https://doi.org/10.31891/csit-2022-2-2
UDC 004.75:004.49

Bohdan SAVENKO, Antonina KASHTALIAN
Khmelnytskyi National University

A METHOD FOR DETERMINING THE EFFECTIVENESS OF A DISTRIBUTED SYSTEM FOR DETECTING ABNORMAL MANIFESTATIONS

Studying the effectiveness of self-organized distributed anomaly detection systems in computer systems is an important mandatory step, which is carried out to confirm the correctness, feasibility and feasibility of the developed solutions, including architecture, method of maintaining system integrity.

In order to conduct a study on the effectiveness of the use of self-organized distributed systems, anomalies in computer systems were identified in relation to the evaluation criteria. Determining the specifics of the application of the system also affects the choice of criteria for evaluating effectiveness.

The method of determining the effectiveness of the proposed solutions for the developed self-organized distributed system for detecting anomalies in computer systems has been developed. Software has been developed to ensure the functioning of a self-organized distributed anomaly detection system in computer systems to confirm the feasibility of the proposed solutions.

Experimental studies with the developed implementation of a self-organized distributed system for detecting anomalies in computer systems according to the obtained coefficients confirmed the effectiveness of the proposed solutions and the developed distributed system for its operation in the computer network.

Keywords: efficiency, method, criterion, distributed systems, intrusion detection systems.

Богдан САВЕНКО, Антоніна КАШТАЛЬЯН
Хмельницький національний університет

МЕТОД ВИЗНАЧЕННЯ ЕФЕКТИВНОСТІ РОЗПОДІЛЕНОЇ СИСТЕМИ ВИЯВЛЕННЯ АНОМАЛЬНИХ ПРОЯВІВ

Дослідження ефективності застосування самоорганізованих розподілених систем виявлення аномалії в комп'ютерних системах є важливим обов'язковим етапом, який проводиться з метою підтвердження коректності, доцільності та можливості реалізації розроблених рішень, зокрема щодо архітектури, методу підтримки цілісності системи.

Для проведення дослідження з ефективності застосування самоорганізованих розподілених систем виявлення аномалії в комп'ютерних системах було визначено щодо критеріїв для оцінювання. Визначення особливостей застосування системи впливає, також, і на вибір критеріїв для оцінювання ефективності.

Здійснено розробку методу визначення ефективності запропонованих рішень для розробленої самоорганізованої розподіленої системи виявлення аномалій в комп'ютерних системах. Розроблено програмне забезпечення для забезпечення функціонування самоорганізованої розподіленої системи виявлення аномалій в комп'ютерних системах для підтвердження можливості реалізації запропонованих рішень.

Проведені експериментальні дослідження з розробленою реалізацією самоорганізованої розподіленої системи виявлення аномалій в комп'ютерних системах згідно отриманих коефіцієнтів підтвердили ефективність запропонованих рішень і розробленої розподіленої системи щодо її функціонування в комп'ютерній мережі.

Ключові слова: ефективність, метод, критерій, розподілені системи, системи виявлення вторгень.

Introduction

Distributed computing, which is organized and implemented with the support of distributed systems, depends significantly in terms of their efficiency on the time spent on delivering tasks to a particular component of the distributed system.

The influence of time on the efficiency of distributed systems is very significant for such a class of them as intrusion detection systems, bait systems, distributed processor emulators, and so on. Therefore, an important stage in their development is to establish efficiency, the factors that affect it and study the possibility of improving efficiency.

The development of appropriate methods for calculating the effectiveness of distributed systems for the tasks of detecting computer attacks and malicious software is relevant. Because not all generally accepted methods are suitable for the specifics of this direction, and not all factors of influence may be significant in them.

Therefore, the development of a method for determining the efficiency of distributed systems is important and relevant. It will make it possible to compare the obtained result with the results of known methods.

Subject area analysis and related decisions

Distributed computing and the systems that support them are given a lot in scientific papers [1-5] and this area of research remains promising. Consider some features of distributed systems, which are presented in scientific articles.

In [6], the authors investigated the processes that exchange data through shared memory in distributed systems, in order to establish the possibility of randomly obtaining them from the main scheduler. They present a general method of calculating these values by classifying distributed algorithms according to their scheme of access to shared memory. In [7-9] the same problem is investigated, which is related to the generation of random numbers.

In [10], the authors present an approach to managing computer network resources under the condition of establishing trust in system components.

