Method of random dynamic control transfer between network nodes for a partially centralized OS security system*

Yuriy Stetsyuk [1,*,†], Mykola Stetsyuk [1,†], Bohdan Savenko [1,†], Andrzej Kwiecien [2,†], Lukasz Kopania [3,†]

[1] *Khmelnytskyi National University, Khmelnytskyi, Instytutska street 11, 29016, Ukraine*
2 Silesian University of Technology, Akademicka str., 2A, Gliwice, Poland
3 Kazimierz Pulaski University of Technology and Humanities in Radom, Department of Informatics, Jacek Malczewski str., 29, Radom, 26600, Poland

Abstract

This paper analyzes the current threat level posed by malicious software to protected specialized OSs in relation to their resistance to leaks of confidential information. A model describing the interaction of protective mechanisms within a centralized security system for a network node OS is presented, which allows building a security architecture of a network OS, in which the problem of leakage of confidential information during attacks on the system's RAM is minimized. This is achieved by incorporating a marking mechanism that controls the channels through which confidential information passes at the memory page level. The principles of building centralized and partially centralized OS security systems and methods of organizing the operation of their security mechanisms are considered. A method of random dynamic transfer of control between nodes of a computer network is proposed as a way to increase the resistance of a computer network to leakage of confidential information and eliminate the bottleneck of a centralized system. Its feature is the synchronization and updating of the central database of privileges and the network state by replicating the local databases of network nodes and ensuring the availability of a backup control node, which is in constant readiness in case of failure. A graph model of the states of the central node of the network is presented, in which the main sequence of its states involved in the dynamic transfer of control between nodes of the computer network and the states corresponding to the most probable possible emergency situations are highlighted. A number of experiments were conducted and a comparative analysis of the effectiveness of centralized and partially centralized OS security systems was performed.

Keywords

operating system, protective mechanism, information leak, centralized system

Intelitsis'25: The 6th International Workshop on Intelligent Information Technologies & Systems of Information Security, April 04, 2025, Khmelnytskyi, Ukraine
* Corresponding author.
† These authors contributed equally.
yuriy.stetsuk@khmnu.edu.ua (Y. Stetsyuk); mykola.stetsiuk@khmnu.edu.ua (M. Stetsyuk); savenko_bohdan@ukr.net (B. Savenko); andrzej.kwiecien@polsl.pl (A. Kwiecien); l.kopania@uthrad.pl (L. Kopania)
0000-0003-0312-2276 (Y. Stetsyuk); 0000-0003-3875-0416 (M. Stetsyuk); 0000-0001-5647-9979 (B. Savenko); 0000-0003-1447-3303 (A. Kwiecien); 0000-0002-7318-4803 (L. Kopania)

1. Introduction

The revolution in the field of information technology has resulted in many aspects of human activity becoming critically dependent on various physical or virtual technological advancements, which is especially true for information technology. Planetary society has entered an era of its existence, where information has become the dominant resource that determines the success of development in any field of its activity. Enormous resources of the modern world economy are directed to its processing, storage, and provision of information services.

At the same time, it must be recognized that all information activities of humanity are under the constant negative influence of attackers and malicious software, which is also in constant development. Therefore, the issue of ensuring information protection and the stability of information systems to its leakage becomes particularly relevant.

The analysis of the situation in the information systems protection sector shows that the task of ensuring information protection remains extremely relevant. Despite the large number of methods developed by scientists to protect information systems, it cannot be considered solved.

In this paper, the emphasis of the information protection problem is aimed at increasing the efficiency of network OS security systems, increasing their resistance to leaks of confidential information through random dynamic transfer of control between network nodes.

2. Related work

Operating systems (OSs) have evolved significantly, transitioning from universal OSs to specialized secure OSs, which are used in critical environments where information loss is unacceptable under any circumstances. These include government and military systems, various financial institutions, and corporate information systems, where data security is of paramount importance. Today, this is not only a purely scientific or technical concept, but also a legal one. The requirements for such systems are outlined in the US standard developed by the National Institute of Standards and Technology (NIST), as well as in international standards [1].

Today, there are many incident response systems and approaches that can be used to improve the level of security [2, 3] in networks in terms of counteracting the leakage of confidential information. It has been established that OSs operating as part of cyber-physical systems that use CBTC wireless data channels are vulnerable to attacks due to the low resilience of their intrusion detection mechanism (IDS) based on machine learning [4], which can detect only known attacks contained in training data sets and is insensitive to new types of attacks. In order to eliminate the above problem, in [5], a more advanced IDS mechanism based on transfer learning for the CBTC system is proposed. It uses an optimized one-dimensional convolutional neural network block and long-term short-term memory to automatically extract spatial and temporal characteristics from the source data and a new method of knowledge transfer, which made it possible to detect new types of attacks. In [6, 7], an improvement of intrusion detection systems (IDS) when operating OS with data in clouds is proposed based on the use of a combination of the fuzzy clustering kernel (KFCM) and the optimal fuzzy neural network of type 2 (OT2FNN), as well as the E-SGX protection mechanism [8].

In [9], a distributed network system with an intrusion prevention system based on the Spark structure and the dynamic model of information security theory is proposed. Its architecture and functional structure are built by improving the Spark algorithm. Another similar mechanism for mitigating attacks on computer networks is proposed in [10, 11].

Another option for building a system for early detection of potential channels of attack on computer systems is shown in [12, 13], the architecture of which includes agents of different types and a certain number of instances, specializing in solving the problems of detecting potential channels of attack on a computer system and interacting with each other, which jointly solve the general problem of detecting intrusion into a computer system [14]. In [15], a method for detecting ADRs at endpoints based on event identifiers using deep learning is proposed. Event identifiers are understood as the behavior of ADRs, which are monitored and collected in the OS kernel [16].

In [17, 18], the use of intelligent deep learning methods in intrusion detection systems in computer networks is proposed. Within the framework of this approach, the WNIDS method is implemented, within which an SDN controller operates, which effectively monitors and manages network devices, analyzes data sets. It is stated that a deep learning method called Bidirectional LSTM (BiLSTM) is implemented based on the WNIDS model, which allows for effective intrusion detection in the SDN-IoT network. In [19], two variants of the FTNPA algorithm are presented: decentralized and centralized. The first variant implements local recovery, when a mobile node is placed in a certain place to help the affected area. The second variant uses a centralized detection method, where the recovery method creates alternative paths of mobile nodes to the destination node.

Much attention has been paid by scientists to improving the operation of OS protection mechanisms, namely its kernel [20, 21]. A model of a centralized OS security system for new OS architectures has been proposed [22].

An interesting solution is to supplement the OS security systems with the DiSAuth authorization system [23], which provides a secure algorithm based on the Domain Name System (DNS) and blockchain, ensuring its resistance to tampering, preserving confidentiality, and protecting data from leakage. [24] raises the issue of minimizing the granting of access privileges to resources as a way to increase resilience to information leaks.

The integration of artificial intelligence into computer networks is one of the advanced technologies. It is assumed that the concept of applying artificial intelligence will solve the problems faced by computer networks, especially with regard to eliminating the threat of information leakage [25]. Methods for increasing network resilience by using specialized cryptographic modules at the transport layer (TLS) together with secure hardware security modules (HSMs) are also proposed [26].

3. Formulation of the problem

A study of methods to enhance the resistance of specialized OSs to information leaks has shown that, despite modern OSs formally meeting the aforementioned criteria, they still require additional protection mechanisms when used in information systems designed to process confidential data.

