ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МІНІСТЕРСТВА ОСВІТИ І НАУКИ УКРАЇНИ

Кваліфікаційна наукова праця
на правах рукопису

САВЕНКО БОГДАН ОЛЕГОВИЧ

УДК: 004.75:004.49:004.3

ДИСЕРТАЦІЯ

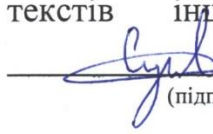
МЕТОД ТА ЧАСТКОВО ЦЕНТРАЛІЗОВАНІ СИСТЕМИ ВИЯВЛЕННЯ ЗЛОВМИСНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В КОМП'ЮТЕРНИХ МЕРЕЖАХ

123 Комп'ютерна інженерія

12 Інформаційні технології

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело



_____ Б.О. Савенко
(підпис)

Науковий керівник: Лисенко Сергій Миколайович, доктор технічних наук, професор

Хмельницький – 2024

АНОТАЦІЯ

Савенко Богдан Олегович. Метод та частково централізовані системи виявлення зловмисного програмного забезпечення в комп'ютерних мережах. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії з галузі знань 12 Інформаційні технології за спеціальністю 123 Комп'ютерна інженерія. – Хмельницький національний університет, Хмельницький. 2024.

Розв'язання задачі з покращення ефективності функціонування розподілених систем з частковою централізацією, самоорганізацією та адаптивністю для виявлення зловмисного програмного забезпечення і комп'ютерних атак в комп'ютерних мережах та виявлення зловмисного програмного забезпечення з їх використанням за рахунок синтезу їх архітектури таким чином, щоб принципи функціонування таких систем ускладнювали зловмисниками їх розуміння є актуальною науковою задачею.

У дисертації здійснено аналіз методів синтезу архітектури розподілених систем, моделей показників оточуючого середовища для розподілених систем в корпоративних мережах, методів організації функціонування розподілених систем та методів виявлення зловмисного програмного забезпечення, зокрема worm-вірусів. В роботі розроблено метод синтезу математичних моделей рівнів безпеки компонентів системи для отримання нових аналітичних виразів комплексного опису оточуючого середовища корпоративних мереж і процесів, які відбуватимуться в частково централізованих розподілених системах, удосконалено модель частково централізованих розподілених систем, розроблено метод організації функціонування частково централізованих розподілених систем, розроблено метод виявлення worm-вірусів з використанням поділу їх на класи за спільними ознаками і визначеними критеріями за багатьма класами, а також розроблено відповідну розподілену систему, здійснено постановку експериментів і проведено експериментальні дослідження з розробленою системою.

Об'єкт дослідження – процес синтезу частково централізованих розподілених систем виявлення зловмисного програмного забезпечення.

Предмет дослідження – методи і розподілені системи з частковою

централізацією для виявлення зловмисного програмного забезпечення в комп’ютерних мережах.

Метою дисертаційного дослідження є покращення ефективності функціонування розподілених систем виявлення зловмисного програмного забезпечення в комп’ютерних мережах за рахунок синтезу в їх архітектурі принципів часткової централізації, самоорганізації та адаптивності.

Наукова новизна одержаних результатів полягає в наступному:

1) удосконалено модель частково централізованих розподілених систем виявлення зловмисного програмного забезпечення, в якій на відміну від відомих моделей синтезовано принципи самоорганізації та адаптивності таким чином, що така модель дала змогу створювати згідно неї системи виявлення зловмисного програмного забезпечення, функціонування яких ускладнює розуміння їх зловмисниками, дозволяє самостійно здійснювати прийняття рішень та гнучку перебудову архітектури, що покращує їх стійкість до зловмисних дій і виявлення зловмисного програмного забезпечення;

2) вперше розроблено метод синтезу математичних моделей рівнів безпеки компонентів системи для отримання нових аналітичних виразів комплексного опису оточуючого середовища корпоративних мереж і процесів, які відбуватимуться в частково централізованих розподілених системах, що дало змогу узгодити між собою характеристичні показники, які задані дискретними та неперевними величинами, та для формування нових характеристик аналітичними виразами з врахуванням їх при визначенні рівнів безпеки в компонентах і системах в цілому;

3) розроблено новий метод організації функціонування частково централізованих розподілених систем, в якому проведено розподіл компонент системи по відношенню до центру прийняття рішень для реалізації часткової централізації, самоорганізації та адаптивності, що дало змогу задати механізми ускладнення розуміння принципу їх функціонування, самостійного прийняття рішень щодо подальших кроків, перебудови їх архітектури та наповнення системи методами виявлення зловмисного програмного забезпечення;

4) розроблено новий метод виявлення worm-вірусів, суть якого в здійсненні поділу їх на класи за спільними ознаками і визначеними критеріями за багатьма класами ознак та прийнятті рішення щодо віднесення worm-вірусу до певного класу

частково централізованою розподіленою системою, що покращило достовірність виявлення, зокрема за рахунок приховування принципів функціонування системи.

Практичне значення отриманих результатів. Розроблена частково централізована розподілена система виявлення зловмисного програмного забезпечення, зокрема worm-вірусів. має можливість її наповнення різними методами попередження, виявлення та протидії зловмисному програмному забезпеченню і комп'ютерним атакам, а також вона забезпечує належну стійкість та стабільність при функціонуванні в комп'ютерних мережах її компонентів. Особливістю розробленої частково централізованої розподіленої системи є складність в розумінні її функціонування зловмисниками, самостійне та гнучке забезпечення переміщення центру між компонентами в процесі функціонування системи, самостійне прийняття рішення щодо подальших кроків та не потребують при цьому залучення адміністратора. Крім того, реалізований метод виявлення worm-вірусів базується на багатокласовій класифікації об'єктів і результати його застосування для виявлення становлять більше 95% та підтверджують ефективність запропонованого рішення.

У результаті проведених експериментальних досліджень з розробленою системою було підтверджене коректне функціонування частково централізованої розподіленої системи, можливість застосування її до виявлення worm-вірусів, а також належні рівні стійкості та деградації системи.

Теоретичні та практичні результати дослідження впроваджені в ТОВ «ІТТ» (м. Хмельницький), Державному підприємстві «Новатор» (м. Хмельницький), ПП «НОЛТ ТЕХНОЛОДЖИС» (м. Хмельницький), а також, в освітньому процесі Хмельницького національного університету при викладанні дисциплін на кафедрі комп'ютерної інженерії та інформаційних систем для спеціальностей 123 Комп'ютерна інженерія, 126 Інформаційні системи та технології, зокрема в курсах «Теорія і проєктування комп'ютерних та кіберфізичних систем і мереж», «Теорія і технології проєктування спеціалізованих операційних систем», «Методи розв'язування наукових задач комп'ютерної інженерії» та «Технології та методи забезпечення надійності та безпеки інформаційних систем та технологій».

У вступі представлено обгрунтування актуальності наукової задачі покращення ефективності виявлення зловмисного програмного забезпечення і комп'ютерних

атак розподіленими системами. Перспективним напрямом досліджень визначено розподілені системи і синтез їх архітектури з реалізацією принципів часткової централізації, самоорганізації та адаптивності. Також, представлено зв’язок тематики дослідження з напрямками наукових досліджень дослідників цієї проблеми в світі та подано основні наукові результати роботи, її практичне значення, перелік підприємств та установ, в яких впроваджено результати роботи.

У першому розділі здійснено аналіз предметної області дослідження, існуючих комерційних і дослідницьких розподілених систем, відомих методів та характерних особливостей розробки розподілених систем попередження, виявлення та протидії зловмисному програмному забезпеченню і комп’ютерним атакам, методи виявлення worm-вірусів в корпоративних мережах. Також, підведено підсумки проведеного аналізу та здійснено постановку задачі дослідження.

У другому розділі представлено розроблену удосконалену модель частково централізованих розподілених систем, що є основою формування систем, які створюють проблеми зловмисникам щодо визначення ними центру їх системи, принципів функціонування. Архітектуру систем подано моделлю систем, в якій закладена можливість динамічної зміни конфігурації, поділу центру прийняття рішень, розподілу компонентів за можливостями з наявності центру прийняття рішень в них. Також, розроблено архітектуру компонент частково централізованих розподілених систем, що базується на отриманих аналітичних виразах, які є математичними моделями характеристичних показників значень рівнів безпеки компонентів. Вони формалізують архітектуру компонент системи згідно наявних в них функцій, їх призначення, взаємодії, місця виконання, формування центру прийняття рішень та оцінювання рівня безпеки виконуваних обчислень. Значення характеристичних показників рівнів безпеки компонентів частково централізованих систем використані для формування рішень щодо її подальших кроків та визначення зловмисного програмного забезпечення.

У третьому розділі представлено вперше розроблений метод синтезу математичних моделей рівнів безпеки компонентів системи, який дає змогу отримувати нові аналітичні вирази комплексного опису об’єктів та процесів, що будуть відбуватись в частково централізованих розподілених системах і будуть відноситись до оцінювання безпеки компонент системи. Він може бути

застосований для дискретних та неперервних величин характеристичних показників. Отримані згідно них значення характеристичних показників рівнів безпеки в компонентах системи будуть використані для оцінювання результатів розподілених обчислень, отриманих з різних компонентів системи, з метою визначення ступеня довіри до них. Також, представлено розроблений новий метод організації функціонування частково централізованих розподілених систем, який дає змогу створювати такі системи. В ньому для функціонування такого типу систем проведено розподіл компонент за відношенням до центру прийняття рішень, що дало змогу реалізувати часткову централізацію сумісно з принципами самоорганізації та адаптивності, які задають механізми до ускладнення розуміння таких систем зловмисниками, самостійного прийняття рішень системою щодо подальших кроків та перебудови її архітектури за потреби. Для дослідження ефективності функціонування систем було розроблено новий метод виявлення worm-вірусів, який базується на їх поділі за багатокласовою класифікацією згідно характерних ознак.

У четвертому розділі представлено методику визначення ефективності функціонування розподілених систем з частковою централізацією, саиоорганізацією та адаптивністю, постановку і проведення експериментальних досліджень із застосування розробленої частково централізованої розподіленої системи, опис експериментального середовища, оцінювання ефективності функціонування розподіленої системи, а також підведено підсумки з отриманих результатів.

У висновках представлено отримані наукові та практичні результати дослідження.

У додатках представлено наукові публікації, в яких відображено наукові результати роботи, акти впровадження результатів роботи, лістинг програмного забезпечення, таблиці з результатами експериментів.

Ключові слова: розподілені системи, комп’ютерні мережі, часткова централізація, самоорганізація, адаптивність, зловмисне програмне забезпечення, worm-вірус.

ANNOTATION

Savenko Bogdan Olehovych. Method and partially centralized systems for detection of malicious software in computer networks. – Qualifying scientific work on manuscript rights.

Dissertation for obtaining the scientific degree of Doctor of Philosophy in the field of knowledge 12 Information technologies in the specialty 123 Computer engineering. – Khmelnytskyi National University, Khmelnytskyi. 2024.

Solving the problem of improving the efficiency of the functioning of distributed systems with partial centralization, self-organization and adaptability for the detection of malicious software and computer attacks in computer networks and the detection of malicious software with their use due to the synthesis of their architecture in such a way that the principles of the operation of such systems made it difficult for attackers to understand them is an urgent scientific problem.

The dissertation analyzes the methods of synthesis of the architecture of distributed systems, models of indicators of the environment for distributed systems in corporate networks, methods of organizing the functioning of distributed systems and methods of detecting malware, in particular, worm viruses. The work developed a method of synthesizing mathematical models of security levels of system components to obtain new analytical expressions for a comprehensive description of the environment of corporate networks and processes that will take place in partially centralized distributed systems, improved the model of partially centralized distributed systems, developed a method of organizing the functioning of partially centralized distributed systems, a method for detecting worm viruses using their division into classes based on common features and defined criteria for many classes was developed, as well as a corresponding distributed system was developed, experiments were set up and experimental research was carried out with the developed system.

The object of the study is the process of synthesis of partially centralized distributed malware detection systems.

The subject of research is methods and distributed systems with partial centralization for detecting malicious software in computer networks.

The aim of the dissertation research is to improve the effectiveness of distributed

systems for detecting malicious software in computer networks due to the synthesis of the principles of partial centralization, self-organization and adaptability in their architecture.

The scientific novelty of the obtained results is as follows:

1) the model of partially centralized distributed malware detection systems was improved, in which, unlike known models, the principles of self-organization and adaptability were synthesized in such a way that such a model made it possible to create malware detection systems according to it, the functioning of which makes it difficult for attackers to understand them, allows independent decision-making and flexible restructuring of its architecture, which improves its resistance to malicious actions and detection of malicious software;

2) for the first time, a method of synthesizing mathematical models of the security levels of system components was developed to obtain new analytical expressions for a comprehensive description of the environment of corporate networks and processes that will take place in partially centralized distributed systems, which made it possible to reconcile the characteristic indicators, which are set by discrete and continuous values , and for the formation of new characteristics by analytical expressions, taking them into account when determining security levels in components and systems as a whole;

3) a new method of organizing the functioning of partially centralized distributed systems was developed, in which, for the functioning of this type of system, the distribution of components was carried out according to the relationship to the decision-making center for the implementation of partial centralization compatible with the principles of self-organization and adaptability, which made it possible to set mechanisms to complicate the understanding of the principle of their functioning , independent decision-making by them regarding further steps, rebuilding their architecture as needed and filling the system with methods of detecting malicious software;

4) a new method of detecting worm viruses was developed, the essence of which is to divide them into classes based on common features and defined criteria for many classes and to make a decision to assign a worm virus to a certain class by a partially centralized distributed system, which improved the reliability of detection, in particular by the account of hiding the principles of the system's functioning.

Practical significance of the obtained results. A partially centralized distributed system for detection of malware, in particular worm viruses, has been developed. it has the

possibility of filling it with various methods of prevention, detection and countermeasures against anti-aircraft and anti-aircraft weapons, and it also ensures proper stability and stability when functioning in computer networks of its components. A feature of the developed partially centralized distributed system is the difficulty in understanding its functioning by attackers, the independent and flexible provision of moving the center between components during the functioning of the system, the independent decision-making in it regarding its further steps and do not require the involvement of an administrator. In addition, the worm virus detection method implemented in it is based on a multi-class classification of objects, and the results of its application for detection confirm the effectiveness of the proposed solution. In addition, the implemented method of detecting worm viruses is based on multi-class classification of objects and the results of its application for detection are more than 95% and confirm the effectiveness of the proposed solution.

As a result of the experimental studies with the developed system, the correct functioning of the partially centralized distributed system, the possibility of its application to the detection of worm viruses, as well as the appropriate levels of stability and degradation of the system were confirmed.

The theoretical and practical results of the research were implemented in ITT LLC (Khmelnytskyi), State Enterprise "Novator" (Khmelnytskyi), PE "NOLT TECHNOLOGY" (Khmelnytskyi), as well as, in the educational process of the Khmelnytskyi National University when teaching disciplines at the Department of Computer Engineering and Information Systems for the specialty 123 Computer Engineering, 126 Information Systems and Technologies, in particular in the courses "Theory and Design of Computer and Cyber-Physical Systems and Networks", "Theory and technologies of designing specialized operating systems", "Methods of solving scientific problems of computer engineering" and "Technologies and methods of ensuring reliability and security of information systems and technologies".

The introduction presents the justification of the relevance of the scientific task of improving the effectiveness of detection of malicious software and computer attacks by distributed systems. Distributed systems and the synthesis of their architecture with the implementation of the principles of partial centralization, self-organization and adaptability are defined as a promising direction of research. Also, the connection of the

research topic with the directions of scientific research of researchers of this problem in the world is presented, and the main scientific results of the work, its practical significance, the list of enterprises and institutions in which the results of the work are implemented are presented.

In the first section, an analysis of the subject area of the study, existing commercial and research distributed systems, known methods and features of the development of distributed systems for warning, detection and countermeasures against of malicious software and computer attacks, methods of detecting worm viruses in corporate networks was carried out. Also, the results of the conducted analysis are summarized and the research task is formulated.

The second chapter presents the developed and improved model of partially centralized distributed systems, which is the basis of the formation of systems that create problems for attackers in determining the center of their system, the principles of operation. The architecture of the systems is provided by the systems model, which includes the possibility of dynamic configuration changes, the division of the decision-making center, the distribution of components according to the capabilities of the decision-making center in them. Also, the architecture of components of partially centralized distributed systems is developed, which is based on the obtained analytical expressions, which are mathematical models of the characteristic indicators of the values of the security levels of the components. They formalize the architecture of the system components according to their available functions, their purpose, interactions, places of execution, formation of the decision-making center and assessment of the security level of the performed calculations. The values of the characteristic indicators of the security levels of components of partially centralized systems are used to form decisions regarding its further steps and to identify malicious software.

The third chapter presents the first developed method of synthesizing mathematical models of the security levels of system components, which makes it possible to obtain new analytical expressions of the complex description of objects and processes that will occur in partially centralized distributed systems and will relate to the evaluation of the security of system components. It can be applied for discrete and continuous values of characteristic indicators. According to them, the values of the characteristic indicators of the security levels in the system components will be used to evaluate the results of

distributed calculations obtained from various system components, in order to determine the degree of trust in them. Also, a new developed method of organizing the functioning of partially centralized distributed systems is presented, which makes it possible to create such systems. In it, for the functioning of this type of system, the distribution of components was carried out according to the relationship to the decision-making center, which made it possible to implement partial centralization compatible with the principles of self-organization and adaptability, which set mechanisms to complicate the understanding of such systems by attackers, independent decision-making by the system regarding further steps and its reconstruction architecture as needed. In order to study the effectiveness of system functioning, a new method of detection of worm viruses was developed, which is based on their division by multi-class classification according to characteristic features.

The fourth chapter presents the methodology for determining the effectiveness of the functioning of distributed systems with partial centralization, self-organization and adaptability, setting up and conducting experimental studies using the developed partially centralized distributed system, describing the experimental environment, evaluating the effectiveness of the functioning of the distributed system, and summarizing the results obtained.

The scientific and practical results of the research are presented in the conclusions.

The appendices present scientific publications that reflect scientific work results, acts of implementation of work results, software listings, tables with experimental results.

Keywords: distributed systems, computer networks, partial centralization, self-organization, adaptability, malicious software, worm virus.

Список публікацій здобувача за темою дисертації

Наукові праці, в яких опубліковані основні наукові результати дисертації

1. Lysenko S., Savenko B. Distributed Discrete Malware Detection Systems Based on Partial Centralization and Self-Organization. *International Journal of Computing*. 2023. Vol. 22. Pp. 117-139. DOI: https://doi.org/10.47839/ijc.22.2.3082

2. Kashtalian A., Lysenko S., Savenko B., Sochor T., Kysil T. Principle and method of deception systems synthesizing for malware and computer attacks detection.

Radioelectronic and Computer Systems. 2023. Vol. *0*(4). Pp. 112-151. DOI: https://doi.org/10.32620/reks.2023.4.10

3. Савенко Б. О. Метод синтезу математичних моделей рівнів безпеки для частково централізованих розподілених систем виявлення зловмисного програмного забезпечення. *Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки*. 2023. № 3. Ч. 1. С. 217-227. DOI: https://doi.org/10.32782/2663-5941/2023.3.1/34

4. Савенко Б. О. Метод виявлення worm-вірусів згідно багатокласової класифікації. *Вісник Хмельницького національного університету. Серія: Технічні науки*. 2024. № 1 (331). С. 18-28. DOI: https://doi.org/10.31891/2307-5732-2024-331-2

Праці, які засвідчують апробацію матеріалів дисертації

5. Савенко Б. О. Розподілена частково централізована система виявлення зловмисного програмного забезпечення в комп'ютерних мережах. *Актуальні проблеми комп'ютерних наук АПКН-2022* : матеріали XIV всеукр. наук.-практ. конф., м. Хмельницький, 18-19 лист. 2022 р. / Хмельницький національний університет. Хмельницький, 2022. С. 251–253. URL: https://kn.khmnu.edu.ua/wp-content/uploads/sites/18/apkn2022_corpuspaper.pdf

6. Савенко Б. О. Самоорганізована розподілена система виявлення зловмисного програмного забезпечення в комп'ютерних мережах. *Математичне та програмне забезпечення інтелектуальних систем (МПЗІС-2022)* : тези доповідей XX міжнар. наук.-практ. конф., м. Дніпро, 23-25 лист. 2022 р. / під заг. ред. О.М. Кісельової. Дніпро, ДНУ, 2022. С. 172–173. URL: http://mpzis.dnu.dp.ua/wp-content/uploads/2022/12/MPZIS-2022-1.pdf

7. Савенко Б. О. Розподілені системи виявлення зловмисного програмного забезпечення. *2022 International Conference on Innovative Solutions in Software Engineering (ICISSE-2022)* : Conference Proceedings, Ivano-Frankivsk, Ukraine, November 29-30, 2022 / Kuz M., Kozenko M. eds. Ivano-Frankivsk, VSPNU, 2022. Pp. 22–25. URL: https://kit.pnu.edu.ua/wp-content/uploads/sites/70/2023/01/2022_International_Conference_on_Innovative_Solutions_in_Software.pdf

8. Савенко Б. О. Модель архітектури частково розподілених систем та їх

компонентів в комп’ютерних мережах. *2023 Інформаційні технології та інженерія*: Тези доп. Всеукраїнської науково-практичної конференції молодих вчених, аспірантів і студентів, м. Миколаїв, 7–10 лютого 2023 р. / ЧНУ імені Петра Могили, м. Миколаїв, 2023. С. 81-82. https://dspace.chmnu.edu.ua/jspui/handle/123456789/875

9. Савенко Б., Каштальян А., Петляк Н. Розподілені системи виявлення worm-вірусів. *2023 ITSec: Безпека інформаційних технологій:* матеріали XII Міжнар. наук.-техн. конф., м. Ужгород, 2-4 трав. 2023 р. / НАУ, м. Київ, 2023. С. 37-39. http://bit.nau.edu.ua/wp-content/uploads/2023/05/2023-ITSec_zbirnyk-1.pdf

10. Savenko B., Kashtalian A., Lysenko S., Savenko O. Malware Detection By Distributed Systems with Partial Centralization. *2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Dortmund, Germany, 2023 / Pp. 265-270. DOI: https://doi.org/10.1109/IDAACS58523.2023.10348773

Публікації, які додатково відображають наукові результати дисертації

11. Савенко Б. О. А. с. 124840, Україна. Комп'ютерна програма «Проміжне програмне забезпечення частково централізованих розподілених систем виявлення зловмисного програмного забезпечення в комп’ютерних мережах». Дата реєстрації 07.03.2024.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

AAFID – Autonomous Agents for Intrusion Detection (автономні агенти для виявлення вторгнень) [4]

API – Application Programming Interface (прикладний програмний інтерфейс)

IDS – Intrusion Detection System (система виявлення вторгнень)

IPS – Intrusion Prevention System (система запобігання вторгненням)

NAC – Network Access Control (контроль доступу до мережі) [59]

NBA – Network Behavior Analysis Systems (системи аналізу поведінки мережі) [33]

STAT – State Transition Analysis Tool (засіб аналізу систем переходів) [78]

SEP – Symantec Endpoint Protection [79]

WIPS – Wireless Intrusion Prevention Systems (системи запобігання вторгненням) [87]

АЗ – Антивірусні засоби

АПЗ – Антивірусні програмні засоби

ЗПЗ – Зловмисне програмне забезпечення

КА – Комп’ютерні атаки

КС – Комп‘ютерна система

ЛКМ – Локальна комп‘ютерна мережа

ОЗП – Оперативний запам’ятовуючий пристрій

ОС – Операційна система

ПЗ – Програмне забезпечення

РС – Розподілена система

СВВ – Системи виявлення вторгнень

ЦПРС – Центр прийняття рішень системи

ВСТУП

Актуальність роботи. Розповсюдження зловмисного програмного забезпечення (ЗПЗ) відбувається постійно та зростає. Сучасні засоби та системи попередження, виявлення та протидії ЗПЗ і комп'ютерним атакам (КА) є досить ефективними, забезпечують великий відсоток виявлення та функціонують на належному рівні. Але зловмисники постійно вивчають спроможності таких засобів та систем, вдосконалюють ЗПЗ та здійснення КА і досягають певних результатів. Тому, розробники засобів та систем попередження, виявлення та протидії ЗПЗ і КА повинні постійно їх вдосконалювати та покращувати ефективність їх функціонування. Особливо актуальним є захист корпоративних мереж, які в сукупності є типовим класом об'єктів, до якого можуть бути застосовані ефективні типові рішення. Цей клас об'єктів порівняно з одиничними комп'ютерними станціями може бути ефективно конфігурований для збільшення обчислювальних ресурсів при вирішенні завдань попередження, виявлення та протидії ЗПЗ та КА.

Крім нових чи удосконалення відомих методів попередження, виявлення та протидії ЗПЗ та КА, важливим та перспективним напрямом залишається напрям з дослідження, удосконалення чи створення принципово нової архітектури засобів та систем попередження, виявлення та протидії ЗПЗ та КА. Така архітектура повинна включати можливості систем до інтеграції в неї методів виявлення і результатом такого поєднання повинна бути система, в якій наявний центр для прийняття рішень, методи виявлення та підсистема залучення обчислювальних ресурсів комп'ютерних станцій, з яких її сформовано. Також, така архітектура повинна бути основою для розроблення систем, які будуть важко зрозумілими та прогнозованими щодо функціонування для зловмисників, бо зловмисники можуть бути присутніми і в межах периметру захисту корпоративної мережі. В цьому контексті важливою вимогою до системи є її спроможність приймати рішення без втручання користувача. Все це в сукупності вимагає синтезувати в архітектурі таких систем ефективний центр прийняття рішень, який міг би, також, переміщуватись в залежності від зміни стану в корпоративній мережі та безпосередньо в системі.

Дослідження та розроблення архітектури розподілених систем попередження, виявлення та протидії ЗПЗ та КА саме зі спрямуванням на особливості та варіанти

їх центру прийняття рішень, а також дослідження, відповідно, впливу варіантів архітектури на ефективність функціонування та достовірність таких систем, є недостатнім. Крім того, не тільки безпосередньо центр прийняття рішень як цілісна частина системи впливатиме на її функціонування, а саме його архітектура та принцип реалізації є перспективним напрямом для дослідження. Найбільш дослідженими є системи з централізованою та децентралізованою архітектурою в контексті завдань з попередження, виявлення та протидії ЗПЗ та КА. Але на сьогодні недостатньо дослідженою є архітектура розподілених систем з частковою централізацією. Вона актуальна для систем з приховуванням їх особливостей та розуміння їх функціонування зловмисниками.

