CEUR-WS.org/Vol-3899/paper20.pdf

A model of a centralized security system, as an information technology for the synthesis of an OS architecture protected against the leakage of confidential information*

Yuriy Stetsyuk [1,*,†], Mykola Stetsyuk [1,†], Kyrylo Voznyi [1,†], Vadym Paiuk [1,†], Miroslav Kvassay [2,†]

[1] *Khmelnitsky National University, Khmelnitsky, Instytutska street 11, 29016, Ukraine*
[2] *Zilina University, Univerzitná 8215, 010 26 Žilina, Slovakia*

Abstract

The construction of subsystem models of a decentralized and centralized OS security system designed to work as part of a protected system for processing confidential information in a multi-machine network computer system is considered. An analysis of publications related to the construction of OSs resistant to the leakage of confidential information and, in general, the protection of the information processed in them, was performed. Their protection mechanisms and methods of improving the efficiency of their work within the framework of OS security systems are considered. The principles of building decentralized and centralized OS security systems and the principles of organizing the operation of their security mechanisms are considered. A comparative analysis of the effectiveness of centralized and decentralized security systems was performed. A graphical model of a centralized security system for construction is presented. The key aspect, according to the adopted approach, is to find a balanced OS security subsystem architecture that can effectively ensure the OS's resilience to information leakage and its information protection in general.

Keywords

operating system, information protection, centralized system, security mechanisms

1. Introduction

The development of information technologies has led to the fact that almost all aspects of human activity have become critically dependent on various achievements in computers, computing systems and their mathematical support. The successful operation of computer systems, in turn, is a hostage to their qualities, such as reliability, fault tolerance and, most importantly, information security.

Their work is based on operating systems of different types and purposes. The fundamental nature of the OS is to abstract the hardware from the user of the information system, allowing the user not to experience all the complexity of the multifaceted hardware platform of a modern computer system, allowing him to focus on solving his application task. OS, managing the work of the computer system, solves very important system-wide tasks related to the distribution of hardware resources, multitasking, productivity and, most importantly, ensuring information security.

AdvAIT-2024: 1st International Workshop on Advanced Applied Information Technologies, December 5, 2024, Khmelnytskyi, Ukraine - Zilina, Slovakia

[*] Corresponding author.

[†] These authors contributed equally.

yuriy.stetsuk@khmnu.edu.ua (Y. Stetsyuk); mykola.stetsiuk@khmnu.edu.ua (M. Stetsyuk); k.vozniy@gmail.com (K. Voznyi); vadympaiuk@gmail.com (V. Paiuk); miroslav.kvassay@fri.uniza.sk (M. Kvassay)

0000-0003-0312-2276 (Y. Stetsyuk); 0000-0003-3875-0416 (M. Stetsyuk); 0009-0007-2545-565 (K. Voznyi); 0000-0002-9969-8239 (V. Paiuk); 0000-0002-6450-5417 (M. Kvassay);

© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

2. Analysis of known solutions

It immediately became clear that only the anticipatory development of OS protective mechanisms will allow them to provide their functionality without significant information losses. The search for models of OS architectures resistant to various types of destruction has become a constant process. An important stage in the formation of the OS was the presentation of several abstract models of protection systems, which became their fundamental bases. One of the first was the 1977 Biba model [1]. According to it, all subjects and objects of some system are previously divided into several levels of access, with the imposition of restrictions on their interaction [2].

The next step in the development of abstract models of OS security systems was the 1982 Goguen-Meseguer model, based on the theory of automata [3]. In 1986, the Sutherland model of protection was presented, which emphasizes the interaction of subjects and information flows. According to it, as in the previous model, the system can only be in predefined states [4]. An important role in the theory of information protection is played by the Clark-Wilson (Clark-Wilson) protection model of 1987 [5, 6], which is based on the use of transactions and on the balanced granting of access rights of subjects to objects.

In addition to purely abstract models of building OS protection systems, there are many practical developments [14] embodied in physical OSes. The process of development of operating systems, as a class of software, began with universal OSes and led to the separation of subclasses in them based on the principle of increasing the importance of some operational parameters. Thus, the advantage of security parameters led to the emergence of a subclass of protected OSes, which must meet certain standards and use specialized mechanisms to counter threats [9]. Today, this is not only a purely scientific or technical concept, but also a legal one. Requirements for such systems are defined in national standards [7, 8]. Thus, the US standard developed by the National Institute of Standards and Technologies (NIST) defines a protected OS in the context of requirements for information systems at the federal level [10].

Protected OS, unlike universal ones, includes more effective, mandatory protection mechanisms against various threats and a wider range of them. The main ones are: mandatory access control (MAC), minimization of privileges, auditing and security monitoring, OS kernel security, process isolation, memory protection, encryption, backup, network control, and data integrity and authenticity control. Many scientific works have been devoted to the development of methods for increasing their efficiency and removing various kinds of vulnerabilities [11, 12].

Minimizing privileges allows you to significantly reduce the number of potentially vulnerable system components [15]. New methods of authentication can use, in addition to a password, an electronic key, a smart card, and biometric data [17].

Auditing plays a critical role in ensuring data security and system integrity. Its improvement in terms of registration and analysis of actions of users who have access to system resources is considered in [15, 16]. New mechanisms of combating ZPP are proposed [24].

As you know, the kernel is the central part of the OS and serves as an interface between the OS hardware resources and applications. Improvement of the application switching mechanism, loading/unloading of their contexts is considered in [18]. The organization of input-output processes, file system operation, processing of processor interruptions, process dispatching is given in [19].

