16.09.2021 8:54:12		3050	server IP 192.168.168.1 is not available	192.168.168.1	192.168.168.10
210308	50	16.09.2021 8:54:38			blocking connection to SQL server reserve IP 192.168.168.2	192.168.168.2	192.168.168.10
210309	3	16.09.2021 8:55:13		3050	The server is not responding	192.168.168.1	192.168.168.5
210310	50	16.09.2021 8:55:17			change address bar for clients to IP 192.168.168.2	192.168.168.2	192.168.168.10
210311	50	16.09.2021 8:55:25			Unblocking the connection to the SQL server of the reserve IP	192.168.168.2	192.168.168.10
210312	76	16.09.2021 8:55:57	16.09.2021 10:15:52		redirect to backup server	192.168.168.2	192.168.168.14
210313	3	16.09.2021 8:58:06	16.09.2021 10:08:27		redirect to backup server	192.168.168.2	192.168.168.5
210314	76	16.09.2021 10:12:22	16.09.2021 10:25:31			192.168.168.2	192.168.168.14
210315	3	16.09.2021 10:21:32				192.168.168.2	192.168.168.5
210316	76	16.09.2021	16.09.2021			192.168.168.2	192.168.168.14

Figure 2: Log-file of events in the information systems

The results of experimental studies confirmed the effective information systems of the developed method of ensuring the fault tolerance of IT malware and computer attacks. Calculations of estimated values for frills and redundancy according to the data from the experiment on the developed information systems indicate about 87 percent more compared to the information systems in which the developed method is not implemented.

6. Conclusions

The proposed abstract model allows us to consider objects of the computer system that can be affected by malware and computer attacks. And, therefore, it acts as the basis of the developed new method of ensuring the fault tolerance of specialized IT in the face of malware and computer attacks. As a result, the use of the developed method is carried out in a system that has mechanisms for restructuring and uses excesses. For the study of the developed method, a methodology for assessing its effectiveness in terms of excesses and redundancy has been developed.

The conducted experimental studies and evaluation calculations confirm the effective information systems of the developed method of ensuring the fault tolerance of IT in the conditions of exposure to malware and computer attacks.

7. References

[1] T. Herault, Y. Robert Fault-Tolerance Techniques for High-Performance Computing, Computer Communications and Networks, Springer, Cham, 2015, 320 p. doi: 10.1007/978-3-319-20943-2.

[2] S. Patranabis, D. Mukhopadhyay Fault Tolerant Architectures for Cryptography and Hardware Security Series: Computer Architecture and Design Methodologies, Springer Singapore, 1st ed. 2018, 240 p. doi: 10.1007/978-981-10-1387-4.

[3] M. Blanke, M. Kinnaert, J. Lunze, M. Staroswiecki, Diagnosis and Fault-Tolerant Control, Springer-Verlag, Berlin, Heidelberg, 3rd ed. 2016, 695 p. doi: 10.1007/978-3-662-47943-8.

[4] S. X. Ding, Data-driven Design of Fault Diagnosis and Fault-tolerant Control Systems Series: Advances in Industrial Control, Springer-Verlag, London, 2014, 300 p. doi: 10.1007/978-1-4471-6410-4.

[5] O. Pomorova, O. Savenko, S. Lysenko, A. Nicheporuk Metamorphic Viruses Detection Technique based on the the Modified Emulators, CEUR Workshop Proceedings 1614 (2016) 375-383.

[6] 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.

[7] 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.

[8] A. Zolghadri, D. Henry, J. Cieslak, D. Efimov, P. Goupil, Fault Diagnosis and Fault-Tolerant Control and Guidance for Aerospace Vehicles From Theory to Application, Advances in Industrial Control, Springer-Verlag, London, 2014, 216 p. doi: 10.1007/978-1-4471-5313-9.

[9] R. S. Chakraborty, J. Mathew, A. V. Vasilakos Security and Fault Tolerance in Internet of Things, Internet of Things, Springer, Cham, 2019, 214 p. doi:10.1007/978-3-030-02807-7.

[10] S. X. Ding. Advanced methods for fault diagnosis and fault-tolerant control, Springer, Berlin, Heidelberg, 2021, 658 p. doi: 10.1007/978-3-662-62004-5.

[11] P. Sokol, M. Zuzcak, T. Sochor, Definition of attack in the context of low-level interaction server honeypots, Computer Science and its Applications. Lecture Notes in Electrical Engineering, Springer, Berlin, Heidelberg, 2015. p. 499-504. doi: 10.1007/978-3-662-45402-2_74.

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

[13] R.Martínez-Guerra, F. Meléndez-Vázquez, I. Trejo-Zúñiga, Fault-tolerant Control and Diagnosis for Integer and Fractional-order Systems Fundamentals of Fractional Calculus and Differential Algebra with Real-Time Applications, Studies in Systems, Decision and Control, Springer, Cham, 2021, 192 p. doi:10.1007/978-3-030-62094-3.

