Implementation of Control by Parameters of Client Automated Workplaces of Specialized Information Systems for Neutralization malware

Mykola Stetsyuk[a], Vasyl Stetsyuk[a], Bohdan Savenko[a], Oleg Savenko[a], Maciej Dobrowolski[b]

[a] *Khmelnytskyi National University, Institutska str., 11, Khmelnytskyi, 29016, Ukraine*
[b] *Kazimierz Pułaski Technology and Humanitarian University, Malczewskiego St 29, Radom, 26-600, Poland*

Abstract

The paper presents a topical scientific problem for the development of information technology, which automatically allows you to neutralize the manifestations of malicious software on specialized information systems.

The risks of malicious software attacks depending on the executable file formats are analyzed. The analysis of methods to ensure fault tolerance and survivability of specialized IP showed that the current methods and technologies do not fully ensure their fault tolerance and survivability in terms of counteracting the impact of malware. Despite the invariance of the methods used, the counteraction procedure is reduced to the organization of a single-level scheme at the system-wide level.

This is enough to ensure the functionality of an ordinary computer system that provides general information needs, but not enough to ensure access to the functionality of specialized IP at any time.

A method of parametric control of client automated workstations (AWP) of specialized information systems to neutralize the effects of malicious software has been developed. The proposed technology to ensure fault tolerance and survivability of automated workstations of specialized IT and developed a method of parametric control of the relevance of client workstation software provide a high level of IP stability in general against the effects of malware. In fact, it realizes the second line of counteraction to malicious software, in comparison with the system-wide one, where it is not always possible to neutralize the destruction by malicious software. At the same time, being combined with the software support service, it does not require additional costs to support its operation. Experimental studies were conducted with the developed information system, which confirmed the improvement of its efficiency, reliability and proposed solutions.

Keywords

malfunctioning software, information system, information technology, performance, software comparator, vitality

1. Introduction

Today, it is difficult to identify areas where the total use of information technology has not found its recognition. Information technology has penetrated into almost all spheres of modern society, including such specialized as financial activities, medical, military.

[1] IntelITSIS'2021: 2nd International Workshop on Intelligent Information Technologies and Systems of Information Security, March 24–26, 2021, Khmelnytskyi, Ukraine
EMAIL: mikstt777@gmail.com (M. Stetsyuk); swmuau@gmail.com (V. Stetsyuk); Savenko_bohdan@ukr.net (B. Savenko); savenko_oleg_st@ukr.net (O. Savenko); m.dobrowolski@uthrad.pl (M. Dobrowolski)
ORCID: 0000-0003-3875-0416 (M. Stetsyuk); 0000-0001-9880-2666 (V. Stetsyuk); 0000-0001-5647-9979 (B. Savenko); 0000-0002-4104-745X (O. Savenko); ORCID 0000-0003-0296-9651 (M. Dobrowolski)

But along with the positive aspects of their use, we have to accept the idea that new information technologies are very sensitive to various kinds of destruction, one of which is the various ways of malicious software, which in the absence of properly organized counteraction paralyzes the information system which entails a lot of negative consequences [1-7].

Therefore, the task of organizing the work of specialized information systems in the face of malicious software, which in turn is part of a more global task of ensuring fault tolerance and survivability of the information system.

This task is considered to have a continuous solution and, at the same time, being complex, includes a number of sub-tasks, such as legal, organizational and, of course, software and hardware, which are responsible for developing mechanisms to counter the effects of malware. It is the latter that have become the subject of this article.

2. Analysis of known solutions

The analysis of methods to ensure fault tolerance and survivability of specialized information systems showed that the current methods and technologies do not fully ensure their fault tolerance and survivability in terms of counteracting the impact of malicious software. Despite the invariance of the methods used, the counteraction procedure is reduced to the organization of a single-level scheme at the system-wide level.

This is sufficient to ensure the operability of an ordinary computer system that provides general information needs, but not enough to guarantee access to the functionality of a specialized information system at any time.

In [1] presents approaches to responding to accidents in computer systems under the influence of malicious software. This is important, including the operation of information systems in computer networks. [2] explains the features of hardware security. In [3] such type of malicious software as botnets is analyzed. Using them causes significant harm to users of computers connected to the Internet. In [4] the security features of IP networks are analyzed. [5] presents forecasts on trends in the development of threats from malicious software. In [6] the influence on the possibility of detecting this type of viruses as metamorphic was analyzed. Their masking complicates the processes of their effective detection. In [7] the possibilities of protection of the hardware and software of the user from external influences are presented. Considered various aspects of the problem area to ensure security in computer systems, indicate the existence of an unresolved problem to ensure the security of processes in them due to the impact of malicious software.

If the object of the attack is a specific information system, and the goal is to block its work, then one level of resistance, as the events of 2016 have shown, may not be enough. This is confirmed by successful malware attacks recorded on December 6, 2016 [8]. Their targets were the internal telecommunications networks of the Ministry of Finance, the State Treasury, the Pension Fund and, as a result, blocking access to critical databases, which led to delays in budget payments. On December 15, an attack was made on Ukrzaliznytsia's information system, as a result of which its work was completely blocked during the day.

Another aspect that was considered in the analysis of the construction of anti-malware systems is that the construction of such systems is by typification and standardization [9-11]. This is a natural way of developing the defense mechanisms of computer systems, which has many positive, no doubt, moments, but at the same time, its undeniable drawback is that the typing process itself facilitates the creation of mechanisms to overcome the means of protection. And here there is a collision, when on the one hand, we can not give up the benefits of typification and standardization in creating mechanisms to counter the effects of malicious software, and on the other hand, we can not accept the fact that such an approach effectively, in turn, simplifies the creation of mechanisms to overcome the protection of the information system. Thus, standardization in the development of IP makes it easier for attackers to develop malicious software focused on such IP.