In [20], a process isolation method is proposed, which is a fundamental approach to ensuring the safety, stability, and efficiency of protected OSes, which is based on the protection of the OS against malicious or incorrect actions of user programs.

In [21], issues of centralization of encryption and the use of new cryptography tools for memory protection are considered. In protected OS, the presence of a backup mechanism is mandatory, as it significantly increases the OS's resistance to all types of failures. Implementation methods are given in [22].

The mechanism of control of incoming and outgoing traffic according to the specified rules is presented in [23, 25]. This is quite an important point in view of the significant increase in the number of services that actively use networks. Mechanisms for combating BOTNET are proposed [26, 27, 28].

The analysis of the state of use of the mechanisms for ensuring the stability of the OS against the leakage of confidential information and the protection of information in the OS in general covered almost all levels of system construction - from their hardware platform to the OS core. He showed that despite the presentation of a large number of fairly effective methods of ensuring OS resistance to the leakage of confidential information and, in general, information protection, their application is restrained by the ever-increasing complexity of implementation and the limitation of the comprehensive application of protective mechanisms at all levels of the OS architecture.

3. Formulation of the problem

The continuous development of computer information technologies has led to the need for widespread use of protected OSes designed for processing confidential data, which requires a new approach in building their security subsystems. They must provide their functionality with simultaneously high levels of fault tolerance, survivability and protection of the information processed in them.

The task is to find such a model of the architecture of the OS protection subsystem, which would integrate the most effective mechanisms for ensuring its resistance to leaks of confidential information, information protection in general. In this way, the scientific problem solved can be characterized as relevant and as one that has a fairly wide practical application.

4. The main part

4.1. Architecture model of the decentralized OS protection subsystem

The model of a decentralized OS security system is based on the principle of distributing security functions between different components or segments of the system. In such an architecture, each component independently manages its own security policies, protection mechanisms, monitoring, and privilege management.

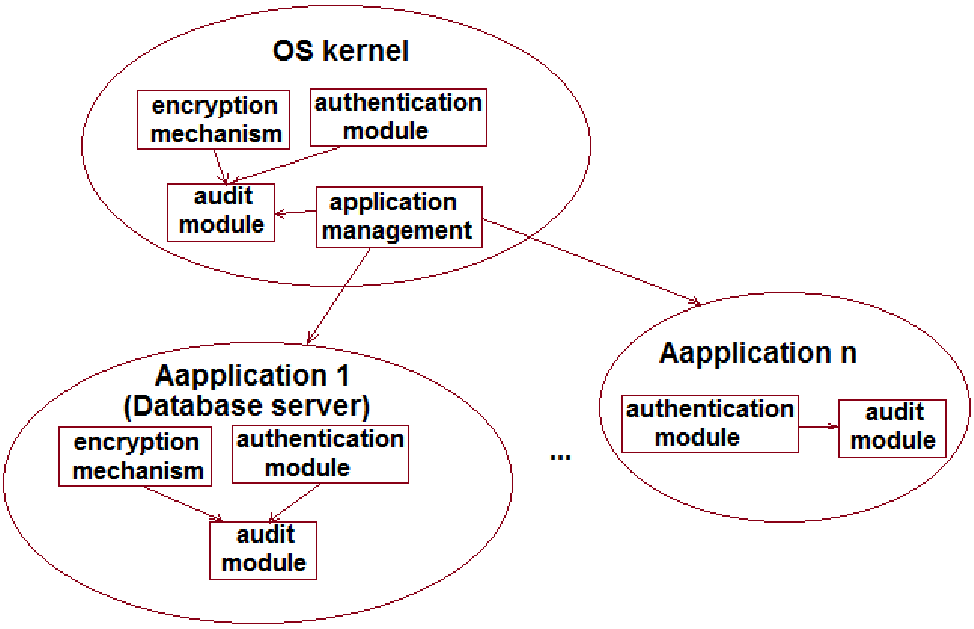


Figure 1: Model of the decentralized OS security system.

Decentralized OS security systems are characterized by the presence of local access control systems, where each system component (database server, application, or process) has its own access policies implemented through local access control lists (ACLs).

Provision of access rights is carried out on the basis of roles or attributes using local authentication mechanisms. An example can be various types of servers operating under OS management and file systems of the same OS that use their own access policies for each user (Figure 1).

The main feature of the decentralized model is the absence of a single security management center. The situation when each component is responsible for its own protection leads to an increase in its vulnerabilities: security management is complicated, the probability of configuration errors is increased, and coordination between protection mechanisms is limited.

4.2. Architecture model of the centralized OS protection subsystem

To overcome the above problems, we will introduce the central security management module CSMM (central security management module) into the OS architecture, where we will concentrate OS security management. Its functions include defining security policies, monitoring events in the system, responding to incidents, and auditing. The model of such organization of the architecture of the OS security system is presented in Figure 2. In order for the central CSMM security module to receive information about the state of security from all important nodes of the OS, we will introduce into their architecture the peripheral security modules PSM1 - PSMn, which are installed on each node of the OS.