Розвитком теорії розподілених систем займаються Таненбаум А. [49], Мухін В. [36, 54-56], Вулгаріс С. [85], Фішер О. [26], Авама Е. [5]. Розробленням систем виявлення та ідентифікації комп'ютерних вірусів, протидії ЗПЗ та КА, зокрема систем з приманками та пастками, а також забезпечення стійкості систем в корпоративних мережах, займаються Сочор Т. [38, 93], Харченко В. [40, 53, 82, 94, 106], Хорошко В. [41, 42], Дудикевич В. [98], Корченко А. [101], Лукова-Чуйко Н. [105], Терейковський І. [120], Летичевський О. [44], Корченко О. [120].

Тому, актуальною науковою задачею є розроблення методів для покращення ефективності функціонування розподілених систем з частковою централізацією, самоорганізацією та адаптивністю для виявлення ЗПЗ та КА в комп'ютерних мережах та виявлення ЗПЗ з їх використанням за рахунок синтезу архітектури таким чином, щоб принципи функціонування таких систем ускладнювали зловмисникам їх розуміння.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційне дослідження виконувалось у рамках науково-дослідної тематики Хмельницького національного університету: держбюджетної науково-дослідної теми 1Б-2021 «Самоорганізована розподілена система виявлення зловмисного програмного забезпечення в комп'ютерних мережах» (ДР № 0121U109936); держбюджетної науково-дослідної теми №2Б-2024 «Система виявлення ЗПЗ та комп'ютерних атак в корпоративних мережах з використанням хибних об'єктів атак та пасток» (№ держреєстрації 0124U000980), в яких автор дисертації був виконавцем.

Мета і завдання дослідження.

Об'єкт дослідження – процес синтезу частково централізованих розподілених систем виявлення зловмисного програмного забезпечення.

Предмет дослідження – методи і розподілені системи з частковою централізацією для виявлення зловмисного програмного забезпечення в комп'ютерних мережах.

Метою дисертаційного дослідження є покращення ефективності функціонування розподілених систем виявлення зловмисного програмного забезпечення в комп'ютерних мережах за рахунок синтезу в їх архітектурі принципів часткової централізації, самоорганізації та адаптивності.

Завдання дослідження формулюються в роботі наступним чином:

1) провести аналіз методів синтезу архітектури розподілених систем, моделей показників оточуючого середовища для розподілених систем в корпоративних мережах, методів організації функціонування розподілених систем та методів виявлення ЗПЗ, зокрема worm-вірусів;

2) розробити формальний опис середовища функціонування розподілених систем через характеристичні показники, які повинні враховуватись при визначенні рівнів безпеки компонентів частково централізованих розподілених систем та формування рішень щодо її подальших кроків і виявлення ЗПЗ;

3) розробити метод синтезу математичних моделей рівнів безпеки компонентів системи для отримання нових аналітичних виразів для комплексного опису оточуючого середовища і процесів, які відбуватимуться в частково централізованих розподілених системах;

4) удосконалити модель частково централізованих розподілених систем, в яких синтезувати принципи самоорганізації і адаптивності та врахувати характеристичні показники оточуючого середовища корпоративних мереж і процесів в розподілених системах, з метою розроблення згідно неї таких засобів, що будуть створювати проблеми зловмисникам щодо визначення ними центру їх системи, принципів функціонування, прийняття самостійних рішень та гнучкої перебудови їх архітектури;

5) розробити метод організації функціонування частково централізованих розподілених систем, в якому для функціонування такого типу систем провести

розподіл компонент за відношенням до центру прийняття рішень, щоб реалізувати часткову централізацію сумісно з принципами самоорганізації та адаптивності, які задають механізми до самостійного прийняття рішень щодо подальших кроків системою та перебудови її архітектури за потреби;

6) розробити метод виявлення worm-вірусів з використанням поділу їх на класи за спільними ознаками і визначеними критеріями та імплементувати його в архітектуру частково централізованих розподілених систем для прийняття рішення системою щодо виявлення ЗПЗ;

7) розробити частково централізовану розподілену систему виявлення worm-вірусів, провести з нею експериментальні дослідження щодо встановлення ефективності функціонування і достовірності виявлення worm-вірусів та впровадити її у виробництво.

Методи дослідження. Для розв'язання поставлених задач використовуються основні положення теорії множин, теорії графів, теорії розподілених систем, теорії комп'ютерних мереж, методи виявлення ЗПЗ в комп'ютерних системах:

1) теорії множин і теорії графів для деталізованого подання елементів моделі архітектури частково централізованих розподілених систем та їх компонентів;

2) теорії розподілених систем для синтезу архітектури частково централізованих розподілених систем та їх компонентів, згідно яких можуть бути розроблені системи виявлення ЗПЗ в комп'ютерних мережах;

3) теорії комп'ютерних мереж для задання функціонування частково централізованих розподілених систем, зокрема їх компонентів;

4) методи виявлення ЗПЗ в комп'ютерних системах, зокрема worm-вірусів, які базуються на класифікації об'єктів.

Наукова новизна одержаних результатів полягає в наступному:

1) удосконалено модель частково централізованих розподілених систем виявлення зловмисного програмного забезпечення, в якій на відміну від відомих моделей синтезовано принципи самоорганізації та адаптивності таким чином, що така модель дала змогу створювати згідно неї системи виявлення зловмисного програмного забезпечення, функціонування яких ускладнює розуміння їх зловмисниками, дозволяє самостійно здійснювати прийняття рішень та гнучку перебудову архітектури, що покращує їх стійкість до зловмисних дій та виявлення

зловмисного програмного забезпечення;

2) вперше розроблено метод синтезу математичних моделей рівнів безпеки компонентів системи для отримання нових аналітичних виразів комплексного опису оточуючого середовища корпоративних мереж і процесів, які відбуватимуться в частково централізованих розподілених системах, що дало змогу узгодити між собою характеристичні показники, які задані дискретними та неперервними величинами, та для формування нових характеристик аналітичними виразами з врахуванням їх при визначенні рівнів безпеки в компонентах і системах в цілому;

3) розроблено новий метод організації функціонування частково централізованих розподілених систем, в якому проведено розподіл компонент системи по відношенню до центру прийняття рішень для реалізації часткової централізації, самоорганізації та адаптивності, що дало змогу задати механізми ускладнення розуміння принципу їх функціонування, самостійного прийняття рішень щодо подальших кроків, перебудови їх архітектури та наповнення системи методами виявлення зловмисного програмного забезпечення;

4) розроблено новий метод виявлення worm-вірусів, суть якого в здійсненні поділу їх на класи за спільними ознаками і визначеними критеріями за багатьма класами ознак та прийнятті рішення щодо віднесення worm-вірусу до певного класу частково централізованою розподіленою системою, що покращило достовірність виявлення, зокрема за рахунок приховування принципів функціонування системи.

Обґрунтованість і достовірність наукових положень, висновків і рекомендацій. Наукові положення, висновки і рекомендації дисертації обґрунтовані коректним та доцільним використанням математичного апарату, зокрема математичні моделі характеристик оточуючого середовища задані розробленими аналітичними виразами для дискретних і неперервних величин, а розроблені методи базуються на математичних моделях характеристик оточуючого середовища, реалізацією розробленої частково централізованої розподіленої системи виявлення ЗПЗ, ефективним практичним впровадженням результатів дисертаційного дослідження на підприємствах, що використовують такі розподілені системи, яке продемонструвало відповідність результатів теоретичних досліджень з реальними результатами застосування.

Практичне значення отриманих результатів. Розроблена частково централізована розподілена система виявлення ЗПЗ, зокрема worm-вірусів, має можливість її наповнення різними методами попередження, виявлення та протидії ЗПЗ та КА, а також забезпечує належну стійкість та стабільність при функціонуванні в комп'ютерних мережах її компонентів. Особливістю розробленої частково централізованої розподіленої системи є складність в розумінні її функціонування зловмисниками, самостійне та гнучке забезпечення переміщення центру між компонентами в процесі функціонування системи, самостійне прийняття рішення щодо подальших кроків та не потребують при цьому залучення адміністратора. Крім того, реалізований метод виявлення worm-вірусів базується на багатокласовій класифікації об'єктів і результати його застосування для виявлення становлять більше 95% та підтверджують ефективність запропонованого рішення.

У результаті проведених експериментальних досліджень з розробленою системою було підтверджене коректне функціонування частково централізованої розподіленої системи, можливість застосування її до виявлення worm-вірусів, а також належні рівні стійкості та деградації системи.

Теоретичні та практичні результати дослідження впроваджені в ТОВ «ІТТ» (м. Хмельницький), Державному підприємстві «Новатор» (м. Хмельницький), ПП «НОЛТ ТЕХНОЛОДЖИС» (м. Хмельницький), а також, в освітньому процесі Хмельницького національного університету при викладанні дисциплін на кафедрі комп'ютерної інженерії та інформаційних систем для спеціальностей 123 Комп'ютерна інженерія, 126 Інформаційні системи та технології, зокрема в курсах «Теорія і проєктування комп'ютерних та кіберфізичних систем і мереж», «Теорія і технології проєктування спеціалізованих операційних систем», «Методи розв'язування наукових задач комп'ютерної інженерії» та «Технології та методи забезпечення надійності та безпеки інформаційних систем та технологій».

Особистий внесок здобувача. Всі основні результати дисертаційного дослідження, які представлені до захисту, одержані автором особисто. В роботах, опублікованих одноосібно автором, отримано наступні результати: [113-116] – розроблено модель частково централізованих розподілених систем; [112] – розроблено метод синтезу математичних моделей рівнів безпеки компонентів системи; [111] – розроблено метод виявлення worm-вірусів. У роботах, які

опубліковані у співавторстві, автору належать основні ідеї, теоретична та практична розробка положень, відображених у характеристиці наукової новизни отриманих результатів, а саме: [47] – розроблено модель частково централізованих розподілених систем та математичні моделі характеристичних показників значень рівнів безпеки компонентів; [38] – розроблено метод організації функціонування частково централізованих розподілених систем; [71] – розроблено метод синтезу математичних моделей рівнів безпеки компонентів системи; [107, 109] – розроблено реалізацію частково централізованої розподіленої системи.

Апробація результатів дисертації. Апробацію основних положень, ідей, висновків дисертаційної роботи проведено на науковому семінарі кафедри комп'ютерної інженерії та інформаційних систем у Хмельницькому національному університеті. Наукові результати роботи доповідалися на таких конференціях: 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS, IEEE), Dortmund, Germany, 2023; XIV всеукраїнської науково-практичній конференції «Актуальні проблеми комп'ютерних наук АПКН-2022» – Хмельницький: ХНУ, 18-19 листопада, 2022; International Conference on Innovative Solutions in Software Engineering. – Ivano-Frankivsk, Ukraine, November 29-30, 2022; XX міжнародній науково-практичній конференції «Математичне та програмне забезпечення інтелектуальних систем», м. Дніпро, 23-25 лист. 2022; Всеукраїнській науково-практичній конференції молодих вчених, аспірантів і студентів «Інформаційні технології та інженерія», м. Миколаїв, 7–10 лютого 2023; XII Міжнародній науково-технічній конференції «Безпека інформаційних технологій (ITSec)», м. Ужгород, 2-4 травня 2023.

Публікації. За результатами проведених досліджень основні наукові результати опубліковано у 4 наукових статтях у трьох фахових наукових журналах України [38, 111, 112] та міжнародному науковому журналі [47], дві з яких в наукових журналах [38, 47], проіндексованих в наукометричній базі Scopus. Апробація засвідчена публікаціями 6 праць в матеріалах міжнародних та всеукраїнських конференцій [71, 109, 113-116], з яких одна праця проіндексована у наукометричній базі Scopus [71]. Опубліковано та отримано одне свідоцтво про реєстрацію авторського права на твір (програму) [107].

Структура та обсяг дисертації. Дисертаційна робота складається з анотації, змісту, переліку умовних скорочень, вступу, чотирьох розділів, висновку, списку використаних джерел та шести додатків. Повний обсяг роботи містить 245 сторінок друкованого тексту, з них анотація – на 12 стор., зміст – на 2 стор., перелік умовних скорочень – на 1 стор., основний текст – на 162 стор., список із 121 використаних джерел – на 13 стор., додатки – на 53 стор. Дисертація містить 15 рисунків та 85 таблиць.

РОЗДІЛ 1.
АНАЛІЗ РОЗПОДІЛЕНИХ СИСТЕМ ПОПЕРЕДЖЕННЯ, ВИЯВЛЕННЯ І ПРОТИДІЇ ЗЛОВМИСНОМУ ПРОГРАМНОМУ ЗАБЕЗПЕЧЕННЮ ТА КОМП'ЮТЕРНИМ АТАКАМ

1.1. Аналіз існуючих розподілених систем виявлення зловмисного програмного забезпечення

Зловмисники продовжують здійснювати комп'ютерні атаки (КА) на інформаційні ресурси різних корпоративних мереж. Їх кількість постійно зростає. Вони проводяться із використанням різних засобів. Засоби проведення атак постійно удосконалюються. Для здійснення атак широко застосовують зловмисне програмне забезпечення (ЗПЗ). Раніше його використання було хаотичним. В останні десятиліття його розробляють для здійснення цілеспрямованих атак, зокрема, цю особливість, також, враховують розробники в його архітектурі. ЗПЗ створює проблеми користувачам, особливо користувачам корпоративних мереж. Тому, адміністратори комп'ютерних мереж повинні враховувати це при їх експлуатації. Наявні засоби попередження, виявлення та протидії комп'ютерним атакам, а також правильне конфігурування комп'ютерних мереж із покращеними характеристиками їх захисту та організаційні заходи з управління інформаційною та кібербезпекою на підприємстві переважно забезпечують належний рівень захисту корпоративних мереж. Але в зв'язку з тривалою постійною появою нових засобів і методів проведення комп'ютерних атак та недотриманням повного комплексу вимог щодо кіберзахисту, проблеми із захистом ресурсів корпоративних мереж залишаються і потребують удосконалення та розробки принципово нових засобів попередження, виявлення та протидії КА та ЗПЗ.

Незалежні від розробників лабораторії з тестування засобів попередження, виявлення та протидії КА і ЗПЗ підтверджують зростання кількості ЗПЗ [74, 84] та потенційно небезпечних програм [84]. Також, в їх звітах щодо достовірності виявлення ЗПЗ різними сучасними антивірусними засобами (АЗ) підтверджується високий рівень виявлення. Але АЗ, які б забезпечували виявлення повністю всього ЗПЗ в аналізованих ними засобах немає. ЗПЗ є різноспрямованим і містить різні

технології проникнення. А також ЗПЗ спрямоване на перебування в різних за призначенням і архітектурою обчислювальних ресурсах. Тому, для попередження, протидії та виявлення ЗПЗ антивірусні засоби повинні бути універсальними щодо застосування. Але наявні АЗ не є повністю універсальними. Тому, ЗПЗ проникає в корпоративні мережі, також, з цієї причини. Крім того, розробники АЗ повинні постійно удосконалювати наявні та розробляти принципово нові методи створення таких засобів і методи виявлення ЗПЗ. При цьому ці засоби можуть покращувати загальні результати, але можуть на деяких типах ЗПЗ і погіршувати попередні версії, бо вони використовуються в постійно змінюваному середовищі.

На практиці, як правило, в корпоративних мережах використовують комплексні стратегії та технології. В них використовують різноспрямовані антивірусні засоби та системи попередження, виявлення та протидії КА і ЗПЗ. Таким чином, в корпоративних мережах формують комплексну систему.

Важливим напрямом досліджень та розвитку систем попередження, виявлення та протидії КА і ЗПЗ є удосконалення та розробка їх нової архітектури. Це важливо в контексті використання таких систем, які складно зрозуміти зловмисникам. Системи, які використовують в корпоративних мережах, можуть бути окремими і не пов’язаними між собою та призначатись для різних етапів попередження, виявлення та протидії КА і ЗПЗ. Для забезпечення захисту інформації в корпоративних мережах доцільно використовувати різноспрямовані засоби та системи, а також для частини з них організовувати взаємодію між ними, а решту використовувати винятково як самостійні. Це потрібно для того, щоб зловмисник не зміг в результаті проведення атаки зрозуміти будову захисту корпоративних мереж. При нерозумінні йому складно буде досягти результату. Як правило, перший етап атаки здійснюється за планом розвідки. Якщо все-таки зловмисник пройшов успішно етап розвідки, то на основному етапі, тобто етапі проведення атаки, його повинні зустріти системи попередження, виявлення та протидії такі, яких він не бачив на попередньому етапі. Зокрема, це можуть бути системи з приманками чи пастками [38]. Таким чином, якщо для захисту корпоративних мереж використовують різноспрямовані системи та засоби на різних рівнях виявлення і для кожного етапу використовують призначені для нього системи і засоби, то результат попередження, виявлення та протидії КА повинен бути відповідним. Аналогічний підхід стосується і для ЗПЗ,

зокрема і призначеного для проведення КА. Зловмисник, також, може здійснювати вторгнення з середини корпоративної мережі. Тоді, застосування систем попередження, виявлення та протидії його ЗПЗ теж буде ефективним, якщо він не розуміє як побудована система захисту. В загальному випадку наявний перелік можливих засобів і систем попередження, виявлення та протидії КА та ЗПЗ є невеликим і зловмисник може здійснити їх дослідження та вивчення принципів роботи. Також, зловмисник може отримати доступ до відомостей про організацію захисту корпоративних мереж. Наприклад, атака на «Київстар» [121] відбулась завдяки використанню облікового запису співробітника. Тому, які б системи не були і для яких етапів захисту корпоративних вони б не призначались цілеспрямованість зловмисників може давати позитивний для них результат. Таким чином, вплив зловмисників з використанням КА та ЗПЗ ззовні чи зсередини корпоративної мережі може досягти мети.

В зв’язку з цим, крім удосконалення та розроблення принципово нових методів для попередження, виявлення та протидії ЗПЗ і КА, перспективним напрямом досліджень є розробка нової архітектури систем попередження, виявлення та протидії ЗПЗ і КА. Така архітектура повинна була б дати змогу власникам корпоративних мереж убезпечити себе за рахунок того, що згідно неї можна створити системи, які б змінювали свою архітектуру самостійно без втручання адміністратора в процесі свого функціонування. Тоді, в такі системи можна було б втілювати різні методи попередження, виявлення та протидії ЗПЗ і КА та застосовувати їх на різних етапах захисту корпоративних мереж.

Розглянемо існуючі системи попередження, виявлення та протидії ЗПЗ і КА для корпоративних мереж [74, 84]. Їх можна поділити на комерційні та дослідницькі (некомерційні).

Комерційні системи, на відміну від дослідницьких, пропонують різноспрямовані і різнофункційні варіанти захисту корпоративних мереж, а також можуть забезпечувати комплексні рішення, та для різних рівнів, на які може проникати ЗПЗ.

Компанія ESET [24] пропонує багаторівневе та ефективне запобігання поширенню загроз. Поєднання сучасних технологій та професійних сервісів забезпечують систему захисту інформації та кібербезпеки для підприємств.

Платформа ESET PROTECT — це рішення для запобігання, виявлення та реагування на кіберзагрози. Спрямування платформи ESET PROTECT: провідний захист робочих станцій від програм-вимагачів та «0-денних» загроз, а також безпека даних; комплексний захист робочих станцій, хмарних застосунків та пошти; багаторівневий захист у поєднанні з можливостями запобігання, виявлення та реагування; багаторівневий захист IT-середовища; управління ризиками та допомога провідних фахівців ESET. Наявні функції та особливості ESET PROTECT: консольний інтерфейс; захист робочих станцій; захист файлових серверів; повнодискове шифрування; розширений аналіз у хмарі; захист хмарних застосунків; захист поштових серверів; управління вразливостями та виправленнями; розширене виявлення та реагування; багатофакторна автентифікація; сервіси з виявлення та реагування; розгортання та модернізація; преміум-підтримка; інструменти для офіцера кібербезпеки.

Розглянемо систему Symantec Endpoint Protection (SEP) [79]. Ця система є клієнт-серверною. Вона захищає ноутбуки, настільні ПК і сервери у мережі від ЗПЗ, ризиків і вразливостей. SEP поєднує захист від вірусів із розширеним захистом від загроз, щоб проактивно захистити клієнтські комп'ютери від відомих і невідомих загроз, таких як віруси, worm-віруси, троянські програми та рекламне ПЗ. Розробниками SEP заявлено, що ця система забезпечує захист навіть від найскладніших атак, які обходять традиційні заходи безпеки, наприклад руткіти, атаки нульового дня та ЗПЗ, яке може змінюватись. SEP використовує пошаровий підхід до захисту. Розробники SEP використовують комплексний підхід до захисту мережі, що полягає в забезпеченні захисту до, під час і після атаки. SEP використовує цілісний підхід до безпеки для захисту середовища по всьому ланцюжку атак, враховуючи наступні його етапи: вторгнення, інфікування та ексфільтрація, а також відновлення та щеплення.

Malwarebytes [50] - система для захисту апаратного забезпечення, яка поєднує 21 рівень захисту, аналітику загроз та людський досвід у простому для використання рішенні для захисту організацій, яким не потрібний великий штат IT-персоналу для захисту від новітніх загроз, включаючи програми-вимагачі, шкідливе програмне забезпечення, віруси та інші атаки. Рівні захисту: захист від експлойтів; веб-захист; проактивний захист від загроз (аналітика загроз, загрози спільноти); режим

самозахисту агенту; захист від атак методом підбору ключа (протокол віддаленого робочого столу); захист від програм-вимагачів; виявлення аномалій; сканування на наявність шкідливих програм; виявлення програм-вимагачів; аналіз корисного навантаження; виявлення безфайлових індикаторів компрометації; ретроспективний пошук загроз; пошук та перевірка повідомлень; пошук безфайлових індикаторів компрометації; автоматичне виправлення (карантин); експертне виправлення (аналітик); ізоляція (мережа, процес, настільні системи); аналіз підозрілих файлів (хмарна пісочниця); відкат установки програми-вимагача; виправлення бекдору; посібник з виправлення.

Рішення Network Access Control (NAC) [59] для контролю доступу до мережі підтримують видимість мережі та керування доступом через застосування політики до пристроїв і користувачів корпоративних мереж. Оскільки зараз організаціям доводиться враховувати експоненціальне зростання кількості мобільних пристроїв, які отримують доступ до їхніх мереж, і є ризики для безпеки, які вони створюють. Надзвичайно важливо мати інструменти, які забезпечують видимість, контроль доступу та відповідні можливості, які необхідні для посилення інфраструктури безпеки мережі. Система NAC може заборонити доступ до мережі невідповідним пристроям, помістити їх у зону карантину або надати їм лише обмежений доступ до обчислювальних ресурсів, утримуючи таким чином незахищені вузли від зараження мережі.

Snort [27] — це система запобігання вторгнень (IPS) із відкритим кодом. Snort IPS використовує серію правил, які допомагають визначити зловмисну мережну активність, використовує ці правила для пошуку пакетів, які збігаються з ними, та генерує сповіщення для користувачів. Snort, також, можна розгорнути всередині, щоб зупинити ці пакети. Snort має три основні призначення: виявлення сніффер пакетів; реєстратор пакетів; повномасштабна система запобігання вторгненню в мережу. Набір правил знаходиться у вільному доступі для всіх користувачів.

Система виявлення вторгнень (IDS) [60] відстежує події в комп’ютерній системі чи мережі та аналізує їх на наявність ознак можливих інцидентів, які є порушеннями чи неминучою загрозою порушення політик кібербезпеки, політик прийнятного використання або стандартних методів безпеки. IDS бувають двох основних типів: мережні та хост-базовані. Мережні IDS відстежують мережний

трафік і запускають його через систему правил, щоб визначити, чи є він зловмисним відповідно до набору сигнатур. Якщо трафік вважається зловмисним, сповіщення спрацює та повідомить систему моніторингу. Хостові IDS відстежують активність на окремих хостах і порівнюють її з відомим базовим рівнем. IDS часто поєднують із системами запобігання вторгненням (IPS) для виявлення та блокування трафіку, який може бути пов’язаний зі зловмисними діями. IDS та IPS можуть бути інтегровані з брандмауером і їх потрібно розглядати для розгортання на периметрах мережі, а також у мережі у відповідних мережних вузлах. В системах захисту корпоративних мереж [2, 8, 11, 15, 29, 65, 77] теж наявні IDS, як мережні так і хостові. Згідно відомостей в [2, 8, 11, 15, 29, 65, 77] про системи захисту інформації встановлено, що розробниками комерційних систем пропонуються комплексні рішення для приватних комп’ютерів і корпоративних мереж. З наявної інформації на їх офіційних ресурсах для корпоративних мереж ними пропонуються системи, які розгортаються у вузлах комп’ютерних мереж. Крім того, можуть також долучати для захисту корпоративних мереж і розроблені ними апаратні пристрої, які є аналогами мережних екранів. Щодо внутрішньої архітектури систем з [2, 8, 11, 15, 29, 65, 77], то можна встановити, що переважна більшість з них є централізованими із залученням адміністратора до прийняття частини рішень. Хоча пропонуються і варіанти із децентралізованою архітектурою таких систем або із змішаною. Для корпоративних мереж пропонуються рішення із захистом комп’ютерних станцій окремо і поєднанням результатів роботи компонентів систем в хостових вузлах між собою та узгодження прийняття результуючого рішення на мережному рівні.

Розглянемо некомерційні системи виявлення комп’ютерних атак: AAFID [4, 16]; ASAX [3]; NetSTAT [58]; Prelude [70]; STAT [78]; OSSEC HIDS [64].

Система AAFID (Autonomous Agents for Intrusion Detection) має розподілену архітектуру та відноситься до систем виявлення атак [4, 7, 16]. Основою системи є автономні агенти виявлення. В основі її архітектури планувалось використати автономні агенти, що працювали б згідно генетичних алгоритмів і адаптувались би до поведінки користувачів. Ідея використання генетичних алгоритмів реалізованою не була, але ідеї в архітектурі були втілені в системі AAFID. Основними компонентами системи є агенти, трансивери, монітори, фільтри. Система AAFID є повністю розподіленою. У будь-якому вузлі локальної комп’ютерної мережі (ЛКМ)

може бути розміщена будь-яка кількість агентів, що спостерігають за цікавими з їх точки зору подіями в даному вузлі.

Система ASAX [3] орієнтована на виявлення зловмисних зловживань і побудована на використанні методу виявлення згідно моделювання правил. В ній реалізована нескладна експертна система.

Система NetSTAT [58] працює під керуванням ОС Linux і Solaris і реалізована мовою C++. Система Prelude [70] є системою з відкритими вихідними текстами. Вона розподілена і складається з наступних компонентів: мережні сенсори; вузлові сенсори; модулі керування; агенти реагування.

Аналіз переходу системи із стану в стан реалізовано в системі STAT (State Transition Analysis Tool) [78]. Основою використовуваного системою методу є опис вихідної інформації захищуваної системи у вигляді набору станів компонентів і подальший аналіз переходів зі стану в стан в результаті активних зовнішніх впливів. Стан захищуваної системи визначається при налаштуванні і конфігурації системи виявлення атак. Для кожного стану визначається характеристика захищеності.