Thus, the development of a distributed anomaly detection system in computer systems is based on two components: the development of a distributed system; use and improvement of anomaly detection methods. Analysis of scientific results in these areas is the basis for creating a distributed system for detecting anomalies in computer systems. At the same time, determining the effectiveness of the created systems by comparing them with known solutions remains an urgent issue that needs to be investigated. To achieve this goal requires information about the architecture of known distributed systems, which are analogous to the developed distributed systems. Therefore, we will analyze the existing systems for detecting anomalies and intrusions into computer systems.

The closest software solutions to the developed system of detection of anomalies in local computer networks according to the set task are network systems of detection of intrusions, network anti-virus programs and non-commercial developments of the corresponding purpose.

The network system for detecting malicious software or computer attacks mainly has a centralized management module [11-14]. This allows the system administrator to manage updates and settings of all network settings from a single console. Network anti-virus tools are usually used in conjunction with anti-virus protection tools for network nodes as the second level of protection [15]. With such an architecture of network systems, it is recommended to use network and host parts from different manufacturers. For example, the system of protection of hosts in corporate networks is used as a security technology in network systems [12, 16]. Its elements allow you to control applications, web traffic and devices that connect. System functions are controlled from a single console. The network system from Dr.Web [17, 18] is the application "Dr.Web CureNet!". It is built according to a centralized architecture. Symantec Endpoint Protection is a network antivirus system. It provides the network administrator with the necessary tools to deploy antivirus tools on the network [13]. A hardware-software system for detecting malicious software and computer attacks was developed by Palo Alto Networks [19, 20]. The Malwarebytes Endpoint Security network system [21] provides an extended set of local tools for detecting and removing threats to computers on the network. Cisco® Network Admission Control (NAC) technology was developed to protect all hosts on the network [17, 18]. The Kaspersky Administration Kit network system implements independent work without intervention at the stage of research or identification of the administrator [22].

Host tools designed to detect malware or computer attacks: Avast! [23], AVG Antivirus [24], AntiVir (Avira) [25], BitDefender [26], Clam AntiVirus [20], etc.

Thus, the results of the analysis, in particular in [8, 9], show that host tools and network systems to detect malicious software or computer attacks do not ensure its complete detection. Both manufacturers and users of these tools agree with these results. The development and use of network detection systems based on the detection of anomalies in nodes in the network is promising for improving the reliability of detection. But at the same time there is a problem to prove the effectiveness of the developed distributed system. Therefore, a separate study requires the study of parameters that affect efficiency, and the direct development of a method for calculating the efficiency of the designed distributed systems.

The method for determining the efficiency of a distributed system

Research on the effectiveness of self-organized distributed anomaly detection systems in computer systems is an important mandatory step, which is carried out to confirm the correctness, feasibility and feasibility of developed solutions, including improved architecture, system integrity, anomaly detection method and implemented system. .

To conduct a study on the effectiveness of the use of self-organized distributed systems to detect anomalies in computer systems, it is necessary to determine the criteria for evaluation. Determining the specifics of the application of the system also affects the choice of criteria for evaluating effectiveness.

Since the system in question is self-organized, it will function in virtually all functions without user intervention and, therefore, the criterion that satisfies the user's requirements is to maximize the time in the system when it does not involve the user to decide on further steps. work or its component, as well as changes in its architecture. As a variable, ie an argument, in the function that will describe and set the requirements in this criterion will be time, because it can be measured during the operation of the system. The time can be determined to determine the entire period of operation of the system from the beginning of the system startup, during the working day, as well as the time spent on the user's processing of certain system states. The time spent processing certain states of the system by the user (system or network administrator) can be divided by the time determined when the system is completely shut down and when part of it is running. This division allows you to take into account the peculiarities of the processes that will occur during the period of operation of the system, and provides an opportunity to correlate different time intervals.

Let t_1^1 – the time during which the self-organized distributed system functioned. Time will be determined in hours. This is all the time spent on the work of the entire system, which includes the functioning of its individual modules offline during the absence of the center and the functioning of the system in a disabled state, that is, the time in a passive state between the periods of operation of the system. Thus, this is the time of the full cycle of

operation of the system. We will introduce $t_{2,i}^{1}$. as the time of functioning of the system or its individual components in the active state during i – the day where $i = 1, 2, \dots, d$, d – the number of days of functioning of the system, then the fair ratio $t_1^1 =< 24 * d$. Since, during the day, the system can be active or be inactive, when all the computer stations in which it is installed are disabled, it is advisable to divide the time between these cases, and also, we can consider the frequency for it, if you take a day as a time interval. As the time of stay of the system or its individual components in an inactive state, when the computer stations in which they are installed are disabled, we will enter the $t_{3,i}^{1}$. during i – the day where $i = 1, 2, \dots, d$, d – the number of days of operation of the system, then the following ratios will be fair:

$$t_{2,i}^{1} + t_{3,i}^{1} = 24, (t_{2,i}^{1} + t_{3,i}^{1}) * d \geq t_1^1, \quad (1)$$

consider the time t_1^2 - the time during which the self-organized distributed system functioned without the intervention of the user or system administrator. Then, time t_1^3 - the time during which the self-organized distributed system functioned with user or system administrator intervention at the request of the system. The time when the system functioned and there was interference of the user or system administrator will not be considered and will not be taken into account in this case, since it refers to the time during which the anomalies related to the specifics of the system will be investigated. Generally investigated time intervals are designated as t_1^1, t_1^2, t_1^3, $t_{2,i}^{1}$, $t_{3,i}^{1}$. can be attributed to the external characteristics of the system, such as the time when a user or system administrator intervened without asking for such an action by the system's internal system characteristic. Given time values associated with the ratio:

$$t_1^2 + t_1^3 = t_1^1. \quad (2)$$

If we consider the time value t_1^3, as taking into account the time during which the system or its component was serviced by the user or system administrator and was completely inactive, we denote the value of $t_1^{3,1}$, and the time during which the system continued to work, and were served by the user or system administrator on its request components, which is denoted by the value $t_1^{3,2}$. In the case of determining the value $t_1^{3,2}$, the presence of a working component with a decision-making center of the highest level of hierarchy in the system is mandatory. Divide the value $t_1^{3,2}$ at different time values that will characterize the time intervals in different cases: $t_1^{3,2,1}$ – system uptime or components with a decision-making center of the highest level of hierarchy in the system, when the user or system administrator serves at the request of the system part of its components; $t_1^{3,2,2}$ – maintenance time components that do not contain a top-level decision center in the system, user, or system administrator.

Using the entered numeric characteristics of the system, we determine the value r_1^1, which will characterize the share of time spent by the user or system administrator to maintain components that do not contain a higher-level decision center in the system, using the formula:

$$r_1^1 = \frac{t_1^{3,2,2}}{t_1^{3,2,1} - t_1^{3,2,2}}. \quad (3)$$

Value r_1^1 characterizes the case when a part of the system with a decision center of the highest level of the hierarchy works and at the same time part of the system is served by the system administrator or user, but it is considered that the time spent on maintenance affects the system and, therefore, the time of full independent functioning of the system is considered. If it is not taken into account during the independent functioning of the system, then we will introduce the appropriate value r_1^2 and calculate it by the formula:

$$r_1^2 = \frac{t_1^{3,2,2}}{t_1^{3,2,1}}. \quad (4)$$

We will introduce a criterion for evaluating the effectiveness of work according to the coefficient K_1, based on its ability to independently make decisions without involving a user or a system administrator. Values r_1^1 and r_1^2 will be convergence to zero if time is minimized $t_1^{3,2,2}$, but the convergence rate depends very much on the value $t_1^{3,2,1}$, so given that they are close in value between them we will take them into account in the final representations to determine the coefficient K_1 criterion for evaluating the effectiveness of work as follows:

$$K_1 = \frac{r_1^1 + r_1^2}{2}. \quad (5)$$

The value of the coefficient K_1 calculated by the formula (5) will be averaged and more accurately describes the given time value as a characteristic in the system. The criterion for assessing the effectiveness of the system is as follows:

$$K_1 = f\left(t_1^{3,2,1}, t_1^{3,2,2}\right) = \frac{\frac{t_1^{3,2,2}}{t_1^{3,2,1} - t_1^{3,2,2}} + \frac{t_1^{3,2,2}}{t_1^{3,2,1}}}{2} = 2 \cdot \frac{t_1^{3,2,1} \cdot t_1^{3,2,2} - t_1^{3,2,2} \cdot t_1^{3,2,2}}{t_1^{3,2,1} \cdot (t_1^{3,2,1} - t_1^{3,2,2})},$$

$$K_1 \to 0, \text{при} \min\left(t_1^{3,2,2}\right), \max\left(t_1^{3,2,1}\right). \quad (6)$$

So, determining the coefficient K_1 the formula (6) allows you to evaluate the effectiveness of the system. For example, if $t_1^{3,2,1} \leq t_1^{3,2,2}$, then it is obvious that $K_1 > 1$ and the functioning of the system during the study was ineffective.