The analysis of the current state of the level of threats from malicious software to today's protected OS confirms that most of them are due to the conceptual approach implemented in

the OS, which is based on the implementation of a distributed administration scheme for protection mechanisms. Within the framework of this approach, the user in the OS is considered as a trusted person who is an element of the administration scheme and has the ability to assign access delimitation rules. At the same time, he is not perceived as a potential attacker who can knowingly or unknowingly carry out unauthorized access to information.

Thus, the OSs currently in existence are unable to guarantee one hundred percent resistance to information leakage due to the existence of vulnerabilities on the one hand, and on the other hand, due to being under constant threat from the PPS, which is only increasing. Without the implementation of additional OS protection mechanisms based on centralized and partially centralized security systems, countering information leakage and, in general, unauthorized access to confidential information will be ineffective.

4. Methods and material

4.1. Centralized and partially centralized network OS security systems

Consider the graph model of the OS security system shown in Figure 1. Here, the functional modules of the OS security system correspond to the vertices of the graph, and the directed edges indicate the interaction between the modules, determining the algorithm of its operation. The main interactions:

- the central security management module (CSMM) transmits security policies, including confidential information marking policies, through the Policy Distribution module to the peripheral security modules PSM1 – PSMn;
- the peripheral security modules PSM1 - PSMn transmit monitoring data through the Monitoring module to the central CSMM module for analysis;
- the Monitoring module transmits the results of monitoring the correctness of confidentiality markers to the Labeling module;
- the Labeling module transmits information about the marking of confidential information to the agents to the peripheral security modules PSM1 - PSMn for further use at the memory page level;
- AuthM authentication, AuthZ authorization, EncM encryption, and NetSec network security modules interact with the Labeling module when performing their functions.

The proposed model of a centralized security management system allows us to obtain an OS security architecture that will be free from the problem of confidential information leakage during low-level malware attacks. This is achieved by including a marking mechanism in it, the feature of which is the control of channels for passing confidential information at the level of memory pages.

If we take a machine with an operating system with the given security system model as the basic node of a computer network, we will get a network where centralized privilege management will be implemented in each individual node, but the network itself will have decentralized management, with all the security problems inherent in this model. If we transfer network management to one of the network nodes, we will get a network system with centralized management. This management model has a number of significant advantages, such as:

- simplified management, since there is no need to coordinate actions between several management modules - administration is carried out through a single CSMM module (Figure 1);
- unification of privilege management policies and resource management according to the same rules, security policies and protocols are installed in one place, which ensures the unity of network management, facilitates administration, since changes in policies are applied simultaneously to the entire network;

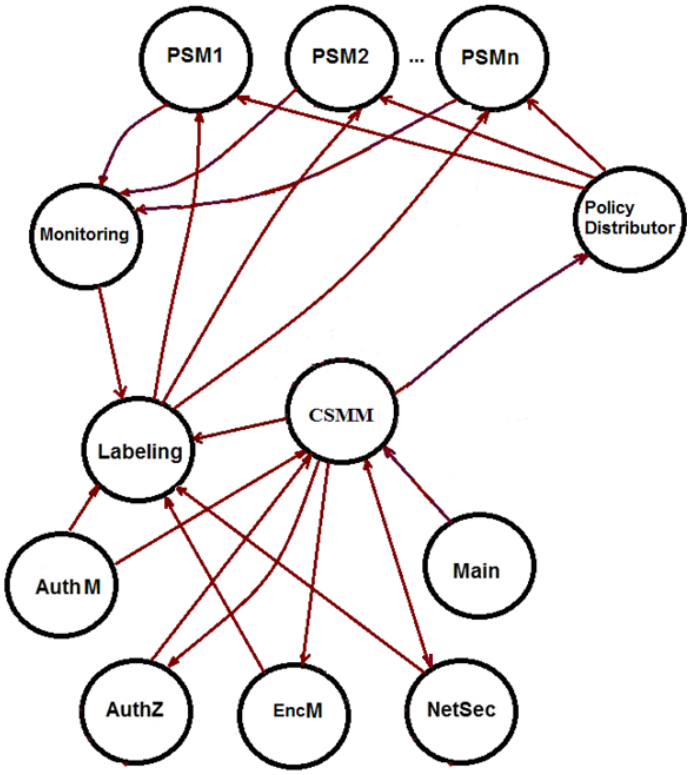

Figure 1. A model of a centralized OS security system with a low-level labeling mechanism for confidential information.

- centralized monitoring, means that all information about the state of the network and its nodes comes to a single module, which greatly facilitates the detection of threats, simplifies analysis and decision-making.
- However, a centralized control system has a number of potentially threatening problems. The main ones are:
- the so-called "single point of failure" (Single Point of Failure), when the failure of the central module makes the network vulnerable, creating a high risk of its continuous operation;
- vulnerability to malware attacks. The very presence of a central control node makes it a prime target for various kinds of destruction, as a successful attack can give access to the entire network.

The purpose of this study is to find a solution that would allow preserving the main advantages of a centralized system, while getting rid of its biggest problems.

4.2. Method of dynamic transfer of control between network nodes with replication of the network OS security privilege base

As one of the ways to overcome the shortcomings of the centralized OS security system while preserving its positive qualities, a method of dynamic control transfer between network nodes during its operation is proposed (Figure 2). The method is based on the constant migration of the control center across network nodes, which allows getting rid of the main drawback of a purely centralized approach - the presence of a bottleneck in the form of vulnerabilities in the central control module.

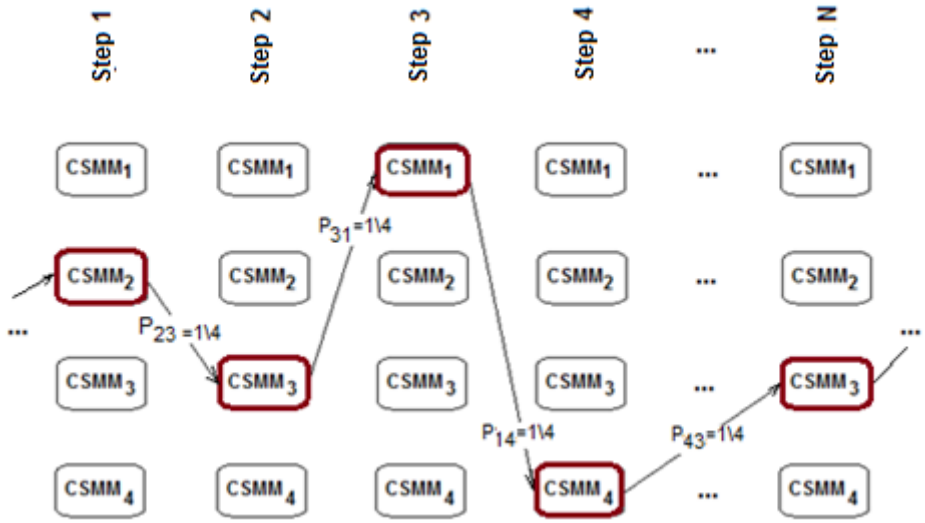

Figure 2. Fragment of the process of transferring control of network nodes with period P.

According to the proposed method, a computer network is represented as a set of nodes M_{CSMM}, whose OS security systems have central control modules CSMM:

$$M_{CSMM} = \{CSMM_1, CSMM_2, \dots, CSMM_n\} \quad (1)$$

where: n is the number of network nodes.