Система OSSEC HIDS [64] є відкритою вузловою системою виявлення атак. До її завдань входить: аналіз журналів; контроль цілісності; виявлення закладок; оповіщення про атаки; активна реакція на атаки. Система може бути встановлена як в одиночній конфігурації в одному вузлі, так і в розподіленій конфігурації в кількох вузлах. В останньому випадку одна з інсталяцій стає сервером, а решта - агентами системи. У ній керування агентами виконується централізовано із сервера.

В описах некомерційних систем виявлення КА і ЗПЗ більше деталізовано про їх внутрішню архітектуру порівняно з комерційними розробками. Але і в комерційних і дослідницьких системах основна увага приділена методам попередження, виявлення, протидії та реагування на КА і ЗПЗ і менше деталізовано про унікальність архітектурних рішень. Саме креативні рішення в частині архітектури систем могли б покращити ефективність їх функціонування при застосуванні та виявленні КА і ЗПЗ. Особливо, це актуально для систем, які використовують приманки і пастки.

Також, в контексті аналізу систем [2-4, 7, 8, 11, 15, 16, 24, 27, 29, 38, 50, 58, 59, 60, 64, 65, 70, 74, 77-79, 84] за типами можна поділити на серверні, мережні і змішані, за вибором способу аналізу на неперервні та інтервальні, за типом централізації в

архітектурі на централізовані, децентралізовані та змішані (комбіновані, гібридні), за типами поведінки після атаки на активні та пасивні.

В результаті проведеного аналізу з джерел [2-4, 7, 8, 11, 15, 16, 24, 27, 29, 38, 50, 58, 59, 60, 64, 65, 70, 74, 77-79, 84, 121] встановлено, що зловмисники продовжують активно розробляти ЗПЗ та здійснювати КА, наявні комерційні системи мають як правило комплексні рішення щодо захисту корпоративних мереж, аналіз незалежних лабораторій підтверджує відсутність систем, які здійснюють виявлення всього наявного ЗПЗ та виявлення всіх КА, розробники некомерційних систем зосереджуються переважно на розробленні архітектури систем в цілому та архітектурі їх сенсорів, у відомостях про комерційні системи від їх розробників акцент зроблено на особливості систем з виявлення і частково про особливості архітектури, відсутній аналіз впливу особливостей архітектури систем на ефективність виявлення та аналіз цілісного поєднання в архітектурі систем методів виявлення з її особливостями архітектури в корпоративній мережі. Тому, потребує дослідження архітектура систем попередження, виявлення та протидії ЗПЗ і КА в частині її цілісного поєднання з методами виявлення з метою покращення ефективності функціонування та виявлення, розроблення архітектури систем в частині формування нею самостійних рішень без залучення адміністратора та розроблення методу організації функціонування таких систем.

1.2. Методи та особливості створення розподілених систем

Архітектура систем попередження, виявлення та протидії ЗПЗ і КА в корпоративних мережах є розподіленою, тому для її аналізу і удосконалення розглянемо принципи, методи та способи розроблення такої розподіленої архітектури включно з компонентами і елементами.

Загальні принципи, методи та способи розроблення розподілених систем, які подані в [49], є концептуальними. Зокрема, виділено різні класифікації розподілених систем. Ці класифікації відображають різні аспекти їх створення. Наприклад, поділ архітектури розподілених систем за типами шарів, змішаної і з поверненням. Також, подані протоколи, які необхідні для підтримки функціонування і, відповідно, цілісності розподіленої системи. Складнощі при синхронізації виконання завдань

виділено як проблемну задачу при розробленні розподілених систем і її розв'язування базується переважно за критерієм часу виконання. Аналогічно, подано інші проблемні задачі, які виникають при функціонуванні розподілених систем, і пропонуються варіанти архітектури систем та стратегії для зменшення кількості недоліків, що впливають на функціонування. Крім того, приділено увагу варіантам комунікації між компонентами розподілених систем та захисту інформації в процесі обміну повідомленнями. Готовність, як важливий критерій, розподілених систем подано через характеристики стійкості та надійності. Але не виокремлено питання виділення центру системи в архітектурі розподіленої системи та не представлено дослідження його різних варіантів, які можуть впливати на ефективність функціонування систем.

В архітектурі розподілених систем можуть бути синтезовані властивості, які забезпечуватимуть безпосередньо ефективність функціонування самих систем з орієнтацією безпосередньо на їх призначення. Такими властивостями, наприклад, є самоорганізація, адаптивність тощо. Тому, розглянемо наукові дослідження з таких особливостей архітектури розподілених систем. Математичні аспекти для організації самоорганізованих динамічних систем подано в роботі [81]. Організацію таким систем запропоновано описувати через характеристики процесів, які будуть відбуватись в системі або контрольованому нею середовищі. У роботі [45] подано розподілену сервіс-орієнтовану мультиагентну систему. Агенти повинні співпрацювати один з одним для виконання завдань децентралізованого виявлення послуг. Структура системи впливає на ефективність виявлення послуг. Тому, потрібно використовувати механізм структурної самоорганізації, щоб полегшити децентралізоване виявлення послуг у системі. В роботі [9] подано важливість самоорганізації при вирішення проблеми подолання складності. Розроблений підхід до реалізації самоорганізації є важливим для його використання, також, в технічних системах. В роботі [43] подано дослідження щодо динамічних систем вищого порядку та динамічної топології. В результаті запропоновано аналіз відносини між взаємодіями вищого порядку та колективною поведінкою. Моделювання в роботі [67] показує, що поступові зміни в системі можуть призвести до стану самоорганізованої критичності. Процес при наближенні до цього стану може зіткнутись зі змінами, які можуть спричинити нелінійні прояви складності процесу.

В роботі [39] доведено що на межі при функціонуванні нейромереж система коливається навколо критичної точки. Це важливо в контексті її стабільності і прогнозу щодо подальших кроків. В роботі [37] проаналізовано самоорганізовану квазікритичність, яка працює за спонтанною появою універсальних фактів. В результаті досягнення такого стану дає змогу системі вирішувати питання щодо кількості компонентів в ній. В роботі [32] метою дослідження була розробка та випробування нової концепції моделювання розподіленої системи та управління розподіленою системою для реалізації вузлів розподіленого виробництва. В роботі [61] для ефективного функціонування кіберфізичних систем розроблено згідно їх моделі систему комунікації між ними. В роботах [20, 21] базуючись на принципах, визначених програмним забезпеченням, запропоновано цілісну архітектуру для кіберфізичних систем та застосунків Інтернету речей. Досліджено питання масштабованості, гнучкості, надійності, сумісності та кібербезпеки. Архітектура використовує обчислювальні блоки, якими володіють інтелектуальні агенти, що можна використовувати для децентралізованого контролю та обробки даних у мережі. Крім того, запропоновано рівень проміжного програмного забезпечення для інкапсуляції пристроїв і служб для критичних за часом операцій у високодинамічних середовищах. Новий клас самоорганізованих кіберфізичних систем отримано в роботі [10] через об’єднання кіберфізичних систем і самоорганізованих обчислень. Таким чином, створення системи із значно збільшеною керованою автономією є основною вимогою для багатьох нових і майбутніх програм і технологій. Самоорганізовані кіберфізичні системи розташовані у фізичному середовищі та обмежені у своїх ресурсах. Вони розуміють свій власний стан та середовище. Грунтуючись на цьому розумінні, вони здатні самостійно приймати рішення під час виконання. Авторами розкрито п’ять ключових проблем для самоорганізованих кіберфізичних систем. В роботах [17, 25] проаналізовано автономні агенти, які здатні працювати над поставленою метою і взаємодіяти з навколишнім середовищем та іншими системами самостійно. При цьому вони повинні розуміти своє оточення, свої цілі, завдання, а також інших агентів, які знаходяться поряд. В роботі [31] формалізуються процеси під колективну поведінку, що спостерігається в живих системах. Їх можна локалізувати в просторі та вони не обмежені геометричними обмеженнями. В контексті опису

складних систем така робота розкриває підходи до формування самоорганізованих систем. Про організацію складних систем подано в роботі [80]. Зокрема, подано врахування складових інформації, як їх будувати і які складнощі можуть при цьому виникати. В роботі [88] розглянуто самоорганізацію як загальний механізм створення нової структурної моделі систем. Характеристики самоорганізованої поведінки, такі як відкритість, нелінійність, внутрішня випадковість, внутрішній зворотний зв'язок, інформаційна мережа, забезпечують відповідні умови та основу для самоорганізаційної еволюції системи з аспектів інформаційної функції середовища, підтримки і побудови загальної інформаційної основи системи, і дослідження нового інформаційного режиму системи. Таким чином, властивості самоорганізації в складних системах є важливими і можуть бути реалізовані. Сучасні мережні динамічні системи мають особливості, які дозволяють реалізувати в їх архітектурі принципи самоорганізації і досягти, відповідно, покращення ефективності функціонування таких систем.

Розглянемо особливості елементів та компонентів при створенні розподілених систем. В [55] подано розроблену методологію синтезу розподіленої комп'ютерної системи, яка включає базу даних. Запропоновано нові моделі оцінки параметрів обробки даних, а також аналізу та прогнозування функціональних параметрів. Запропонована методика визначення кількості вузлів обробки даних у системах з динамічною структурою згідно мережного центричного механізму управління дозволяє підвищити ефективність гетерогенних розподілених баз даних. Реалізація запропонованої методології дозволяє синтезувати розподілену систему обробки даних для обробки запитів до різнорідних баз даних з урахуванням параметрів вузлів і пропускної здатності каналів між вузлами. В роботі [14] досліджено моделі контролю доступу, які є важливим інструментом, розробленим для захисту сучасних систем даних. Моделі контролю доступу не можуть бути реалізовані кваліфіковано через те, що умови для визначення запитів користувачів на доступ до ресурсів, розподілених на різних серверах, один з яких послідовно прив'язаний до іншого, бо перевірку та авторизацію цих запитів користувачів, і можливість відстежувати дії користувачів не можна постійно налаштовувати ефективним способом. Запропоновано автоматично обчислити дозволи та рівні доступу всіх користувачів, визначених у системах розподілених баз даних для об'єктів. В роботі [63]

розглянуто підхід до децентралізованих систем і запропоновано узагальнену модель для побудови децентралізованих систем управління згідно визначення пріоритетів. Розподілене рішення [28] проблеми планування завдань з урахуванням безпеки в інфраструктурах хмарних обчислень базується на тому, що розподіл доступних ресурсів регулюється виключно спеціалізованими брокерами, призначеними окремим користувачам, які надсилають свої завдання в систему. Згідно цієї схеми потрібно виділити обмежену кількість ресурсів для певної кількості завдань, мінімізуючи ймовірність їх невдалого виконання та загальний час виконання. Запропоновано реалізовувати найкращі стратегії планування на рівні окремого користувача.

Великі дані, хмарні обчислення, мережні обчислення та Інтернет речей змінюють поточні системи та методи обробки даних. ІТ-експерти прагнуть використовувати потужність [34] розподілених систем для підвищення безпеки та запобігання шахрайству.

В роботі [56] розроблено метод визначення необхідної кількості вузлів бази даних, який базується на механізмі визначення вагового коефіцієнта типів запитів. Це дозволило отримати необхідну кількість вузлів бази даних у розподіленій системі для задоволення вимог до параметрів обробки запитів і підвищення швидкості їх обробки. Запропоновано модель [54] оцінки параметрів обробки інформації в розподілених системах з динамічною структурою, які реалізують неоднорідні розподілені бази даних. Модель базується на виборі кортежу параметрів розподіленої системи та враховує обмеження, що дозволяє формалізувати структуру керування обробкою запитів у розподілених базах даних. Багато установ і компаній [1], які займаються технологічними розробками, виробляють великі обсяги структурованих і неструктурованих даних. Для роботи з такими даними потрібні спеціальні бази даних. Тому, потрібні нові архітектури, щоб зберігати все більше різних типів даних. У роботі перевірено базу даних ключів і значень, щоб виміряти її продуктивність з точки зору пропускної здатності та затримки, де великі обсяги даних зберігаються в різних розмірах у середовищі розподіленої бази даних. За допомогою розподілених систем [6] різні користувачі можуть миттєво отримувати доступ до даних з різних місць і виконувати деякі операції з даними. Однак несанкціонований доступ кількох користувачів до системи з різних точок одночасно

може призвести до небезпечних результатів з точки зору безпеки та конфіденційності даних. Це дослідження базується на системах виявлення та запобігання вторгненням, побудованих згідно розподілених баз даних, і класифікує методи, які використовуються для порівняльного аналізу. У роботі [36] запропоновано підхід та механізми комплексного аналізу параметрів розподіленої комп'ютерної системи з урахуванням кількох критеріїв функціонування для вибору ефективної конфігурації ресурсів. Існують аналітичні оцінки параметрів: продуктивність; безпека; надійність; швидкість передачі даних. Запропоновано комплексну аналітичну модель для параметрів. У багатьох завданнях [68], пов'язаних зі спостереженням за об'єктом або моніторингом у реальному часі, потрібен збір тимчасових мультимодальних даних. Такі набори даних семантично пов'язані, оскільки відображають різні аспекти одного об'єкта. Однак набори даних різних модальностей зазвичай зберігаються та обробляються незалежно. Представлення тимчасових мультимодальних наборів даних задає процедури обробки даних, оскільки забезпечує надійний семантичний зв'язок між даними, що описують різні особливості одного об'єкта, процесу або явища. В роботі [35] подано протоколи другого рівня для платіжних мереж. які можуть виконуватися в мережі, не вимагаючи будь-якої інформації про його топологію. В роботі [99] розглянуто проблему комплексного подання мультимодальних даних про об'єкт спостереження, характеристики якого вимірюються та досліджуються з урахуванням часу та взаємозв'язку між даними різних модальностей. Метою дослідження в [53] є представлення системного підходу до аналізу стійкості систем штучного інтелекту. Пропонована методологія включає формування набору факторів стійкості, організацію та визначення таксономічних і онтологічних зв'язків для факторів стійкості систем штучного інтелекту, а також аналіз відповідних рішень стійкості та викликів. В останні роки досягнення стійкості [12] стало важливою метою багатьох зусиль щодо розробки систем. Стійкість визначається як здатність системи забезпечувати необхідні задані функційні можливості. В роботі [44] запропоновано алгебраїчний підхід для вирішення проблеми виявлення вразливості та перевірки стійкості систем до кібератак.

В роботі [82] подано реалізацію захищеного дистанційного керування накладеними мережами, які представлені у вигляді інтелектуальних мобільних

об’єктів. Специфікою цього класу мереж є використання в якості середовища передачі даних існуючих мереж стільникового зв'язку. Безпека процесів передачі даних між вузлами оверлейної мережі забезпечується використанням технології приватних віртуальних мереж. У роботі [42] розглядаються питання безпеки та топології в комп’ютерних мережах, де практичні методи захисту інформації не мають достатньої теоретичної бази. Для оптимального проектування топології комп’ютерних мереж та мінімізації обчислювальних ресурсів пропонується використовувати методи декомпозиції. Децентралізована синхронізація на рівні мережі в мобільних спеціальних мережах подана в роботі [85]. В роботі [19] подано про цілі, особливості та застосування розподілених систем. У роботах [26, 66] пропонуються методи вирішення проблеми побудови орієнтованого мінімального остовного дерева для застосування при створенні розподілених систем. У роботі [30] показано, як побудувати накладену мережу постійного ступеня та заданого діаметру за допустимий час, починаючи з довільного слабозв’язного графа. В роботі [18] проаналізовано синхронну динаміку k-більшості, де в кожному вузлі з дискретним часом рівномірно випадковим чином відібрано k сусідів із заміною. В роботі [5] проаналізовано паралельні обчислення як особливу тісно пов’язану форму розподілених обчислень. Розподілені обчислення розглядаються як слабко пов’язана форма паралельних обчислень. Розширення цього поняття на інтернет речей подано в роботі [13]. В роботах [52, 86] представлено теорію, що використовується для моделювання конкретного класу розподілених великомасштабних систем.

При розробленні розподілених систем потрібно врахувати багато аспектів, особливостей та характеристик систем: самоорганізація; адаптивність; стійкість; резилентність; перенесення базових функцій в компоненти; врахування часу; топології мереж; збір інформації з об’єктів в мережі тощо [47]. Використання наукових результатів дослідників, які подані в роботах [1, 5, 6, 9, 10, 12-14, 17-21, 25, 26, 28, 30-32, 34-37, 39, 42-45, 49, 52-56, 61, 63, 66-68, 80-82, 85, 86, 88, 99], потрібно враховувати при створенні розподілених систем в корпоративних мережах. Така властивість системи, як самоорганізація, надає системі можливості з визначення своїх наступних кроків без необхідності втручання користувача або адміністратора. Властивість адаптивності синтезована в архітектурі системи надала

б можливості перебудовувати архітектуру в залежності від поточного стану, зовнішніх подій у комп'ютерних системах, в яких вона встановлена. Незважаючи на останні досягнення в розвитку теорії розподілених систем, розробка програмного забезпечення для ефективного функціонування самих систем залишається складним завданням через складність функційних завдань, які вони повинні вирішувати. Стійкість та надійність є визначальними характеристиками розподілених систем, які залишаються предметом дослідження для покращення функціонування таких систем. Таким чином, універсальних розподілених систем, які б задовільняли вимогам завдань, що вирішуватимуться в комп'ютерних системах, немає. Ці універсальні розподілені системи могли б наповнюватись відповідними методами. Оскільки для складних завдань, особливо в комп'ютерних мережах, таких систем універсального використання немає, в які можна імплементувати розроблені методи, то їх необхідно синтезувати з врахуванням місця функціонування та завдань. Підтвердженням відсутності таких систем універсального призначення є активні дослідження в цій предметній області. При синтезі таких систем з метою досягнення їх ефективної роботи доцільно використовувати результати досліджень, які подані в роботах [1, 5, 6, 9, 10, 12-14, 17-21, 25, 26, 28, 30-32, 34-37, 39, 42-45, 49, 52-56, 61, 63, 66-68, 80-82, 85, 86, 88, 99].

1.3. Особливості розподілених систем для виявлення мережного зловмисного програмного забезпечення

Розподілені системи попередження, виявлення та протидії ЗПЗ функціонують в корпоративних мережах. Тому, розглянемо їх спрямування щодо попередження, виявлення, запобігання, протидії та реагування до мережного типу ЗПЗ. Хоча розподілені системи можуть здійснювати виявлення ЗПЗ, також, в окремих комп'ютерних станціях. До мережного ЗПЗ відноситься багато різноманітних типів за різними критеріями поділу. Але найбільш широко позиціонованим мережним ЗПЗ є множина worm-вірусів. Вони поширюються комп'ютерними мережами, часто стають основою розбудови бот-мереж, можуть бути цілеспрямованим, можуть переносити частини іншого ЗПЗ, принципово відрізняються за будовою від багатьох класів комп'ютерних вірусів, які спрямовані на інфікування виконуваних PE-файлів.

Тому, розглядатимемо множину worm-вірусів як об’єкт для дослідження розподіленими системи, оскільки переважна більшість процесів для розподілених систем та worm-вірусів буде відбуватись саме в комп’ютерних мережах. Синтез розподілених систем потрібно здійснювати так [38, 114, 115], щоб в їх архітектуру можна було втілювати не одиничні методи попередження, виявлення, запобігання, протидії та реагування на дії ЗПЗ, а багато різних і для різних типів ЗПЗ та КА. Може бути так, що для одного класу певного типу ЗПЗ потрібно декілька методів. ЗПЗ може бути таким, що тривалий час приховує свою присутність і розподілена система повинна вміти протидіяти і таким загрозам. Наприклад [119], в США було виявлено ЗПЗ, яке надавало таємний доступ до комп'ютерів жертв, дозволяючи пристроям, в яких воно функціонувало, таємно спілкуватися один з одним і діючи як плацдарм для додаткової зловмисної активності. Тому, системи повинні бути комплексними і їх архітектура, відповідно, повинна враховувати потреби в її наповнені багатьма методами. Це підтверджено фахівцями, наприклад в [83, 91]. Зокрема, в [83] акцент зроблено на комплексні рішення для управління безпекою за допомогою комплексних можливостей запобігання, виявлення та реагування згідно штучного інтелекту, провідних досліджень загроз і розвідки. А система Zeek (Bro) [91], яка є платформою для аналізу трафіку, орієнтована пріоритетно на відстеження подій, пов'язаних з безпекою, не обмежується тільки цим застосуванням. В ній наявні модулі для аналізу і опрацювання різних мережних протоколів програм, що враховують стан з'єднань і дозволяють формувати детальний журнал (архів) мережної активності.

Розглянемо worm-віруси [112] в контексті їх особливостей, застосування та будови. В роботах [22, 57] представлено декомпозицію вірусів і worm-програм згідно їх основних функційних компонентів. В роботі [76] здійснено аналіз окремого ЗПЗ «троянський кінь», який під час виконання може змінювати інші комп’ютерні програми, наприклад, копіюючи себе (або частину) у них. В роботі [75] зловмисні програми аналізуються на наявність ознак вірусів, worm-вірусів, троянських програм і руткітів та пропонуються конкретні контрзаходи для їх розпізнавання. У роботі [69] побудовано модель SIQR розповсюдження worm-вірусу залежно від двофакторної моделі. У роботі [62] зроблено припущення про існування багатовекторних worm-вірусів. В ній подано пару з них за слідами нападу, які зібрані

в приманці. У роботі [23] проаналізовано приклади ЗПЗ: віруси; worm-віруси; троянські програми; шпигунське програмне забезпечення; клавіатурні шпигуни; бот-мережі; руткіти; програмне забезпечення для вимагання та випадкові завантаження. Розробники ЗПЗ стають професійнішими в здатності розробляти ефективне ЗПЗ, яке складно виявляти [23, 96, 119], і тому розробникам систем протидії ЗПЗ потрібно постійно залишатися інноваційними, щоб вдосконалювати методи та системи протидії ЗПЗ. Підтвердження нестандартних рішень, наприклад, щодо троянських програм подано в роботах [89, 90]. Апаратні трояни вважаються одним з найнебезпечніших видів зловмисного порушення цілісності систем на основі FPGA. Дослідження довело, що апаратні трояни можуть бути імплантовані в систему (або проект системи) під час її планової модифікації. Зокрема, це відбувається, коли не працює моніторинг цілісності, базований на використанні хеш-суми. Перед запуском моніторингу цілісності слід переконатися, що апаратні трояни не імплантовані. Автори запропонували метод виявлення розташування апаратних троянів у просторі компонентів на основі FPGA критичних для безпеки систем. Дослідження троянських програм в роботі [103] підтверджує використання стандартних засобів для забезпечення спілкування між скомпрометованими вузлами за різними моделями зловмисників [73, 103]. Worm-віруси можуть бути застосовні у взаємодії із троянськими програмами, а також для побудови бот-мереж. В роботі [102] розглянуто різні моделі бот-мереж та запропоновано рішення, яке сформоване з використанням мультиагентної системи, для дослідження зловмисної активності в мережах.

Для забезпечення ефективного результату з виявлення ЗПЗ та КА потрібно, крім методів виявлення, також і ефективні системи. В роботі [98] подано комплексне рішення щодо перспективи забезпечення кібербезпеки, яке включає також і погляд на систему, яка потребує захисту, та побудову системи захисту. В роботі [93] пропонується використовувати приманки для виявлення ЗПЗ і приділено увагу системі, в якій вони реалізовані. Така система є розподіленою [38, 93]. В роботі [51] подано еволюційну мережну модель тестування розподілених систем, яка може бути використана для проєктування таких систем. Перспективність такого напряму досліджень і розвитку підтверджується, також, в [33, 87]. Системи запобігання вторгненням для бездротових мереж (WIPS) [87] відстежують активність у

бездротових мережах. А системи аналізу поведінки мережі (NBA) [33] аналізують мережний трафік для виявлення незвичних моделей. Аналіз поведінки мережі визначається як процес збору та аналізу корпоративних мережних даних для виявлення незвичної поведінки об'єктів, яка може свідчити про зловмисну діяльність. Взагалі існує дві основні стратегії в сфері виявлення атак: виявлення зловживань (misuse detection); виявлення аномалій (anomaly detection) [72, 101, 108, 110]. Але в обох випадках для корпоративних мереж основою їх реалізації є розподілені системи. Методи розподіленого управління корпоративними комп'ютерними мережам подано в [117]. Вимоги до розподілених обчислень задані в стандарті [97]. Методи ідентифікації аномальних станів [100] для систем виявлення вторгнень подано в роботі [101]. Методи виявлення зловживань, зокрема з використанням ЗПЗ та КА, подано в роботах [48, 71].

Для розподілених систем важливо забезпечити їх стійкість, особливо в умовах впливу КА та ЗПЗ безпосередньо на них. Методологічні основи та інформаційна технологія забезпечення резильєнтності комп'ютерних систем в умовах кіберзагроз подана в роботах [46, 104]. Методи забезпечення відмовостійкості та живучості розподілених систем в умовах впливів ЗПЗ подано в роботі [118]. Процеси кіберзахисту [41] відносяться до випадкових багатовимірних, динамічних нестаціонарних, активних (цілеспрямованих), що ускладнює завдання прогнозування показників кіберзахищенності. В роботі [41] запропоновано алгоритм вибору показників прогнозування кіберзахищенності комп'ютерних систем. У роботі [40] пропонується стратегія для оцінки надійності, доступності і кібербезпеки хмари та системи Інтернету речей на основі безперервного збору, порівняння, вибору та поєднання марковських і напівмарковських моделей. Отриманими результатами були алгоритми для збору та аналізу даних, вибору та поєднання відповідних моделей та їх різних типів, таких як багатофрагментні та багатофазові моделі, враховуючи зміну рівня відмов, параметрів кібератак, періодичного обслуговування тощо. Методологічні основи забезпечення функційної стійкості розподілених систем до кібернетичних загроз подано в роботі [105]. Таким чином, розроблення розподілених систем потрібно здійснювати з врахуванням забезпечення їх стійкості не тільки через забезпечення функційної безпеки, але й в умовах впливу КА та ЗПЗ безпосередньо на них.

Worm-вірус характеризується тим, що це тип ЗПЗ, який має визначальну мету, що полягає в поширенні його на велику кількість комп’ютерів з використанням мереж. В українській антивірусній компанії «Zillya Антивірус» [92] для worm-вірусів виділено такі методи розмноження: через вразливості програмного забезпечення; за допомогою програм для спілкування; через мережні ресурси; через P2P мережі каналами файлообмінних пірінгових мереж. Засоби, які при цьому використовуються worm-вірусами, що закладені в них зловмисниками, та функції повинні бути предметом аналізу для їх виявлення [109]. Наприклад [95], "Хробак Моріса" намагався підібрати паролі до облікових записів. Для цього використовував ім'я користувача і список із 400 найбільш популярних слів. "Хробак" використовував маскування, щоб приховати свою присутність у комп'ютері. Він видаляв свій виконуваний файл, перейменовував свій процес у sh [95]. Тому, розробники систем протидії ЗПЗ повинні включати в них аналіз функцій, що забезпечують мережну комунікацію, та їх комбінацій [111].