Results obtained for efficiency criterion according to coefficient K_1 relate to the case for the total time of the system functioning and can be clarified taking into account each component of the system. That is, for each component of the system we will calculate the time of its functioning in different cases and present in the criteria for evaluating the effectiveness of the system according to the characteristics in each of the components. The component of a self-organized distributed system in a computer station in a network when analyzing its functioning according to the time indicator can be attributed to one of the cases: functioning simultaneously with the component in which the decision-making center of the highest level of hierarchy is located; functioning compatible with other components of the system without the component in which the decision-making center of the higher level of hierarchy is located; non-functioning and located in a computer system that is disabled; is non-functioning because it is being processed by a user or system administrator. Taking into account that the components of the system create parallel executable processes in different nodes in the network and the time intervals in which they are active are different, the study of the functioning of the system according to the time indicator must be carried out throughout the time of the system functioning as a whole or on average data during the day, since the active functioning of the system is periodic.

We will introduce for each j – the components of the system such values that will characterize its functioning according to time indicators, and $j = 1, 2, \dots, N$, N – the number of components of the system. For the case of $j = 0$, that is, the components that contain a decision center of a higher level of the hierarchy, we will define the time by its different states separately. Let $t_{1,j}^1$ – the time during which j functioned – that component of the self-organized distributed system. That is, it is all the time spent on the work of j – that component of the system, which includes the functioning in the following cases: offline in the presence of components with a decision center of the highest level of hierarchy; in the absence of the center; functioning of the system components in a disabled state, that is, the time in a passive state between the periods of operation of the system; A time period when a component was not functioning and was maintained by a user or system administrator. Thus, we consider these four time intervals as describing the time of the full cycle of operation j – that component of the system. Then, let's introduce $t_{2,i,j}^1$ as the time of functioning of j – the component of the system in the active state in the presence of components with a decision-making center of the highest level of hierarchy during i – the day where $i = 1, 2, \dots, d$, d – the number of days of operation of the system. To characterize the period of absence of the center, but the functioning of the system components, we will enter the value $t_{3,i,j}^1$ as the time of functioning of j – the component of the system in an active state in the absence of components with a decision-making center of the higher level of the hierarchy during i – the day where $i = 1, 2, \dots, d$, d – the number of days of operation of the system. The functioning of j is the component of the system in the disabled state, that is, the time in a passive state between the periods of operation of the system to mean the characteristics of the value $t_{4,i,j}^1$. The time interval when j – and the component did not function, but was maintained by the user or system administrator, we set the value of $t_{5,i,j}^1$, in which this characteristic is defined during i – the day when $i = 1, 2, \dots, d$, d – the number of days of operation of the system.

Similarly, we will introduce time characteristics for components with a higher level decision center of the hierarchy, that is, when $j = 0$. Enter the value $t_{2,i,0}^1$ as the time of functioning of the components with the decision-making center of the higher level of the hierarchy during i – the day where $i = 1, 2, \dots, d$, d – the number of days of operation of the system. To characterize the absence period, components with a higher-level decision center of the hierarchy due to its inaccessibility to the rest of the system components due to the definition of performing functions to determine the further actions or steps of the system, we will enter the value $t_{3,i,0}^1$ during i – the day where $i = 1, 2, \dots, d$, d – the number of days of operation of the system. Functioning of the system components with a decision center of the higher level of hierarchy in the disabled state, that is, the time in a passive state between the periods of functioning of the system to mean the characteristics of the value $t_{4,i,0}^1$. The time interval when the system component with the decision center of the higher level of the hierarchy did not function, but was served by the user or system administrator, we set the value of $t_{5,i,0}^1$, in which this characteristic is defined during i – the day when

$i = 1, 2, \dots, d$, d – the number of days of operation of the system. We will establish the following ratio between the time characteristics of the system components:

$$\frac{\sum_{i=1}^{d}\sum_{j=0}^{N}\sum_{k=2}^{5} t_{k,i,j}^{1}}{N} = t_{1,j}^{1}. \quad (7)$$

All components of the system are given time characteristics in such a way that they are parts of the system for the entire functioning of the system and this is specified by the formula (7). But in the process of prolonged functioning of the system, some of its components can be removed from it and for this case the following ratios are fair:

$$\sum_{j=0}^{N} t_{1,j}^{1} = \sum_{i=1}^{d}\sum_{j=0}^{N}\sum_{k=2}^{5} t_{k,i,j}^{1}, \quad (8)$$

$$t_{1}^{1,c} = \frac{\sum_{i=1}^{d}\sum_{j=0}^{N}\sum_{k=2}^{5} t_{k,i,j}^{1}}{N} \geq t_{1,j}^{1}, \quad (9)$$

where $t_1^{1,c}$ – the average arithmetic value of the time components in the system throughout the entire period of its functioning.