In this case, each CSMM module of the node is associated with the privilege base $B_{privileges}$ which in the current mode serves as the local base of the node, and when the node in the process of dynamic transfer of control takes over the function of the central network control module, it is updated to the level of the main network privilege base. All local privilege bases of network nodes can be represented by the set:

$$B_{privileges} = \{B_{privileges\,1}, B_{privileges\,2}, \dots, B_{privileges\,n}\} \quad (2)$$

Let us define the state of the system S_{System} at time t, which corresponds to the moment preceding the transfer of control, as:

$$S_{System}(t) = (CSMM_{Active}(t), CSMM_{Reserve}(t), T_{Governance}) \quad (3)$$

where: $CSMM_{Active}(t) \in M_{CSMM}$ - a node from the M_{CSMM} set that owns a token at time t (is the network control center), $CSMM_{Reserve}(t) \in M_{CSMM}$ - backup control node at time t in the network, $T_{Governance}$ – management token.

As can be seen from the definition of the network state (formula 3), a feature of the method is the transfer of the functions of the central control module to the next node by sending it a control token. We describe the transfer of the token as a transition:

$$S_{System}(t_k) = (CSMM_{Active}(t_k), CSMM_{Reserve}(t_k), T_{Governance}) \rightarrow S_{System}(t_{k+1}) \quad (4)$$
$$= (CSMM_{Active}(t_{k+1}), CSMM_{Reserve}(t_{k+1}), T_{Governance})$$

where: $CSMM_{t_{k+1}}$ - a new control node, randomly selected from the set $M_{CSMM/\{CSMM_k\}}$, whose selection rule is given by the formula:

$$CSMM_{k+1} \approx Uniform(M_{CSMM/\{CSMM_k\}}) \quad (5)$$

where: $Uniform()$ – a function that provides the same probability of choosing any node from among the active ones, excluding the current node, as the control node $CSMM_k$, $CSMM_k$ - the current control node, after transferring control to the $CSMM_{k+1}$ module, performs the function of a backup control node, $T_{Governance}$ - management token.

The control token has a simple structure (Figure 3) and is physically a data packet that is transmitted between network nodes to determine the current control module (CSMM). At one point in time, only one network node owns the token. Thus, it serves to coordinate the network state, transfer control, avoid conflicts, and prevent multiple nodes from simultaneously performing control functions. In order to maintain its integrity, the token is digitally signed.

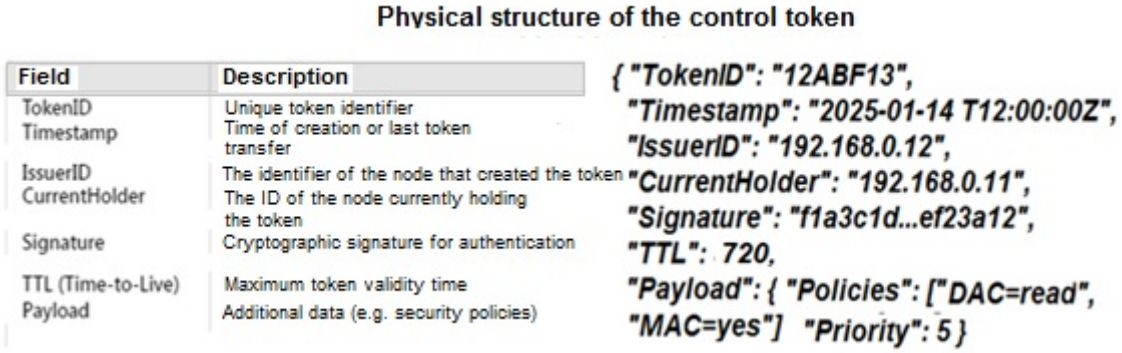

Figure 3. Structure of the dynamic control method control token.

An important point of the dynamic control transfer method is the updating of the main privilege base $B_{privileges\,k+1}$ for the node to which control of the computer network is transferred. It is obtained during the replication procedure of the local privilege bases $B_{local\,i}$ of active network nodes at the time of changing the network control center.

In the current mode of operation, the node uses the local privilege base to the maximum, and when transferring the functions of the central control node to it, it activates the main base. It is assumed that each node stores its local privilege base, and the main base of the control

M_{CSMM}

are allowed for replication, creating a set of privilege bases for replication $B_{privileges\ replication}$, which in the general case may not coincide with the set $B_{privileges}$:

$$B_{privileges\ replication} \leftarrow Filter(B_{local\ i}, Active(CSMM_i)) \quad (6)$$

where: $Filter\left(B_{local\ i}, Active(CSMM_i)\right)$ - filter local bases $B_{local\ i}$ by condition, $Conditium = Active(CSMM_i)$ - the condition for including the local base $B_{local\ i}$ of the i-th node $CSMM_i$ in the replication set when filtering local bases is its membership in the active node.

Replication will be successful if local databases that are up-to-date participate in it. For someB_{local}, its up-to-dateness can be determined through the Validate() function:

$Validate(B_{local\ i}) = 1$, here the value of the function equal to one means that it is relevant. The condition for the relevance of the $B_{local}\ i$ basis is that it falls under the following conditions:

- $Sinc(B_{local\ i}) = 1$ - database synchronized;
- $t_{update}(B_{local\ i}) \leq\triangle t_{max}$ - the time of its last update does not exceed a certain set maximum period $\triangle t_{max}$;
- $Integririty(B_{local\ i})$- the local privilege base is not corrupted.

Given that each local privilege base B_{local} must be up-to-date, the rule for including new elements in the set $B_{privileges\ replication}$ will now look like this:

$$B_{privileges\ replication} \leftarrow Filter(B_{local\ i}, Active(CSMM_i) \wedge Validate(B_{local\ i}) = 1) \quad (7)$$

Taking into account this rule, we obtain a formula for all elements of the set of local privilege bases subject to replication:

$$B_{privileges\ replication} = \{B_{local\ i}\ | i \in M_{CSMM}, Active(CSMM_i) \wedge Validate(B_{local\ i})\} \quad (8)$$

Now, when performing the replication operation on the elements of the set $B_{privileges\ replication}$, the current main privilege base for the central network management node will be obtained, which does not contain incorrect data from local, out-of-date privilege bases and local privilege bases of inactive nodes:

$$B_{privileges\ k+1} = Merge(B_{privileges\ replication}) \quad (9)$$

The transfer of control from the current central module to the next one will be successful if the computer network system is in the $S_{System}(t_k)$ state and the next k+1-th control node has an up-to-date primary privilege base $B_{privileges\ k+1}$. In this case, the node that has just transferred control to the next central node goes into reserve, maintaining its ability to return to the central node function.

This approach to implementing the method practically makes it impossible to lose the privilege base and does not require constant access to the global network, as in the case of its

storage in cloud storage, or implementation in the form of transactions in the blockchain network, maintaining a high level of autonomy inherent in corporate networks. At the same time, high resistance of the computer network as a whole to the effects of various types of destruction is also ensured and, accordingly, the probability of leakage of confidential information is reduced. Thus, the method ensures the continuity of network operation, data consistency and system security.

Now we can formulate the main steps of the dynamic control transfer method:

Step 1. Preparation for control transfer. The state in which the current central security management module $CSMM_i$ (Figure.1) of the set M_{CSMM} is located is evaluated, where the presence of signs (timeout, system overload, scheduled maintenance) is checked. If at least one of them is active, the current central module must transfer control functions to another network node $CSMM_{i+1}$. If at least one sign is present, the current module determines the next $CSMM_{i+1}$ module from the number of active nodes that will take control of the network. Then the current module $CSMM_i$ checks the relevance of the main privilege base $B_{privileges\ j+1}$ of the node that is to take over the functions of the central control node. If necessary, its replication or synchronization is performed.