Згідно аналізу наукових результатів [22, 23, 57, 62, 69, 75, 76, 96, 112, 119] встановлено різноманітність worm-вірусів, яка проявляється не тільки за основним типом розповсюдження, але й за використанням з різними іншими комп’ютерними вірусами та троянськими програмами, а також, можлива наявність у worm-вірусів багатовекторності.

Таким чином, універсальні підходи та стратегії до створення розподілених систем виявлення ЗПЗ не можуть бути застосовані, зокрема і для мережного ЗПЗ. Оскільки зловмисники після ознайомлення з ними можуть зрозуміти як працюють такі системи і використати це для здійснення КА та в ЗПЗ. Тобто, створити декілька стандартних розподілених систем універсального призначення з різною архітектурою і наповнювати їх різними функціями (методами), зокрема і тими, які виявляють чи протидіють ЗПЗ, не доцільно. Тому, для розподілених систем саме такого призначення потрібно синтезувати особливий набір характеристик в їх архітектурі і цим вони будуть відрізнятись від універсальних розподілених систем. Важливою характеристикою розподілених систем такого призначення є їх стійкість до впливів ЗПЗ та КА і, тому її потрібно враховувати при створенні розподілених систем. Як для вирішення завдань попередження, виявлення та протидії ЗПЗ і КА, так і для забезпечення стійкості функціонування розподілених систем потрібно

сформувати набір показників [47] в корпоративній мережі, які б система могла аналізувати для подальшого прийняття рішень.

Зловмисники продовжують створювати ефективне ЗПЗ і такі дії мають стійку тенденцію до зростання, як кількісно, так і за охопленням різних типів. Для корпоративних мереж використовуються відомі засоби, але вони не забезпечують повного виявлення та надійної протидії ЗПЗ. Це підтверджується відповідними результатами незалежних антивірусних лабораторій та самими розробниками. Тому, є потреба в подальшій розробці нових систем та методів для попередження, виявлення та протидії ЗПЗ в корпоративних мережах, які повинні бути розподіленими. Розроблені, таким чином, розподілені системи могли б бути наповнені різними методами попередження, виявлення та протидії ЗПЗ і КА та, при цьому, могли б мати різне призначення. Вони могли б бути системами попередження, або системами виявлення, комплексними системами, системами з приманками тощо.

Оскільки розроблення методів стосується створення розподілених систем, які будуть функціонувати в корпоративних мережах, то для врахування особливостей ЗПЗ, якому вони будуть протидіяти, проаналізовано мережне ЗПЗ, зокрема worm-віруси. Зловмисники при їх реалізації використовують стандартний набір доступних функцій та засобів співвіднесено до середовища їх функціонування, зокрема корпоративних мереж та їх особливостей. Тому, саме множина worm-вірусів максимально охоплює мережні особливості. Решта типів ЗПЗ теж може мати при створенні такі або частину функцій та засобів як і worm-віруси. Але в них буде інше спрямування і це зменшуватиме для всієї їх множини відсоток застосування таких функцій і засобів порівняно з worm-вірусами. Тому, для дослідження ефективності методів створення розподілених систем і на їх основі самих систем будемо розглядати worm-віруси.

1.4. Постановка задачі дослідження

Для розв’язання задачі покращення ефективності функціонування розподілених систем виявлення ЗПЗ в корпоративних мережах необхідно здійснити розроблення методів синтезу таких систем, синтезу моделей показників оточуючого

середовища розподілених систем, організації функціонування розподілених систем і вирішити такі завдання:

1) провести аналіз методів синтезу архітектури розподілених систем, моделей показників оточуючого середовища для розподілених систем в корпоративних мережах, методів організації функціонування розподілених систем та методів виявлення ЗПЗ, зокрема worm-вірусів;

2) розробити формальний опис середовища функціонування розподілених систем через характеристичні показники, які повинні враховуватись при визначенні рівнів безпеки компонентів частково централізованих розподілених систем та формування рішень щодо її подальших кроків і виявлення ЗПЗ;

3) розробити метод синтезу математичних моделей рівнів безпеки компонентів системи для отримання нових аналітичних виразів для комплексного опису оточуючого середовища і процесів, які відбуватимуться в частково централізованих розподілених системах;

4) удосконалити модель частково централізованих розподілених систем, в яких синтезувати принципи самоорганізації і адаптивності та врахувати характеристичні показники оточуючого середовища корпоративних мереж і процесів в розподілених системах, з метою розроблення згідно неї таких засобів, що будуть створювати проблеми зловмисникам щодо визначення ними центру їх системи, принципів функціонування, прийняття самостійних рішень та гнучкої перебудови їх архітектури;

5) розробити метод організації функціонування частково централізованих розподілених систем, в якому для функціонування такого типу систем провести розподіл компонент за відношенням до центру прийняття рішень, щоб реалізувати часткову централізацію сумісно з принципами самоорганізації та адаптивності, які задають механізми до самостійного прийняття рішень щодо подальших кроків системою та перебудови її архітектури за потреби;

6) розробити метод виявлення worm-вірусів з використанням поділу їх на класи за спільними ознаками і визначеними критеріями та імплементувати його в архітектуру частково централізованих розподілених систем для прийняття рішення системою щодо виявлення ЗПЗ;

7) розробити частково централізовану розподілену систему виявлення worm-вірусів, провести з нею експериментальні дослідження щодо встановлення ефективності функціонування і достовірності виявлення worm-вірусів та впровадити її у виробництво.

Під ефективністю функціонування розподілених систем виявлення ЗПЗ в комп'ютерних мережах будемо розглядати комплексне відображення кінцевих результатів їх функціонування за певний проміжок часу. Зокрема, ефективність функціонування розподілених систем буде спрямована на рівні архітектури систем, показники стійкості та деградації систем, а також, достовірність виявлення ЗПЗ. Удосконалення, зміна або синтез принципово нової архітектури розподілених систем виявлення ЗПЗ, які порівняно з існуючими системами дадуть переваги користувачам, розглядатимемо як покращення ефективності функціонування таких систем.

1.5. Висновки до першого розділу

Дослідження існуючих комерційних та дослідницьких розподілених систем, методів та особливостей розробки розподілених систем попередження, виявлення та протидії ЗПЗ і КА, показало наступні результати:

1. Зловмисники продовжують створювати і розповсюджувати ефективне ЗПЗ та здійснювати КА і такі дії мають стійку тенденцію до зростання, як кількісно, так і якісно.

2. Аналіз розподілених систем попередження, виявлення та протидії ЗПЗ і КА підтверджує відсутність їх повного виявлення, потребу в постійному удосконаленні таких систем.

3. Методи створення розподілених систем попередження, виявлення та протидії ЗПЗ і КА потребують удосконалення і повинні враховувати можливість імплементування методів виявлення ЗПЗ в архітектурі таких систем, в якій окремого дослідження потребує розміщення, будова та вплив центру прийняття рішень на ефективність функціонування розподілених систем.

4. В результаті проведеного дослідження було встановлено необхідність формалізованого опису показників оточуючого середовища для розподілених

систем в корпоративних мережах і врахування цих показників в процесі прийняття рішень системами.

5. В мережному ЗПЗ в результаті проведеного аналізу було виділено worm-віруси, які можуть бути самостійними, можуть поєднуватись з іншими типами ЗПЗ, можуть бути основою для бот-мереж, можуть переміщувати комп'ютерними мережами цілісні фрагменти іншого ЗПЗ і, через включення в їх функціонал функцій та засобів для підтримки їх функціонування в корпоративних мережах, встановлено, що вони найбільш повно охоплюють такі функції і засоби порівняно з рештою типів ЗПЗ.

6. Методи виявлення worm-вірусів в корпоративних мережах потребують удосконалення і повинні бути імплементовані в архітектурі розподілених систем.

7. Перспективним напрямом згідно проведеного дослідження є розроблення методу створення розподілених систем, дослідження впливу і, відповідно, розроблення центру прийняття рішень системи, здійснення опису показників оточуючого середовища в корпоративній мережі для прийняття рішення розподіленою системою згідно них, розроблення методу виявлення worm-вірусів з втіленням його в архітектуру розподілених систем для покращення їх ефективності функціонування та виявлення такими системами.

Основні результати розділу опубліковані у працях [38, 47, 71, 109, 111, 112, 114, 115].

РОЗДІЛ 2.
АРХІТЕКТУРА ЧАСТКОВО ЦЕНТРАЛІЗОВАНИХ РОЗПОДІЛЕНИХ СИСТЕМ ВИЯВЛЕННЯ ЗЛОВМИСНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В КОМП'ЮТЕРНИХ МЕРЕЖАХ

2.1. Модель частково централізованих розподілених систем виявлення зловмисного програмного забезпечення

Засоби систем виявлення ЗПЗ в комп'ютерних мережах, а також в їх хостах, повинні базуватись не тільки на сучасних актуальних методах виявлення, але і бути імплементованими в такі архітектури засобів, які б залучали свої елементи до покращення виявлення сумісно із методами виявлення. Реагування систем виявлення ЗПЗ в комп'ютерних мережах завдяки динамічній перебудові своєї архітектури в умовах зловмисних впливів та аномальних проявів створює додаткові перешкоди для зловмисників та ЗПЗ. Така динамічна перебудова архітектури повинна координуватись та узгоджуватись із застосуванням методів виявлення. Створення для зловмисників та ЗПЗ перешкод в розумінні функціонування та поведінки засобів виявлення на архітектурному рівні надає перевагу користувачам комп'ютерних систем та мереж. Досягнення переваги потребує, крім методів, які орієнтовані безпосередньо на виявлення ЗПЗ, забезпечення в складі засобів виявлення компоненти або елементи, які змінюватимуть архітектуру системи виявлення сумісно з методами виявлення, але при цьому вони не повинні бути орієнтовані саме на виявлення конкретних типів ЗПЗ чи конкретного ЗПЗ. Такі компоненти чи елементи повинні забезпечувати функціонування системи виявлення ЗПЗ без втручання адміністратора системи чи користувача при прийнятті рішень щодо подальшого функціонування в умовах впливів ЗПЗ і не бути прогнозованими в своїх подальших діях для зловмисників та користувачів.

Перспективною архітектурою для таких засобів є архітектура, яка дає змогу створювати згідно неї самоорганізовані та адаптивні системи. Самоорганізація системи дозволятиме визначати її подальші кроки без участі користувача чи адміністратора. Адаптивність системи забезпечуватиме можливість оперативної перебудови її архітектури в залежності від стану системи, зовнішніх подій в

комп’ютерних системах та мережах, в які вона встановлена. Неврахування в архітектурі системи виявлення ЗПЗ таких складових характеристик, які забезпечать самоорганізацію та адаптивність, може надати переваги зловмисникам та ЗПЗ, бо система виявлення швидко досліджуватиметься ними і відповідно блокуватимуться її можливості з виявлення і протидії ЗПЗ.

Важливим рішенням при проєктуванні систем виявлення ЗПЗ є рішення щодо центру прийняття рішень системи для визначення його подальшого функціонування. Такий центр прийняття рішень системи може міститись в одному місці, тобто бути єдиним, або в декількох, тобто бути розподіленим. Також, він може бути однаковим в кожній компоненті системи, тоді вся система буде децентралізованою. Розробляючи системи виявлення ЗПЗ потрібно враховувати, що зловмисники чи ЗПЗ обов’язково намагатимуться втрутитись в роботу таких спеціалізованих систем і, тому, досліджуватимуть місце знаходження центру системи. Використання децентралізованої архітектури для систем виявлення ЗПЗ є ефективним, але час на прийняття рішень в них є більшим, ніж для систем, які розроблені з використанням централізованої архітектури, що в сучасних умовах протікання процесів є суттєвою характеристикою ефективності функціонування. Крім того, дослідження виявлення ЗПЗ в окремих компонентах децентралізованих систем може бути постійним і це впливатиме на значну кількість компонент системи, в яких відсутнє ЗПЗ. Це в цілому завантажуватиме обчислювальні ресурси всіх комп’ютерних станцій в мережі. Тому, вибір архітектури для системи виявлення ЗПЗ здійснюватимемо з централізованої та гібридної (комбінованої, змішаної) архітектури. Для великої розподіленої в комп’ютерній мережі системи виявлення ЗПЗ наявність одного центру створює проблему, бо виведення його з ладу або комп’ютерної системи, в яку він встановлений, призведе до втрати роботи системи, як цілісної, так і компонент системи, які встановлені в решті комп’ютерних станцій і, при цьому, втратять ефективність виявлення ЗПЗ. Тому, доцільною для розгляду є централізована архітектура з частковою централізацією, що в сукупності має узгоджуватись та поєднуватись з самоорганізацією та адаптивністю системи в цілому. Часткову централізацію розглядатимемо як варіант архітектури гібридного (комбінованого, змішаного) типу, в якому центр прийняття рішень в системі буде розподілятись між невеликою кількістю компонентів та за потреби мігруватиме

повністю між ними. Така архітектурна організація щодо центру прийняття рішень в системі створить зловмисникам та ЗПЗ проблеми у його пошуку. Крім того, компоненти в системі підтримуватимуть зв’язки як горизонтально, тобто між собою, так і вертикально з компонентами, в яких буде центр прийняття рішень системи чи буде відбуватись переміщення центру. Кількість зв’язків за умови вимоги наявності горизонтальних і вертикальних зв’язків при децентралізованій архітектурі між компонентами системи суттєво більша, ніж при централізованій архітектурі. Пропонована частково централізована архітектура міститиме теж меншу кількість зв’язків порівняно з децентралізованою архітектурою, а з централізованою – більше приблизно в ту кількість, в яку виділено кількість компонент для переміщення центру системи. Для підтримки зв’язків між компонентами системи потрібні витрати ресурсів і особливо час. При втраті певних зв’язків витратиться час, також, на уточнення чи перепідтвердження з’єднань, що вплине на ефективність функціонування системи.

Часткова централізація, яка пропонується до реалізації в архітектурі системи, та її представлення через наявність горизонтальних зв’язків передбачатиме динамічну перебудову системи від децентралізації, в якій присутня часткова централізація до збільшення централізації за потреби. Наприклад, при вилученні частини компонентів, які не містять центру системи прийняття рішень, що збільшить співвідношення кількості компонентів з центром системи по відношенню до кількості компонентів без центру. Крім того, центром прийняття рішень системи може прийматись рішення про вилучення частини горизонтальних з’єднань між компонентами системи за потреби. Тому, рівень централізації в системі може змінюватись динамічно в залежності від стану системи в цілому в певний момент часу та функціонуючих процесів в комп’ютерних станціях та мережі.

Таким чином, в основу системи виявлення ЗПЗ застосуємо архітектуру, в якій буде синтезовано часткову централізацію, можливість до самоорганізації та адаптивності. Це дасть змогу створювати системи виявлення ЗПЗ такого класу, які не залежатимуть від користувача чи адміністратора при прийнятті рішень щодо своєї подальшої роботи чи наступних кроків, які матимуть змогу гнучко перебудовуватись та які будуть містити декілька компонент з центром прийняття рішень системи для покращення стійкості системи в цілому до впливів ЗПЗ. Крім

того, ці характерні можливості в архітектурі системи будуть узгоджуватись з конкретними методами виявлення ЗПЗ і вони залучатимуться до процесу виявлення з рівня архітектури системи.

Нехай S – проєктована частково централізована розподілена система. Її компоненти будуть розміщені у вузлах корпоративної мережі і всі разом в сукупності вони будуть формувати систему S. Задамо її так:

$$S = \{(S_1, 1), (S_2, 2), \dots, (S_N, N)\}, \quad (2.1)$$

де (S_i, i) – пара, яка відображає компоненту S_i в i – ій комп'ютерній станції; S_i - i – та компонента системи S; $i = 1, 2, \dots, N$; N – кількість компонент в системі S, які встановлені в комп'ютерні станції в мережі; другий елемент пари (S_i, i) позначає номер комп'ютерної станції.

Позначення компоненти S_i відображає компоненту з номером i, але при цьому номер комп'ютерній станції буде збіжним з номером компоненти. Якщо ж аналізувати компоненти за наявним в них функціоналом, то кількість компонент певного типу може бути меншою. Тоді, індекс компоненти буде використовуватись тільки для нумерування компонент. Таким чином, позначимо номер компоненти та номер комп'ютерної станції однаковими числами незалежно від позиціонування компоненти, її функціоналу, розміщення, перебування в активному чи пасивному станах, перебуванням у вимкненій чи увімкненій комп'ютерній станції.

Архітектуру системи S зобразимо схемою її основних складових компонентів з виділенням компонент, в яких може бути розміщений центр прийняття рішень системи, на рис. 2.1.

Частина компонент системи S міститиме центр системи або його частини. Також, в певних компонентах центр може переміщуватись. Решта компонент системи не міститиме центр системи, але за потреби матиме таку можливість. Тобто, не обов'язково всі компоненти системи одночасно міститимуть центр системи. Інакше, якщо в поточний момент часу всі компоненти системи міститимуть центр системи, то вона буде вироджена в децентралізовану систему. Якщо ж центр системи в певний момент часу буде в одній компоненті, то вона стане централізованою. Таким чином, центр системи може бути або в одній компоненті і переміщуватись, а може за потреби розподілятись між декількома компонентами.

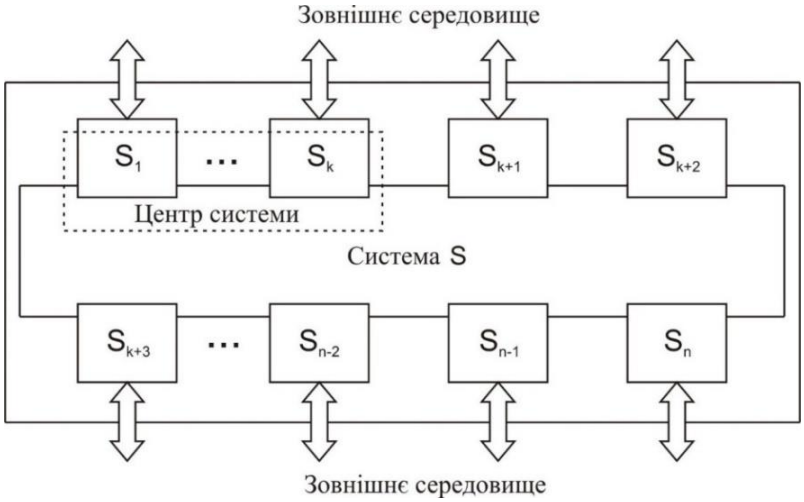


Рисунок 2.1 - Схема архітектури системи S з виділеними компонентами

Також, виходячи із таких варіантів архітектурних особливостей центру, вся система може бути частково централізованою, але у граничних випадках може вироджуватись у централізовану або децентралізовану.

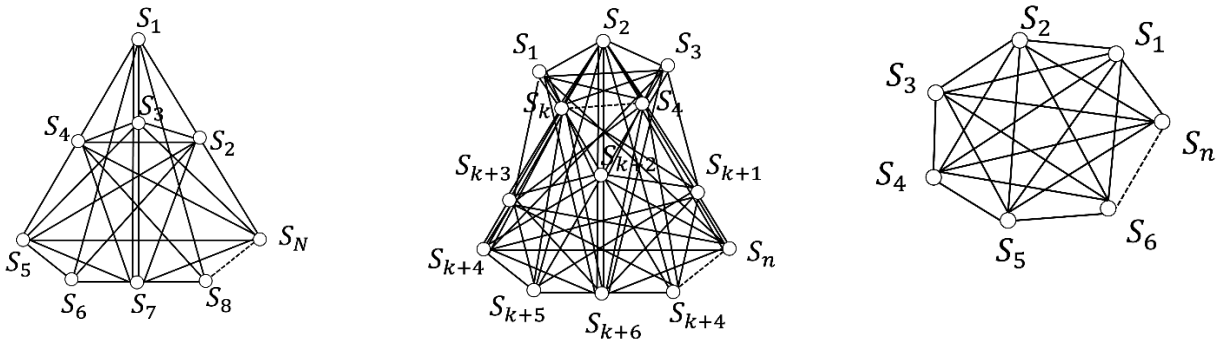
Тоді, задамо систему S з врахуванням двох типів її компонент так:

$$S = \{(S_1, 1), (S_2, 2), \dots, (S_k, k), (S_{k+1}, k+1), \dots, (S_N, N)\}, \quad (2.2)$$

де (S_i, i) – пара, яка відображає компоненту S_i в i – ій комп'ютерній станції; S_i - i – та компонента системи S; $i = 1, 2, \dots, N$; N – кількість компонент в системі S, які встановлені в комп'ютерні станції в мережі; другий елемент пари (S_i, i) позначає номер комп'ютерної станції; компоненти з 1 по k містять центр системи; компоненти з $k+1$ до N не містять центр системи.

Якщо $k = N$, тоді система стає децентралізованою. Якщо $k = 1$, тоді система стає централізованою. Якщо в системі S центр системи буде в одній компоненті, тоді вона стає централізованою, а якщо центр системи переміщується між декількома компонентами або розподіляється між декількома компонентами, тоді така система стає k – централізованою і містить більше зв'язків між компонентами додатково для центру. Тому, для проєктованої системи вибір частково централізованої архітектури базуватиметься для значення k так: $1 < k < N$. Представимо систему графом згідно формули (2.2), враховуючи також горизонтальні з'єднання між компонентами. На рис. 2.2 зображено трьома графами можливі архітектури для проєктованої системи. На рис. 2.2 а) зображена централізована архітектура, на рис. 2.2 в) – децентралізована архітектура, а на рис. 2.2 б) – пропонована для проєктованої системи частково централізована архітектура. Такий варіант архітектури, як

пропонується на рис. 2.2 б) має універсальність, що дозволяє від нього перейти до архітектури з рис. 2.2 а) та рис. 2.2 в). Це дозволить гнучко в динамічному режимі адаптувати систему під поточні завдання, які виникатимуть в комп'ютерній мережі при впливах зловмисників чи ЗПЗ.



а) б) в)

Рисунок 2.2 - Архітектура систем: а) централізована; б) частково централізована; в) децентралізована

Для фіксованої кількості компонент в системі, наприклад N, кількість вертикальних і горизонтальних зв'язків для всіх трьох варіантів архітектури з рис. 2.2 подано в табл. 2.1.

Таблиця 2.1

Кількість зв'язків в різних типах архітектури

Тип архітектури системи, значення k	Тип зв'язку, кількість з'єднань		
	вертикальний	горизонтальний	разом
централізована, $k = 1$	$N - 1$	$\frac{(N-1)\cdot(N-2)}{2}$	$\frac{(N-1)\cdot(N-2)}{2} + N - 1$
частково централізована, $k = 2$	$2 \cdot (N-2)$	$\frac{(N-2)\cdot(N-3)}{2} + 1$	$2 \cdot (N-2) + \frac{(N-2)\cdot(N-3)}{2} + 1$
частково централізована, $k = 3$	$3 \cdot (N-3)$	$\frac{(N-3)\cdot(N-4)}{2} + 3$	$3 \cdot (N-3) + \frac{(N-3)\cdot(N-4)}{2} + 3$
частково централізована, k	$k \cdot (N-k)$	$\frac{(N-k)\cdot(N-k-1)}{2} + \frac{(k-1)\cdot k}{2}$	$k \cdot (N-k) + \frac{(N-k)\cdot(N-k-1)}{2} + \frac{(k-1)\cdot k}{2}$
децентралізована, $k = N$	-	$\frac{(N-1)\cdot N}{2}$	$\frac{(N-1)\cdot N}{2}$

Таким чином, дані з табл. 2.1 щодо кількості зв’язків в різних типах архітектури підтверджують можливість ефективної реалізації системи згідно частково централізованої архітектури, бо кількість зв’язків в ній хоча і буде більша, але не перевищує суттєво за кількістю зв’язків інші типи архітектури. Всі розглянуті типи архітектури визначаються квадратичною залежністю кількості зв’язків, що є прийнятним для їх практичної реалізації з врахуванням обчислювальної складності за кількістю елементарних операцій для підтримки з’єднань і передачі повідомлень між компонентами системи.

Тоді, модель M_S частково централізованої розподіленої системи S задамо згідно її компонентів та зв’язків між ними так:

$$M_S = \langle S, G_S \rangle, \tag{2.3}$$

де G_S – граф з рис. 2.2 б), що відображає зв’язки між компонентами частково централізованої розподіленої системи S.

Враховуючи поділ компонент системи на дві підмножини за критерієм наявності центру в них і без нього, отримуємо уточнену модель $M_{S,k}$ системи S згідно формули:

$$M_{S,k} = \langle (\langle S_1, S_2, \dots, S_k \rangle, \langle S_{k+1}, S_{k+2}, \dots, S_N \rangle), G_S \rangle, \tag{2.4}$$

де G_S – граф з рис. 2.2 б), що відображає зв’язки між компонентами системи S; k – кількість компонент системи, в яких може бути центр прийняття рішень системи; $S = \{(S_1, 1), (S_2, 2), \dots, (S_k, k), (S_{k+1}, k+1), \dots, (S_N, N)\}$ (формула (2.2)); N – кількість компонент в системі, які встановлені в комп’ютерні станції в мережі.

В запропонованій, таким чином, моделі $M_{S,k}$ частково централізованої розподіленої системи, крім розподілу компонент системи, які задано підмножинами і відповідно вершинами, в залежності від можливостей містити центр, виділено також три типи зв’язків між компонентами, які поєднують компоненти з центром системи, компоненти без центру системи та компоненти з центром і без центру між собою. Граф G_S при такому заданні є повним, тобто з’єднання між компонентами система наявні між ними усіма. Але для ефективнішої роботи та приховування можливостей системи з’єднання між компонентами системи можуть бути задані різними деревами графа G_S і, таким чином, їх кількість буде зменшено, а також,

приховуватимуться від зловмисника або ЗПЗ очікувані повідомлення. Визначення варіантів дерев графа G_S встановлюватиметься центром прийняття рішень системи.

Задамо варіанти з'єднань між різного типу компонентами системи через відповідні їм функції. Таких функцій буде три і будуть, також, відповідні їм обернені. Визначення функцій необхідно для подальшої деталізації моделі системи та розроблення відповідного протоколу комунікації між ними. Функції задамо так:

$$F_1: S_i \overset{G_S}{\rightarrow} S_j, 0 < i, j \leq k, i \neq j;$$
$$F_2: S_i \overset{G_S}{\rightarrow} S_j, 0 < i \leq k, k < j \leq N; \quad (2.5)$$
$$F_3: S_i \overset{G_S}{\rightarrow} S_j, \ \ k < i, j \leq N, i \neq j,$$

де S_i - i – та компонента системи; j – та компонента системи; k – кількість компонент системи, в яких може бути центр системи; $S = \{(S_1, 1), (S_2, 2), \dots, (S_k, k), (S_{k+1}, k+1), \dots, (S_N, N)\}$ (формула (2.2)); N – кількість компонент в системі, які встановлені в комп'ютерні станції в мережі; G_S – граф з рис. 2.2 б), що відображає зв'язки між компонентами системи S.