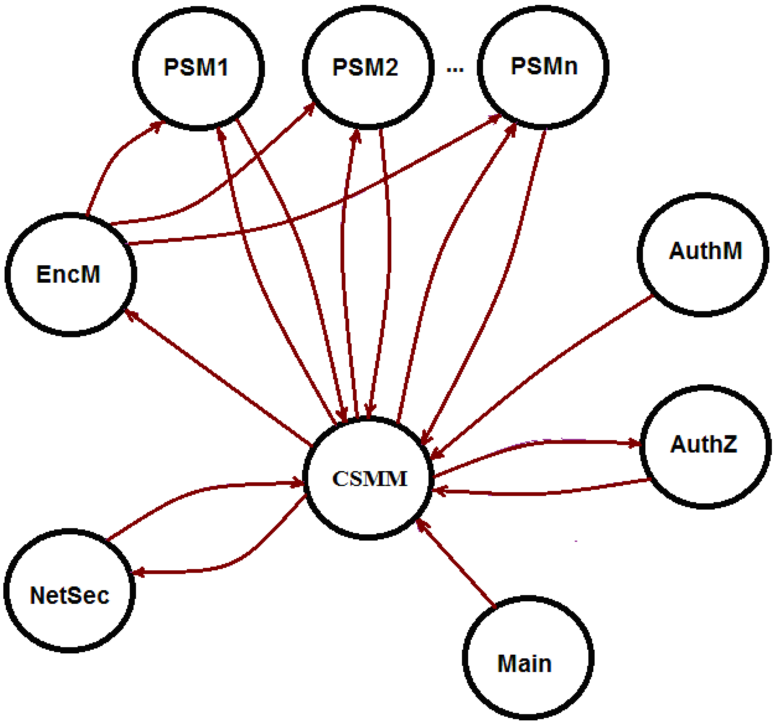


Figure 2: Graph model of the centralized security system of the abstract OS.

In the given centralized OS security model presented in the form of a graph, the vertices correspond to the modules of the centralized OS security subsystem, and the directed edges of the graph indicate the corresponding interaction between the modules, determining the algorithm of its operation. Main interactions:

- CSMM sends security policies to peripheral security modules PSM1 - PSMn;

- PSM1 - PSMn modules send monitoring data and reports to the CSMM central control module.

The model can be extended by including other security mechanisms as needed.

4.3. Analysis of the effectiveness of architectures with regard to their resistance to leakage of confidential information

A single center for managing access and security policies allows you to ensure consistency of all access rules to confidential information. Policies can be easily applied to all users and resources, reducing the chance of configuration errors that could lead to information leakage. Centralized privilege management allows you to restrict access to confidential data based on roles, which minimizes the risks of unauthorized access.

Architectures based on the use of centralized OS security systems show high efficiency of all security mechanisms. Centralized management allows you to monitor all events in the system, which facilitates the detection of potential information leaks. A monitoring system (for example, MaxPatrol SIEM in Windows Server 2019) allows you to detect abnormal activity in real time and quickly respond to threats. Centralized event and audit logging provides a more complete picture of access to sensitive data. A single encryption standard allows you to ensure data protection at all levels of the system.

The redundancy mechanism within the framework of the centralized security system allows to ensure the protection of confidential data in case of failures or attacks of malicious software more fully than in the decentralized system.

A centralized OS security system allows for a more prompt response to malware attacks and lower administration costs.

The centralized OS security system allows for a more prompt response to malware attacks and lower administration costs. However, centralized systems are vulnerable to their central node. and centralized OS security systems, but it is always easier to protect a central node than several in decentralized systems.

Therefore, regardless of the mentioned vulnerability, the centralized security systems of the OS provide more effective protection of information in general and against leakage of confidential information in particular.

5. Experiments

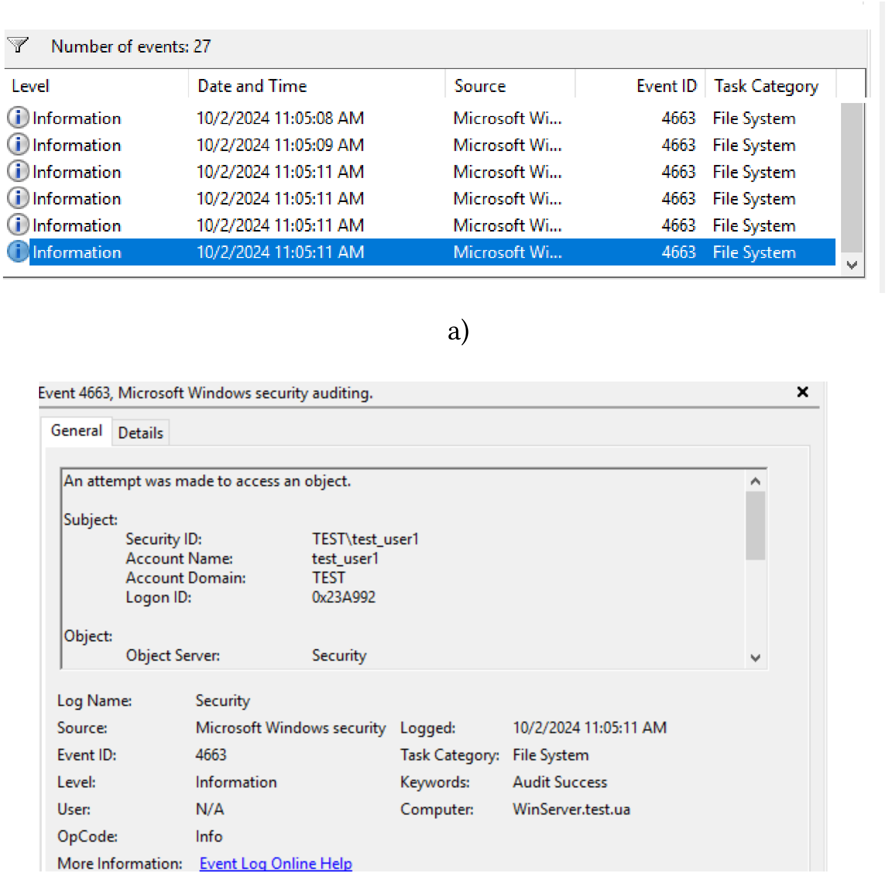
5.1. Access control testing

The purpose of the experiment: to test how effectively a centralized security system restricts access to resources based on policies set by administrators.