Step 2. Transfer of the current system state, which includes the transfer of active sessions, security policies and system monitoring status, event log data. The current module $CSMM_i$ initiates a check of the correctness of the received data by the successor module $CSMM_{i+1}$ and the relevance of the privilege base $B_{privileges\ j+1}$, after which it notifies the current module $CSMM_i$ of its readiness to perform the functions of the central node.

Step 3. Activation of the new module. The current $CSMM_i$ module sends a notification to the successor in the form of a management token $T_{Governance}$, while all other active nodes of the network are transmitted information about the new NM module. An important feature is that at any given time only one node from the set $CSMM_{i+1}$ owns the management token $T_{Governance}$, which ensures its management role in the network.

The old module is deactivated, but at the same time remains ready as a backup module, periodically synchronizing its database of privileges and event logs in the system with the database and logs of the new central node.

Step 4. Having received the $T_{Governance}$ control token (Figure 2), the node activates itself as the central module $CSMM_{i+1}$, taking control of the system. Its first step is to record in the logs all actions related to the fact of transferring control to it, including the time of transfer, the identifiers of the CSMM modules of the active nodes, the state of the privilege base. In addition, the new module $CSMM_{i+1}$ notifies all active nodes of the system that it is ready to perform the functions of the network security system manager.

Step 5. System monitoring and control. In this step, the new $CSMM_{i+1}$ module monitors the security status of the system in order to detect possible failures or destruction. Constantly informs the backup $CSMM_i$ module about changes in the main privilege database in order to maintain them in a state of maximum synchronization. Tracks the status of the signs of transferring network control to the next node.

4.3. Algorithm for dynamic control transfer between network nodes

In order to further detail the work of the proposed method, we will display the steps of the method in the form of an algorithm (Figure 4 and Figure 5).

It should be emphasized that this algorithm is a law of the functioning of each node of a computer network, but at each moment of time they are at different stages of its execution. Its important feature in the implementation of dynamic transfer of control between the central modules of the CSMM system of network nodes is based on the rule of continuity of network operation and the prevention of leaks of confidential information.

In general, the process of operating a computer network based on the dynamic control transfer method is divided into several stages - the network startup process, the main stage, and the failure recovery stage.

The main stage is a cyclic movement along the trajectory, which includes the vertices of the state graph (Figure 6) 3 – 4 – 5 – 6 - 3 … . Here, the third vertex of the graph corresponds to the standby mode. This is the normal mode in which the network node is most of the time, performing the functions assigned to it. Its central control module CSMM is subordinate to the node CSMM module, which currently manages the security system of the entire network and provides local control of the node security system in accordance with the security policies implemented by the central module.

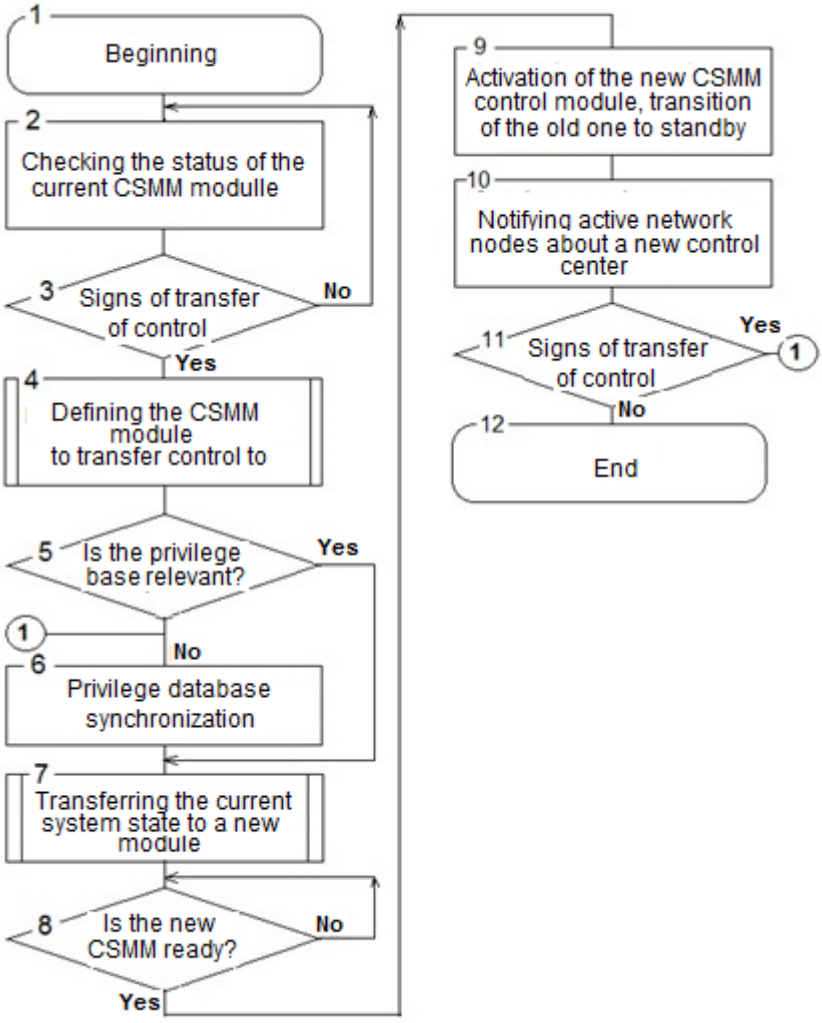

Figure 4. Algorithm for transferring control between computer network nodes.

From this state, the network node can enter the transition mode (vertex 4 Figure 4). In this mode, the node's CSMM module has received a control token and is in the process of assuming control of the network OS security system - it receives information about the system state, active sessions and the privilege base, checks the consistency of data between nodes, which corresponds to the execution of the algorithm (operators 6-8 Figure 4).

From the state of the transition mode, the node, during the normal course of events, transitions to a state in which it performs the functions of the central module CSMM (vertex 5 Figure 5), in another way - it controls the security system of the network OS, which corresponds to the execution of the algorithm (operators 9 - 11 Figure 5). The node is in this state for the period for which it is assigned to perform the functions of the central module of the network security system control, or until the onset of an emergency situation (edge 5-7 Figure 6). If, during the execution of the algorithm (operator 11 Figure 3), at least one sign of transfer of control is detected (expiration of the period, intervention of the system administrator), then the node transitions to the reserve state (vertex 6 Figure 4). In this mode, it is in a state of readiness to return to performing the functions of the central module CSMM.