An important area of ensuring the stability of IP under the influence of malware is the choice of an appropriate effective mathematical apparatus as a basis for the search for abnormal or malicious manifestations [12-15]. Malicious software controlled by an attacker, which is a botnet [16-17], is aimed precisely at taking control of reptiles by user computer systems and gaining access to information systems [18-25]. The authors of the article [24, 25] consider cloud programs, which are considered to be components of several components of cloud services that interact with each other, where each component performs certain functionalities. A comprehensive recovery scheme based on software rejuvenation for cloud applications is proposed, which consists of three important parts: adaptive fault detection, aging assessment, and component-based rejuvenation checkpoint. In the article [26] the use of the clonal selection algorithm as a mathematical apparatus is considered. Therefore, the choice of mathematical software as the basis of methods for detecting abnormal or malicious manifestations, when creating IP that must meet the requirements of fault tolerance and survivability in the face of malware, is an important task.

We will also consider other strategic approaches to solving the problem of ensuring the stability and survivability of IP in the face of malware. In [27] the approach for avoiding functional failures during execution in component application systems is presented. The approach uses the internal redundancy of components to find workarounds as alternative sequences of operations to avoid failures.

In articles [28, 29] methods of ensuring reliability and functional security of software packages in real time are offered. In [30], tolerance to failure is a major problem in ensuring the availability and reliability of critical services, as well as program implementation. In order to minimize the impact of failures on the system and the implementation of applications, it is necessary to anticipate deviations and take measures for them. Failure tolerance methods are used to predict these failures and take appropriate action before the failures actually occur. In [31] the use of application software interface calls in malware detection problems is shown. This is required for inclusion in detection systems or as part of certain IPs.

In [32, 33], cyber resilience and viability are presented as closely related concepts with similar technologies and practices. For historical reasons, these concepts have been embedded in different frameworks that define different constructs to describe problems and areas of solution.

In [34-40] shows the impact on the resilience and survival of IT of various types of malware and computer attacks. Paper [39] presents and discusses a method for classifying Android applications to detect malware. Based on the use of an artificial immune system and artificial neural networks, an antivirus system has been proposed, especially for the Android system, which can detect and block unwanted and malicious programs. This system can be characterized by self-adaptation and self-evolution and can detect even unknown and previously unseen malware. That is, the proposed approaches allow the system to respond dynamically to events.

Problems to ensure fault tolerance and survivability of specialized information technologies in the face of malicious software and computer attacks are issues of research, including the hardware infrastructure where they operate. Work [40] shows a study of such a class of devices as a router. This document examines the spread of DDoS attacks on the router subsystems of the Smart Office system. This paper analyzes and solves the problem of optimizing the search for the minimum path of attack on the router subsystems. The result of this work is to determine the most vulnerable subsystems of the router to the consequences of DDoS-attacks.

Paper [41-44] considers the problems of hidden faults that are inherited in security systems aimed at ensuring the functional safety of high-risk facilities to combat accidents, which is also important and should be taken into account when ensuring resilience and survivability of specialized information technologies.

The problem of hidden faults is considered in terms of resource-oriented approach as a problem of growth from the lowest level of replication to the next level of diversification in the development of models, methods and tools [45-46].

To consider malicious manifestations and methods of counteraction to them allow to use these results at creation of information technologies with the increased level of maintenance of fault tolerance and survivability in the conditions of influences of malicious software.

3. Formulation of the problem

The practice of using information technology has shown that viable methods to ensure fault tolerance and survivability of the information system are those that are characterized not only by potentially high efficiency parameters, but at the same time, remain simple and cheap to use. This fully applies to measures to neutralize the effects of malicious software.

The events that took place in Ukraine in the period from 2013 to 2018 showed the vulnerability of modern information technologies, which in turn makes it urgent to develop methods to improve the efficiency of fault tolerance and survivability of the information system and, even more, specialized information systems in which critical data is usually processed.

It is proposed to supplement the existing methods of ensuring the resilience and survivability of the information system in terms of neutralizing the impact of malicious software technology, which is based on the idea of ease of implementation and ensuring high efficiency of specialized information technology in the effects of malicious software. In this case, despite the simplicity of implementation, a feature of the new technology of information system protection is its operation in automatic mode.

An important point of its operation is the inclusion in its tasks to document the identified manifestations of malware, which allows for constant analysis of information about events in the information system in order to improve methods of counteraction.

4. Main part

One of the ways to solve this problem is to use two levels of counteraction to the effects of malicious software, the first of which, system-wide, is built using conventional counteraction mechanisms, and the second, local, is implemented within the most specialized information system, using its nuances of operation and architecture (Figure 1).