As a laboratory installation, we will use a virtual machine with a centralized security system. Setting up the OS for conducting the experiment consists in creating users test_user1 and test_user2 with different levels of access to resources in the OS, which will work from a computer with the domain name winserver.test.ua.

We set centralized access policies through Active Directory of the resource with confidential data 1111.txt.

During the experiment, users try to access the file with confidential data 1111.txt, to check the operation of the centralized security system.



Number of events: 27

Level	Date and Time	Source	Event ID	Task Category
Information	10/2/2024 11:05:08 AM	Microsoft Wi...	4663	File System
Information	10/2/2024 11:05:09 AM	Microsoft Wi...	4663	File System
Information	10/2/2024 11:05:11 AM	Microsoft Wi...	4663	File System
Information	10/2/2024 11:05:11 AM	Microsoft Wi...	4663	File System
Information	10/2/2024 11:05:11 AM	Microsoft Wi...	4663	File System
Information	10/2/2024 11:05:11 AM	Microsoft Wi...	4663	File System

a)

Event 4663, Microsoft Windows security auditing.

```
An attempt was made to access an object.

Subject:
        Security ID:            TEST\test_user1
        Account Name:           test_user1
        Account Domain:         TEST
        Logon ID:               0x23A992

Object:
        Object Server:          Security
```

```
Log Name:          Security
Source:            Microsoft Windows security   Logged:          10/2/2024 11:05:11 AM
Event ID:          4663                         Task Category:   File System
Level:             Information                  Keywords:        Audit Success
User:              N/A                          Computer:        WinServer.test.ua
OpCode:            Info
More Information:  Event Log Online Help
```

b)

Figure 3: The reaction of the OS security system to an attempt to access a file system object without the appropriate rights.

The result of the experiment. The user test_user1 with missing rights to the resource 1111.txt was blocked when he tried to access it, as a result of which an entry was made in the audit log dated 10.2.2024 11:05:11 "An attempt was made to access the object." (Figure 3 a)), and its details are given in figure. 3 b). This test confirmed the effectiveness of centralized access control policies.

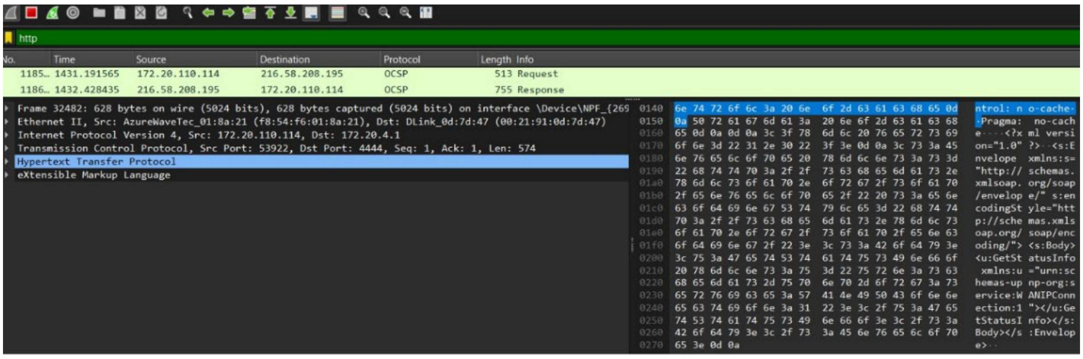
5.2. Data encryption testing

The purpose of the experiment: to check the reliability of the centralized encryption system, namely during transmission over the network.

1st stage. Let's try to intercept data by physically accessing network traffic. To do this, using the Wireshark network analyzer, with the encryption system turned off, we will view the content of the network data packet taken from the machine with IP 172.20.110.114. The result is shown in figure 4. As you can see, in such a situation, the data are available, which means that their protection is absent.

2nd stage. Enable the data encryption system in the OS using BitLocker and configure encryption policies for network connections (SSL/TLS). We will intercept the network packet from the same IP address.

The result is shown in figure 5. As can be seen from it, analysis of the packet data is impossible due to their encryption protection, which confirms the effectiveness of the centralized encryption system.