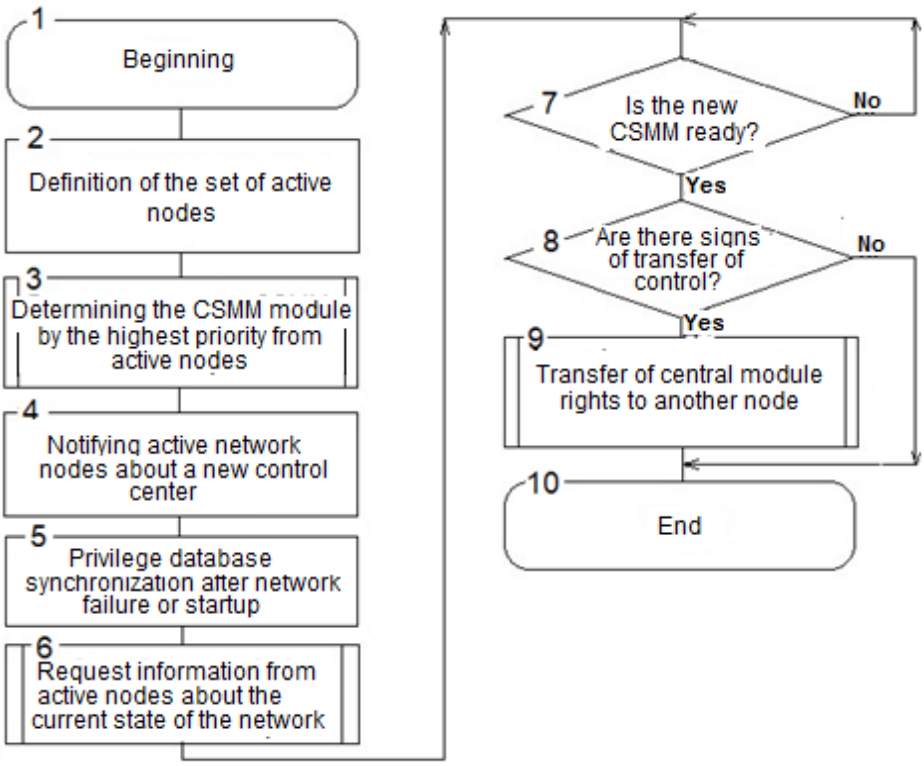

Figure 5. Algorithm for establishing a control center at startup or during network failures.

Це This means that the main privilege base of this node is constantly maintained in an up-to-date state, which allows for the fastest possible replacement of the current central node if necessary in the event of an abnormal development of events (edge 6-5 Figure 6). In the case of a normal development of events, after the next iteration of the dynamic change of the control center, the node will return to the standby mode (vertex 3 Figure 6), closing the main sequence of normal states in which a computer network node can be.

In the event of an emergency situation of a local failure, any node, regardless of its current state, will transition to the emergency state (vertex 7 Figure 6). In this case, it will be excluded from the number of active nodes in the network. From this state, according to the state graph (Figure 6), the node transitions to the unavailability state (vertex 1 Figure 6).

In this state, after eliminating the cause of the failure, it waits for a reconnection to the network. When connecting to the network, it automatically enters the synchronization state (vertex 2 Figure 6). In this state, the local privilege and policy database is updated to the current state, and consistency with other nodes is checked. The synchronization processes are completed by including the node in the active list, and the node itself enters the waiting state (vertex 3 Figure 6).

When starting the network or in case of large-scale failures, the network operation is restored according to the algorithm shown in figure 5. It takes into account situations that may arise when the network operation is restored after a failure. Since not all nodes restore their operation in the network at the same time when starting the network, the first node that will take control will be determined as the node with the highest priority among the nodes that are currently active (operator 2 Figure 5).

The node thus determined, whose CSMM module will perform the function of the central module of the OS security system, generates a control token and sends it to all other active nodes of the network that it is the control center of the system (operator 4 Figure 5).

In the subsequent steps, the algorithm involves synchronizing the privilege database with the local databases of active nodes in order to recreate the current main privilege database (operator 5 Figure 5). The next step of the algorithm is to collect information about session opening and other information about the network status (operator 6 Figure 5). In the future, the algorithm of the central control module when starting the network after a failure coincides with the algorithm shown in figure 5.

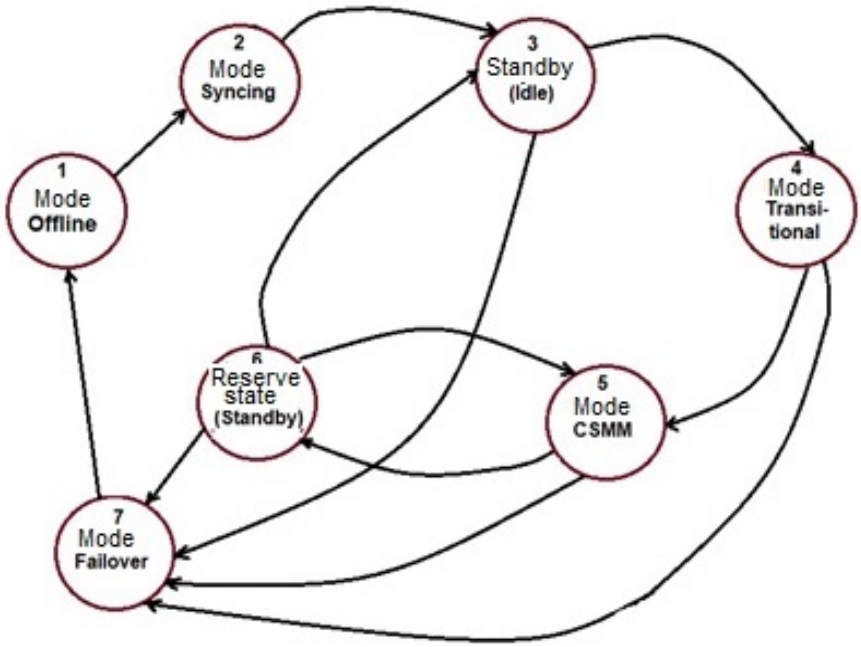

Figure 6. State graph of a network node.

The main sequence of states of an active node in the network is described by transitions between the vertices of the state graph 3-4-5-6-3-... (Figure 6). Thus, each node has four main states. To simplify understanding, vertices 4-5 of the graph can be conditionally combined into one. Then the main sequence of transitions of the graph (Figure 6) for the node will be limited to three states:

- Standby mode $CSMM_{current\ mode}$;
- Network Control Center Mode $CSMM_{control\ center}$;
- Standby module mode $CSMM_{Reserv}$.

Other states of the transition graph have their own specialization, which lies outside the main sequence. Thus, the states corresponding to vertices 1 and 2 of the graph (Figure 6) serve to bring the node to the main trajectory. Vertex 7 corresponds to the emergency state of the node, which always occurs in a physical system. Exit from this state is possible through vertices 1 and 2 of the graph.

4.4. Determining the effectiveness of the dynamic control transfer method

The partially centralized OS security system is based on the application of the proposed method of dynamic control transfer between computer network nodes. To assess its effectiveness, we will use a set of mathematical and probabilistic models that take into account the following indicators:

- the probability of leakage of confidential information, which takes into account the effectiveness of the protective mechanisms of the network node's OS and the vulnerability of the computer system P_{Leak};
- threat detection time $T_{AttackDetection}$;
- security transaction processing time $T_{\text{Performance}}$.

Taking into account the above, the formula for the efficiency of a computer network, in which all three parameters will be combined, taking into account their weight, will take the form:

$$E_{System} = w_1 \times (1 - P_{Leak}) + w_2 \times (\frac{1}{T_{AttackDetection}}) + w_2 \times (\frac{1}{T_{Perfomance}}) \quad (10)$$

where: w_1, w_2, w_3 - weighting factors of efficiency parameters. P_{Leak} - the likelihood of leakage of confidential information. $T_{AttackDetection}$ - threat detection time. $T_{Perfomance}$ - network control node performance.