Determine the coefficient K_2 for the criterion of estimation of the effectiveness of the system, based on the entered characteristics and the obtained ratios specified by the formulas (7)-(9), to take into account the characteristics that depend on the components of the system.

Using the entered numeric characteristics of the system components, we determine the value $r_{1,j}^{1}$, which will characterize the share of time spent by the user or system administrator to serve the j-th component by the formula:

$$r_{1,j}^{1} = \frac{t_{1,j}^{3,2,2}}{t_{1,j}^{3,2,1} - t_{1,j}^{3,2,2}}. \quad (10)$$

where $t_{1,j}^{3,2,1}$ – the time of operation of j-th components with a decision-making center of the highest level of the hierarchy in the system, when the user or system administrator serves at the request of the system part of its components, including the j-th component; $t_{1,j}^{3,2,2}$ – maintenance time for j-part that does not contain a top-level decision center in the system, user, or system administrator.

Value $r_{1,j}^{1}$ characterizes the case when a part of the system with a decision center of the highest level of the hierarchy works and at the same time part of the system is served by the system administrator or user, but it is considered that the time spent on maintenance affects the system and, therefore, the time of full independent functioning of the system is considered. If it is not taken into account during the time of self-operation of the system, then we will introduce the appropriate value $r_{1,j}^{2}$. For j-part and calculate it by formula:

$$r_{1,j}^{2} = \frac{t_{1,j}^{3,2,2}}{t_{1,j}^{3,2,1}}. \quad (11)$$

We will introduce a criterion for evaluating the effectiveness of work for j-th component according to the coefficient $K_{2,j}$, based on the possibility of part of the system to independently make decisions without involving the user or system administrator. Values $r_{1,j}^{1}$ and $r_{1,j}^{2}$ will be convergent to zero if time is minimized $t_{1,j}^{3,2,2}$, but the convergence rate is very dependent on the value $t_{1,j}^{3,2,1}$, so given that they are close in value between them we will take them into account in the final representations to determine the coefficient $K_{2,j}$. criterion for evaluating the effectiveness of the work of j-th component as follows:

$$K_{2,j} = \frac{r_{1,j}^{1} + r_{1,j}^{2}}{2}. \quad (12)$$

The value of the coefficient $K_{2,j}$, calculated by the formula (12) will be averaged and more accurately describe the given time value as a characteristic of j-of-that component in the system. The criterion for assessing the efficiency of the j-part of the system is set as follows:

$$K_{2,j} \to 0, \text{при } \min\left(t_{1,j}^{3,2,2}\right), \max\left(t_{1,j}^{3,2,1}\right). \quad (13)$$

So, determining the coefficient $K_{2,j}$ the formula (13) allows you to evaluate the effectiveness of the functioning of the j-th component of the system. The general efficiency factor of the system, taking into account the

time characteristics of the system components and the values of coefficients obtained from the formula (13), is found by the formula:

$$K_2 = \max\left(K_{2,0}, K_{2,1}, \dots, K_{2,N}\right). \quad (14)$$

The system performance criterion, which is set using the coefficient value K_2, similar to the case of K_1, will display the best result when minimizing its value. Thus, the value of coefficients K_1 and K_2 will coincide to zero, provided that there are no long-term failures in the system and its components, as well as in the absence of destructive effects of malware and computer attacks on nodes on the network in which the system components are located, and short-term maintenance work on the part of the user or system administrator. In addition, the obtained values of coefficients that are calculated for a certain period of time will be taken into account when determining the further actions of the system.

Experimental research on the use of a self-organized distributed system

The purpose of experimental research is to study the effectiveness of the functioning of the self-organized distributed system and the reliability of anomaly detection in computer systems. The study of the effectiveness of the functioning of the self-organized distributed system is necessary to establish the implementation of its functions to maintain the integrity and coordination of the work of the system components. The reliability of the detection of an anomaly in computer systems requires research to establish the possibility of its use in real conditions. The part of the distributed system responsible for detecting an anomaly is implemented as an appropriate method in it, and its testing will allow to evaluate the reliability of the detection of an anomaly in computer systems.

Therefore, consider testing the distributed system first with the disabled module responsible for detecting an anomaly. Indicators that will be investigated are grouped by the following characteristics: communication time between individual components; time of communication between the components of the system and the component in which the decision-making center of the higher level of the hierarchy is located; communication time between components depending on the number of active components that form the system; the time spent on the work when the center distribution was carried out in the system, i.e. distributed center testing, and the time spent when the center is not divided between the levels of the hierarchy, but is completely in one component.