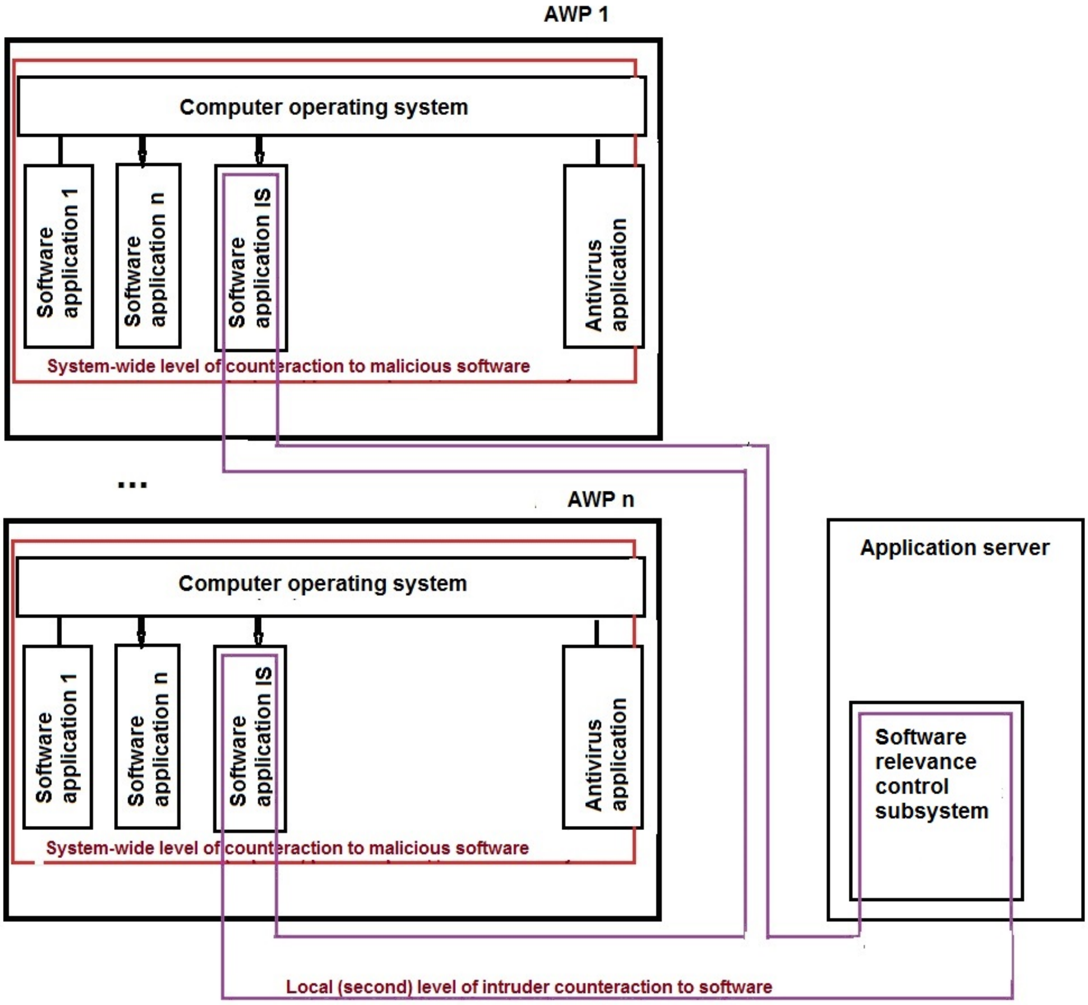

AWP 1
Computer operating system
Software application 1
Software application n
Software application IS
Antivirus application
System-wide level of counteraction to malicious software
...
AWP n
Computer operating system
Software application 1
Software application n
Software application IS
Antivirus application
System-wide level of counteraction to malicious software
Application server
Software relevance control subsystem
Local (second) level of intruder counteraction to software

Figure 1: A two-tier anti-malware scheme for client automated workstations of a specialized information system.

This approach will increase the likelihood of neutralizing the target attack of malicious software on the objects of a specialized information system, make the work of malicious software as difficult as possible in order to increase the availability of the information system at any time. It is proposed to use as a mechanism to counter attacks, already existing in most developed specialized information systems, the service of maintaining the relevance of client software, giving it new qualities, described in the following method.

5. A method of ensuring the fault tolerance and survivability of the information system in the face of malicious software using parametric control of the relevance of software modules of client automated workstations and their masking

To solve the problem of restoring the functionality of the information system in order to prevent the effects of malicious software, increase the degree of warranty of the information system, a method of ensuring fault tolerance and survivability is proposed, the model of which is shown in Figure 2.

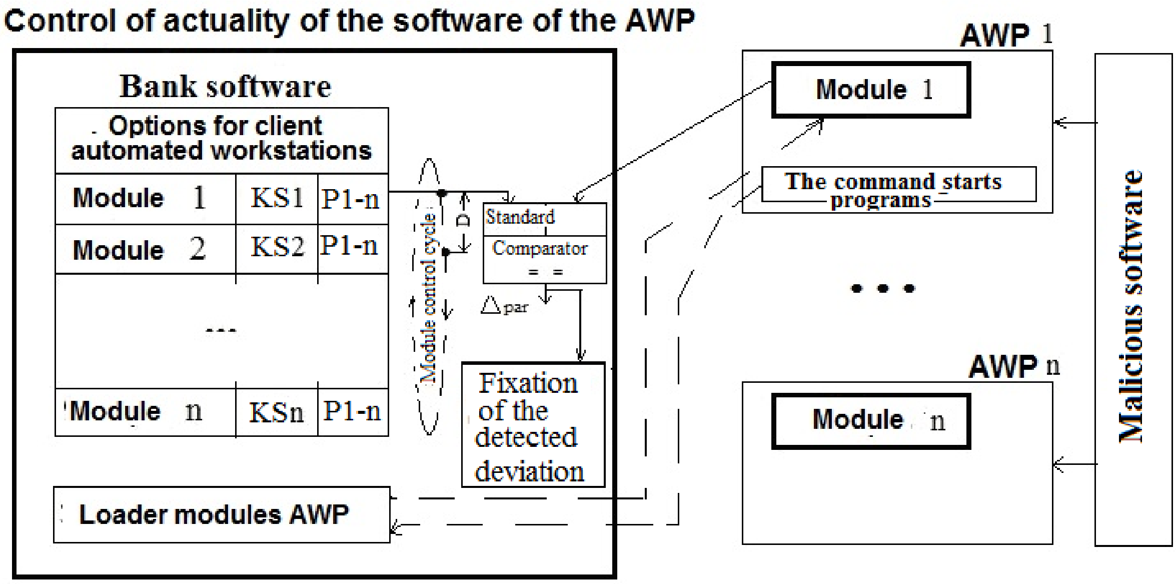


Figure 2: Model of the method of ensuring fault tolerance and survivability of the information system under the influence of malicious software using parametric control of the relevance of software modules of client automated workstations and their masking

The essence of the method is to carry out constant cyclic control of the parameters of the modules of client automated workstations with a given discreteness D.

Discreteness is a parameter whose value is chosen based on the level of system-wide performance of the computer network and client computers on which automated workstations are based. This allows you to adapt this technology to the hardware platforms of the information system of different performance.

To ensure the robustness of the method, its model includes a software bank that contains software modules of all client automated workstations of the information system and their reference parameters, such as checksums KS1 - KSn code pages, calculated according to a given rule. The set of control parameters can be changed according to the structure of the controlled software modules. In the process of monitoring the relevance of the state of the software module of the client automated workplace, its availability is checked in a given way, its parameters are calculated and compared with the reference.

The task of parametric control of relevance of modules within the limits of this method is assigned to the software implemented comparator.