The expression of formula (10) corresponds to all the relations for the centralized OS security system, so let us rewrite it taking into account the notations adopted for it:

$$E_{System,centralize} = w_1 * (1 - P_{Leak,centralize}) + w_2 * \frac{1}{T_{AttackDetection,centralize}} + \\ + w_3 * \frac{1}{T_{Perfomance,centralize}} \quad (11)$$

Since the operation of a partially centralized system is based on the use of a method of dynamic transfer of control between network nodes, this fact should be reflected in its efficiency formula. This method reduces the vulnerability of the OS security system, but at the same time, it also reduces to some extent the performance of the system as a whole. Dynamic change of the control center requires time to synchronize the privilege bases. Let us denote it as $T_{\text{Synchronization, partially}}$. Taking into account this parameter, the efficiency formula for a partially centralized system will take the form:

$$E_{System,partyally} = w_1 * \left(1 - P_{Leak,partyally}\right) + w_2 * \frac{1}{T_{AttackDetection,partially}} + \\ + w_3 * \frac{1}{T_{Perfomance,partially} + T_{Synchronization,partially}} \quad (12)$$

Now let's compare the performance indicators of partially centralized and centralized OS security systems:

$$E_{System,partyally} - E_{System,centralize} > 0 \quad (13)$$

Thus, if expression (13) has a value greater than zero, this will mean that the method of dynamic transfer of the control center between nodes provides greater system efficiency compared to a centralized security system.

5. Experiments

5.1. Setting up experiments

In order to determine the effectiveness of the dynamic control transfer method, several experiments were conducted. For this purpose, a test environment was deployed based on the use of a virtual computer network, which includes virtual nodes VM01 - VM04, which served as target machines during the experiments (Figure 7). In addition to them, the network includes an attacking machine VM00, with the help of whose software various types of attacks on possible ways of leaking confidential information will be simulated. The attacking machine is equipped with the Kali Linux OS and the Metasploit Framework attack simulator. The target machines are controlled by the FreeBSD 13.1 OS with a centralized security system.

All virtual machines run under the control of the Virtual Box hypervisor. With its help, virtual machines are combined into a separate local virtual network necessary for conducting experiments.

The purpose of the experiment is to obtain data on the comprehensive stability of protective mechanisms under the influence of threats acting on possible information leakage paths that take into account such parameters as the probability of information leakage, the time of detection of a malicious software attack, the performance of the central network node and the time of control transfer between network nodes. The experiment includes simulation of computer attacks of various types directed on all possible paths of confidential information leakage of all nodes of the virtual network.

To simulate various types of attacks, the Metasploit Framework simulator of the attacking machine VM00 (Figure 7) was used. Under its control, a developed script file was launched, which contains instructions in the Ruby language. The script file algorithm includes cyclic simulation of attacks along all main leakage paths across all network nodes in the address range

192.168.10.12 - 15. At the time of the script file operation, the /tmp/k1/test.txt files were opened on all target machines, simulating resources with confidential information. The results of the experiment are information about the number of successful and unsuccessful attacks on the corresponding nodes of the virtual computer network.

In addition to the above information on the side of the attacking machine VM00, the Result_atack.txt file recorded the start data of each attack, the IP address of the target machine, and the type of attack. If the security system of the corresponding target machine, to which the attack vector was directed, detects a threat, then after blocking it, it will make entries in its system logs security.log and audit.log upon this event. The difference between the start time of the simulation recorded in the Result_atack.txt file of the attacking machine and the time of its detection recorded in the security.log log of the corresponding target machine will correspond to the time of threat detection.

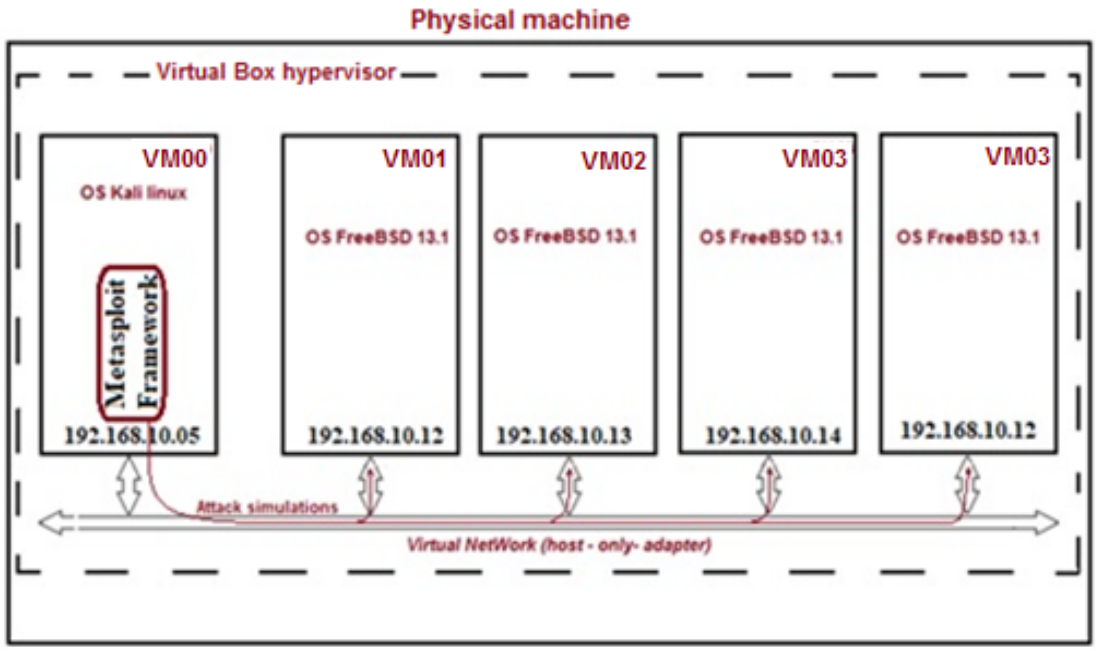

Figure 7. Scheme of the experimental setup for determining the effectiveness of the method of dynamic transfer of control of network nodes**.**

The results of comparing data from the Result_atack.txt file of the attacking machine VM00 (Figure 8) and the security.log files (Figure 9) of the corresponding target machines VM01-VM04 are presented in Table 1 in an averaged and normalized form. Since the attacking machine and the target machines are in the same time space (one time scale) of the physical machine, the error in measuring the duration of the attack detection period will be minimal. The situation when the record of the start of the simulation recorded in the Result_atack.txt file does not have an associated record in the security.log logs of all target machines means that a successful attack has occurred. An example of the comparison is highlighted in red (Figure 8, Figure 9).

Another stage of experiments was aimed at obtaining data that would allow determining the performance of the control node of the $T_{Performance}$

$T_{\text{Request at startup}}$ and $T_{\text{Shutdown prompt}}$, respectively. Then the performance of the central node of the network will be determined as:

$$T_{Performance} = \frac{T_{Shutdownpromt}}{T_{Reguestat\ starup}}. \quad (14)$$

The processed experimental data are given in Table 1.