Тоді, введемо функцію F, яку задамо як множину всіх функцій, які встановлюватимуть різні варіанти з'єднань між різного типу компонентами системи, за формулою:

$$F = \{F_1, F_2, F_3, F_1^{-1}, F_2^{-1}, F_3^{-1}\}, \quad (2.6)$$

де F_q – функція встановлення зв'язку між компонентами згідно формул (2.5); $q = 1, 2, 3$; F_q^{-1} – обернена функція до функції F_q, що відображає зворотнє виконання дій між компонентами системи.

Введемо множину предикатів P для бінарного представлення результатів виконання функцій з множини F. Такий результат, отриманий з множини предикатів надасть змогу встановлювати архітектуру дерева в графі G_S, яка використовуватиметься для визначення подальших кроків системи. Множину предикатів P задамо за формулою:

$$P = \{P_1, P_2, P_3, P_1^{-1}, P_2^{-1}, P_3^{-1}\}, \quad (2.7)$$

де P_s та P_l^{-1}– предикати, що відображають успішність/неуспішність виконання функцій встановлення зв'язку між компонентами з множини F згідно формули (2.5);

$s = 1, 2, 3;\ l = 1, 2, 3;$ предикати P_l^{-1} не є зворотніми до предикатів P_s як для відповідних функцій F_q та F_q^{-1}, а є лише відображенням їх успішного/неуспішного виконання; F_q^{-1} обернена функція до функції F_q, що відображає зворотнє виконання дій між компонентами системи, виконання яких може бути і неуспішним; $q = 1, 2, 3$.

Предикати P_l^{-1}, $l = 1, 2, 3$ можуть бути відображеннями результатів виконання обернених функцій F_q^{-1}, $q = 1, 2, 3$, які можуть бути виконані чи можуть виконуватись одночасно з функціями F_q, тобто всі шість функцій на певному кроці системи можуть виконуватись. Побудова системи з так заданими функціями збільшує кількість варіантів в системі для визначення її подальших кроків, що також фіксується набором предикатів. Таким чином, з використанням опису компонентів системи як її підмножин, введеними функціями для відображення дій між компонентами системи та предикатів для відображення успішного/неуспішного виконання функцій отримуємо опис всієї системи S.

Визначення архітектури частково централізованої системи S її моделлю заданою формулами (2.3) і (2.4) та деталізацією щодо функцій і предикатів за формулами (2.5)-(2.7) відповідає вимогам щодо можливості динамічної зміни конфігурації, поділу центру прийняття рішень, розподілу компонентів за можливостями з наявності центру прийняття рішень в них та зображеної архітектури з рис. 2.2 б) у вигляді графа G_S і, тому, є основою для подальшого синтезу в ній властивостей адаптивності і самоорганізації, реалізація яких буде здійснена безпосередньо в компонентах системи, в основному тих з них, в яких знаходитиметься центр прийняття рішень системи. Тобто, для можливості системи із здійснення самоорганізації та адаптування в залежності від оточуючого середовища та процесів в комп'ютерних системах і мережах необхідно синтезувати ці властивості саме в центрі прийняття рішень системи.

2.2. Архітектура компонентів частково централізованих розподілених систем виявлення зловмисного програмного забезпечення

Синтез вимог щодо адаптивності і самоорганізації в частково централізованих системах виявлення ЗПЗ потребує його реалізації в центрі прийняття рішень як його частини. Також, центрів прийняття рішень в системі може бути декілька і вони

узгоджуватимуть між собою подальші кроки системи. Тоді, ці вимоги будуть окремо в кожній з компонент, що містить центр системи. Оскільки, центр прийняття рішень системи знаходиться в частині компонент системи, тоді розглянемо архітектуру компонент системи.

Компоненти частково централізованих розподілених систем S згідно графа з рис. 2.2 б) розподілені на два типи. Розглянемо їх архітектури окремо та встановимо можливу наявність в них центру прийняття рішень системи, враховуючи їх призначення в системі. Сформуємо архітектуру компоненти системи узагальнено для двох типів компонент, оскільки ці компоненти матимуть однакові функції, бо є компонентами однієї системи, і відрізнятимуться тільки додатковою наявністю функції центру. Компоненти, які міститимуть центр прийняття рішень системи, можуть в певні проміжки часу не бути такими, в яких приймаються рішення, а виконувати функції компонент без центру прийняття рішень. До складу компоненти введемо функцію для забезпечення функціонування компоненти в окремому вузлі комп'ютерної мережі, функцію F (формула (2.6)) для встановлення різних варіантів з'єднань між різного типу компонентами системи, функцію обробки зовнішніх повідомлень від решти компонент системи S, функцію прийняття рішень в компоненті системи щодо подальших кроків функціонування системи, функцію з виявлення ЗПЗ, функцію для забезпечення прийняття рішень в системі S, функцію узгодження прийнятого рішення з рештою компонент з центром системи, функцію для повідомлення всім компонентам системи S щодо подальших кроків. Ці функції компоненти міститимуть, також, функції для виконання основних завдань за призначенням та допоміжних супровідних завдань. Таким чином, вони задаватимуться як множини функцій за певним призначенням і розподілятимуться на підмножини функцій для виконання конкретних завдань. Тому, функції компоненти системи S розподілимо на два рівні: функції-множини; функції-підмножини. Верхній рівень міститимуть функції-множини. Нижній рівень міститимуть функції-підмножини. Задамо компоненту системи S переліком її функцій-множин так:

$$\Psi_{S_i} = \bigcup_{j=1}^{N_{S_i}} \Psi_{S_i,j}, \tag{2.8}$$

де Ψ_{S_i} – сукупність (об'єднання) функцій-множин S_i компоненти системи S; i –

номер компоненти системи; $i = 1, \dots, N$; N – кількість компонент в системі, які встановлені в комп'ютерні станції в мережі; $\Psi_{S_i,j} - j$ - та функція-підмножина, яка забезпечує виконання одного функційного завдання в i - тій компоненті; $j = 1,2, \dots, N_{S_i}$; N_{S_i} – кількість функцій-підмножин в S_i компоненті, які об'єднано в функції-множині Ψ_{S_i}.

Зокрема, нехай $\Psi_{S_i,1}$ - функція для забезпечення функціонування компоненти в окремому вузлі комп'ютерної мережі, $\Psi_{S_i,2}$ - функція (функція F; формула (2.6)) для встановлення різних варіантів з'єднань між різного типу компонентами системи, $\Psi_{S_i,3}$ - функція обробки зовнішніх повідомлень від решти компонентів системи S, $\Psi_{S_i,4}$ - функція прийняття рішень в компоненті системи щодо подальших кроків функціонування системи, $\Psi_{S_i,5}$ - функція для виявлення ЗПЗ, $\Psi_{S_i,6}$ - функція для забезпечення прийняття рішень в системі S, $\Psi_{S_i,7}$ - функція узгодження прийнятого рішення з рештою компонент з центром системи, $\Psi_{S_i,8}$ - функція для повідомлення всім компонентам системи S щодо подальших кроків та інші функції, які розширюють можливості компоненти системи у вузлі комп'ютерної мережі.

Функції-множини поділимо на два типи. До першого типу віднесемо ті з них, в яких містяться всі функції-підмножини, які передбачені для компонентів системи і містять засоби центру прийняття рішень в компоненті. До другого типу віднесемо ті з них, в яких присутні всі функції-підмножини, крім тих, які відповідають за формування центру прийняття рішень системи в компоненті.

Розподілимо функції-підмножини на три групи. До першої групи віднесемо функції-підмножини, які відповідають в компоненті за прийняття рішень. Наприклад, це функції $\Psi_{S_i,2}$, $\Psi_{S_i,4}$, $\Psi_{S_i,6}$, $\Psi_{S_i,7}$. До другої групи віднесемо функції-підмножини, що забезпечують підтримку функціонування компоненти системи у вузлі комп'ютерної мережі. Наприклад, це функції $\Psi_{S_i,1}$, $\Psi_{S_i,3}$, $\Psi_{S_i,8}$. До третьої групи віднесемо функції-підмножини, які виконують спеціалізовані завдання з виявлення ЗПЗ. Зокрема, це функція $\Psi_{S_i,5}$. Такий поділ функцій-підмножин на групи надає змогу створити в компонентах системи, в яких може функціонувати центр системи, базу відомостей про стани виконання функцій-підмножин в усіх компонентах, які враховуватимуться при визначенні рішення про подальші кроки системи.

Задамо матрицею уточнену модель $M_{S,k,\Psi}$ системи S згідно наявних в ній

функцій-підмножин в компонентах системи з врахуванням формул (2.4) та (2.8):

$$M_{S,k,\Psi} = \begin{pmatrix} \Psi_{S_1,1} & \cdots & \Psi_{S_k,1} & \cdots & \Psi_{S_N,1} \\ \Psi_{S_1,2} & \cdots & \Psi_{S_k,2} & \cdots & \Psi_{S_N,2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \Psi_{S_1,N_{S_{max}}} & \cdots & \Psi_{S_k,N_{S_{max}}} & \cdots & \Psi_{S_N,N_{S_{max}}} \end{pmatrix}, \quad (2.9)$$

де $\Psi_{S_k,j}$ – j - функція-підмножина, яка входить до складу функції-множини Ψ_{S_k} компоненти S_k; $k = 1,2,\dots,N$; N – кількість компонент в системі, які встановлені в комп'ютерні станції в мережі; k – кількість компонент системи, в яких може бути центр системи; $S = \{(S_1,1),(S_2,2),\dots,(S_k,k),(S_{k+1},k+1),\dots,(S_N,N)\}$ (формула (2.2)); $j = 1,2,\dots,N_{S_{max}}$; $N_{S_{max}}$ – найбільша кількість функцій-підмножин, яку можна встановити в компоненті S_k.

Частина розроблених для компонент функцій-підмножин може не бути встановленою. Тобто, в компоненті системи може бути менше функцій. Зокрема, наприклад, можуть бути відсутніми функції, які забезпечують прийняття рішень системою, тобто в компоненті відсутній центр прийняття рішень чи його частина. Відсутність функції в компоненті системи у певному вузлі в мережі задаватимемо нуль-функцією.

Для матриці, яку задано формулою (2.9) введемо бітову матрицю, що відображатиме наявність функцій-підмножин, з яких сформовані компоненти системи. Відсутність в компонентах функцій-підмножин задамо $\{0\}$, наявність - $\{1\}$. Тоді, бітову матрицю задамо так:

$$P(M_{S,k,\Psi}) = \begin{pmatrix} P(\Psi_{S_1,1}) & \cdots & P(\Psi_{S_k,1}) & \cdots & P(\Psi_{S_N,1}) \\ P(\Psi_{S_1,2}) & \cdots & P(\Psi_{S_k,2}) & \cdots & P(\Psi_{S_N,2}) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ P(\Psi_{S_1,N_{S_{max}}}) & \cdots & P(\Psi_{S_k,N_{S_{max}}}) & \cdots & P(\Psi_{S_N,N_{S_{max}}}) \end{pmatrix}, \quad (2.10)$$

де $P(\Psi_{S_k,j}) = \begin{cases} 0, \text{якщо функція } \Psi_{S_k,j} \text{ відсутня в компоненті;} \\ 1, \text{якщо функція } \Psi_{S_k,j} \text{ наявна в компоненті.} \end{cases}$, $\Psi_{S_k,j}$ – j - функція-підмножина, яка входить до складу функції-множини Ψ_{S_k} компоненти S_k; $k = 1,2,\dots,N$; N – кількість компонент в системі, які встановлені в комп'ютерні станції в мережі; k – кількість компонент системи, в яких може бути центр системи; $S = \{(S_1,1),(S_2,2),\dots,(S_k,k),(S_{k+1},k+1),\dots,(S_N,N)\}$ (формула (2.2)); $j = 1,2,\dots,N_{S_{max}}$; $N_{S_{max}}$ – найбільша кількість функцій-підмножин, яка може бути

встановлена в компоненті S_k.

Також, для матриці, яку задано формулою (2.9) введемо матрицю, яка відображатиме активність функцій-підмножин в поточний момент часу та їх наявність, так:

$$P_t(M_{S,k,\Psi}) = \begin{pmatrix} P_t(\Psi_{S_1,1}) & \cdots & P_t(\Psi_{S_k,1}) & \cdots & P_t(\Psi_{S_N,1}) \\ P_t(\Psi_{S_1,2}) & \cdots & P_t(\Psi_{S_k,2}) & \cdots & P_t(\Psi_{S_N,2}) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ P_t(\Psi_{S_1,N_{S_{max}}}) & \cdots & P_t(\Psi_{S_k,N_{S_{max}}}) & \cdots & P_t(\Psi_{S_N,N_{S_{max}}}) \end{pmatrix}, \quad (2.11)$$

де $P_t(\Psi_{S_k,j}) = \begin{cases} 0, \text{якщо функція } \Psi_{S_k,j} \text{ відсутня в компоненті } S_k; \\ 1, \text{якщо функція } \Psi_{S_k,j} \text{ наявна в компоненті та неактивна;} \\ 2, \text{якщо функція } \Psi_{S_k,j} \text{ наявна в компоненті та виконується.} \end{cases}$,

$\Psi_{S_k,j}$ – j - функція-підмножина, яка входить до складу функції-множини Ψ_{S_k} компоненти S_k; $k = 1,2,\dots,N$; N – кількість компонент в системі, які встановлені в комп'ютерні станції в мережі; k – кількість компонент системи, в яких може бути центр системи; $S = \{(S_1,1),(S_2,2),\dots,(S_k,k),(S_{k+1},k+1),\dots,(S_N,N)\}$ (формула (2.2)); $j = 1,2,\dots,N_{S_{max}}$; $N_{S_{max}}$ – найбільша кількість функцій-підмножин, яка може бути встановлена в компоненті S_k.

Бітова карта активних функцій згідно формули (2.11) задає уточнену модель $M_{S,k,\Psi}$ системи S згідно наявних в ній функцій-підмножин на рівні їх виконання в поточний момент часу. Функції можуть виконуватись псевдопаралельно, тобто можуть одночасно в одній компоненті виконуватись дві і більше функцій. Таким чином, матриця згідно формули (2.11) відображатиме поточний стан виконання функцій в усіх компонентах. Тобто, в ній буде актуальна інформація про стан компоненти та компонент в цілому. Така інформація буде використовуватись центром прийняття рішень системи.

В матриці (формула (2.11)) перші k стовпців відображають активність функцій-підмножин центру прийняття рішень системи в поточний момент часу. Центр системи може міститись безпосередньо в усіх k компонентах і виконувати завдання залучаючи всі свої частини з k компонент. При цьому може відбуватись дублювання виконання поставлених завдань в різних компонентах з подальшим аналізом отриманих результатів, опрацюванням їх для формування та фіксації результуючого рішення. Або в k компонентах можуть залучатись до вирішення поставлених перед

системою завдань лише певні функції-підмножини, тобто не всі однакові функції-підмножини в різних компонентах будуть одночасно виконуватись. Порядок їх залучення в цьому випадку визначатиметься цим же центром прийняття рішень системи. Центр прийняття рішень системи може бути активним не одночасно в усіх k компонентах у вузлах в мережі, а в меншій кількості компонент. При цьому функціонал для активації центру прийняття рішень міститься відповідно в усіх k компонентах системи і активується системою в процесі її функціонування. Така активізація функціоналу, який відповідає за прийняття рішень системою, в компоненті надає змогу в процесі функціонування системи здійснювати переміщення центру прийняття рішень системи між різними компонентами системи із визначених k компонент. Так організоване переміщення центру прийняття рішень з розподіленням між компонентами, а також з можливістю переміщення повністю в одну з компонент, є частиною синтезу в системі властивості адаптивності. Міграція центру прийняття рішень системи покращуватиме її стійкість в процесі функціонування та за наявності загроз. Сформуємо множину з можливих різноманітних варіантів розміщення центру прийняття рішень системи в залежності від кількості компонент в системі та функцій-підмножин, що відносяться до функцій забезпечення функціонування центру прийняття рішень. На цьому етапі формування множини не враховуватимемо рівні безпеки в компонентах, в яких міститиметься центр прийняття рішень системи. Врахування рівнів безпеки в цих компонентах буде здійснено при виборі варіантів розміщення центру прийняття рішень в процесі функціонування системи.

Тоді, нехай m – кількість активних компонент в системі, в яких в поточний момент часу функціонує центр прийняття рішень системи. Крім того, $2 \leq m \leq k$, де k – кількість компонент системи, в яких може бути центр системи. Кількість функцій-підмножин, які формуватимуть повну функційну складову частину центру прийняття рішень в компоненті в системі, нехай дорівнює $n_{S_k,max}$, причому $1 \leq n_{S_k,max} < N_{S_{max}}$, де $N_{S_{max}}$ – найбільша кількість функцій-підмножин, яка може бути встановлена в компоненті S_k. Матрицю, яка буде відображати активність функцій-підмножин, що забезпечують функціонування центру прийняття рішень, в поточний момент часу та їх наявність, згідно формули (2.11) задамо так:

$$P_{t,n_{S_k,max}}(M_{S,k,\Psi,}) = \begin{pmatrix} P_t(\Psi_{S_1,1}) & \cdots & P_t(\Psi_{S_k,1}) \\ P_t(\Psi_{S_1,2}) & \cdots & P_t(\Psi_{S_k,2}) \\ \cdots & \cdots & \cdots \\ P_t(\Psi_{S_1,n_{S_k,max}}) & \cdots & P_t(\Psi_{S_k,n_{S_k,max}}) \end{pmatrix}, \quad (2.12)$$

де $P_t(\Psi_{S_k,j}) = \begin{cases} 0, \text{якщо функція } \Psi_{S_k,j} \text{ відсутня в компоненті } S_k; \\ 1, \text{якщо функція } \Psi_{S_k,j} \text{ наявна в компоненті та неактивна;} \\ 2, \text{якщо функція } \Psi_{S_k,j} \text{ наявна в компоненті та виконується.} \end{cases}$,

$\Psi_{S_k,j}$ – j - функція-підмножина, яка входить до складу функції-множини Ψ_{S_k} компоненти S_k; k – кількість компонент системи, в яких може бути центр системи.

Для формування множини варіантів розміщення центру прийняття рішень системи розділимо на підмножини типові з них. Тоді, задамо підмножини варіантів розміщення центру прийняття рішень системи так:

1) підмножина Q_1 містить варіанти з вибором будь-яких m компонент з наявних активних компонентів з всеможливих k компонентів;

2) підмножина Q_2 містить m сформованих динамічних компонент з функцій різних k компонент без повторення в різних компонентах функцій-підмножин певної компоненти, причому m статичних компонент, які будуть наповнюватись функціями-підмножинами решти компонент, вибиратимуться з наявних активних компонентів з можливих k компонентів.

Наприклад, один з варіантів зображення динамічної компоненти з функцій-підмножин різних k компонент подано на рис. 2.3.

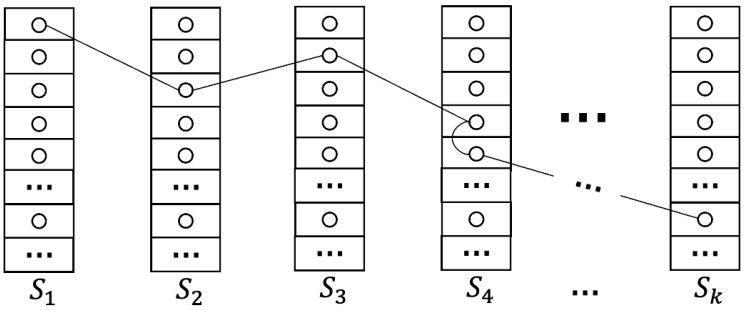


Рисунок 2.3 - Приклад одного варіанту динамічної компоненти з функцій-підмножин різних k компонент

Функції-підмножини зображено на рис. 2.3 вершинами, а зв’язки між ними ребрами. На рис. 2.4 зображення всіх сформованих динамічних компонент у вигляді лісу дерев.

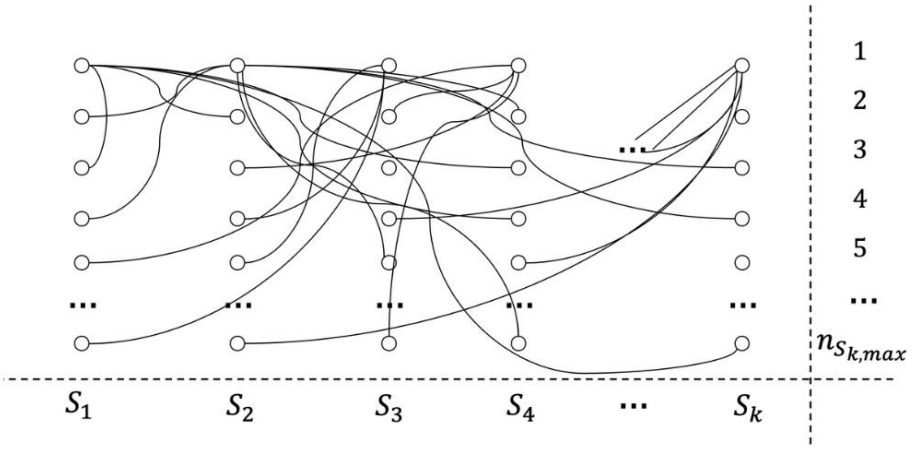


Рисунок 2.4 - Приклад сформованих динамічних компонент у вигляді лісу дерев

Формування динамічних компонент, в яких міститиметься центр прийняття рішень системи, здійснено у вигляді дерев, а разом вони формують ліс. Зв'язки між цими деревами як компонентами встановлюються в іншій підсистемі компоненти, яка не пов'язана безпосередньо із центром прийняття рішень та не приймає участі в формуванні рішення. Так сформовані динамічні компоненти дають змогу здійснювати незалежні розподілені обчислення, що дозволяє мінімізувати вплив щодо формування рішень, зокрема із зловмисною метою.

Множина всіх можливих варіантів розміщення центру прийняття рішень системи $Q = Q_1 \cup Q_2$. Кількість можливих варіантів для підмножини Q_1 визначатимемо так:

$$k_{Q_1} = \sum_{m=2}^{k} C(n,m) = \sum_{m=2}^{k} \frac{n!}{m! \cdot (n-m)!}, \quad (2.13)$$

де $C(n,m)$ – кількість комбінацій без повторень з n по m; $2 \le m \le k$; k – кількість компонент з функціоналом центру прийняття рішень; m – кількість компонент з активним функціоналом центру прийняття рішень.

Для підмножини Q_2 кількість можливих варіантів визначатиметься з урахуванням результатів згідно формули (2.13), оскільки на першому кроці визначатимуться статичні компоненти, а на другому кроці визначатиметься можлива кількість варіантів з вибору функцій-підмножин серед всіх наявних активних компонент, в яких вони встановлені. Кожну з функцій-підмножин можна вибрати серед всіх однакових функцій-підмножин різних компонент k способами. Їх треба вибрати m для встановленої кількості компонент для центру прийняття

рішень системи. В такий спосіб треба наповнити кожну з m компонент кількістю функцій-підмножин $n_{S_k,max}$, яка може бути встановлена в компонентах, що містять центр прийняття рішень системи. Тоді, кожну з функцій-підмножин можна вибрати кількісно як обчислення кількості за формулою, а оскільки їх може бути $n_{S_k,max}$, то кількість варіантів з врахуванням вибору компонент окремо і без врахування порядку входження компонент та функцій-підмножин в компоненти для підмножини Q_1 визначатиметься так:

$$k_{Q_2} = \left(n_{S_k,max} + 1\right) \cdot k_{Q_1} . \tag{2.14}$$

Серед кількості варіантів для підмножини Q_2 можливі як граничні випадки такі варіанти, які для підмножини Q_1. Тому, загальна кількість варіантів для множини Q визначатиметься з врахуванням формул (2.13) і (2.14) так:

$$k_Q = k_{Q_2} = \left(n_{S_k,max} + 1\right) \cdot \sum_{m=2}^{k} \frac{n!}{m! \cdot (n-m)!} , \tag{2.15}$$

де $n_{S_k,max}$ - кількість функцій-підмножин, які формуватимуть повну функціональну складову частину центру прийняття рішень в компоненті в системі.

В системі S елементи множини Q зберігатимуться в форматі двозв’язних списків. Формування цієї множини здійснюватиметься автоматично після встановлення всіх компонентів системи у вузлах в мережі. При зміні конфігурації системи в частині додавання нових компонентів чи вилучення наявних компонентів система переформовує множину Q. Наявність такої множини Q зі списків варіантів розміщення центру прийняття рішень в компонентах системи представимо базою варіантів, причому в кожному з тих компонентів, в яких можливе розміщення центру прийняття рішень системи. Це дасть змогу оперативно і незалежно від решти компонент центру прийняття рішень системи опрацьовувати наступний варіант свого розміщення в компонентах системи. Також, враховуючи специфіку виконуваних задач системи та її розподілення функції-підмножини повинні мати максимально допустиму модульність для можливості здійснення їх віддаленого запуску, тобто виконання процедури віддаленого запуску. Таким чином створені функції-підмножини та збереження інформації про них в двозв’язних списках необхідно для формування підмножини Q_2.

Події в системі S, зокрема і виконання функцій та отримання для обробки зовнішніх впливів, не будуть мати однаковий ступінь довіри, не будуть виконуватись в однакових умовах в різних вузлах в мережі, в яких встановленні компоненти, і не матимуть однакових можливостей при виконанні в одній комп'ютерній станції та згідно віддаленого виклику процедур. Тому, при кожному виконанні функцій в компонентах для формування остаточного результату повинна враховуватись довіра до проміжних результатів, яка може виражатись певною часткою або відсотком порівняно з результатом, отриманим в ідеальних умовах. Також, комп'ютерні станції в мережі, в які встановлені компоненти частково централізованої системи, та виконувані в них процеси можуть бути під впливом різних робочих навантажень, а також ЗПЗ, що призводитиме до отримання в системі значень виконуваних функцій системи з різними часовими термінами та за певних умов спотвореними результатами значень. В зв'язку з цим потребують оцінювання як комп'ютерні станції, компоненти системи, так і значення виконаних розподілених обчислень. Всі ці оцінювання повинні бути враховані центром прийняття рішень системи S. Для їх оцінювання введемо критерії, які формуватимуть рівень довіри до компонент системи, підсистем компонент, функцій-множин, функцій-підмножин та отриманих обчислених значень за результатами виконання функцій. Рівень довіри до результатів розподілених обчислень, які отримані з різних компонент системи S, може бути різним і ці значення повинні впливати на результуючі обчислення.