In the absence of a controlled module in a given place, or a difference between the actual and reference parameters Δpar at the output of the comparator, the software of the client automated workstation is restored using the reference software stored in the bank. The very fact of detecting discrepancies Δpar is automatically documented while maintaining the necessary parameters for further analysis in the database.

If no discrepancies are found between the standard and the module that has passed the relevance check, it can, in addition, be masked by renaming. This will reduce the likelihood of the module being attacked by malicious software, which is known to primarily affect executable files. This method allows you to control the relevance of the modules of client automated workstations in automatic mode, which in turn allows you to ensure fault tolerance and survivability of the information system under the influence of malicious software.

In this case, the nature of the attack of malicious software on client software does not play a special role. Because no information system data is stored on client computers, malicious software can only damage program files. This fact is manifested in the process of monitoring the relevance of software modules for automated workstations and they are replaced by reference.

6. Technology to ensure fault tolerance and survivability of the information system under the influence of malicious software using the method of parametric control of software relevance of client automated workstations

It is known that all information systems are characterized by a long life cycle, during which their software, under the influence of many external and internal factors is subject to change. And these changes are all the more significant the larger the subject area covered by the information system.

The task of maintaining the relevance of the software of client workstations of a specialized information system is quite extensive, so, as a rule, the natural course of development of the information system leads to the transition to an automated subsystem to restore the relevance of software. In the future, we will call it the software support service for client automated workstations. Its task is to detect software updates for client automated workstations of the information system and perform its replication to all workstations with settings for a specific automated workstation.

The main component of the software relevance support service is the reference software bank. In the process of improving the information system, changes are made to the software, which in turn leads to the replacement of the standard software in the bank. The task of the support service is to identify the fact of changing the standard of the software and, in response to this, to start the procedure of updating the software of all client workstations where it is used.

If we compare the above method of ensuring the fault tolerance and survivability of the information system under the influence of malicious software with the work of the information system software support service, we will see that their implementation will be based on similar algorithms. The only difference is that the algorithms of the software relevance service respond to the change of the software standard, and the algorithms of the method of ensuring the fault tolerance and survivability of the information system to the loss of compliance with the software instance standard n-th client workplace. The reaction in both cases will be the same - the restoration of the software of the client workplace of its standard in the bank.

Therefore, the basis of the proposed technology to ensure fault tolerance and survivability of a specialized information system under the influence of malicious software is to put the idea of using the functionality of the existing service to maintain the relevance of software for client automated workstations. To do this, its algorithms have been improved by including in its composition functions that ensure fault tolerance and survivability of the client part of the information system in the conditions of malicious software.

The method of counteracting attacks of malicious software, the model of which is shown in Figure 2, implemented in the form of information technology, which includes several interacting at the software level processes. This technology is in addition to the already known ways to protect the information system by counteracting malicious software attacks. It includes a background process, the algorithm of which is shown in a simplified form in Fig. 3, during which, in fact, the relevance of software modules of client automated workstations and a special procedure are checked launching software modules for automated workstations for execution (Figure 4).

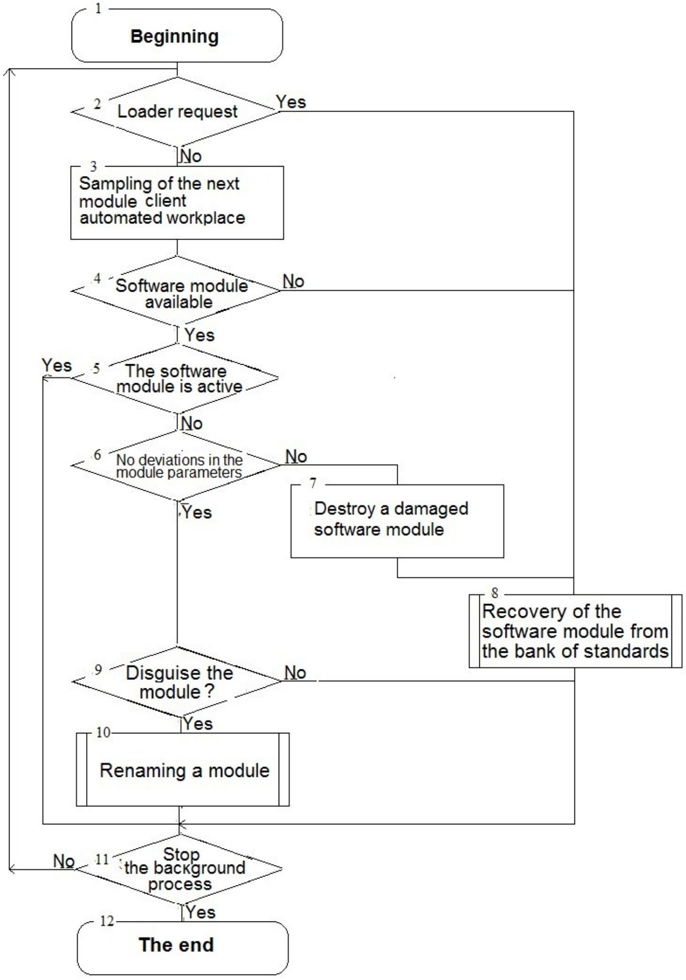

1
Beginning
2
Loader request
Yes
No
3
Sampling of the next module client automated workplace
4
Software module available
No
Yes
Yes
5
The software module is active
No
6
No deviations in the module parameters
No
Yes
7
Destroy a damaged software module
8
Recovery of the software module from the bank of standards
9
Disguise the module ?
No
Yes
10
Renaming a module
No
11
Stop the background process
Yes
12
The end

Figure 3: Algorithm of the background process of the software relevance support service in terms of ensuring fault tolerance and survivability of the client part of the specialized information system under the influence of malicious software.

1. The background process checks the availability of a service request from the loader of the client automated workstation modules. This is necessary in order that the reaction to the fact of failure to start some software module of the automated workplace was the fastest (Figure 3, operator 2).