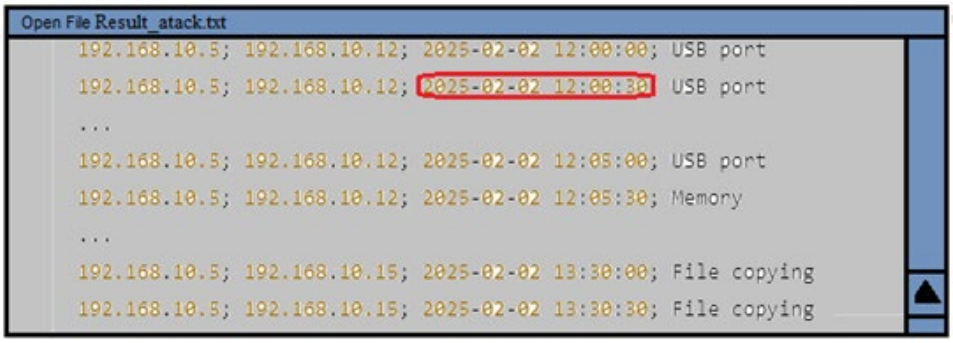

Figure 8. The contents of the Result_atack.txt file generated by the Metasploit Framework simulator script file.

Figure 9. Content of the security.log file of the target machine VM1.

Table 1.
Averaged experimental data (absolute value\normalized value)

Система безпеки ОС	P_{Leak}	$T_{\text{AttackDetection,A}}$ сек	$T_{\text{Shutdown prompt}}$ сек	$T_{\text{Request at startup}}$ сек	$T_{\text{Synchronization}}$ сек
Централізована	0,0133	5,2 (0,4)	0.126 (0,887)	0,142	-
Частково-централізована	0,0115	4,925 (0,358)	0,117 (0,921)	0,127	2.118 (0,688)

5.2. Calculations of the effectiveness of centralized and partially centralized OS security systems

Using formula (14) and experimental data, we will calculate the performance of the central node for a centralized system, for which we will substitute the corresponding values from Table 1:

$$T_{Perfomane,Centralize} = \frac{T_{Shutdown\ promt}}{T_{Reguestat\ startup}} = \frac{0{,}126}{0{,}142} = 0{,}887 \quad (15)$$

The performance of the central node for a partially centralized system is calculated in the same way:

$$T_{Perfomane,Partially} = \frac{T_{Shutdown\ promt}}{T_{Reguestat\ startup}} = \frac{0{,}117}{0{,}127} = 0{,}921 \quad (16)$$

To calculate the effectiveness of the security system, we will use formula (10). To do this, we will determine the values of the weight coefficients $w_1 - w_3$ present in this formula. To determine their weight, we will use the linear hierarchy method, within which we will apply an importance scale from one to nine: 1 - equal importance; 3 - weak advantage of one factor over another; 5 - strong advantage; 7 - very strong advantage; 9 - absolute advantage. Guided by this scale, we build a matrix of pairwise comparisons:

Guided by this scale, we build a matrix of pairwise comparisons:

$M(w) = \begin{bmatrix} 1 & 3 & 5 \\ 1/3 & 1 & 3 \\ 1/5 & 1/3 & 1 \end{bmatrix}$. Here, the columns and rows of the matrix are weight coefficients $w_1 - w_3$, and its elements are the result of pairwise comparison. After performing the following steps, we obtain: w_1=0.633; w_2=0.261; w_3=0.106 . Now we can directly calculate the effectiveness of the security system, for which we substitute the values into formula (11) for the centralized OS security system:

$$E_{System,centralize} = w_1 * (1 - P_{Leak,centralize}) + w_2 * \frac{1}{T_{AttackDetection,centralize}} +$$
$$+ w_3 * \frac{1}{T_{Perfomance,centralize}} =$$
$$= 0{,}633 * (1 - 0{,}0133) + 0{,}261 * \frac{1}{0{,}4} + 0{,}106 * \frac{1}{0{,}887} = 1{,}397$$

Similarly, for a partially centralized OS security system, we use formula (12):

$$E_{System,partyally} = w_1 * (1 - P_{Leak,partyally}) + w_2 * \frac{1}{T_{AttackDetection,partially}} +$$
$$+ w_3 * \frac{1}{T_{Perfomance,partially} + T_{Synchronization,partially}} =$$
$$= 0{,}633 * (1 - 0{,}0115) + 0{,}261 * \frac{1}{0{,}308} + 0{,}106 * \frac{1}{0{,}921 + 0{,}688} = 1{,}5389$$

Let us use inequality (3.32) to compare the calculated efficiency values of two variants of OS security systems:

$$E_{System,partyally - E_{System,centralize}} = 1{,}5389 - 1{,}397 = 0{,}142$$

Let's display the result in percentages:

$$\left(1 - \frac{E_{System,centralize}}{E_{System,partyally}}\right) * 100\% = \left(1 - \frac{1{,}397}{1{,}539}\right) * 100\% = 9{,}2\%$$

The calculation shows that the effectiveness of a partially centralized OS security system, despite its complexity, is significantly higher. In turn, the higher effectiveness of such a security system means greater resistance to SPP, which in turn provides lower vulnerability to the

leakage of confidential information. As can be seen from the graph (Figure 10), the effectiveness of a partially centralized OS security system is 9.2 percent higher than that of a centralized system.

6. Conclusions

The proposed method of dynamic control transfer between network nodes with support for replication of the main privilege base for a partially centralized OS security system made it possible to preserve the main advantages of a centralized security system, especially with regard to the system's resistance to information leakage, and at the same time get rid of its biggest drawback - a bottleneck in the security system in the form of a central security module, the failure of which leads to loss of control over the entire system.

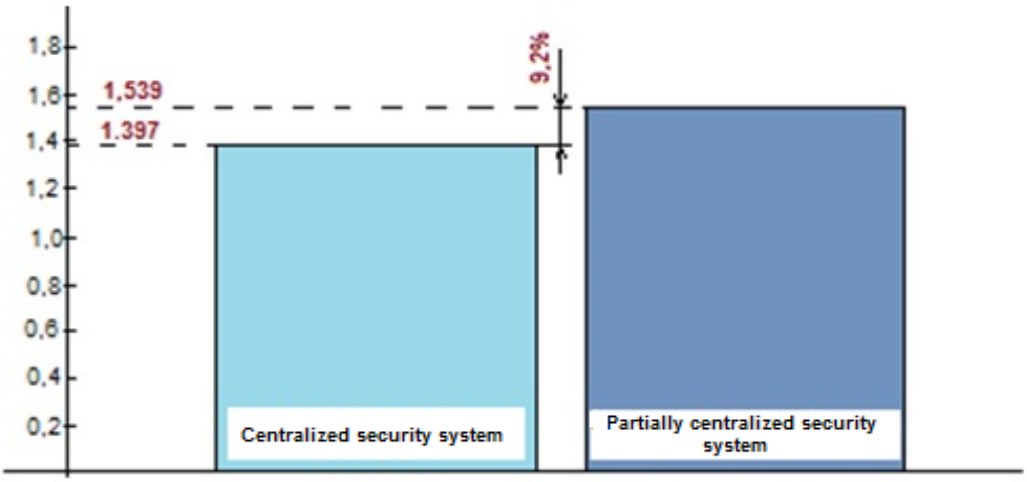

Figure 10. OS security systems effectiveness chart.

The efficiency of the OS security system using this method of computer network management is at least 9.2 percent higher than a purely centralized approach.

Another advantage of this method of dynamic control transfer is that periodic random changes to the network control node with a high probability make the results of the attacker's network reconnaissance irrelevant, thereby disrupting attacks on the network before they even begin. This is explained by the fact that the network reconnaissance time is much longer than the period of changing the network control center, which gives it the opportunity to evade the attack in this way.

Declaration on Generative AI

The authors did not use artificial intelligence tools.