[14] M. Witczak, Fault Diagnosis and Fault-Tolerant Control Strategies for Non-Linear Systems Analytical and Soft Computing Approaches, Lecture Notes in Electrical Engineering, Springer, Cham, 2014, 229 p. doi:10.1007/978-3-319-03014-2.

[15] D. Du, S. Xu, V. Cocquempot, Observer-Based Fault Diagnosis and Fault-Tolerant Control for Switched Systems, Studies in Systems, Decision and Control., Springer, Singapore, 2021, 280 p. doi: 10.1007/978-981-15-9073-3.

[16] Y. Xiang, L. Hui, Y. Xianfei, Y. Chen, S. Haifeng An adaptive method based on contextual anomaly detection in Internet of Things through wireless sensor networks, International Journal of Distributed Sensor Networks 16(5) (2020). doi: 10.1177/1550147720920478.

[17] A. Rawat, R. Sushil, A. Agarwal, A. Sikander, R. S. Bhadoria. A New Adaptive Fault Tolerant Framework in the Cloud, IETE Journal of Research (2021) 1-13. doi:10.1080/03772063.2021.1907231.

[18] A. Haqiq, B. Bounabat, Data Sciences Towards Integration of Fault Tolerance in Agent-based Systems, Procedia Computer Science. 127 (2018) 264-273. doi:10.1016/j.procs.2018.01.122.

[19] S. Lysenko, K. Bobrovnikova, S. Matiukh, I. Hurman, O. Savenko, Detection of the botnets' low-rate DDoS attacks based on self-similarity, International Journal of Electrical and Computer Engineering, 10(4), (2020) 3651-3659. doi:10.11591/ijece.v10i4.pp3651-3659

[20] O. Pomorova, O. Savenko, S. Lysenko & A. Kryshchuk, Multi-agent based approach for botnet detection in a corporate area network using fuzzy logic. In International Conference on Computer Networks. Springer, Berlin, Heidelberg (2013) 146-156.

[21] M. T. Hamayun, C. Edwards, H. Alwi, Fault Tolerant Control Schemes Using Integral Sliding Modes, Studies in Systems, Decision and Control, Springer, 2016, 199 p. doi:10.1007/978-3-319-32238-4.

[22] T. Jain, J. J. Yamé, D. Sauter, Active Fault-Tolerant Control Systems A Behavioral System Theoretic Perspective, Studies in Systems, Decision and Control, Springer, 2018, p. 152 p. doi. 10.1007/978-3-319-68829-9.

[23] T. Sochor, P. Smolka1, Z. Priscakova, P. Jedlicka & D. Dlabolova, Survey on the usage of public cloud services with copyrighted contents. In AIP Conference Proceedings Proceedings, 2019. doi: 10.1063/1.5137967.

[24] N.K. Dewangan, T. Prakash, J. K. Tandekar, K. K. Gupta, Open-circuit fault-tolerance in multilevel inverters with reduced component count, Electr Eng, 102**,** (2020) 409–419 doi:10.1007/s00202-019-00884-9.

[25] M. Yang, G. Hua, Y. Feng, J. Gong. Fault-Tolerance Techniques for Spacecraft Control Computers, Wiley, 344 p.

[26] J. Andersson, V. Grassi, R. Mirandola, D. Perez-Palacin, A Distilled Characterization of Resilience and Its Embraced Properties Based on State-Spaces, Lecture Notes in Computer Science, Vol. 11732, Springer, Cham. doi:10.1007/978-3-030-30856-8_2.

[27] B. Savenko, S. Lysenko, K. Bobrovnikova, O. Savenko, G. Markowsky, Detection DNS Tunneling Botnets. In 2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS) 2021, pp. 64-69.

[28] J. Parri, F. Patara, S. Sampietro, E. Vicario, JARVIS, A Hardware/Software Framework for Resilient Industry 4.0 Systems, Lecture Notes in Computer Science, Vol. 11732. Springer, Cham. doi:10.1007/978-3-030-30856-8_6

[29] M.A. Olivero, A. Bertolino, F.J. Dominguez-Mayo, M.J. Escalona, I. Matteucci, Addressing Security Properties in Systems of Systems: Challenges and Ideas, Lecture Notes in Computer Science, Vol. 11732. Springer, Cham. doi:10.1007/978-3-030-30856-8_10.

[30] M. Zuzcak, T. Sochor, M. Zenka, Intrusion Detection System for Home Windows based Computers, KSII Transactions on Internet and Information Systems, Korean Society for Internet Information (KSII), Vol. 13, No. 9., pp. 4706-4726. doi: 10.3837/tiis.2019.09.021.