Experimental studies with the developed self-organized distributed system were carried out in a local computer network created using Ethernet technology with a data transfer rate of 1 GB/s between nodes in the network. Since the messages for transmission between the components of the system contain a very small amount of information, they are formed in short packages and we assume that each of them was transmitted in one package of 64 bytes.

The self-organized distributed system studied contained eight components, one of which was a decision-making center of the highest level of hierarchy. Experimental studies were carried out within 50 days separately for the case when the decision-making center was distributed among all components taking into account two levels of hierarchy and also 50 days for the case when the decision-making center was placed only in one component. During the entire time of the experiment, the time of sending messages, their number and recording of the time of receipt were recorded. This was carried out in order to conduct a study of the time spent on communication in the middle of the system itself. The results of experimental studies are presented in Table 1.

Table 1

Results of experimental studies on the effectiveness of the functioning of the self-organized distributed system

№	Time value characteristics	Time interval for the case when the time spent on the work of the system with the distribution of the center, i.e. testing with a distributed center, s	Time interval for the time spent on the system without distributing the center between the components, s
1	The time between individual components	1,42 – 2,61	1,41 – 2,56
2	Communication time between the system components and the component that hosts the top-level decision-making center of the hierarchy.	1,81 – 2,87	1,67 – 2,23
3	Communication time between components depending on the number of active components that form the system. Cases 3.1. Number of components 2-4. 3.2. Number of components 5-8.	1,41 – 2,51 1,83 – 2,81	1,27 – 2,39 1,72 – 2,43
4	Number of packages moved for the event under study 1.	7546	5788
5	Number of packages moved for the event under investigation 2.	12458	8386

The results of experimental studies presented in Table 1 confirm the increase in the number of messages transmitted for the case when the time spent on the work of the system with the distribution of the center, that is, testing with a distributed center, compared to the case when the time spent on the system without the distribution of the center among the components. In addition, the processing time of transmitted message packets increases because the time is wasted.

To the work of the system with the distribution of the center, messages from the system components in parallel at one time are sent to the components, which houses the decision-making center and is sent to all components, unlike when the decision-making center, which is represented only in one component, independently decides on communication with the rest of the component, which reduces significantly the number of messages between components. But at the same time, the time costs for the transmission of messages are significantly small in both cases, so the use of architecture with the distribution of the center between the levels of hierarchy in two different types of system components is effective, which is confirmed by the results of experiments.

The next step in the processing of experimental studies is to determine the ratio of processing time and functioning of the self-organized distributed system according to the introduced coefficients of its efficiency of work K_1 and K_2. Since the time characteristics of the processes that took place in the self-organized distributed system under study are fixed, in addition to the fact that they were used by the system itself to determine further actions, we will use them to build function graphs according to its efficiency coefficients K_1 and K_2. For the experiment, a variant of the system was used with the distribution of the center between two levels of the hierarchy. Initially, the case when the user or system administrator did not intervene was analyzed, then the time spent processing was zero and, indeed, there were no costs associated with servicing part of the components in the system. The second case in the experiment involved and was realized when the time of maintenance of system components by the user or system administrator was significantly shorter than the entire system and when it was longer. The results of such experiments are presented in Fig. 1. They refer to two parameters that characterize the system. These settings are the system components' time in the site and the time it took the user or system administrator to process the event in a computer station. Determining such time parameters will allow calculating the coefficients K_1 and K_2. The time intervals in the graphs are represented by time charts. Of the eight components, only three (4,5,7) required user or system administrator interventions. The rest were not needed. The last ninth graph displays a summary time chart of the entire system.