References

[1] Security and Privacy Controls for Information Systems and Organizations, National Institute of Standards and Technology Walter Copan, NIST Director and Under Secretary

of Commerce for Standards and Technolog, Publication 800-53 Revision 5. doi:10.6028/NIST.SP.800-53r5

[2] S. Leventopoulos, K. Pipyros, D. Gritzalis, Retaliating against cyber-attacks: a decision-taking framework for policy-makers and enforcers of international and cybersecurity law, Int. Cybersecur, Law Rev., 5 (2024) 237–262. doi:10.1365/s43439-024-00113-5

[3] N. D. Viet, D.D. Quan, Proposing A New Approach for Detecting Malware Based on the Event Analysis Technique, International Journal of Innovative Technology and Exploring Engineering, 8 (2023) 21–27. doi:10.35940/ijitee.h9651.0712823

[4] Lysenko S., Bobrovnikova K., Savenko O., Kryshchuk A. BotGRABBER: SVM-Based Self-Adaptive System for the Network Resilience Against the Botnets' Cyberattacks, Communications in Computer and Information Science, 1039 (2019) 127-143.

[5] H. Lu, Y. Zhao, Y. Song, Y. Yang, G. He, H. Yu, Y. Ren, A transfer learning-based intrusion detection system for zero-day attack in communication-based train control system, Cluster Comput, (2024). doi:10.1007/s10586-024-04376-9

[6] D. Srilatha, G.K. Shyam, Cloud-based intrusion detection using kernel fuzzy clustering and optimal type-2 fuzzy neural network, Cluster Comput, 24 (2021) 2657–2672. doi:10.1007/s10586-021-03281-9

[7] B. Hajimirzaei, N.J. Navimipour, Intrusion detection for cloud computing using neural networks and artificial bee colony optimization algorithm, ICT Express, 5 (2019) 56–59. doi:10.1016/j.icte.2018.01.014

[8] F. Lang, H. Li, W. Wang, J. Lin, F. Zhang, W. Pan, Q. Wang, E-SGX: Effective Cache Side-Channel Protection for Intel SGX on Untrusted OS, In: Y. Wu, M. Yung, (Eds), Information Security and Cryptology, Lecture Notes in Computer Science, volume 12612, Springer, Cham, Switzerland AG, 2021, pp. 221–243. doi:10.1007/978-3-030-71852-7_15

[9] L. An, J. Qiu, H. Zhang, C. Liu, Design of distributed network intrusion prevention system based on Spark and P2DR models, Cluster Comput, (2024) 12-13. doi:10.1007/s10586-024-04487-3

[10] A.K. Jain, H. Shukla, D. Goel, A comprehensive survey on DDoS detection, mitigation, and defense strategies in software-defined networks, Cluster Comput, (2024). doi:10.1007/s10586-024-04596-z

[11] M. Mushtaq, M.M. Yousaf, M.K. Bhatti, V. Lapotre, Gogniat, G. The Kingsguard OS-level mitigation against cache side-channel attacks using runtime detection. Annals of Telecommun, 77 (2022) 731–747. doi:10.1007/s12243-021-00906-3

[12] M. Soltani, K. Khajavi, M. Jafari Siavoshani, A. Hossein Jahangir, A multi-agent adaptive deep learning framework for online intrusion detection, Cybersecurity, 7(9) (2024). doi:10.1186/s42400-023-00199-0

[13] S. Wang, Y. Jing, K. Wang, X. Wang, Multi-Agent Systems for Collaborative Inference Based on Deep Policy Q-Inference Network, J Grid Computing, 22(38) (2024). doi:10.1007/s10723-024-09750-w

[14] O. Karataiev, Towards multi-agent platform development. Computer Systems and Information Technologies, 4 (2024) 98–106. doi: 10.31891/csit-2024-4-12

[15] N. Pradeep, Vaishnavi, V. Varsha, K. Vivek, The Application of Artificial Intelligence in Computer Network Technology, International Journal of Advanced Research in Science Communication and Technology, (2024) 588–592. doi:10.48175/IJARSCT-22770

[16] O. Pomorova, O. Savenko, S. Lysenko, A. Kryshchuk, Multi-Agent Based Approach for Botnet Detection in a Corporate Area Network Using Fuzzy Logic, in: Communications in Computer and Information Science 370 (2013) 243-254.
[17] G. Srividhya, R. Nagarajan, A novel bidirectional LSTM model for network intrusion detection in SDN-IoT network, Computing, 106 (2024) 2613–2642 doi:10.1007/s00607-024-01295-w.
[18] N. Temene, A. Naoum, C. Sergiou, C. Georgiou, V. Vassiliou, A fault tolerant node placement algorithm for WSNs and IoT networks, Computer Networks, 254 (2024) 110835. doi:10.1016/j.comnet.2024.110835.
[19] Y. Zhang, L. Wang, D. Liu, Y. Su, Y. Zhu, X. Zhang, Design of Information Security Protection System for Cloud Business System. In: R. Kountchev, S. Patnaik, K. Nakamatsu, R, Kountcheva (Eds), Proceedings of International Conference on Artificial Intelligence and Communication Technologies (ICAICT 2023), volume 369, Smart Innovation, Systems and Technologies, Singapore, 2024, pp. 105–122. doi:10.1007/978-981-99-6956-2_10.
[20] T. Yamauchi, Y. Akao, R. Yoshitani, Y. Nakamura, M. Hashimoto, Additional kernel observer: privilege escalation attack prevention mechanism focusing on system call privilege changes, Int. J. Inf. Secur., 20 (2021) 461-473. doi:10.1007/s10207-020-00514-7.
[21] O. Savenko, K. Bobrovnikova, S. Lysenko, DDoS Botnet Detection Technique Based on the Use of the Semi-Supervised Fuzzy c-Means Clustering, In: V. Ermolayev, V. Yakovyna, V. Kharchenko, V. Kobets, H. Kravtsov (Eds), Proceedings of the 14th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer, volume 2104, Kiyv Ukraine, 2018, pp. 688-695.
[22] L. Ying, W. Jiuqi, F. Ziyu, F. Yufan, L. Xiaodong, DiSAuth: A DNS-based secure authorization framework for protecting data decoupled from applications, Computer Networks, 254 (2024) 110774. doi:10.1016/j.comnet.2024.110774.
[23] Q. Zhou, X. Jia, J. Chen, Q. Huang, H. Du, LightArmor: A Lightweight Trusted Operating System Isolation Approach for Mobile Systems, In: N. Pitropakis, S. Katsikas, S. Furnell, K. Markantonakis, (Eds), ICT Systems Security and Privacy Protection. SEC 2024. IFIP Advances in Information and Communication Technology, volume 710, Springer, Cham., Switzerland, 2024, pp. 206–220. doi:10.1007/978-3-031-65175-5_15.
[24] S. Lysenko, O. Savenko, K. Bobrovnikova, A. Kryshchuk, B. Savenko, Information technology for botnets detection based on their behaviour in the corporate area network, Communications in Computer and Information Science, 718 (2017), 166–181.
[25] T. Gueye, Y. Wang, M. Rehman, R.T. Mushtaq, S. Zahoor, A novel method to detect cyber-attacks in IoT/IIoT devices on the modbus protocol using deep learning, Cluster Comput 26 (2023) 2947–2973. doi:10.1007/s10586-023-04028-4.
[26] O. Kehret, A. Walz, A. Sikora, Integration of hardware security modules into a deeply embedded tls stacs, International Journal of Computing, 15(1) (2016) 22-30. doi:10.47839/ijc.15.1.827.