The background process of the software relevance service, running at a specified frequency, monitors the software modules of each registered in the information system client automated workstation. In each iteration, the process performs a given sequence of operations, which implements the algorithm for monitoring the relevance of the software of client automated workstations:

This situation in the information system can occur when the operator of an automated workstation tries to run its client program for execution, and it is for some reason unavailable. The program downloader, at the time of attempting to run it, detects this fact and submits an application to the background process for the primary recovery of the software specified in the application automated workstation.

Upon receipt of the application, the background process reads from it the number of the software module, which requires priority maintenance and goes to step 6.

2. If there is no request for emergency maintenance, the background process proceeds to check the next module in the queue (Figure 3, operator 3).

3. In the next step, it is checked whether the module of the client automated workstation under analysis is located in a certain place in the file directory of the client computer. In case of its absence for any reason the transition to point 6 (Figure 3, operator 4) is carried out.

4. If the module is available and in a given place, then check its activity (Figure 3, operator 5). At this stage, it is determined whether it is loaded into the memory of the PC and performs the function assigned to it within the information system. If at the time of testing the module is active, then go to step 7.

5. Control of parameters of the next software module of the n-th automated workplace on conformity to the reference stored in bank of service of actuality of the software (figure 3, operator 6). If no deviations from the reference parameters are detected, the file is masked by renaming (Figure 3, operators 9,10). This allows you to remove it from a possible attack by malware, knowing that it attacks executable files, focusing on their extension. Then go to step 7, otherwise to the next.

6. Replace the damaged or restore the missing software module with a reference from the software bank (Figure 3, operator 8).

7. The supply of the command to stop the background process is checked (Figure 3, operator 11). If not, the current iteration is completed, followed by step 1.

8. Completion of the background process.

Since this technology is intended for specialized information systems, which in themselves can be the object of targeted attack, the algorithm (Figure 3) may include another function, the task of which is to form the location of the executable modules of client automated working places special files-traps that should serve as false objects of attack for malware, while the real modules are disguised.

Trap files are no different from software modules in automated information system workstations except that they can never be downloaded for execution. Before starting the real module, they are destroyed and then re-created after the module completes its work.

Along with the function of masking the software modules of automated workstations, the function of creating false objects of attack allows you to direct the destructive actions of malicious software in a direction that does not threaten the functioning of the information system.

The process of launching software modules of client automated workstations for execution also has its own feature - it is performed in two stages. First, from the client PC, a short bootloader program is launched, which is permanently stored in a secure directory of the software service of the specialized information system. The bootloader starts, finds the file of the corresponding module in a certain place, restores its name and transfers control to it.

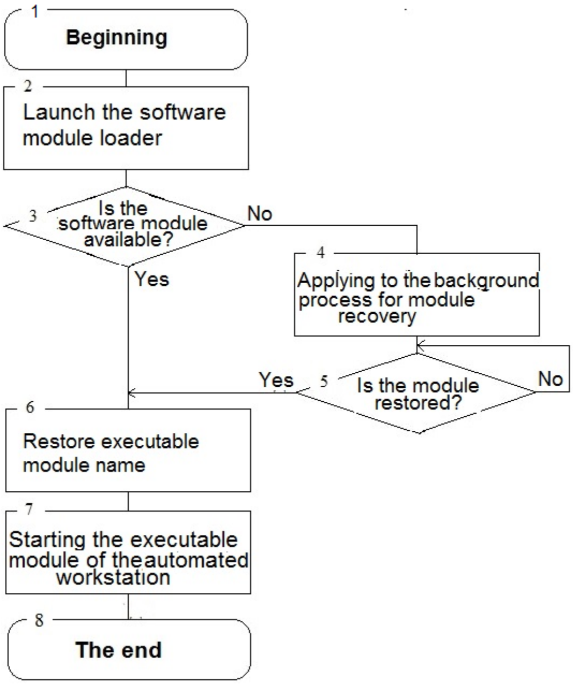

1
Beginning
2
Launch the software module loader
3
Is the software module available?
No
Yes
4
Applying to the background process for module recovery
Yes
5
Is the module restored?
No
6
Restore executable module name
7
Starting the executable module of the automated workstation
8
The end

Figure 4: Algorithm of work of the loader of modules of the automated workplaces of service of support of actuality of the client software.

If the file for some reason is not found, the module loader will issue to the background process a request for an extraordinary software update of the specified automated workstation and will go into standby mode (Figure 4. operators 4,5). After the background process executes the loader's request, it will in turn repeat the procedure of starting the corresponding module (Figure 4 operators 6,7).

The ability of this technology to restore damaged and destroyed software of client automated workstations allows you to eliminate the effects of malicious software attacks that penetrate into the module of automated workstations information system, and those whose destructive actions are manifested in data encryption.

7. Influence of executable file format on the frequency of malware attacks

In order to find ways to improve the efficiency of the proposed method of ensuring the resilience and survivability of specialized information systems, an analysis of the frequency of malware attacks depending on the type of executable files within the operating system MS Windows. This operating system allows you to work with a fairly wide range of executable files. These are primarily COM and EXE program files. Next are the system drivers. They have the extension SYS or BIN. Executable files also include batch files. These are called BAT files.

Executable files also include overlay files and dynamically loaded libraries that are used by programs as needed.

The analysis (Table 1) showed that malicious software most often uses files in COM and EXE formats as objects of its attack. They are followed by CMD and BAT files and SYS and BIN driver files.

But the use of other file types (INF, INS, MSC, MSI, PIF, REG, VBS, MDB, MDE) for destructive purposes by malicious software is rare.

Table 1

Dependence of the frequency of malware attacks on the format of the executable file

File type	Risk of being attacked by malware
EXE	Highest
COM	High
CMD & BAT	Average
SYS & BIN	Low
MDB, MDE	The lowest
Other types	Not analyzed

Of all the above files, we will be interested only in the file with the MDE format of the MS Access package. Executed with functionality that allows you to implement a software system of any complexity, it has the lowest risk of being attacked by malicious software. This means that it is ignored by the developers of the malicious code and no cases of infection have been found.