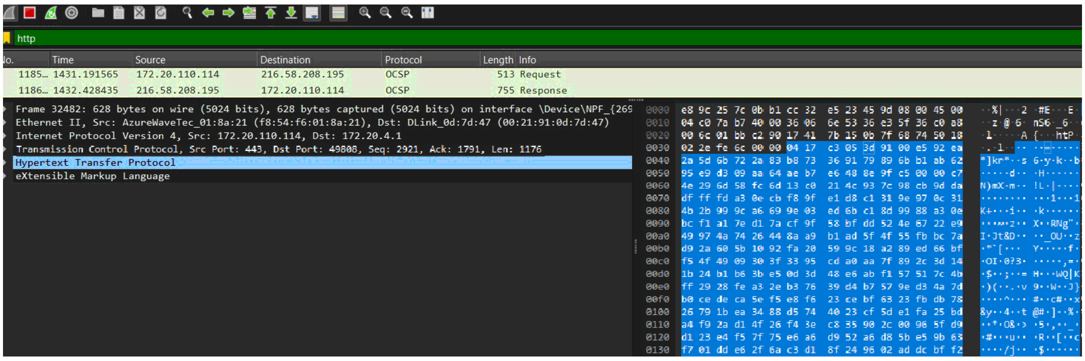


No.	Time	Source	Destination	Protocol	Length	Info
1185…	1431.191565	172.20.110.114	216.58.208.195	OCSP	513	Request
1186…	1432.428435	216.58.208.195	172.20.110.114	OCSP	755	Response

```
Frame 32482: 628 bytes on wire (5024 bits), 628 bytes captured (5024 bits) on interface \Device\NPF_{269
Ethernet II, Src: AzureWaveTec_01:8a:21 (f8:54:f6:01:8a:21), Dst: DLink_0d:7d:47 (00:21:91:0d:7d:47)
Internet Protocol Version 4, Src: 172.20.110.114, Dst: 172.20.4.1
Transmission Control Protocol, Src Port: 53922, Dst Port: 4444, Seq: 1, Ack: 1, Len: 574
Hypertext Transfer Protocol
eXtensible Markup Language
```

```
0140  6e 74 72 6f 6c 3a 20 6e  6f 2d 63 61 63 68 65 0d   ntrol: n o-cache·
0150  0a 50 72 61 67 6d 61 3a  20 6e 6f 2d 63 61 63 68   ·Pragma:  no-cach
0160  65 0d 0a 0d 0a 3c 3f 78  6d 6c 20 76 65 72 73 69   e····<?x ml versi
0170  6f 6e 3d 22 31 2e 30 22  3f 3e 0d 0a 3c 73 3a 45   on="1.0" ?>··<s:E
0180  6e 76 65 6c 6f 70 65 20  78 6d 6c 6e 73 3a 73 3d   nvelope  xmlns:s=
0190  22 68 74 74 70 3a 2f 2f  73 63 68 65 6d 61 73 2e   "http:// schemas.
01a0  78 6d 6c 73 6f 61 70 2e  6f 72 67 2f 73 6f 61 70   xmlsoap. org/soap
01b0  2f 65 6e 76 65 6c 6f 70  65 2f 22 20 73 3a 65 6e   /envelop e/" s:en
01c0  63 6f 64 69 6e 67 53 74  79 6c 65 3d 22 68 74 74   codingSt yle="htt
01d0  70 3a 2f 2f 73 63 68 65  6d 61 73 2e 78 6d 6c 73   p://sche mas.xmls
01e0  6f 61 70 2e 6f 72 67 2f  73 6f 61 70 2f 65 6e 63   oap.org/ soap/enc
01f0  6f 64 69 6e 67 2f 22 3e  3c 73 3a 42 6f 64 79 3e   oding/"> <s:Body>
0200  3c 75 3a 47 65 74 53 74  61 74 75 73 49 6e 66 6f   <u:GetSt atusInfo
0210  20 78 6d 6c 6e 73 3a 75  3d 22 75 72 6e 3a 73 63    xmlns:u ="urn:sc
0220  68 65 6d 61 73 2d 75 70  6e 70 2d 6f 72 67 3a 73   hemas-up np-org:s
0230  65 72 76 69 63 65 3a 57  41 4e 49 50 43 6f 6e 6e   ervice:W ANIPConn
0240  65 63 74 69 6f 6e 3a 31  22 3e 3c 2f 75 3a 47 65   ection:1 "></u:Ge
0250  74 53 74 61 74 75 73 49  6e 66 6f 3e 3c 2f 73 3a   tStatusI nfo></s:
0260  42 6f 64 79 3e 3c 2f 73  3a 45 6e 76 65 6c 6f 70   Body></s :Envelop
0270  65 3e 0d 0a                                        e>··
```

Figure 4: Network packet with the centralized encryption system disabled.



No.	Time	Source	Destination	Protocol	Length	Info
1185…	1431.191565	172.20.110.114	216.58.208.195	OCSP	513	Request
1186…	1432.428435	216.58.208.195	172.20.110.114	OCSP	755	Response