Задамо рівень довіри R_S до результатів обчислень, який використовуватиметься для розподілених обчислень безпосередньо в системі, із значеннями з проміжку $[0; 1]$, тобто $0 \leq R_S \leq 1$. Приймемо за більше значення з двох значень R_S за таке, в якому ступінь довіри результатів обчислень є більшим.

Компоненти системи S поділені на ті, в яких може бути центр прийняття рішень системи, та ті, в яких засоби його функціонування відсутні. Крім того, компоненти, в яких може бути центр прийняття рішень системи, поділяються на дві підмножини: центр прийняття рішень функціонує в компоненті; центр прийняття рішень не функціонує в компоненті. Тому, рівень довіри до результатів обчислень з компонент системи визначатимемо з урахуванням такого поділу. Нехай рівень довіри до результатів обчислень, які отримано з кожної компоненти, задано значеннями з

проміжку $[0; 1]$, тобто $0 \leq R_{S,S_k} \leq 1$, де S_k – компонента системи, $k = 1, 2, \dots, N$, N – кількість компонент в системі, які встановлені в комп'ютерні станції в мережі, k – кількість компонент системи, в яких може бути центр прийняття рішень системи.

Обчислення в конкретній компоненті можуть відбуватись з врахуванням належності функцій-підмножин підмножинам за трьома варіантами:

1) обчислення, які відносяться винятково до центру прийняття рішень системи і виконуються відповідними функціями-підмножинами центру прийняття рішень;

2) обчислення, які виконуються функціями-підмножинами, що не є функціями-підмножинами центру прийняття рішень системи;

3) обчислення, які виконуються функціями-підмножинами як центру прийняття рішень системи, так і рештою функцій.

Перший і другий варіант передбачають виконання функцій-підмножин саме для центру прийняття рішень в першому варіанті або для решти функцій-підмножин компоненти. Третій варіант передбачає виконання функцій-підмножин, які не відносяться до частини, яка формується з функцій-підмножин центру прийняття рішень, але після їх виконання і за потреби реагування на можливу наявність ЗПЗ, звертаються для додаткових обчислень до функцій-підмножин центру прийняття рішень. Результатом таких обчислень може бути, наприклад, значення, яке вимагатиме або змінити архітектуру системи, або місце розміщення центру прийняття рішень системи або виконати інші, закладені в систему, кроки. Варіант, в якому функції-підмножини здійснюють обчислення та після цього звертаються до певної чи певних функцій-підмножин, які не відносяться до центру прийняття рішень, вважатимемо таким, що задає послідовне виконання першого та другого варіантів і відповідно розподіляється. Виділення такого варіанту в окремий є недоцільним, бо він не викликає змін в системі в цілому за один такт, як це може відбуватись в третьому варіанті. Результати обчислень, які отримані за другим варіантом, надсилаються до центру прийняття рішень системи. Приклад виконання розподілених обчислень в різних динамічних компонентах системи із залученням функцій різних статичних компонентів системи зображено на рис. 2.5.

Для виконання конкретних завдань в динамічних компонентах можуть виконуватись функції-підмножини кількісно за різними варіантами так: одна функція-підмножина; декілька функцій-підмножин; всі функції-підмножини. При

цьому так само будуть виконуватись функції-підмножини, які формуватимуть центр прийняття рішень системи. Відповідно, в такому випадку такі варіанти виконання впливатимуть на рівень довіри до результатів обчислень і повинні бути враховані при його визначенні.

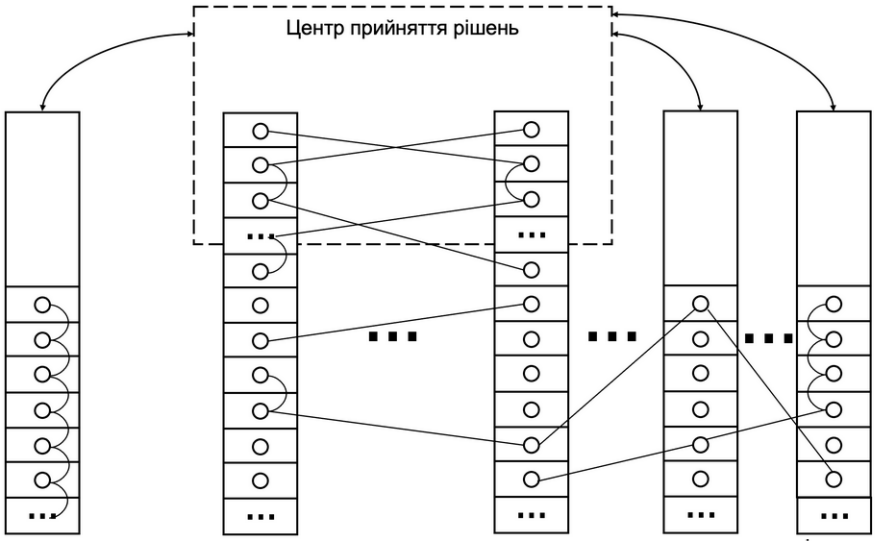


Рисунок 2.5 - Варіант частково сформованих динамічних компонент системи

Задамо матрицею коефіцієнти ваги функцій-підмножин в компонентах, в яких може бути розміщений центр прийняття рішень системи, так:

$$M_{\alpha_{1,S_k}} = \begin{pmatrix} \alpha_{1,S_1,1} & \cdots & \alpha_{1,S_k,1} \\ \vdots & \ddots & \vdots \\ \alpha_{1,S_1,n_{S_k,max}} & \cdots & \alpha_{1,S_k,n_{S_k,max}} \end{pmatrix}, \quad (2.16)$$

де $\alpha_{1,S_i,j}$ – коефіцієнт ваги функцій-підмножин в компонентах, в яких може бути розміщений центр прийняття рішень системи; $j = 1, 2, \dots, n_{S_k,max}$; $n_{S_k,max}$ - кількість функцій-підмножин, які формуватимуть повну функціональну складову частину центру прийняття рішень в компоненті в системі; $i = 1, 2, \dots, k$; k – кількість компонент системи, в яких може бути центр прийняття рішень системи.

Якщо функція-підмножина входить до динамічної компоненти і знаходиться в тій же статичній компоненті, тоді значення коефіцієнта ваги для неї встановимо таким, що дорівнює одиниці $\alpha_{1,S_i,j} = 1$. Якщо функція-підмножина входить до динамічної компоненти і знаходиться в іншій статичній компоненті, тоді значення коефіцієнта ваги для неї встановимо в інтервалі $(0; 1)$: $0 < \alpha_{1,S_i,j} < 1$. Якщо функція-підмножина входить до складу динамічної компоненти, але не була задіяна для виконання завдання, тоді значення коефіцієнта встановимо таким, що дорівнює

нулеві $\alpha_{1,S_i,j} = 0$. Таким чином, при виконанні певного завдання щодо прийняття рішень в системі в компонентах будуть активізовані на виконання певні функції і для цих компонент буде сформовано матрицею коефіцієнтів ваги функцій-підмножин в компонентах, значення яких будуть використовуватись при визначенні рівня довіри до результатів обчислень.

Визначимо рівень довіри r_{1,S_i} до результатів обчислень при виконанні завдання в одній з динамічних компонент S_i системи, в якій знаходиться центр прийняття рішень, так:

$$r_{1,S_i} = \frac{\sum_{j=1}^{n_{S_k,max}} \alpha'_{1,S_i,j}}{n'_{1,S_i}}, \tag{2.17}$$

де $\alpha'_{1,S_i,j}$ - коефіцієнт ваги функції-підмножини динамічної компоненти; n'_{1,S_i} - кількість функцій-підмножин, що були виконані при обчисленнях; $j = 1, 2, \dots, n_{S_k,max}$; $n_{S_k,max}$ - кількість функцій-підмножин, які формуватимуть повну функціональну складову частину центру прийняття рішень в компоненті в системі; $i = 1, 2, \dots, k$; k – кількість компонент системи, в яких може бути центр прийняття рішень системи; $i <= k$.

Якщо всі функції-підмножини певної динамічної компоненти S_i знаходяться в одному і тому ж вузлі в мережі, тоді всі коефіцієнти ваг функцій-підмножин, які виконані для реалізації певного завдання, $\alpha'_{1,S_i,j} = 1$. Отже, в такому випадку за формулою (2.17) $r_{1,S_i} = 1$. Якщо, наприклад, при реалізації певного завдання виконана всього одна функція-підмножина (позначимо її p), яка знаходиться в іншій, відмінній від S_i компоненти, компоненті, тоді $r_{1,S_i} = \alpha'_{1,S_i,p}$. Якщо для реалізації певного завдання було виконано дві функції-підмножини, одна з яких міститься в тій же компоненті S_i, а друга - в іншій (позначимо її p), тоді $r_{1,S_i} = \frac{1+\alpha'_{1,S_i,p}}{2}$. Таким чином, результат обчислення в певній компоненті системи буде мати оціночне значення в частині рівня довіри до нього з врахуванням відповідних коефіцієнтів ваг функцій-підмножин.

Архітектура компонент системи S деталізована до їх варіантів з статичних та динамічних компонент, рівня функцій, рівнів довіри до результатів розподілених обчислень, що дало змогу задати матрицею уточнену модель $M_{S,k,\Psi}$ системи S згідно

наявних в ній функцій. Відомості з такої матриці та рівні довіри до результатів обчислень у компонентах необхідні для аналізу центром прийняття рішень системи щодо її подальших кроків.

2.3. Характеристичні показники безпеки компонентів та середовища корпоративної мережі

Оточуюче середовище корпоративної мережі буде впливати на функціонування системи S та її компонентів. Тому, його потрібно подати в формалізованому вигляді з врахуванням усіх особливостей та характерних ознак для його подальшого аналізу центром прийняття рішень системи. Характерні ознаки будуть впливати на довіру до результатів розподілених обчислень в компонентах, тому деталізуємо їх саме в контексті визначення оціночного значення рівня довіри до нього з врахуванням відповідних коефіцієнтів ваг функцій-підмножин.

Враховуватимемо в значеннях коефіцієнтів порядок виконання функцій-підмножин в динамічних компонентах, час витрачений на пересилання результатів виконання, рівень безпеки вузла, виконання функцій-підмножин в компонентах з неактивними підсистемами центру прийняття рішень системи, кількість функцій-підмножин, що приймали участь у виконанні завдання. Тоді, коефіцієнти ваг функцій-підмножин в динамічних компонентах S_i задамо як функції з п’ятьма аргументами так:

$$\alpha'_{1,S_i} = f_{\alpha'_{1,S_i}}\left(\alpha'_{1,S_i,1}, \alpha'_{1,S_i,2}, \alpha'_{1,S_i,3}, \alpha'_{1,S_i,4}, \alpha'_{1,S_i,5}\right), \quad (2.18)$$

де $\alpha'_{1,S_i,1}$ - значення, яке враховує порядок виконання функцій-підмножин в динамічних компонентах; $\alpha'_{1,S_i,2}$ - значення, яке враховує відносний час витрачений на пересилання результатів виконання; $\alpha'_{1,S_i,3}$ - значення, яке враховує рівень безпеки вузла в мережі; $\alpha'_{1,S_i,4}$ - значення, яке враховує виконання функцій-підмножин в компонентах з неактивними підсистемами центру прийняття рішень системи; $\alpha'_{1,S_i,5}$ - значення, що враховує кількість функцій-підмножин, які приймали участь у виконанні завдання; S_i - i – та компонента системи; $i = 1, 2, \dots, k$; k – кількість компонент системи, в яких може бути центр прийняття рішень системи; $i <= k$.

Значення $\alpha'_{1,S_i,1}$ враховує порядок виконання функцій-підмножин в динамічних компонентах і залежить від порядку виконання конкретної j функції-підмножини в загальному порядку виконання функцій-підмножин. Припустимо, що для виконання поставленого завдання були задіяні всі функції-підмножини, які формують центр прийняття рішень системи і всі вони знаходились в одному вузлі в мережі. Кількість функцій-підмножин визначається значенням $n_{S_k,max}$, тоді вектором $v_{\alpha'_{1,S_i}}$ задаємо порядок виконання функцій-підмножин в S_i компоненті за певний період часу при повному виконанні певного завдання. Введемо функцію $f_{nom}(j)$, значенням якої є номер j – тої функції-підмножини в списку виконання функцій-підмножин. В результаті такого визначення вектор $v_{\alpha'_{1,S_i}} = (f_{nom}(1), f_{nom}(2), \dots, f_{nom}(n_{S_k,max}))$. Номера порядку виконання функцій-підмножин є числами з проміжку $[1, n_{S_k,max}]$, а їх сума буде значенням арифметичної прогресії. Введемо для значень $\alpha'_{1,S_i,1}$ проміжок, в якому буде регулюватись нижня межа в залежності від параметру рівня значущості $\alpha_1^{r,1}$ так: $[1 - \alpha_1^{r,1}; 1]$. За рівень значущості приймемо частку від одиниці, яка відображатиме відхилення від рівня довіри до результату обчислень внаслідок певних подій, архітектурної особливості компоненти тощо. Значення $\alpha'_{1,S_i,1}$ визначаємо з врахуванням координат вектору $v_{\alpha'_{1,S_i}}$, які унормовуємо в проміжок $[1 - \alpha_1^{r,1}; 1]$, так:

1) визначаємо найбільше значення з координат вектору $v_{\alpha'_{1,S_i}}$:

$$f_{nom,max} = max(f_{nom}(1), f_{nom}(2), \dots, f_{nom}(n_{S_k,max}));$$

2) найменше значення з координат вектору $v_{\alpha'_{1,S_i}}$: $f_{nom,min} = 1$;

3) визначаємо крок для розміщення унормованих значень з проміжку $[f_{nom,min}; f_{nom,max}]$ на проміжок $[1 - \alpha_1^{r,1}; 1]$, поділяючи проміжок на рівні відрізки і їх кількість повинна відповідати кількості задіяних функцій-підмножин: $\frac{1-(1-\alpha_1^{r,1})}{n_{S_k,max}-1} = \frac{\alpha_1^{r,1}}{n_{S_k,max}-1}$;

4) обчислюємо значення $\alpha'_{1,S_i,j,1}$ для j-тої функції так:

$$\alpha'_{1,S_i,j,1} = 1 - \frac{\left(n_{S_k,max} - f_{nom}(j)\right)}{n_{S_k,max}-1} \cdot \alpha_1^{r,1}, \qquad (2.19)$$

де $j = 1, 2, \dots, n_{S_k,max}$; $n_{S_k,max}$ - кількість функцій-підмножин, які формуватимуть повну функціональну складову частину центру прийняття рішень в компоненті в системі; $i = 1, 2, \dots, k$; k – кількість компонент системи, в яких може бути центр прийняття рішень системи; $i <= k$; $f_{nom}(j)$ - функція, значенням якої є номер j – тої функції-підмножини в списку виконання функцій-підмножин;

5) визначити значення $\alpha'_{1,S_i,1}$ згідно кроку 4 (формули (2.19)):

$$\alpha'_{1,S_i,1} = \frac{\sum_{j=1}^{n_{S_k,max}} \alpha'_{1,S_i,j,i}}{n_{S_k,max}}. \quad (2.20)$$

Значення $\alpha'_{1,S_i,j,1}$ обчислені за формулою (2.19) будуть більшими для тих функцій-підмножин, які виконуватимуться раніше за решту.

Уточнимо формулу (2.20) для загального випадку, коли кількість функцій-підмножин, які виконаються для вирішення поставленого завдання, може бути менше числа $n_{S_k,max}$, частина функцій-підмножин може виконатись багатократно, частина з функцій-підмножин можуть бути та можуть виконуватись в інших компонентах, в яких є активним центр прийняття рішень системи. Якщо кількість функцій-підмножин, які виконаються для вирішення поставленого завдання, менше числа $n_{S_k,max}$, то введемо змінну $n_{S_k,max,f,1}$, яка задаватиме кількість задіяних функцій-підмножин, та змінну $n_{S_k,max,f,2}$, яка задаватиме кількість задіяних функцій, причому враховуватиме багатократне використання функцій-підмножин, якщо таке їх виконання відбуватиметься. Визначимо вектор $v_{\alpha'_{1,S_i},f,1} = (f_{nom,f}(1), f_{nom,f}(2), \dots, f_{nom,f}(n_{S_k,max,f,2}))$ так, що його координати вказуватимуть номер функції-підмножини в динамічній компоненті, яка виконувалась. Номера порядку виконання функцій-підмножин є натуральними числами з проміжку $[1, n_{S_k,max,f,2}]$. Крім того, номери координати вектору $v_{\alpha'_{1,S_i},f,1}$ визначають послідовність виконання функцій-підмножин. Сформуємо вектор номерів координат (значення функції $f_{nom,f,2}$ – номер координати) вектору $v_{\alpha'_{1,S_i},f,1}$ так: $v_{\alpha'_{1,S_i},f,2} = (f_{nom,f,2}(1), f_{nom,f,2}(2), \dots, f_{nom,f,2}(n_{S_k,max,f,2}))$. Згідно значень у сформованих векторах $v_{\alpha'_{1,S_i}}$, $v_{\alpha'_{1,S_i},f,1}$, $v_{\alpha'_{1,S_i},f,2}$ задаємо для кожної функції-підмножини окремі вектори, координатами яких будуть числа, що отримані в

результаті виконання поставленого завдання та задавали послідовність їх виконання, зокрема і багатократного виклику. Для j-тої функції-підмножини вектор матиме кількість координат, яка дорівнює кількості її викликів $K_{f,j}$, та задамо його так: $v_{\alpha'_{1,S_i,j}} = (f_{nom}(j,1), f_{nom}(j,2), \dots, f_{nom}(j,K_{f,j}))$. Порядок координат цього вектору відповідатиме послідовності викликів. Кількість виконуваних функцій-підмножин при вирішенні певного завдання завжди скінченна і число таких викликів функцій-підмножин дорівнює $n_{S_k,max,f,2}$. Для функцій-підмножин, які при виконанні поставленого завдання будуть виконуватись багатократно, буде вибиратись мінімальне значення $\alpha'_{1,S_i,j,1}$ з обчислених. Значення $\alpha'_{1,S_i,j,1}$ для функцій-підмножин, які не були задіяні при виконанні поставленого завдання, задаємо такими, що дорівнюють нулю. Таким чином, обчислення значення $\alpha'_{1,S_i,1}$ з врахуванням доданих вимог можна здійснити так:

$$\alpha'_{1,S_i,j,1} = \min_{q=1,2,\dots,K_{f,j}} \left(1 - \frac{n_{S_k,max,f,2} - f_{nom}(j,q)}{n_{S_k,max,f,2}-1} \cdot \alpha_1^{r,1}\right), \quad (2.21)$$

де після обчислення значень для кожної функції, отримані результати потрібно підставити в формулу (2.20).

Спростимо формулу (2.21) з врахуванням використання мінімального значення $f_{nom}(j,q)$ (при $q = 1,2,\dots,K_{f,j}$). Оскільки перше значення є найменшим, тоді визначення для кожної функції значення $\alpha'_{1,S_i,j,1}$ здійснюватимемо так:

$$\alpha'_{1,S_i,j,1} = 1 - \frac{n_{S_k,max,f,2} - f_{nom}(j,1)}{n_{S_k,max,f,2}-1} \cdot \alpha_1^{r,1}, \quad (2.22)$$

де $f_{nom}(j,1)$ – перше найменше значення координати вектору $v_{\alpha'_{1,S_i,j}}$; $n_{S_k,max,f,2}$ - кількість виконуваних функцій-підмножин при вирішенні певного завдання.

Далі, аналогічно після обчислення значень для кожної функції за формулою (2.22), отримані результати підставляємо в формулу (2.20).

Якщо частина з функцій-підмножин можуть бути та виконуватись в решті компонент системи, в яких є активним центр прийняття рішень системи, тоді рівень довіри до результатів обчислень порівняно з обчисленнями здійсненими в одній компоненті буде відрізнятись. Кількість таких функцій-підмножин впливатиме на загальний результат та окремі результати для кожної з них, причому частина з них може викликатись для виконання багатократно, тому для кожної з них при

визначенні значення $\alpha'_{1,S_i,j,1}$ врахуємо їх кількість, кількість багатократних викликів певних функцій-підмножин та рівень значущості так:

$$\alpha'_{1,S_i,j,1} = 1 - \left(\frac{n_{S_k,max,f,2} - f_{nom}(j,1)}{n_{S_k,max,f,2} - 1} + \frac{K_{f,j}}{n_{S_k,max,f,2}}\right) \cdot \alpha_1^{r,1}, \quad (2.23)$$

де $f_{nom}(j,1)$ – перше найменше значення координати вектору $v_{\alpha'_{1,S_i},j}$; $n_{S_k,max,f,2}$ - кількість залучених на виконання функцій-підмножин при вирішенні певного завдання; $K_{f,j}$ – кількість викликів j-тої функції-підмножини.

Аналогічно після обчислення значень для кожної функції за формулою (2.23), отримані результати підставляємо в формулу (2.20).

Таким чином, згідно формул (2.23) і (2.20) проміжок $[1 - \alpha_1^{r,1}; 1]$ для функцій-підмножин може бути розширений в нижній межі. Для функцій-підмножин, які запущені на виконання паралельно, значення послідовності запусків будуть однакові, а наступна, запущена на виконання за ними, функція-підмножина буде мати номер запуску збільшений на одиницю.

Визначимо значення другої координати вектору в формулі (2.18) $\alpha'_{1,S_i,2}$, що враховує відносний час витрачений на пересилання результатів виконання функцій-підмножин. Припустимо, що функції-підмножини при виконанні завдання були задіяні однократно. Задаємо функцію $f_{nom,t}(j)$, значенням якої є час витрачений на надсилання результатів обчислення j – тої функції-підмножини у динамічну компоненту. В результаті такого визначення сформуємо вектор $v_{\alpha'_{1,S_i},t} = (f_{nom,t}(1), f_{nom,t}(2), \dots, f_{nom,t}(n_{S_k,max}))$. Номери координат вектору $v_{\alpha'_{1,S_i},t}$ є числами з проміжку $[1, n_{S_k,max}]$. Введемо для значень $\alpha'_{1,S_i,j,2}$ j – тої функції-підмножини проміжок $[1 - \alpha_1^{r,2}; 1]$, в якому буде регулюватись нижня межа в залежності від параметру рівня значущості $\alpha_1^{r,2}$ таким числом $1 - \alpha_1^{r,2}$. За рівень значущості приймемо частку від одиниці, яка відображатиме відхилення від рівня довіри до результату обчислень. Значення $\alpha'_{1,S_i,j,2}$ $(j = 1, 2, \dots, n_{S_k,max})$ визначаємо з врахуванням координат вектору $v_{\alpha'_{1,S_i},t}$, які унормуємо в проміжок $[1 - \alpha_1^{r,2}; 1]$, так:

1) визначаємо найбільше значення з координат вектору $v_{\alpha'_{1,S_i},t}$:

$$f_{nom,max,t} = max(f_{nom,t}(1), f_{nom,t}(2), \dots, f_{nom,t}(n_{S_k,max}));$$

2) визначаємо найменше значення з координат вектору $v_{\alpha'_{1,S_i,t}}$:

$$f_{nom,min,t} = min(f_{nom,t}(1), f_{nom,t}(2), \dots, f_{nom,t}(n_{S_k,max}));$$

3) обчислюємо значення $\alpha'_{1,S_i,j,2}$ для j-тої функції при $j = 1, 2, \dots, n_{S_k,max}$ так:

$$\alpha'_{1,S_i,j,2} = 1 - \alpha_1^{r,2} + \frac{(f_{nom,max,t} - f_{nom,t}(j))}{f_{nom,max,t}} \cdot \alpha_1^{r,2} = 1 - \frac{f_{nom,t}(j)}{f_{nom,max,t}} \cdot \alpha_1^{r,2}, \qquad (2.24)$$

де $n_{S_k,max}$ - кількість функцій-підмножин, які формуватимуть повну функціональну складову частину центру прийняття рішень в компоненті в системі; $i = 1, 2, \dots, k$; k – кількість компонент системи, в яких може бути центр прийняття рішень системи; $i <= k$;

4) визначити значення $\alpha'_{1,S_i,2}$ згідно кроку 3 (формули (2.24)):

$$\alpha'_{1,S_i,2} = \frac{\sum_{j=1}^{n_{S_k,max}} \alpha'_{1,S_i,j,i}}{n_{S_k,max}}. \qquad (2.25)$$

Формула (2.25) надає можливість здійснювати обчислення для випадку, коли частина функцій-підмножин розміщена безпосередньо в компоненті у вузлі в мережі, тоді згідно неї обчислене значення $\alpha'_{1,S_i,j,2}$ для таких функцій дорівнює одиниці, бо час на пересилання результатів обчислень не витрачається.