To date, only attempts to destroy the contents of the MDE file by the destructive actions of the Blackmal virus by entering the line "DATA Error" have been detected. But the destruction of the contents of the file is not an infection of the file and, accordingly, such destruction by malicious software does not threaten serious consequences for the data, but only requires the replacement of the distorted file with a new one.

Therefore, this fact (the presence of its own format of the executable MDE-file, little prone to infection with malware), among others, significantly influenced the recommendation to choose MS Access as a tool for software development of client automated workstations of specialized information systems.

The MDE file is a special format of the MS Access database, and in turn is derived from the MDB-type database of MS Access. Its feature is that part of the database components, which may include executable modules - forms, reports, modules, macros - is stored in the middle of the MDE-file in compiled form, which does not allow any changes to their source text, as well as their review, but it remains possible to make changes to the table and queries. It is positioned as a DBMS file with advanced data manipulation capabilities. The database data can be in the same file, or in another MDE file, or MDB-file. It is also possible to work with data contained in any non-MS Access database that supports ODBC data access technology. An MDE file is an executable file in MS Windows and MAC. It can be started by MS Microsoft Access or RUN Time Access.

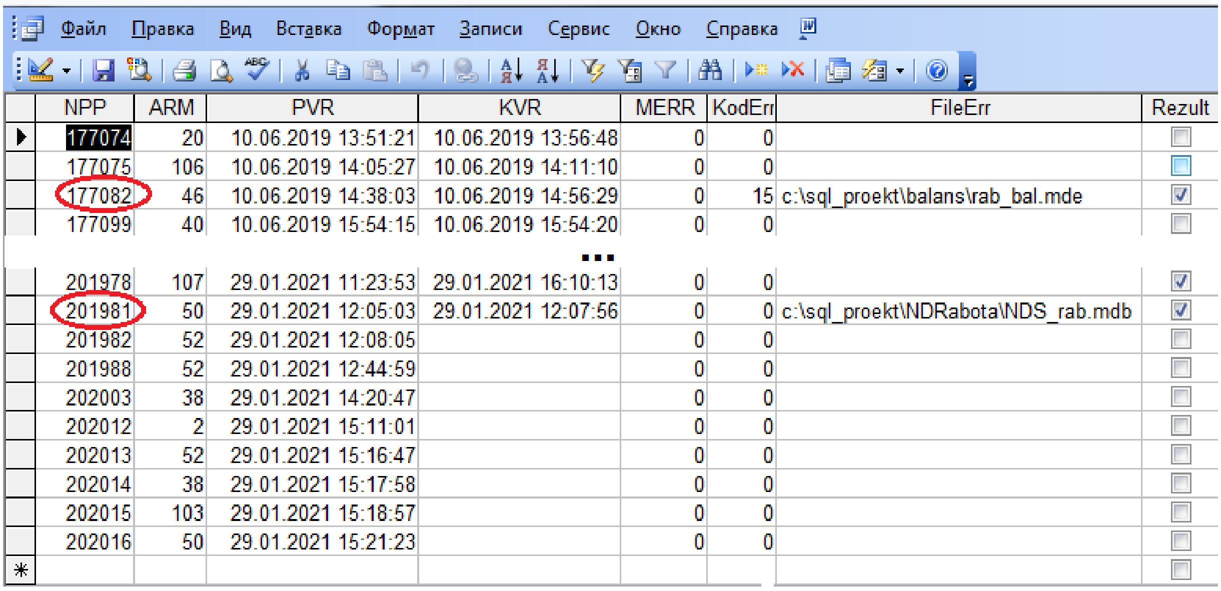
Such properties of the MDE-file can significantly increase the security of the information system as a whole, because its users do not have access to the source code of the software modules of the user's automated workstation components, and therefore their potentially destructive actions against databases.

8. Experimental studies

The subsystem of control of integrity of the client software is realized in the specialized information system "Management of financial resources of KhNU" in its automated workplace "ADMINISTRATOR". The software of this information system was developed in MS Access, which was caused by a number of points, one of which was the desire to reduce the likelihood of its modules to be attacked by malicious software. Unfortunately, this approach only works if this development tool is not widely used.

An experiment was performed with this information system, in which the situation of damage to one of the files of the automated workplace №46 by malicious software was simulated - changes were made to one of its code pages using a HEX editor. As a result, its structure began to differ from the reference.

As can be seen from the fragment of the log file shown in Figure 5, when trying to run at 14:38 10.6.19 the program of the automated workstation "BALANCE" for execution, the software comparator of comparison of files of the subsystem of control of actuality of the client software, deviations from reference parameters were revealed.



	NPP	ARM	PVR	KVR	MERR	KodErr	FileErr	Rezult
▶	177074	20	10.06.2019 13:51:21	10.06.2019 13:56:48	0	0		□
	177075	106	10.06.2019 14:05:27	10.06.2019 14:11:10	0	0		□
	177082	46	10.06.2019 14:38:03	10.06.2019 14:56:29	0	15	c:\sql_proekt\balans\rab_bal.mde	☑
	177099	40	10.06.2019 15:54:15	10.06.2019 15:54:20	0	0		□
				...				
	201978	107	29.01.2021 11:23:53	29.01.2021 16:10:13	0	0		☑
	201981	50	29.01.2021 12:05:03	29.01.2021 12:07:56	0	0	c:\sql_proekt\NDRabota\NDS_rab.mdb	☑
	201982	52	29.01.2021 12:08:05		0	0		□
	201988	52	29.01.2021 12:44:59		0	0		□
	202003	38	29.01.2021 14:20:47		0	0		□
	202012	2	29.01.2021 15:11:01		0	0		□
	202013	52	29.01.2021 15:16:47		0	0		□
	202014	38	29.01.2021 15:17:58		0	0		□
	202015	103	29.01.2021 15:18:57		0	0		□
	202016	50	29.01.2021 15:21:23		0	0		□
*								□

Figure 5: Fragment of the Log-file of documentation of events in the subsystem of control of actuality of the client software of the information system of management of financial resources of KhNU

The situation was recorded in the log-file of this subsystem in the form of a record number 177082 with error code "15".