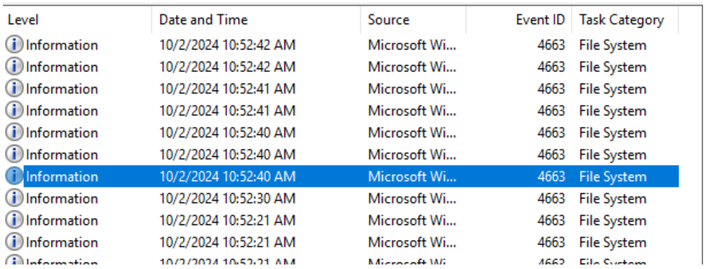
```
Frame 32482: 628 bytes on wire (5024 bits), 628 bytes captured (5024 bits) on interface \Device\NPF_{269
Ethernet II, Src: AzureWaveTec_01:8a:21 (f8:54:f6:01:8a:21), Dst: DLink_0d:7d:47 (00:21:91:0d:7d:47)
Internet Protocol Version 4, Src: 172.20.110.114, Dst: 172.20.4.1
Transmission Control Protocol, Src Port: 443, Dst Port: 49808, Seq: 2921, Ack: 1791, Len: 1176
Hypertext Transfer Protocol
eXtensible Markup Language
```

```
0000  e8 9c 25 7c 0b b1 cc 32  e5 23 45 9d 08 00 45 00   ··%|···2 ·#E···E
0010  04 c0 7a b7 40 00 36 06  6e 53 36 e3 5f 36 c0 a8   ··z·@·6· nS6·_6··
0020  00 6c 01 bb c2 90 17 41  7b 15 0b 7f 68 74 50 18   ·l·····A {···htP·
0030  02 2e fe 6c 00 00 04 17  c3 05 3d 91 00 e5 92 ea   ·.·l···· ··=·····
0040  2a 5d 6b 72 2a 83 b8 73  36 91 79 89 6b b1 ab 62   *]kr*··s 6·y·k··b
0050  95 e9 d3 09 aa 64 ae b7  e6 48 8e 9f c5 00 00 c7   ·····d·· ·H······
0060  4e 29 6d 58 fc 6d 13 c0  21 4c 93 7c 98 cb 9d da   N)mX·m·· !L·|····
0070  df ff fd a3 0e cb f8 9f  e1 d8 c1 31 9e 97 0c 31   ········ ···1···1
0080  4b 2b 99 9c a6 69 9e 03  ed 6b c1 8d 99 88 a3 0e   K+···i·· ·k······
0090  bc f1 a1 7e d1 7a cf 9f  58 bf dd 52 4e 67 22 e9   ···~·z·· X··RNg"·
00a0  49 97 4a 74 26 44 8a a9  b1 ad 5f 4f 55 fb bc 7a   I·Jt&D·· ··_OU··z
00b0  d9 2a 60 5b 10 92 fa 20  59 9c 18 a2 89 ed 66 bf   ·*`[···  Y·····f·
00c0  f5 4f 49 09 30 3f 33 95  cd a0 aa 7f 89 2c 3d 14   ·OI·0?3· ·····,=·
00d0  1b 24 b1 b6 3b e5 0d 3d  48 e6 ab f1 57 51 7c 4b   ·$··;··= H···WQ|K
00e0  ff 29 28 fe a3 2e b3 76  39 d4 b7 57 9e d3 4a 7d   ·)(··.·v 9··W··J}
00f0  b0 ce de ca 5e f5 e8 f6  23 ce bf 63 23 fb db 78   ····^··· #··c#··x
0100  26 79 1b ea 34 88 d5 74  40 23 cf 5d e1 fa 25 bd   &y··4··t @#·]··%·
0110  a4 f9 2a d1 4f 26 f4 3e  c8 35 90 2c 00 96 5f d9   ··*·O&·> ·5·,··_·
0120  d1 23 e4 f5 7f 75 e6 a6  d9 52 a6 d8 5b e5 9b 63   ·#···u·· ·R··[··c
0130  f7 01 dd e6 2f 6a c3 d1  8f 24 96 02 ad dc bf f2   ····/j·· ·$······
```

Figure 5: Network package with the centralized encryption system enabled.

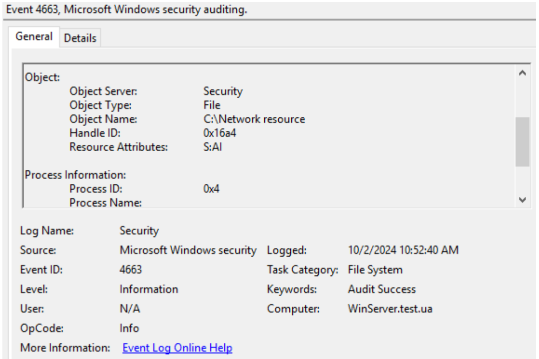
5.3. Audit testing and logging

The purpose of the experiment. It is tested how effectively the central module stores and processes event logs, according to the given security policies, and whether these logs can be used to detect information leaks or suspicious activity.



Level	Date and Time	Source	Event ID	Task Category
Information	10/2/2024 10:52:42 AM	Microsoft Wi...	4663	File System
Information	10/2/2024 10:52:42 AM	Microsoft Wi...	4663	File System
Information	10/2/2024 10:52:41 AM	Microsoft Wi...	4663	File System
Information	10/2/2024 10:52:41 AM	Microsoft Wi...	4663	File System
Information	10/2/2024 10:52:40 AM	Microsoft Wi...	4663	File System
Information	10/2/2024 10:52:40 AM	Microsoft Wi...	4663	File System
Information	10/2/2024 10:52:40 AM	Microsoft Wi...	4663	File System
Information	10/2/2024 10:52:30 AM	Microsoft Wi...	4663	File System
Information	10/2/2024 10:52:21 AM	Microsoft Wi...	4663	File System
Information	10/2/2024 10:52:21 AM	Microsoft Wi...	4663	File System
Information	10/2/2024 10:52:21 AM	Microsoft Wi...	4663	File System

Figure 6:. Security system audit log.



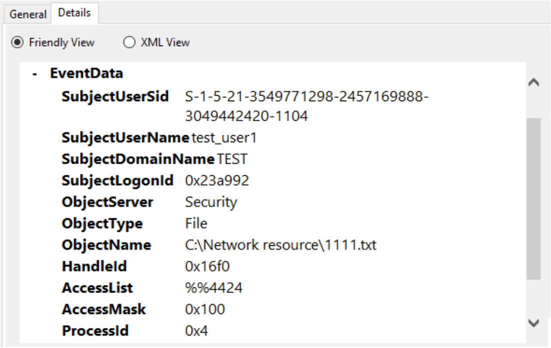
Event 4663, Microsoft Windows security auditing.