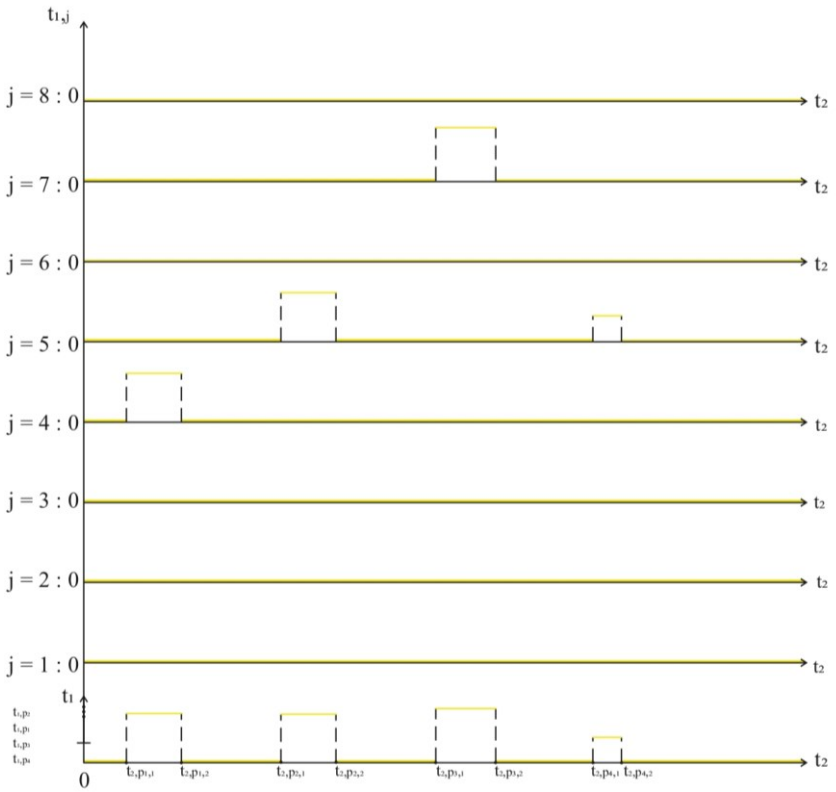

Fig. 1. Time charts based on the results of experimental studies

The value of coefficients K_1 and K_2 after the experiments are presented in Table. 2.

Table 2

Coefficient values K_1 and K_2

№	System components maintenance time by the user or system administrator was significantly shorter than the entire system	System components maintenance time by the user or system administrator was significantly longer than the entire system
	Case 1	Case 2
K_1	0,00987567	0,04873418
K_2	0,00986972	0,04873326

The obtained coefficients confirm the effectiveness of the proposed solutions and the developed distributed system for its functioning in the computer network. In the second case, not large values of coefficients are justified by the fact that for a long time only certain components of the system were served, and most components of the systems functioned. This proves the effectiveness of using distributed systems of this type.

Also, according to the results of the experiment, it was found that when the decision-making center is divided between the levels of the hierarchy, the efficiency in time is better, because processing at the lower level of the hierarchy reduces the time of event handling compared to using one center. Reliability in the processing of an anomaly in computer systems, which were artificially entered, is 0.8356, which is a satisfactory result and confirms the sufficient effectiveness of the proposed solutions.

Conclusions

To evaluate the effectiveness of the proposed solutions, a method for calculating the effectiveness of the use of a self-organized distributed anomalies detection system in computer systems has been developed. It uses two entered criteria, the results of which are determined by the appropriate coefficients. In addition, the obtained values of coefficients that are calculated for a certain period of time are taken into account when determining further actions of the system.

Experimental studies with the developed implementation of a self-organized distributed system for detecting anomalies in computer systems according to the obtained coefficients confirm the effectiveness of the proposed solutions and the developed distributed system for its functioning in the computer network.

References

1. Savenko, O., Nicheporuk, A., Hurman, I., Lysenko, S. - CEUR-WS. – 2019. – Vol. 2393. – P.633-643, ISSN: 1613-0073.

2. Savenko O. Metamorphic Viruses' Detection Technique Based on the Equivalent Functional Block Search / O. Savenko, S. Lysenko, A. Nicheporuk, B. Savenko // CEUR-WS, ISSN: 1613–0073. – 2017. – Vol. 1844. – Pp. 555–569.

3. Lysenko S., Bobrovnikova K., Matiukh S., Hurman I., Savenko O. Detection of the botnets' low-rate DDoS attacks based on self-similarity. *International Journal of Electrical and Computer Engineering*, Vol. 10, Issue 4, 2020, Pages 3651-3659. DOI: http://doi.org/10.11591/ijece.v10i4.pp3651-3659.

4. Pomorova O. Metamorphic Viruses Detection Technique based on the the Modified Emulators [Text] / O. Pomorova, O. Savenko, S. Lysenko, A. Nicheporuk // CEUR-WS. – 2016. – Vol. 1614. – PP.375-383, ISSN: 1613-0073

5. Sergii Lysenko, Kira Bobrovnikova and Oleg Savenko. A Botnet Detection Approach Based on The Clonal Selection Algorithm // Proceedings of 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DeSSerT-2018, Kyiv, Ukraine, May 24-27, 2018) – Pp. 424-428.