Уточнимо формулу (2.24) для випадків, коли функції-підмножини виконуються багатократно та коли кількість функцій-підмножин, які виконаються для вирішення поставленого завдання, може бути менше числа $n_{S_k,max}$. Якщо кількість функцій-підмножин, які виконаються для вирішення поставленого завдання, менше числа $n_{S_k,max}$, то введемо змінну $n_{S_k,max,f,1,t}$, яка задаватиме кількість задіяних функцій-підмножин, та змінну $n_{S_k,max,f,2,t}$, яка задаватиме кількість задіяних функцій, причому враховуватиме багатократне залучення функцій-підмножин, якщо таке їх виконання відбуватиметься. Визначимо вектор $v_{\alpha'_{1,S_i},f,1,t} = (f_{nom,f,t}(1), f_{nom,f,t}(2), \dots, f_{nom,f,t}(n_{S_k,max,f,2,t}))$ так, щоб його координати вказували на номер функції-підмножини в динамічній компоненті, яка була виконана. Номера порядку виконання функцій-підмножин є натуральними числами з проміжку $[1, n_{S_k,max,f,2,t}]$. Крім того, номери координат вектору $v_{\alpha'_{1,S_i},f,1,t}$ визначають послідовність виконання функцій-підмножин. Сформуємо вектор

номерів координат вектору $v_{\alpha'_{1,S_i},f,1,t}$ так: $v_{\alpha'_{1,S_i},f,2,t},f,2,t} = (f_{nom,f,2,t}(1), f_{nom,f,2,t}(2), \dots, f_{nom,f,2,t}(n_{S_k,max,f,2,t}))$. Згідно значень у сформованих векторах $v_{\alpha'_{1,S_i},t}$, $v_{\alpha'_{1,S_i},f,1,t}$, $v_{\alpha'_{1,S_i},f,2,t}$ задаємо для кожної функції-підмножини окремі вектори, координатами яких будуть числа, що були отримані в результаті виконання поставленого завдання та задавали послідовність їх виконання, зокрема і багатократного виклику. Для j-тої функції-підмножини вектор матиме кількість координат, яка дорівнює кількості її викликів $K_{f,j,t}$, та задамо його так: $v_{\alpha'_{1,S_i},j,t} = (f_{nom,t}(j,1), f_{nom,t}(j,2), \dots, f_{nom,t}(j,K_{f,j,t}))$. Порядок координат цього вектору відповідатиме послідовності викликів. Кількість виконуваних функцій-підмножин при вирішенні певного завдання завжди скінченна і число таких викликів функцій-підмножин дорівнює $n_{S_k,max,f,2,t}$. Значення координат вектору $v_{\alpha'_{1,S_i},j,t}$ при $j = 1, 2, \dots, n_{S_k,max}$ будуть значеннями витраченого часу на надсилання результатів обчислень. Для функцій-підмножин, які при виконанні поставленого завдання будуть виконуватись багатократно, буде вибиратись сумарне значення витраченого часу на виконання. Значення $\alpha'_{1,S_i,j,2}$ для функцій-підмножин, які не були задіяні при виконанні поставленого завдання, задаємо такими, що дорівнюють нулю. Таким чином, значення $\alpha'_{1,S_i,j,2}$ визначаємо так:

$$\alpha'_{1,S_i,j,2} = 1 - \frac{\sum_{q=1}^{K_{f,j,t}} f_{nom,t}(j,q)}{\sum_{j=1}^{n_{S_k,max}} \sum_{q=1}^{K_{f,j,t}} f_{nom,t}(j,q)} \cdot \alpha_1^{r,2}, \tag{2.26}$$

де $f_{nom,t}(j,q)$ – значення координати вектору $v_{\alpha'_{1,S_i},j,t}$; $n_{S_k,max}$ – кількість функцій-підмножин в компоненті; $K_{f,j,t}$ – кількість викликів j-тої функції-підмножини.

Згідно формули (2.25), підставляючи значення отримані за формулою (2.26), визначаємо значення $\alpha'_{1,S_i,2}$ для компонент S_i ($i = 1, 2, \dots, k$; k – кількість компонент системи, в яких може бути центр прийняття рішень системи; $i <= k$).

Таким чином, за формулою (2.24) значення $\alpha'_{1,S_i,j,2}$ визначається з врахуванням відношення витраченого часу на надсилання результату обчислень для однієї функції-підмножини до максимального часу, який визначається як максимальне значення з витраченого часу для окремих функцій-підмножин. А за формулою (2.26) визначення значення $\alpha'_{1,S_i,j,2}$ здійснюється через відношення сумарного витраченого

часу однієї функції-підмножини, зокрема і при багатократних викликах, до всього часу витраченого функціями-підмножинами для вирішення поставленого завдання. Тому, значення $\alpha'_{1,S_i,j,2}$, які отримані за формулою (2.26) будуть більшими, ніж за формулою (2.25), для однократного виконання функцій-підмножин.

Визначимо $\alpha'_{1,S_i,3}$ як значення, яке враховує рівень безпеки вузла в мережі та міститиме данні про стан безпеки вузла в мережі і визначатиметься безпосередньо системою. Це значення, також, оновлюватиметься напротязі активності компоненти системи. Саме значення $\alpha'_{1,S_i,3}$ визначатиметься з проміжку $\left[1-\alpha_1^{r,3};1\right]$, де $\alpha_1^{r,3}$ є рівнем значущості. Довіра до результатів обчислень з певної компоненти системи буде залежати від значення $\alpha'_{1,S_i,3}$. За рівень значущості приймемо частку від одиниці. Рівень значущості $\alpha_1^{r,3}$ відноситиметься для обчислень, які здійснюватимуться на рівні центру прийняття рішень. Його значення буде більшим, ніж значення рівня значущості для обчислень, які не відноситимуться до рівня центру прийняття рішень, тобто для решти обчислень. Це пов’язано з тим, що від рішень центру прийняття рішень залежить стійкість всієї системи та її подальші кроки. Значення $\alpha'_{1,S_i,3}$ обчислюватиметься з врахуванням критеріїв для комплексного оцінювання безпеки вузла в мережі, де знаходитиметься S_i компонента системи. Для цього розглянемо окремо такі складові частини, які впливатимуть на значення $\alpha'_{1,S_i,3}$: процеси, які відбуваються на рівні мережі та мережних служб; процеси, які відбуваються в комп’ютерній станції; внутрішні процеси в системі; значення рівня безпеки, яке видаватиме безпосередньо система.

Мережні служби, розміщення станцій в комп’ютерній мережі, конфігурування комп’ютерної мережі та інші характерні складові частини, які віднесені до питань забезпечення функціонування вузлів в мережах, суттєво впливатимуть на забезпечення безпеки, а тому, і на значення $\alpha'_{1,S_i,3}$. Складова частина, яка формуватиме значення $\alpha'_{1,S_i,3}$, і задаватиме окремі проміжні значення, що описуватимуть процеси, які відбуваються на рівні мережі та мережних служб, є багатокомпонентною і кожна з компонент має різні характеристики. Виходячи із планованого місця розміщення системи та завдань, які вона повинна буде виконувати, значення рівня безпеки формуватимемо, починаючи з врахування місця розміщення в комп’ютерних мережах та їх особливостей конфігурації.

Місцем розміщення системи є корпоративна комп'ютерна мережа підприємства, організації чи установи. Певна група користувачів матиме доступ до ресурсів мережі віддалено. Наприклад, в домашніх умовах. Тому, особливості різних типів архітектурних конфігурацій комп'ютерних мереж, які можуть бути на підприємстві, повинні бути враховані при визначенні значень рівня безпеки і система повинна мати про них відомості і значення рівня безпеки в залежності від обладнання і типів конфігурування.

Для визначення рівня безпеки системи врахуємо моделі керування користувачами і мережними ресурсами, які використано безпосередньо в організації комп'ютерних мереж підприємства. Модель розподіленого керування (модель керування робочою групою) порівняно з моделлю централізованого керування (доменна модель керування) при великій кількості комп'ютерних станцій в мережі підприємства буде негативно впливати та безпеку в мережі. Це пов'язано з тим, що робоча група є логічною групою комп'ютерів для надання доступу до ресурсів та використовується в однорангових мережах, де в кожній комп'ютерній станції зберігається локальна база безпеки, зокрема з інформацією про облікові записи користувачів та захист ресурсів. Для невеликої кількості комп'ютерів така модель розподіленого керування має перевагу над моделлю централізованого керування. Як правило, кількість комп'ютерів в такій моделі для забезпечення ефективного користування мережними ресурсами та, відповідно, безпечної роботи повинна становити не більше 8-10. При більшій кількості переваги має мережа, яка побудована з використанням моделі централізованого керування. В підприємствах комп'ютерна мережа, в яку планується встановлення системи S, може бути побудована з використанням різних моделей, в тому числі і неефективно (наприклад, згідно моделі децентралізованого керування з кількістю комп'ютерів більше 20), але система S повинна володіти цією інформацією для визначення своїх подальших кроків, які базуватимуться на оцінюванні стану безпеки в цілому в мережі і в її конкретних вузлах. Введемо параметр для визначення в системі моделі керування і позначимо її $\alpha''_{1,S_i,3,1}$. Значення $\alpha''_{1,S_i,3,1} = 1$, якщо мережа побудована згідно моделі розподіленого керування з кількістю комп'ютерів не більше 8 та $\alpha''_{1,S_i,3,1} = 2$ – не менше 9, і $\alpha''_{1,S_i,3,1} = 3$, якщо мережа побудована згідно моделі централізованого керування. Введемо локальний рівень значущості $\alpha_1^{r,3,1}$ для рівня

значущості $\alpha_1^{r,3}$, який відображатиме відхилення від рівня довіри до результату обчислень. За локальний рівень значущості $\alpha_1^{r,3,1}$ приймемо частку від одиниці, тоді значення довіри до результатів обчислень для всіх вузлів в мережі будуть однакові і знаходитимуться в проміжку $\left[1-\alpha_1^{r,3,1};1\right]$. Значення $\alpha''_{1,S_i,3,1}$ ($i=1,2,\dots,k$; k – кількість компонент системи, в яких може бути центр прийняття рішень системи; $i <= k$) визначаємо так:

$$\alpha'_{1,S_i,3,1}=\begin{cases}1, \text{якщо } \alpha''_{1,S_i,3,1}=1;\\ 1-\frac{n_{1,S_i,3,1,k}-n_{1,S_i,3,1,p}}{n_k}\cdot\alpha_1^{r,3,1}, \text{якщо } \alpha''_{1,S_i,3,1}=2;\\ 1-\frac{n_{1,S_i,3,1,s}-1}{n_{1,S_i,3,1,k}}\cdot\alpha_1^{r,3,1}, \text{якщо } \alpha''_{1,S_i,3,1}=3,\end{cases} \tag{2.27}$$

де $n_{1,S_i,3,1,k}$ – кількість комп'ютерних станцій в мережі; $n_{1,S_i,3,1,p}$ – кількість вимкнених комп'ютерних станцій в мережі; $n_{1,S_i,3,1,s}$ – кількість сегментів в мережі.

Аналогічно введемо решту характеристичних показників для визначення значення рівня безпеки.

При здійсненні аутентифікації користувачів можуть відбуватись невдалі спроби. Накопичення їх певної кількості може бути причиною відмови в доступі, але може, також, бути враховано як спроби зловмисника увійти за логіном авторизованого користувача або часті помилкові спроби некваліфікованого користувача. В першому випадку, значення рівня безпеки не може бути максимальним, оскільки зловмисник отримав можливість фізичного доступу до системи аутентифікації, а в другому – помилкові дії некваліфікованого користувача при аутентифікації вказують на те, що він може і при виконанні решти робіт в інформаційній системі, зокрема і тих, які відносяться до забезпечення безпеки, буде виконувати некваліфіковано. Введемо складову частину $\alpha'_{1,S_i,3,2}$, яка формуватиме значення $\alpha'_{1,S_i,3}$, і його визначене значення буде характеристичним показником аутентифікації. Значення $\alpha'_{1,S_i,3,2}$ визначимо для кожної i-тої комп'ютерної станції так, щоб воно належало проміжку $\left[1-\alpha_1^{r,3,2};1\right]$, де локальний рівень значущості $\alpha_1^{r,3,2}$ є часткою від одиниці і відображає відхилення від рівня довіри до результату обчислень. Обчислення його значення здійснимо за формулою:

$$\alpha'_{1,S_i,3,2}=1-\frac{n_{1,S_i,3,2,k}-n_{1,S_i,3,2,p}}{n_{1,S_i,3,2,k}}\cdot\alpha_1^{r,3,2}, \tag{2.28}$$

де $n_{1,S_i,3,2,k}$ – кількість спроб користувачів пройти аутентифікацію в інформаційній системі в i-тій комп'ютерній станції; $n_{1,S_i,3,2,p}$ – кількість успішних спроб користувачів пройти аутентифікацію в інформаційній системі в i-тій комп'ютерній станції.

Помилкове визнання стороннього користувача комп'ютерною станцією, наслідком якого стає випадковий вхід і доступ в неї, характеризується тим, що в реєстрі користувачів фіксується вхід неавторизованого користувача і надання доступу. Введемо складову частину $\alpha'_{1,S_i,3,3}$, яка формуватиме значення $\alpha'_{1,S_i,3}$, і його обчислене значення буде характеристичним показником помилкового визнання стороннього користувача. Значення $\alpha'_{1,S_i,3,3}$ визначимо для кожної i-тої комп'ютерної станції так, щоб воно належало проміжку $\left[1-\alpha_1^{r,3,3};1\right]$, де локальний рівень значущості $\alpha_1^{r,3,3}$ є часткою від одиниці і відображає відхилення від рівня довіри до результату обчислень. Визначення його значення здійснимо за формулою:

$$\alpha'_{1,S_i,3,3}=1-\frac{n_{1,S_i,3,3,p}}{n_{1,S_i,3,3,k}}\cdot\alpha_1^{r,3,3}, \tag{2.29}$$

де $n_{1,S_i,3,3,k}$ – кількість спроб сторонніх користувачів здійснити вхід в i-ту комп'ютерну станцію; $n_{1,S_i,3,3,p}$ – кількість успішних спроб сторонніх користувачів, які здійснили вхід в i-ту комп'ютерну станцію.

Отримання значень $n_{1,S_i,3,3,k}$ і $n_{1,S_i,3,3,p}$ можливе при встановленні компонент системи S та в процесі певного часу функціонування системи S в усіх вузлах в мережі.

Розглянемо помилкове заперечення авторизованого користувача при намаганні ним отримати доступ до j-тої комп'ютерної станції. Введемо складову частину $\alpha'_{1,S_i,3,4}$, яка формуватиме значення $\alpha'_{1,S_i,3}$, і його значення буде характеристичним показником помилкового заперечення авторизованого користувача. Значення $\alpha'_{1,S_i,3,4}$ визначимо для кожної i-тої комп'ютерної станції так, щоб воно належало проміжку $\left[1-\alpha_1^{r,3,4};1\right]$, де локальний рівень значущості $\alpha_1^{r,3,4}$ є часткою від одиниці і відображає відхилення від рівня довіри до результату обчислень. Визначення значення здійснимо так:

$$\alpha'_{1,S_i,3,4}=1-\frac{n_{1,S_i,3,4,p}}{n_{1,S_i,3,4,k}}\cdot\alpha_1^{r,3,4}, \tag{2.30}$$

де $n_{1,S_i,3,3,k}$ – кількість спроб авторизованих користувачів здійснити вхід в i-ту комп'ютерну станцію; $n_{1,S_i,3,3,p}$ – кількість неуспішних спроб авторизованих користувачів, які здійснювали вхід в i-ту комп'ютерну станцію.

Авторизованим користувачам повинен бути організований розмежований доступ до ресурсів комп'ютерних станцій. Хоча такого розмежування може і не бути. Крім того, кількість авторизованих користувачів, які мають доступ і працюють з однією комп'ютерною станцією може бути різною. В підприємстві можуть бути сформовані різні схеми розмежування доступу аутентифікації користувачів до ресурсів та правила розмежування з використанням монітору звернень. Також, контроль доступу до ресурсів повинен враховувати можливість виконання роботи певних працівників підприємства з домашніх комп'ютерів в певний час. Особливості роботи з контролем доступу та рівнями розмежування авторизованих користувачів повинні бути враховані при обчисленні значення рівня безпеки. Введемо складову частину $\alpha'_{1,S_i,3,5}$, яка формуватиме значення $\alpha'_{1,S_i,3}$, і його значення буде характеристичним показником організації розмежування доступу авторизованих користувачів до ресурсів кожної i-тої комп'ютерної станції в мережі підприємства. Якщо розмежування доступу не здійснюється, тоді значення $\alpha''_{1,S_i,3,5} = 1$, інакше $\alpha''_{1,S_i,3,5} = 2$. Значення $\alpha'_{1,S_i,3,5}$ визначимо для кожної i-тої комп'ютерної станції так, щоб воно належало проміжку $\left[1 - \alpha_1^{r,3,5}; 1\right]$, де локальний рівень значущості $\alpha_1^{r,3,5}$ є часткою від одиниці і відображає відхилення від рівня довіри до результату обчислень. Визначення значення здійснимо так:

$$\alpha'_{1,S_i,3,5} = \begin{cases} 1 - \alpha_1^{r,3,5}, \text{якщо } \alpha''_{1,S_i,3,5} = 1 \\ 1 - \frac{n_{1,S_i,3,5,k}}{n_{1,S_i,3,5,p}} \cdot \alpha_1^{r,3,5}, \text{якщо } \alpha''_{1,S_i,3,5} = 2 \end{cases}, \tag{2.31}$$

де $n_{1,S_i,3,5,k}$ – кількість авторизованих користувачів, яким дозволено та які за певний час функціонування i-тої комп'ютерної станції здійснили в неї вхід з різними рівнями доступу; $n_{1,S_i,3,5,p}$ – кількість авторизованих користувачів, яким дозволено здійснювати вхід в i-ту комп'ютерну станцію з різними рівнями доступу.

Певна частина авторизованих користувачів підприємства може мати дозвіл на під'єднання та вхід в систему з домашніх комп'ютерів. В такому випадку значення локального рівня значущості буде однаковим для всіх вузлів в мережі однаковим.

Введемо складову частину $\alpha'_{1,S_i,3,6}$, яка формуватиме значення $\alpha'_{1,S_i,3}$, і його значення буде характеристичним показником доступу авторизованих користувачів до ресурсів в мережу підприємства з домашніх комп'ютерів. Якщо такий доступ не здійснюється через заборону, тоді значення $\alpha''_{1,S_i,3,6} = 1$, інакше $\alpha''_{1,S_i,3,6} = 2$. Значення $\alpha'_{1,S_i,3,6}$ визначимо для всіх комп'ютерних станцій так, щоб воно належало проміжку $\left[1 - \alpha_1^{r,3,6}; 1\right]$, де локальний рівень значущості $\alpha_1^{r,3,6}$ є часткою від одиниці і відображає відхилення від рівня довіри до результату обчислень. Визначення значення $\alpha'_{1,S_i,3,6}$ здійснимо так:

$$\alpha'_{1,S_i,3,6} = \begin{cases} 1, \text{якщо } \alpha''_{1,S_i,3,6} = 1 \\ 1 - \frac{n_{1,S_i,3,6,k}}{n_{1,S_i,3,6,p}} \cdot \alpha_1^{r,3,5}, \text{якщо } \alpha''_{1,S_i,3,6} = 2 \end{cases}, \quad (2.32)$$

де $n_{1,S_i,3,6,k}$ – кількість авторизованих користувачів, яким дозволено та які за певний час здійснили вхід в систему підприємства ззовні з домашніх комп'ютерів з різними рівнями доступу; $n_{1,S_i,3,6,p}$ – кількість авторизованих користувачів, яким дозволено здійснювати вхід в систему підприємства ззовні з домашніх комп'ютерів з різними рівнями доступу.

Складність побудованих корпоративних мереж впливає на значення рівнів безпеки в цілому в них та в їх вузлах зокрема. Враховуючи різноманітність і різноспрямованість архітектурних особливостей, задамо їх елементами множини $M_{A,KM}$. Комбінації з них будуть відображати сформовані корпоративні мережі з архітектурними особливостями, що визначатимуться переліком елементів підмножини множини $M_{A,KM}$. Таким чином, задамо такі значення елементів множини $M_{A,KM}$: $m_{A,KM,1}$ - корпоративна мережа має одне під'єднання до мережі Інтернет; $m_{A,KM,2}$ - корпоративна мережа має декілька під'єднань до мережі Інтернет; $m_{A,KM,3}$ - корпоративна мережа має тривалий час незмінювану архітектуру; $m_{A,KM,4}$ - корпоративна мережа розташована в одному приміщенні підприємства; $m_{A,KM,5}$ - корпоративна мережа розташована в декількох приміщеннях підприємства безпосередньо на його території і сегменти мережі поєднані провідним зв'язком; $m_{A,KM,6}$ - корпоративна мережа розташована в декількох приміщеннях підприємства безпосередньо на його території і сегменти мережі поєднані безпровідним зв'язком; $m_{A,KM,7}$ – зв'язок між сегментами мережі

провідний; $m_{A,KM,8}$ - зв'язок між сегментами мережі безпровідний; $m_{A,KM,9}$ – зв'язок між частиною вузлів мережі безпровідний; $m_{A,KM,10}$ - зв'язок між усіма вузлами мережі провідний; $m_{A,KM,11}$ - корпоративна мережа має часто змінювану архітектуру, налаштування, користувачів тощо; $m_{A,KM,12}$ - корпоративна мережа складається з територіально розділених складових частин і при цьому ці частини пов'язуються зовнішніми мережами, які знаходяться за межами території підприємства, та обслуговуються сторонніми надавачами мережних послуг; $m_{A,KM,13}$ – сервери корпоративних мереж знаходяться на території підприємства; $m_{A,KM,14}$ – не всі сервери корпоративних мереж знаходяться на території підприємства; $m_{A,KM,15}$ – всі підсистеми та елементи корпоративної мережі обслуговуються та контролюються адміністраторами підприємства; $m_{A,KM,16}$ – не всі підсистеми та елементи корпоративної мережі обслуговуються та контролюються адміністраторами підприємства; $m_{A,KM,17}$ – певний користувач використовує ресурси, які постійно розміщені в одному місці корпоративної мережі; $m_{A,KM,18}$ – певний користувач використовує ресурси, які розміщені в різних місцях корпоративної мережі, зокрема і територіально віддалених, та з різними платформами; $m_{A,KM,19}$ – до мережі та її ресурсів повинен бути організований цілодобовий доступ; $m_{A,KM,20}$ – забезпечення вимоги щодо максимального часу простою, який має бути мінімізований і перебувати в заданих межах, наприклад одна хвилина. Кількість елементів множини $M_{A,KM}$ може бути збільшено за рахунок введення та виділення решти архітектурних особливостей корпоративних мереж. Нехай $n_{M_{A,KM}}$ можлива максимальна кількість архітектурних особливостей корпоратнивних мереж. Згідно так сформованої множини $M_{A,KM}$ задамо, наприклад, її підмножини $M_{A,KM,1} = \{m_{A,KM,2}, m_{A,KM,11}, m_{A,KM,19}\}$ та $M_{A,KM,2} = \{m_{A,KM,2}, m_{A,KM,13}, m_{A,KM,14}, m_{A,KM,20}\}$. Відповідно значення рівнів безпеки в них будуть різними. Нехай значення $\alpha''_{1,S_i,3,7} = 0$ для випадку елементу (наприклад, елементи $m_{A,KM,1}$, $m_{A,KM,3}$, $m_{A,KM,4}$, $m_{A,KM,5}$, $m_{A,KM,7}$, $m_{A,KM,10}$, $m_{A,KM,13}$, $m_{A,KM,15}$, $m_{A,KM,17}$), який не впливає на зниження значення рівня безпеки порівняно з другим альтернативним елементом, що передбачає ускладнення в архітектурі корпоративних мереж. Тоді, значення $\alpha''_{1,S_i,3,7} = 1$ приймемо для альтернативного випадку. Підмножини, які відображатимуть реально можливі архітектури

корпоративних мереж, можуть бути сформовані не з усіх елементів, тобто не всі елементи можуть поєднуватись при формуванні підмножин, а тому певних підмножин може не бути. Введемо функцію $f_{M_{A,KM}}$ значення якої задамо так:

$$f_{M_{A,KM}}(m_{A,KM,q}) = \begin{cases} 0, \text{якщо } \alpha''_{1,S_i,3,7} = 0 \\ 1, \text{якщо } \alpha''_{1,S_i,3,7} = 1 \end{cases}, \quad (2.33)$$

де $q = 1,2,\dots,n_{M_{A,KM}}$; $n_{M_{A,KM}}$ - максимальна кількість архітектурних особливостей корпоративних мереж.

Значення локального рівня значущості для характеристичного показника складності архітектури корпоративних мереж буде однаковим для всіх вузлів в мережі. Введемо складову частину $\alpha'_{1,S_i,3,7}$, яка формуватиме значення $\alpha'_{1,S_i,3}$, і його значення буде характеристичним показником складності архітектури корпоративних мереж. Значення $\alpha'_{1,S_i,3,7}$ визначимо для всіх комп'ютерних станцій так, щоб воно належало проміжку $\left[1-\alpha_1^{r,3,7};1\right]$, де локальний рівень значущості $\alpha_1^{r,3,7}$ є часткою від одиниці і відображає відхилення від рівня довіри до результату обчислень. Визначення значення $\alpha'_{1,S_i,3,7}$ здійснимо так:

$$\alpha'_{1,S_i,3,7} = 1 - \frac{\sum_{w=1}^{n_{M_{A,KM,p}}} f_{M_{A,KL,p}}(m_{A,KM,w})}{\sum_{q=1}^{n_{M_{A,KM}}} f_{M_{A,KL}}(m_{A,KM,q})} \cdot \alpha_1^{r,3,7}, \quad (2.34)$$

де $m_{A,KM,q}$ – елементи множини $M_{A,KL}$; $m_{A,KM,w}$ - елементи підмножини $M_{A,KM,p}$; $n_{M_{A,KM,p}}$ – кількість елементів підмножини $M_{A,KM,p}$; $n_{M_{A,KM}}$ - кількість елементів множини $M_{A,KL}$.

Наявність засобів захисту інформації та забезпечення її безпеки є складовою частиною корпоративних мереж. Важливим елементом сервісу безпеки в корпоративній мережі є міжмережне екранування, причому як комп'ютерної станції так і корпоративної мережі в цілому. Практичних реалізацій міжмережних екранів може бути декілька. Введемо множину $M_{E,KM}$, елементи якої відповідатимуть певним типовим реалізаціям міжмережних екранів: $m_{E,KM,1}$ – з фільтрацією пакетів; $m_{E,KM,2}$ – з подвійним шлюзом; $m_{E,KM,3}$ – з ізольованим хостом; $m_{E,KM,4}$ – з ізольованої підмережею. Нехай $n_{M_{E,KM}}$ – кількість елементів множини $M_{E,KM}$. Підмножини множини $M_{E,KM}$ можуть містити один з елементів або декілька. Цим

визначатиметься складність та безпека корпоративної мережі. Також, в корпоративній мережі може бути декілька варіантів реалізації міжмережних екранів, зокрема однакових, тому такі випадки задаватимемо як об'єднання підмножин. Різні варіанти реалізації можуть бути, наприклад, в територіально розподілених частинах корпоративної мережі. В такому випадку деякі з елементів множини $M_{E,KM}$ можуть повторюватись в різних підмножинах. Визначення значення рівня безпеки здійснюватимемо з врахуванням всіх наявних підмножин, які характеризують реалізації мережних екранів в корпоративній мережі. Локальний рівень значущості для характеристичного показника реалізації мережних екранів в корпоративних мережах буде однаковим для всіх вузлів в частині мережі з однаковою реалізацією. Введемо складову частину $\alpha'_{1,S_i,3,8}$, яка формуватиме значення $\alpha'_{1,S_i,3}$, і його значення буде характеристичним показником реалізації мережних екранів в корпоративній мережі. Значення $\alpha'_{1,S_i,3,8}$ визначимо для всіх комп'ютерних станцій так, щоб воно належало проміжку $\left[1-\alpha_1^{r,3,8};1\right]$, де локальний рівень значущості $\alpha_1^{r,3,8}$ є часткою від одиниці і відображає відхилення від рівня довіри до результату обчислень. Введемо функцію $f_{M_{E,KM}}$ значення якої задамо так:

$$f_{M_{E,KM}}\left(m_{E,KM,q}\right)=\frac{1}{q}, \tag{2.35}$$

де $q=1,2,\dots,n_{M_{E,KM}}$; $n_{M_{E,KM}}$ - кількість елементів множини $M_{E,KM}$.