Object:
Object Server: Security
Object Type: File
Object Name: C:\Network resource
Handle ID: 0x16a4
Resource Attributes: S:AI

Process Information:
Process ID: 0x4
Process Name:

Log Name:	Security		
Source:	Microsoft Windows security	Logged:	10/2/2024 10:52:40 AM
Event ID:	4663	Task Category:	File System
Level:	Information	Keywords:	Audit Success
User:	N/A	Computer:	WinServer.test.ua
OpCode:	Info		
More Information:	Event Log Online Help		

a)



```
- EventData
    SubjectUserSid      S-1-5-21-3549771298-2457169888-
                        3049442420-1104
    SubjectUserName test_user1
    SubjectDomainName TEST
    SubjectLogonId      0x23a992
    ObjectServer        Security
    ObjectType          File
    ObjectName          C:\Network resource\1111.txt
    HandleId            0x16f0
    AccessList          %%4424
    AccessMask          0x100
    ProcessId           0x4
```

b)

Figure 7: Details of the event recorded in the audit log.

The result of the experiment. As can be seen from the audit log, all events, in accordance with established security policies, are recorded in the system (Figure 6, 7). Each event can be detailed, providing information about the event that includes all the main parameters. Such a high degree of detailing of events allows detection of attempts of unauthorized access to resources, suspicious activity of individual users, which increases the ability of the central module to prevent information leakage.

6. Conclusions

A centralized OS security system is not an absolute solution, but when the operating system is focused on preventing the leakage of sensitive information and generally protecting the information processed by its applications, it is, as experiments have shown, a better solution. Its key role in countering the leakage of confidential information is the use of integrated protection mechanisms and centralized management of all aspects of security. This guarantees the coordination of security measures, comprehensive, unlike decentralized systems, activity monitoring, access management and control over user and process actions, which significantly reduces the risk of unauthorized access to resources or data leakage.

The results of information technology research on the construction of a centralized OS security system confirm the improved level of resistance to the leakage of confidential information, the simplification of the management of mechanisms for assigning access rights to resources.

As an alternative approach for future research, developing a centralized OS security system can be used, considering the analysis of its components' importance [29]. In reliability engineering, this approach is known as importance analysis [30]. Machine learning based on the importance analysis of systems can be effective for security systems too [31].

Acknowledgments

This work was partially supported by the Slovak Research and Development Agency under the grant ‘‘Development of a new approach for reliability analysis and risk assessment based on artificial intelligence’’ (reg.no. APVV-23-0033).

Declaration on Generative AI

During the preparation of this work, the authors used Grammarly in order to: grammar and spelling check; DeepL Translate in order to: some phrases translation into English. After using these tools/services, the authors reviewed and edited the content as needed and take full responsibility for the publication’s content.

References

[1] S., Semenov, V., Zmiivska, A.V., Golubenko, Comparative studies of access delineation technologies for data protection in a computer system. Information processing systems, 2015, issue 3 (128) pp. 99 - 102

[2] VPN Unlimited part of MonoDefence. Biba Model. https://www.vpnunlimited.com/ua/help/cybersecurity/biba-model

[3] J. A. Goguen and J. Meseguer, "Unwinding and Inference Control," 1984 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 1984, pp. 75-75, doi: 10.1109/SP.1984.10019

[4] M. A. Hahn, D. R. Oestreicher and R. J. Stevenson, "The Evans & Sutherland view of tomorrow's supercomputing," Digest of Papers. COMPCON Spring 89. Thirty-Fourth IEEE Computer Society International Conference: Intellectual Leverage, San Francisco, CA, USA, 1989, pp. 300-303, doi: 10.1109/CMPCON.1989.301945

[5] H. Fatima, A. Messaoud, D. Rachid and B. M. Mounir, "Formal Modelling and Implementation of Clark-Wilson Security Policy with FoCaLiZe," 2024 6th International Conference on Pattern Analysis and Intelligent Systems (PAIS), EL OUED, Algeria, 2024, pp. 1-5, doi: 10.1109/PAIS62114.2024.10541223

[6] F. Avorgbedor, J. Liu, "Enhancing User Privacy Protection by Enforcing Clark-Wilson Security Model on Facebook," 2020 IEEE International Conference on Electro Information Technology (EIT), Chicago, IL, USA, 2020, pp. 155-161, doi: 10.1109/EIT48999.2020.9208279

[7] Law of Ukraine: On the Protection of Information in Information and Telecommunication Systems No. 80 of 05.07.1994. Bulletin of the Verkhovna Rada of Ukraine. 1994, No. 31, as amended and supplemented.