6. Antoniadis, K., Blanchard, P., Guerraoui, R. *et al.* The entropy of a distributed computation random number generation from memory interleaving. *Distrib. Comput.* **31,** 389–417 (2018). https://doi.org/10.1007/s00446-017-0311-5

7. Alistarh, D., Sauerwald, T., Vojnovic, M.: Lock-free algorithms under stochastic schedulers. In: Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing, PODC 2015, Donostia-San Sebastián, Spain, July 21–23, 2015, pp. 251–260 (2015). doi:10.1145/2767386.2767430

8. Barker, E., Kelsley, J.: Recommendation for random bit generator (rbg) constructions. SP 800-90C (2012)

9. Zhou, H., Bruck, J.: Generalizing the Blum-Elias method for generating random bits from markov chains. In: Proceedings of IEEE International Symposium on Information Theory (ISIT) (2010)

10. Pomorova O. Multi-Agent Based Approach for Botnet Detection in a Corporate Area Network Using Fuzzy Logic [Text] / Oksana Pomorova, Oleg Savenko, Sergii Lysenko, and Andrii Kryshchuk // Communications in Computer and Information Science. – 2013. – Vol. 370. - PP.243-254, ISSN: 1865-0929.

11. B. Savenko, S. Lysenko, K. Bobrovnikova, O. Savenko, G. Markowsky. Detection DNS Tunneling Botnets // Proceedings of the 2021 IEEE 11th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), IDAACS'2021, Cracow, Poland, September 22-25, 2021.

12. Corporate Endpoint Protection Products Group Test: Socially-Engineered Malware Q2 [Electronic resource]: [Web-site]. – Electronic data. – Mode of access: http://www.nsslabs.com/ research/ endpoint-security/ anti-malware/q2-2010-endpoint-protection-product-group-test.html (Viewed on April 3, 2021). – Title from the screen.

13. ESET Endpoint Security [Electronic resource]: [Web-site]. – Electronic data. – Mode of access: http://www.eset.com/ (Viewed on April 3, 2021). – Title from the screen.

14. Symantec Endpoint Protection [Electronic resource]: [Web-site]. – Electronic data. – Mode of access: https://www.anti-malware.ru/reviews/ Symantec_Endpoint_Protection (Viewed on April 3, 2021). – Title from the screen.

15. Branitskiy A., Kotenko I. Hybridization of computational intelligence methods for attack detection in computer networks. Journal of Computational Science. 2017. №23. P. 145–156.

16. Stetsyuk M., Stetsyuk V., Savenko B., Savenko O., Dobrowolski M. Implementation of Control by Parameters of Client Automated Workplaces of Specialized Information Systems for Neutralization malware. CEUR-WS. 2021. Vol. 2853. P. 340-352. URL: http://ceur-ws.org/Vol-2853/paper40.pdf.Securelist. FAQ: Disabling the new Hlux / Kelihos Botnet [Electronic resource]: [Web-site]. – Electronic data. – Mode of access: https://securelist.com/blog/research/32634/faq-disabling-the-new-hluxkelihos-botnet-13/ (Viewed on April 3, 2021). – Title from the screen.

17. COMSS1 [Electronic resource]: [Web-site]. – Electronic data. – Mode of access: http://www.comss.ru/page.php?id=2758 (Viewed on April 3, 2021). – Title from the screen.

18. Enterprise End Point Protection Comparative Analysis - Socially Engineered Malware: Report Overview [Electronic resource]: [Web-site]. – Electronic data. – Mode of access: www.nsslabs.com/reports (Viewed on April 3, 2021). – Title from the screen.

19. Bakotech [Electronic resource]: [Web-site]. – Electronic data. – Mode of access: https://bakotech.ua/ (Viewed on April 3, 2021). – Title from the screen.

20. ClamAV [Electronic resource]: [Web-site]. – Electronic data. – Mode of access: https://www.clamav.net/ (Viewed on April 3, 2021). – Title from the screen.

21. Malwarebytes Endpoint Security [Electronic resource]: [Web-site]. – Electronic data. – Mode of access: https://ru.malware bytes. com/business/endpoint security (Viewed on April 3, 2021). – Title from the screen.

22. Kaspersky Lab [Electronic resource]: [Web-site]. – Electronic data. – Mode of access: http://www.kaspersky.ru (Viewed on April 2, 2019). – Title from the screen.

23. Avast! [Electronic resource]: [Web-site]. – Electronic data. – Mode of access: https://www.avast.com/index (Viewed on April 21, 2020). – Title from the screen.

24. AVG [Electronic resource]: [Web-site]. – Electronic data. – Mode of access: http://www.avg.com (Viewed on April 21, 2020). – Title from the screen.

25. Avira [Electronic resource]: [Web-site]. – Electronic data. – Mode of access: http://www.avira.com (Viewed on April 21, 2020). – Title from the screen.

26. Bitdefender [Electronic resource]: [Web-site]. – Electronic data. – Mode of access: http://www.bitdefender.com/(Viewed on April 21, 2020). – Title from the screen.