This code indicates the mismatch of the checksum of the code module of the file rab_bal.mde workplace №46 to the parameters of the file stored in the database of standards.

As a result of further operation of the client software relevance control subsystem, the file with the damaged part of the code was deleted and replaced with a new one from the database of standards.

Another operation of the comparator, recorded in line 201981 of the listing, documented the event of a discrepancy between the parameters of the standard and the file NDS_rab.mdb in the workplace №50. Error code "0" indicates that the cause of the operation was an update of the software version of this automated workstation.

9. Conclusions

The proposed technology to ensure fault tolerance and survivability of automated workplaces of specialized information systems based on the method of parametric control of software relevance of client automated workplaces provides a high level of stability of the information system as a whole against malware.

In fact, it implements the second line of counteraction to malicious software, compared to the system-wide, which is not always possible to neutralize the destruction of malicious software. At the same time, being combined with the software relevance support service, it does not require additional costs to support its operation.

The direction of further research is to find ways to increase the efficiency of the proposed technology in ensuring fault tolerance and survivability of specialized information systems.

10.References

[1] A. Steve. Applied Incident Response. John Wiley & Sons, Inc., 2020.

[2] S. Bhunia, M. Tehranipoor Hardware Security: A Hands-on Learning Approach. Morgan Kaufmann, 2019

[3] O. Savenko, S. Lysenko, A. Kryschuk Multi-agent based approach of botnet detection in computer systems, Communications in Computer and Information Science 291 (2012) 171-180

[4] B. Swarup, R. Sandip, S.-K. Susmita. Fundamentals of IP and SoC Security: Design, Verification, and Debug. Springer, 2017

[5] R. S. Grinyov, O. V. Severinov, Analysis of trends in viral threats in Ukraine, in: Proceedings of Modern directions of development of information and communication technologies and management tools, Kharkiv, 2019 [in Ukrainian]

[6] O. Savenko, S. Lysenko, A. Nicheporuk, B. Savenko, Metamorphic Viruses' Detection Technique Based on the Equivalent Functional Block Search, CEUR-WS, 1844 (2017) 555–569.

[7] Y. Y. Gromov, O. G. Ivanova, K. V. Starodubov, A. A. Kadykov, Software and hardware means of protection of information systems, TSTU, 2017 [in Russian]

[8] Electronic magazine "Nowoe Vremya". The largest cyber attacks against Ukraine since 2014. Infographics, URL: https://nv.ua/ukraine/events/krupnejshie-kiberataki-protiv-ukrainy-s-2014-goda-infografika-1438924.html [in Russian]

[9] O. S. Savelyeva, O. M. Krasnozhon, O. U. Lebedeva, Using the structural fault-tolerance index in project designing. Odes'kyi Politechnichnyi Universytet. Pratsi, 2 (2014) 130–135. doi: 10.15276/opu.2.44.2014.24.

[10] S. Boranbayev, S. Altayev, A. Boranbayev. Applying the method of diverse redundancy in cloud based systems for increasing reliability, in: Proceedings of the 12th International Conference on Information Technology: New Generations (ITNG 2015), Las Vegas, Nevada, 2015, pp. 796-799.

[11] A. Boranbayev, S. Boranbayev, K. Yersakhanov, A. Nurusheva, R. Taberkhan R, Methods of Ensuring the Reliability and Fault Tolerance of Information Systems, Advances in Intelligent Systems and Computing, 738 (2018)

[12] Y. Kondratenko, N. Kondratenko, Soft Computing Analytic Models for Increasing Efficiency of Fuzzy Information Processing in Decision Support Systems. Chapter in book: Decision Making: Processes, Behavioral Influences and Role in Business Management, R. Hudson (Ed.), Nova Science Publishers, New York, 2015, 41-78

[13] L. Bedratyuk O. Savenko, The Star Sequence and the General First Zagreb Index, MATCH Communications in Mathematical and in Computer Chemistry, 79 2 (2018) 407-414.

[14] M. Chinnaiah, N. Niranjan, Fault tolerant software systems using software configurations for cloud computing, J Cloud Comp 7 3 (2018). doi: https://doi.org/10.1186/s13677-018-0104-9.

[15] D. Fitzpatrick, D. Bodeau, R. Graubart, R. McQuaid, C. Olin and J. Woodill, (DRAFT) Cyber Resiliency Evaluation Framework for Weapon Systems: Foundational Principles and Their Potential Effects on Adversaries, The MITRE Corporation, Bedford, MA, 2019.

[16] O. Pomorova, O. Savenko, S. Lysenko, A. Kryshchuk, Multi-Agent Based Approach for Botnet Detection in a Corporate Area Network Using Fuzzy Logic, Communications in Computer and Information Science 370 (2013) 243-254

[17] S. Lysenko, O. Savenko, A. Kryshchuk, Y. Kljots, Botnet detection technique for corporate area network, in: Proceedings of the 2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems, 2013, pp. 363-368

[18] NIST, "Initial Public Draft of NIST SSP 800-160 Volume 2, Systems Security Engineering: Cyber Resiliency Considerations for the Engineering of Trustworthy Secure Systems, 2018 URL: https://csrc.nist.gov/CSRC/media/Publications/sp/800-160/vol-2/draft/documents/sp800-160-vol2-draft.pdf.

[19] X. Zhu, J. Wang, H. Guo, D. Zhu, L .T. Yang, L. Liu Fault-tolerant scheduling for real-time scientific workflows with elastic resource provisioning in virtualized clouds, IEEE Transactions on Parallel and Distributed Systems, 27 12 (2016) 3501–3517. doi: https://doi.org/10.1109/TPDS.2016.2543731.

[20] A. Bala, I. Chana, Fault tolerance-challenges, techniques and implementation in cloud computing, International Journal of Computer Science Issues 9(1) (2012)

[21] W. Zhao, Z. Wenbing, P. M. Melliar-Smith, L. E. Moser, Fault Tolerance Middleware for Cloud Computing, in: Proceedings of 2010 IEEE 3rd International Conference on Cloud Computing, Miami, USA, 2010, pp. 67–74. doi: https://doi.org/10.1109/CLOUD.2010.26.