[8] DSTU EN ISO/IEC 15408-1:2022 Information technology. Method of protection. Evaluation criteria. Part 1: Introduction and general model

[9] Anderson, R. Security Engineering: A Guide to Building Dependable Distributed Systems. Wiley; 3rd edition, 2020; p 1232 ISBN-13 : 978-1119642787

[10] FIPS PUB 200, Minimum Security Requirements for Federal Information and Information Systems/Federal Information Processing Standards. National Institute of Standarts and Technology USA, 2006

[11] Cho, C.; Seong, Y.; Won, Y. Mandatory Access Control Method for Windows Embedded OS Security \ Electronics, 10, 2021; p12 https://doi.org/10.3390/electronics10202478

[12] Penelova, M. Access Control Models. Bulgarian academy of sciences. Cybernetics and information technologies, Sofia, vol 21, No 4, 2021; pp 77 - 104 http://dx.doi.org/10.2478/cait-2021-0044

[13] Billoir, E.; Laborde, R.; Wazan, A.S.; Benzekri, A. Implementing the principle of least administrative privilege on operating systems: challenges and perspectives, Ann. Telecommun. 2024; https://doi.org/10.1007/s12243-024-01033-5

[14] Devyanin, P.N.; Khoroshilov, A.V.; Kuliamin, V.V. Integrating RBAC, MIC, and MLS in Verified Hierarchical Security Model for Operating System. Program Comput Soft 46, 2020; pp 443–453. https://doi.org/10.1134/S0361768820070026

[15] Calcatinge, A.; Balog, J. Mastering Linux Administration - Second Edition: Take your sysadmin skills to the next level by configuring and maintaining Linux systems 2nd ed. Edition, Packt Publishing, 2024; p 764 ISBN 978-1837630691

[16] D.A. Tevault. Mastering Linux Security and Hardening: A practical guide to protecting your Linux system from cyber attacks*, 3rd Edition, Packt Publishing, 2023, p. 618.

[17] N. Jiang, 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, vol. 710, Springer, Cham, July 2024, pp. 206–220. https://doi.org/10.1007/978-3-031-65175-5_15.

[18] M.L. Scott, T. Brown. Shared-Memory Synchronization*, Springer, Cham, 2024, p. 243. https://doi.org/10.1007/978-3-031-38684-8 .

[19] H.C. Kuo, J. Chen, S. Mohan, T. Xu. Set the Configuration for the Heart of the OS: On the Practicality of Operating System Kernel Debloating. *Communications of the ACM*, vol. 65, no. 5, May 2022, pp. 101–109. http://dx.doi.org/10.1145/3524301.

[20] L. Gerhorst, B. Herzog, S. Reif, W. Schröder-Preikschat, T. Höni. Fast and Flexible System-Call Aggregation. *11th Workshop on Programming Languages and Operating Systems (PLOS '21)*, October 25, 2021, Virtual Event, Germany, vol. 3487267, 2021, p. 6. https://doi.org/10.1145/3477113.

[21] da Rocha, M.; Valadares, D.C.G.; Perkusich, A.; Gorgonio, K.C.; Pagno, R.T.; Will, N.C. Trusted Client-Side Encryption for Cloud Storage. In: Ferguson, D., Pahl, C., Helfert, M. (eds) Cloud Computing and Services Science. CLOSER 2020. Communications in Computer and Information Science, vol 1399, Springer, Cham, March 2021; pp 1-24 https://doi.org/10.1007/978-3-030-72369-9_1.

[22] M. Seddigh, M. Esfahani, S. Bhattacharya, M.R. Aref, H. Soleimany. Breaking KASLR on mobile devices without any use of cache memory (extended version). *Journal of Cryptographic Engineering*, vol. 14, pp. 281–294, January 2024. https://doi.org/10.1007/s13389-023-00344-y.

[23] De Oliveira, D.B.; Casini, D.; Cucinotta, T. Operating System Noise in the Linux Kernel. IEEE Transactions on Computers, № 1, 2023; pp 196-207 https://doi.org/10.1109/tc.2022.3187351.

[24] S. Lysenko, O. Pomorova, O. Savenko, A. Kryshchuk, K. Bobrovnikova. DNS-based Anti-evasion Technique for Botnets Detection. Proceedings of the 8-th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Warsaw (Poland), September 24–26, 2015. Warsaw, 2015. Pp. 453–458.

[25] O. Savenko, A. Sachenko, S. Lysenko, G.N. Markowsky, N. Vasylkiv. Botnet detection approach based on the distributed systems. International Journal of Computing, 19(2), 190-198, 2020. https://doi.org/10.47839/ijc.19.2.1761.

[26] O. Savenko, S. Lysenko, A. Kryshchuk, Y. Klots / Proceedings of the 7-th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Berlin (Germany), September 12–14, 2013. Berlin, 2013. Pp. 363–368. ISBN 978-1-4799-1426-5.

[27] O. Pomorova, O. Savenko, S. Lysenko, A. Kryshchuk, A. Nicheporuk. A Technique for detection of bots which are using polymorphic code. Communications in Computer and Information Science. 2014. Vol. 431. PP.265-276, ISSN: 1865-0929.

[28] O. Pomorova, O. Savenko, S. Lysenko, A. Kryshchuk. Multi-Agent Based Approach for Botnet Detection in a Corporate Area Network Using Fuzzy Logic. Communications in Computer and Information Science. 2013. Vol. 370. PP.243-254, ISSN: 1865-0929.

[29] E. Zaitseva, V. Levashenko, Investigation multi-state system reliability by structure function, Proc. of Int. Conf. on Dependability of Computer Systems, DepCoS – RELCOMEX, Poland, 2007, pp. 81 – 90, https://doi.org/10.1109/DEPCOS-RELCOMEX.2007.28.

[30] E. Zaitseva, V. Levashenko, Importance analysis by logical differential calculus, Automation and Remote Control, 74, 2013, pp. 171 - 182, doi:10.1134/S000511791302001X.

[31] E. Zaitseva, V. Levashenko, J.Rabcan, A new method for analysis of Multi-State systems based on Multi-valued decision diagram under epistemic uncertainty, Reliability Engineering and System Safety, 229, 2023, article number 108868.