[22] I. P. Egwutuoha, S. Chen, D. Levy, B. Selic A fault tolerance framework for high performance computing in cloud, Cluster, Cloud and Grid Computing (CCGrid), in: Proceedings of the 12th IEEE/ACM international symposium. 2012, 709–710. doi: https://doi.org/10.1109/CCGrid.2012.80.

[23] 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 (Q2) 10 4 (2020) 3651-3659

[24] J. Liu, J. Zhou J., R. Buyya, Software rejuvenation based fault tolerance scheme for cloud applications, in: Proceedings of 2015 IEEE 8th International Conference on Cloud Computing, New York, USA, 2015, pp. 1115–1118. https://doi.org/10.1109/CLOUD.2015.164.

[25] J. Liu, S. Wang, A. Zhou, S.A.P. Kumar, F. Yang, R. Buyya, Using Proactive Fault-Tolerance Approach to Enhance Cloud Service Reliability, in: Proceedings of IEEE Trans Cloud Computing PP(99), 2016. doi: http://dx.doi.org/10.1109/TCC.2016.2567392.

[26] S. Lysenko, K. Bobrovnikova, O. Savenko, A Botnet Detection Approach Based on The Clonal Selection Algorithm, in: Proceedings of 2018 IEEE 9th International Conference on Dpendable Systems, Services and Technologies, DeSSerT-2018, Kyiv, Ukraine, 2018, pp. 424-428.

[27] P. Nicolo, A frame work for self-healing software systems, in: Proceedings of the IEEE 35th International Conference on Software Engineering, 2013, pp. 1397–1400. doi: https://doi.org/10.1109/ICSE.2013.6606726.

[28] A. S. Markov V. L. Tsirlov, A. V. Barabanov, Methods for assessing the inconsistency of information security measures, Radio and communication, Echelon-Espadon, 2012 [in Russian]

[29] V. V. Lipaev Reliability and functional security of real-time software packages, Institute of System Programming, Russian Academy of Sciences, 2013 [in Russian]

[30] A. Balyk, M. Karpinski, A. Naglik, G. Shangytbayeva, I. Romanets, Using graphic network simulator 3 for DDoS attacks simulation, International Journal of Computing 16 4 (2017) 219-225

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

[32] S. Pitcher, "New DoD Approaches on the Cyber Survivability of Weapon Systems, 2019. URL: https://www.itea.org/wp-content/uploads/2019/03/Pitcher-Steve.pdf.

[33] D. J. Bodeau, R. D. Graubart, R. M. McQuaid and J. Woodill, Cyber Resiliency Metrics and Scoring in Practice: Use Case Methodology and Examples (MTR 180449), the MITRE Corporation, Bedford, MA, 2018.

[34] K. Alminshid M. N. Omar, Detecting backdoor using stepping stone detection approach, in: Proceedings of the 2013 Second International Conference on Informatics & Applications, Lodz, Poland, 2013, pp. 87-92

[35] J. Zaddach, A. Kurmus, D. Balzarotti, E.-O. Blass, A. Francillon, et al., Implementation and Implications of a Stealth Hard-drive Backdoor, in: Proceedings of the 29th Annual Computer Security Applications Conference, New Orleans, Louisiana, US, 2013.

[36] T. F. Dullien, Weird machines, exploitability, and provable unexploitability, IEEE Transactions on Emerging Topics in Computing 99 (2017) 1-15

[37] S. L. Thomas, T. Chothia, F. D. Garcia, HumIDIFy: A Tool for Hidden Functionality Detection in Firmware, in: Proceedings of the 14th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, Bonn, Germany, 2017, pp. 279-300

[38] S. L. Thomas, T. Chothia, F. D. Garcia, Measuring the Importance of Static Data Comparisons to Detect Backdoors and Undocumented Functionality, in: Proceedings of the 22nd European Symposium on Research in Computer Security. Oslo, Norway, 2017, pp. 513-531

[39] S. Bezobrazov, A. Sachenko, M. Komar, V. Rubanau, The method of artificial intelligence for malicious applications detection in android OS, International Journal of Computing, 15(3) (2016) 184-190

[40] M. Kolisnyk, V. Kharchenko, I. Piskachova, Research of the attacks spread model on the smart office’s router, International Journal of Computing, 19(4) (2020) 629-637.

[41] V. Golovko, Y. Savitsky, T. Laopoulos, A. Sachenko, L. Grandinetti, Technique of learning rate estimation for efficient training of MLP, in: Proceedings of the International Joint Conference on Neural Networks 1, 2000, pp. 323-328

[42] R. Kochan, K. Lee, V. Kochan, A. Sachenko, Development of a dynamically reprogrammable NCAP, in: Proceedings of the Conference Record - IEEE Instrumentation and Measurement Technology Conference 2, 2004, pp. 1188-1192

[43] A. Melnyk, V. Melnyk, Specialized Processors Automatic Design Tools-the Basis of Self-Configurable Computer and Cyber-Physical Systems, in: Proceedings of the 2019 IEEE International Conference on Advanced Trends in Information Theory, ATIT 2019, pp. 326-335. doi:10.1109/ATIT49449.2019.9030481

[44] J. Drozd, A. Drozd, M. Al-dhabi, A resource approach to on-line testing of computing circuits, in: Proceedings of the IEEE East-West Design & Test Symposium, Batumi, Georgia, 2015, pp. 276-281. doi: 10.1109/EWDTS.2015.7493122.

[45] M. Drozd, A. Drozd, “Safety-Related Instrumentation and Control Systems and a Problem of the Hidden Faults,” The 10th International Conference on Digital Technologies 2014, Zhilina, Slovak Republic, 2014, pp. 137–140. DOI: 10.1109/DT.2014.6868692

[46] J. Drozd, A. Drozd, S. Antoshchuk, A. Kushnerov, V. Nikul, “Effectiveness of Matrix and Pipeline FPGA-Based Arithmetic Components of Safety-Related Systems,” The 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Warsaw, Poland, 2015, pp. 785–789. DOI: 10.1109/IDAACS.2015.7341410