Тоді, визначення значення $\alpha'_{1,S_i,3,8}$ здійснимо так:

$$\alpha'_{1,S_i,3,8}=1-\frac{n_{1,S_i,3,8,k}}{n_{1,S_i,3,8,p}}\cdot\prod_{w=1}^{n_{M_{E,KM,p}}}f_{M_{E,KM}}\left(m_{E,KM,w}\right)\cdot\alpha_1^{r,3,8}, \tag{2.36}$$

де $m_{E,KM,w}$ - елементи підмножини $M_{E,KM,p}$; $n_{M_{E,KM,p}}$ – кількість елементів підмножини $M_{E,KM,p}$; $n_{1,S_i,3,8,k}$ – кількість комп'ютерних станцій в мережі, які перебувають в межах одного мережного екрану, тобто реалізація мережного екрану відповідає підмножині $M_{E,KM,p}$; $n_{1,S_i,3,8,p}$ – кількість комп'ютерних станцій в корпоративній мережі.

Значення $\alpha'_{1,S_i,3,8}$, яке обчислене за формулою (2.36) буде однаковим для тих комп'ютерних станцій в мережі, які перебувають в межах одного мережного екрану.

Якщо мережних екранів декілька для різних сегментів корпоративної мережі, тоді значення $\alpha'_{1,S_i,3,8}$, відповідно, будуть для них різними.

Використання СВВ в корпоративних мережах має важливе значення в контексті забезпечення безпеки і, відповідно, захисту інформації. Розглянемо спочатку вузлові СВВ. Їх функціонування базується на використанні п'яти основних типів датчиків, які позначимо як елементи множини $M_{H,KM}$: $m_{H,KM,1}$ – аналізатори журналів; $m_{H,KM,2}$ – датчики ознак; $m_{H,KM,3}$ – контролери цілісності файлів; $m_{H,KM,4}$ – аналізатори поведінки застосунків; $m_{H,KM,5}$ – аналізатори системних викликів. Кількість датчиків може бути більшою, тому нехай $n_{M_{H,KM}}$ – кількість елементів множини $M_{H,KM}$. Крім того, ці елементи мають різну вагу в контексті забезпечення безпеки в корпоративній мережі. Наприклад, аналізатори журналів відстежують події, які можуть бути за їх критеріями такими, що впливатимуть на безпеку. Але при цьому вони можуть відреагувати на подію тільки після її виконання, а не попередити її. Аналогічно, робота датчиків ознак, які призначені для аналізу трафіку, відбувається в процесі здійснення атак. А аналізатори системних викликів можуть запобігати зловмисним діям. В цілісному контексті всі елементи множини $M_{H,KM}$ є важливими, але значущість певних елементів більша. Із елементів множини $M_{H,KM}$ можуть формуватись різні підмножини. Також, в межах певних сегментів можуть бути різні СВВ або однакові чи однаково налаштовані, але в різних фізично територіально розподілених сегментах корпоративної мережі, тоді такі випадки задаватимуться об'єднаннями підмножин. Це впливатиме на забезпечення безпеки корпоративної мережі. Визначення значення рівня безпеки здійснюватимемо з врахуванням всіх наявних підмножин. Локальний рівень значущості для характеристичного показника вузлових СВВ в корпоративних мережах буде однаковим для всіх вузлів в частині мережі з однаковою реалізацією. Введемо складову частину $\alpha'_{1,S_i,3,9}$, яка формуватиме значення $\alpha'_{1,S_i,3}$, і його значення буде характеристичним показником вузлових СВВ в корпоративній мережі. Значення $\alpha'_{1,S_i,3,9}$ визначимо для всіх комп'ютерних станцій так, щоб воно належало проміжку $\left[1-\alpha_1^{r,3,9};1\right]$, де локальний рівень значущості $\alpha_1^{r,3,9}$ є часткою від одиниці. Задамо елементи множини $M_{H,KM}$, як координати вектору $\left(m_{H,KM,1},m_{H,KM,2},m_{H,KM,3},m_{H,KM,4},m_{H,KM,5}\right)$, причому приймемо, що координати з

більшим номером матимуть більшу вагу в контексті забезпечення безпеки. Введемо функцію $f_{M_{H,KM}}$ значення якої задамо так:

$$f_{M_{H,KM}}(m_{E,KM,w}) = \frac{1}{w+1}, \quad (2.37)$$

де $w = 1,2,\dots,n_{M_{E,KM}}$; $n_{M_{H,KM}}$ - кількість елементів множини $M_{H,KM}$.

Обчислення значення $\alpha'_{1,S_i,3,9}$ здійснимо так:

$$\alpha'_{1,S_i,3,9} = 1 - \frac{n_{1,S_i,3,9,k}}{n_{1,S_i,3,9,p}} \cdot \prod_{w=1}^{n_{M_{H,KM,p}}} f_{M_{H,KM}}(m_{H,KM,w}) \cdot \alpha_1^{r,3,9}, \quad (2.38)$$

де $m_{H,KM,w}$ - елементи підмножини $M_{H,KM,p}$; $n_{M_{H,KM,p}}$ – кількість елементів підмножини $M_{H,KM,p}$; $n_{1,S_i,3,9,k}$ – кількість комп'ютерних станцій в мережі, які перебувають в межах одного типу вузлової СВВ, тобто реалізація відповідає підмножині $M_{H,KM,p}$; $n_{1,S_i,3,9,p}$ – кількість комп'ютерних станцій в корпоративній мережі.

Значення $\alpha'_{1,S_i,3,9}$, яке обчислене за формулою (2.38) буде однаковим для тих комп'ютерних станцій в мережі, які перебувають в межах одного типу вузлової СВВ.

Розглянемо мережні СВВ. Вони можуть бути різними і це впливатиме на їх результат роботи. Наприклад, мережні СВВ можуть реалізовуватись у мережі в таких її місцях, звідки можна здійснювати контроль трафіку всіх пристроїв або у підмережі разом з мережними екранами для їх захисту чи у спеціально виділеній комп'ютерній системі. Мережні СВВ здійснюють пошук за ознаками атак і цільове спрямування та підбір таких ознак теж впливає на їх ефективність. Тому, враховуючи різні конфігураційні особливості мережних СВВ, які впливатимуть на безпеку в мережі, введемо множину $M_{N,KM} = \{m_{N,KM,1}, m_{N,KM,2}, \dots, m_{N,KM,n_{M_{N,KM}}}\}$, елементами якої будуть особливості конфігурування, використовувані датчики ознак, охоплення комп'ютерних станцій в мережі. Приймемо, що ці елементи мають різну вагу в контексті забезпечення безпеки в корпоративній мережі, якщо вони будуть застосовані в певній її частині. Із елементів множини $M_{N,KM}$ можуть формуватись різні підмножини. Також, в межах певних сегментів можуть бути різні мережні СВВ зокрема, наприклад, в різних територіально розподілених сегментах корпоративної мережі, тоді такі випадки задаватимуться об'єднаннями підмножин.

Це впливатиме на забезпечення безпеки корпоративної мережі. Значення рівня безпеки обчислюватимемо з врахуванням всіх наявних підмножин. Локальний рівень значущості для характеристичного показника мережних СВВ в корпоративних мережах буде однаковим для всіх вузлів в частині мережі з однаковою реалізацією. Введемо складову частину $\alpha'_{1,S_i,3,10}$, яка формуватиме значення $\alpha'_{1,S_i,3}$, і його значення буде характеристичним показником мережних СВВ. Значення $\alpha'_{1,S_i,3,10}$ визначимо для всіх комп'ютерних станцій так, щоб воно належало проміжку $\left[1-\alpha_1^{r,3,10};1\right]$, де локальний рівень значущості $\alpha_1^{r,3,10}$ є часткою від одиниці. Задамо елементи множини $M_{N,KM}$, як координати вектора $\left(m_{N,KM,1}, m_{N,KM,2}, m_{N,KM,3}, m_{N,KM,4}, m_{N,KM,5}\right)$, причому приймемо, що координати з більшим номером матимуть більшу вагу в контексті забезпечення безпеки. Введемо функцію $f_{M_{N,KM}}$ значення якої задамо так:

$$f_{M_{N,KM}}\left(m_{N,KM,w}\right) = \frac{1}{w+1}, \tag{2.39}$$

де $w = 1,2,\dots,n_{M_{E,KM}}$; $n_{M_{N,KM}}$ - кількість елементів множини $M_{N,KM}$.

Для мережних СВВ обчислення значення $\alpha'_{1,S_i,3,10}$ здійснимо так:

$$\alpha'_{1,S_i,3,10} = 1 - \frac{n_{1,S_i,3,10,k}}{n_{1,S_i,3,10,p}} \cdot \prod_{w=1}^{n_{M_{N,KM,p}}} f_{M_{N,KM}}\left(m_{N,KM,w}\right) \cdot \alpha_1^{r,3,10}, \tag{2.40}$$

де $m_{N,KM,w}$ - елементи підмножини $M_{N,KM,p}$; $n_{M_{N,KM,p}}$ – кількість елементів підмножини $M_{N,KM,p}$; $n_{1,S_i,3,10,k}$ – кількість комп'ютерних станцій в мережі, які перебувають в межах одного типу конфігурування мережної СВВ, тобто реалізація відповідає підмножині $M_{N,KM,p}$; $n_{1,S_i,3,10,p}$ – кількість комп'ютерних станцій в корпоративній мережі.

Розподіл комп'ютерних станцій та серверів в корпоративній мережі за зонами з різним рівнем захищеності впливатиме на значення рівнів безпеки в кожній із зон. Наприклад, частина комп'ютерних станцій може бути в сегментів чи зоні, де здійснюються заходи підвищеного рівня безпеки для користувачів, і при цьому частина користувачів може бути в зоні, в якій, наприклад, є доступ до мережі Інтернет через Wi-fi, можливість користування комп'ютерами без розподілу користувачів на цільові групи. Крім того, в підприємстві комп'ютерна мережа може не поділятись на зони з різними значеннями рівнів безпеки, а всі вузли

перебуватимуть в одній зоні. Введемо складову частину $\alpha'_{1,S_i,3,11}$, яка формуватиме значення $\alpha'_{1,S_i,3}$, і його значення буде характеристичним показником поділу вузлів в мережі між зонами з різним рівнем безпеки. Значення $\alpha'_{1,S_i,3,11}$ визначимо для всіх комп'ютерних станцій так, щоб воно належало проміжку $\left[1-\alpha_1^{r,3,11};1\right]$, де локальний рівень значущості $\alpha_1^{r,3,11}$ є часткою від одиниці. Визначення значення $\alpha'_{1,S_i,3,11}$ здійснимо так:

$$\alpha'_{1,S_i,3,11}=1-\frac{n_{1,S_i,3,11,k}}{n_{1,S_i,3,11,p}}\cdot\alpha_1^{r,3,11}\ , \tag{2.41}$$

де $n_{1,S_i,3,11,k}$ – кількість комп'ютерних станцій в мережі, які перебувають поза зоною підвищеного рівня безпеки; $n_{1,S_i,3,11,p}$ – кількість комп'ютерних станцій в корпоративній мережі.

Якщо в мережі декілька зон з різними значеннями рівнів безпеки, тоді за зону підвищеного рівня безпеки вибирається один сегмент мережі, а всі решта вважатимемо такими, що перебувають поза нею і, відповідно, комп'ютерні станції в мережі перебувають поза зоною підвищеного рівня безпеки.

В корпоративній мережі обов'язково повинні використовуватись для забезпечення безпеки та захисту інформації антивірусні засоби. Як правило, за допомогою таких засобів створюють багаторівневу систему, в якій кожен з рівнів має своє призначення і спрямування. Тому, введемо множину $M_{AZ,KM}=\{m_{AZ,KM,1},m_{AZ,KM,2},\dots,m_{AZ,KM,n_{M_{AZ,KM}}}\}$, елементами якої будуть рівні системи з відповідними спеціалізованими антивірусними засобами в мережі. Кількість елементів множини $M_{AZ,KM}$ позначимо $n_{M_{AZ,KM}}$. Наприклад, на першому рівні може бути здійснена перевірка трафіку мережі, на другому – захист поштових серверів, на третьому – файловий сервер, на четвертому – сканування файлів в конкретній комп'ютерній станції, на п'ятому – здійснення моніторингу подій в конкретній комп'ютерній станції. Приймемо, що ці елементи мають однакову вагу в контексті антивірусного захисту в корпоративній мережі. Крім того, вони можуть бути застосовані до певних сегментів мережі. В певних сегментах мережі можуть бути застосовані всі рівні, а в інших – тільки деякі. Із елементів множини $M_{AZ,KM}$ можуть формуватись різні підмножини. Поєднання декількох елементів множини $M_{AZ,KM}$ в підмножини покращує значення рівня безпеки у вузлах мереж. В межах певних

сегментів можуть бути різні рівні, тому такі випадки задаватимуться об'єднаннями підмножин. Це впливатиме на забезпечення антивірусного захисту корпоративної мережі. Значення рівня безпеки повинно враховувати всі наявні підмножини. Локальний рівень значущості для характеристичного показника багаторівневого антивірусного захисту в корпоративних мережах може бути різним для всіх вузлів в мережі. Це залежить від конкретного антивірусного забезпечення комп'ютерної станції. Введемо складову частину $\alpha'_{1,S_i,3,12}$, яка формуватиме значення $\alpha'_{1,S_i,3}$, і його значення буде характеристичним показником багаторівневого антивірусного захисту в корпоративних мережах. Значення $\alpha'_{1,S_i,3,12}$ визначимо для всіх комп'ютерних станцій так, щоб воно належало проміжку $\left[1-\alpha_1^{r,3,12};1\right]$, де локальний рівень значущості $\alpha_1^{r,3,12}$ є часткою від одиниці. Введемо функцію $f_{M_{AZ,KM}}$ значення якої задамо так:

$$f_{M_{AZ,KM}}\left(m_{AZ,KM,w}\right)=\begin{cases}0, \text{якщо } n_{M_{AZ,KM}}=0;\\ \frac{1}{n_{M_{AZ,KM}}}, \text{якщо } n_{M_{AZ,KM}}>0,\end{cases} \tag{2.42}$$

де $w=1,2,\dots,n_{M_{AZ,KM}}$; $n_{M_{AZ,KM}}$ - кількість елементів множини $M_{AZ,KM}$.

Обчислення значення $\alpha'_{1,S_i,3,12}$ для кожного i-того вузла здійснимо так:

$$\alpha'_{1,S_i,3,12}=1-\frac{n_{1,S_i,3,12,k}}{n_{1,S_i,3,12,p}}\cdot\prod_{w=1}^{n_{M_{AZ,KM,p}}} f_{M_{AZ,KM}}\left(m_{AZ,KM,w}\right)\cdot\alpha_1^{r,3,12}, \tag{2.43}$$

де $m_{AZ,KM,w}$ - елементи підмножини $M_{AZ,KM,p}$; $n_{M_{AZ,KM,p}}$ – кількість елементів підмножини $M_{AZ,KM,p}$; $n_{1,S_i,3,12,k}$ – кількість комп'ютерних станцій в мережі, які перебувають в межах одного типу системи антивірусного захисту, тобто реалізація відповідає підмножині $M_{AZ,KM,p}$; $n_{1,S_i,3,12,p}$ – кількість комп'ютерних станцій в корпоративній мережі.

На значення рівня безпеки у вузлі комп'ютерної мережі впливають, також, особливості навантаження та наявна потужність комп'ютерної станції. Якщо ресурси комп'ютерної станції не відповідають заданим параметрам навантаження, тоді виконання завдань в ній сповільнюється і додаткова обробка подій щодо аномальних подій чи зловмисних дій ускладниться. Такими характеристичними показниками є обсяг вільного простору жорсткого диску, оперативного запам'ятовуючого пристрою, застосування технології віртуальної пам'яті тощо.

Введемо для цих характеристичних показників локальні рівні значущості $\alpha_1^{r,3,13}$, $\alpha_1^{r,3,14}$, $\alpha_1^{r,3,15}$ відповідно. Значення $\alpha'_{1,S_i,3,13}$, $\alpha'_{1,S_i,3,14}$, $\alpha'_{1,S_i,3,15}$ відповідно визначимо для всіх комп'ютерних станцій в мережі так, щоб вони належали проміжкам $\left[1-\alpha_1^{r,3,13};1\right]$, $\left[1-\alpha_1^{r,3,14};1\right]$, $\left[1-\alpha_1^{r,3,15};1\right]$ і визначались для кожного i-го вузла так:

$$\alpha'_{1,S_i,3,13} = 1 - \frac{u_{1,S_i,3,13,k}}{u_{1,S_i,3,13,p}} \cdot \alpha_1^{r,3,13}, \tag{2.44}$$

$$\alpha'_{1,S_i,3,14} = 1 - \frac{u_{1,S_i,3,14,k}}{u_{1,S_i,3,14,p}} \cdot \alpha_1^{r,3,14}, \tag{2.45}$$

$$\alpha'_{1,S_i,3,15} = 1 - \frac{u_{1,S_i,3,15,p}}{u_{1,S_i,3,15,p}+u_{1,S_i,3,15,k}} \cdot \alpha_1^{r,3,15}, \tag{2.46}$$

де $u_{1,S_i,3,13,k}$ – обсяг заповненого простору жорсткого диску i-го вузла в мережі; $u_{1,S_i,3,13,p}$ – загальний обсяг простору жорсткого диску i-го вузла в мережі; $u_{1,S_i,3,14,k}$ – обсяг заповненого простору оперативного запам'ятовуючого пристрою i-го вузла в мережі; $u_{1,S_i,3,14,p}$ – загальний обсяг простору оперативного запам'ятовуючого пристрою i-го вузла в мережі; $u_{1,S_i,3,15,k}$ – обсяг заповненого простору жорсткого диску процесом, яких реалізується згідно технології віртуальної пам'яті, i-го вузла в мережі; $u_{1,S_i,3,15,p}$ – загальний обсяг заповненого простору оперативного запам'ятовуючого пристрою та жорсткого диску процесом, яких реалізується згідно технології віртуальної пам'яті, i-го вузла в мережі.

Обчислювальні ресурси комп'ютерних станцій в частині характеристичного показника їх завантаженості важливі та впливають на значення рівнів безпеки, також, і в контексті встановлення та функціонування в них систем виявлення вторгнень і завантаженість різними системами антивірусного захисту, для виконання процесів яких потрібні ресурси на постійній основі.

Якщо в корпоративній мережі відсутні мережні екрани, системи антивірусного захисту та системи виявлення вторгнень, тоді значення, які відповідають рівням довіри до результату обчислень, встановлюються такими, що дорівнюють нижнім межам відповідних проміжків.

Аналогічно до визначених характеристичних показників, які формуватимуть значення рівня безпеки в певному вузлі в мережі, можуть бути додані інші характеристичні показники, тобто їх кількість може бути збільшено.

Значення $\alpha'_{1,S_i,3}$, яке враховує рівень безпеки вузла в мережі, визначимо з врахуванням його складових значень $\alpha'_{1,S_i,3,1} - \alpha'_{1,S_i,3,15}$ так:

$$\alpha'_{1,S_i,3} = \frac{1}{15} \cdot \sum_{w=1}^{15} \alpha'_{1,S_i,3,w}. \quad (2.47)$$

Значення $\alpha'_{1,S_i,3}$ знаходитиметься в проміжку $\left[1 - \alpha_1^{r,3}; 1\right]$, де $\alpha_1^{r,3}$ є рівнем значущості і визначатиметься так: $\alpha_1^{r,3} = \frac{1}{15} \cdot \sum_{w=1}^{15} \alpha_1^{r,3,w}$.

В системі S в певний час будуть компоненти, в яких може бути центр прийняття рішень системи, але при цьому вони будуть неактивні, тоді значення рівня безпеки в таких компонентах відрізнятимуться від значень в компонентах, в яких активний центр прийняття рішень. З цих компонент в динамічних компонентах можуть використовуватись певні функції, які відносяться до забезпечення функціонування центру прийняття рішень системи, але сама компонента в поточний момент часу не буде активною частиною центру прийняття рішень. Значення $\alpha'_{1,S_i,4}$, яке враховує виконання функцій-підмножин в компонентах з неактивними підсистемами центру прийняття рішень системи та належить проміжку $\left[1 - \alpha_1^{r,4}; 1\right]$, визначимо так:

$$\alpha'_{1,S_i,4} = 1 - \frac{n_{1,S_i,4,p}}{n_{1,S_i,4,k}} \cdot \alpha_1^{r,4}, \quad (2.48)$$

де $n_{1,S_i,4,k}$ - загальна кількість функцій-підмножин в i-ій компоненті; $n_{1,S_i,4,p}$ - кількість функцій-підмножин в i-ій компоненті взятих для її формування з решти компонент системи S; $\alpha_1^{r,4}$ – рівень значущості для i-ої компоненти з функціями-підмножинами неактивних підсистем центру прийняття рішень системи.

Кількість функцій-підмножин, які приймали участь у виконанні поточного завдання, що відноситься до завдань центру прийняття рішень системи S, впливатиме на значення рівня безпеки, бо при виконанні більшої кількості функцій-підмножин при реалізації розподілених обчислень довіра до їх результатів буде менше, ніж при виконанні меншої кількості функцій-підмножин. Значення $\alpha'_{1,S_i,5}$ належить проміжку $\left[1 - \alpha_1^{r,5}; 1\right]$ та визначатимемо його так:

$$\alpha'_{1,S_i,5} = 1 - \frac{n_{1,S_i,5,p}}{n_{1,S_i,5,k}} \cdot \alpha_1^{r,5}, \quad (2.49)$$

де $n_{1,S_i,5,p}$ - кількість функцій-підмножин в i-ій компоненті, які були задіяні при виконанні поточного завдання, що відноситься до завдань центру прийняття рішень системи S; $n_{1,S_i,5,k}$ - найбільша кількість функцій-підмножин, які були задіяні при виконанні поточного завдання, що відноситься до завдань центру прийняття рішень системи S, тобто $n_{1,S_i,5,k} = max\left(n_{1,S_1,5,p}, n_{1,S_2,5,p}, \dots, n_{1,S_k,5,p}\right),\ i = 1,2,\dots,k;\ \alpha_1^{r,5}$ – рівень значущості для i – ої компоненти щодо врахування кількості функцій-підмножин, які приймали участь у виконанні поточного завдання, що відноситься до завдань центру прийняття рішень системи S.

Крім обчислень, які виконуються функціями-підмножинами, що є функціями-підмножинами центру прийняття рішень системи S, розглянемо обчислення, які виконуються функціями-підмножинами, що не є функціями-підмножинами центру прийняття рішень системи S. Ці обчислення, пов’язані із завданнями системи з виявлення зловмисних подій чи аномальних проявів і вони не відносяться до обчислень, які виконуються для завдань центру прийняття рішень. Функції-підмножини для виконання таких обчислень можуть бути резидентними або такими, що викликаються певними іншими функціями-підмножинами. Оскільки вони не впливатимуть безпосереднього на прийняття рішень та в конкретному вузлі в мережі може не бути ознак зловмисних впливів чи аномальних проявів, то значення рівня безпеки для таких обчислень буде вище, ніж у випадку з обчисленнями для центру прийняття рішень системи. Ці обчислення теж переважно будуть розподіленими, і тому, відповідно, вимагатимуть певного часу на виконання та формування результату. В зв’язку з цим значення рівня до результатів таких обчислень теж може бути різним в залежності від порядку виконання функцій-підмножин в динамічних компонентах, часу витраченого на пересилання результатів виконання, рівня безпеки вузла, виконання функцій-підмножин в трьох типах компонентів (з активними наявними підсистемами центру прийняття рішень системи, з неактивними наявними підсистемами центру прийняття рішень системи, з відсутніми підсистемами центру прийняття рішень системи), кількості функцій-підмножин, які були використані при виконанні завдання. Тоді, коефіцієнти ваг функцій-підмножин задамо як функції з п’ятьма аргументами так:

$$\alpha'_{2,S_{k+1,n}} = f_{\alpha'_{2,S_{k+1,n}}}\left(\alpha'_{2,S_{k+1,n},1}, \alpha'_{2,S_{k+1,n},2}, \alpha'_{2,S_{k+1,n},3}, \alpha'_{2,S_{k+1,n},4}, \alpha'_{2,S_{k+1,n},5}\right), \quad (2.50)$$

де $S_{k+1,n}$ – компоненти системи, в яких відсутні функції-підмножини для центру прийняття рішень системи; $(k+1)$ - n – номери компонент, в яких відсутній центр; $\alpha'_{2,S_{k+1,n},1}$ - значення, яке враховує порядок виконання функцій-підмножин в динамічних компонентах; $\alpha'_{2,S_{k+1,n},2}$ - значення, яке враховує відносний час витрачений на пересилання проміжних результатів виконаних обчислень; $\alpha'_{2,S_{k+1,n},3}$ - значення, яке враховує рівень безпеки вузла в мережі; $\alpha'_{2,S_{k+1,n},4}$ - значення, яке враховує виконання функцій-підмножин в трьох типах компонентів (з активними наявними підсистемами центру прийняття рішень системи, з неактивними наявними підсистемами центру прийняття рішень системи, з відсутніми підсистемами центру прийняття рішень системи); $\alpha'_{2,S_{k+1,n},5}$ - значення, що враховує кількість функцій-підмножин, які приймали участь у виконанні завдання.

Значення $\alpha'_{2,S_{k+1,n},1}, \alpha'_{2,S_{k+1,n},2}, \alpha'_{2,S_{k+1,n},3}, \alpha'_{2,S_{k+1,n},4}, \alpha'_{2,S_{k+1,n},5}$ належать проміжкам $[1-\alpha_2^{r,w}; 1]$ $(w = 1,2,\dots,5)$, в яких задано відповідно рівні значущості $\alpha_2^{r,w}$ $(w = 1,2,\dots,5)$ для кожного із них. Визначення значень здійснимо з врахуванням їх кореляції із значеннями аргументів функції $f_{\alpha'_{2,S_i}}$ (формула (2.18)), бо частина характеристичних показників будуть збіжними для кожного з них.

Значення $\alpha'_{2,S_{k+1,n},1}$ залежатиме від кількості функцій-підмножин, які будуть формувати функціонал компоненти системи порівняно з кількістю функцій-підмножин, які призначені для забезпечення функціонування центру прийняття рішень системи. Тому, визначатимемо його з врахуванням значення $\alpha'_{1,S_i,j,1}$ (формула (2.23)) так:

$$\alpha'_{2,S_{k+1,n},j,1} = 1 - \alpha_2^{r,1} + \left(n_{S_{k+1,n},max,f,2} - f_{nom}(j,1)\right) \cdot \frac{\alpha_2^{r,1}}{n_{S_{k+1,n},max,f,2}-1} - \frac{K_{f,j}\cdot\alpha_2^{r,1}}{n_{S_{k+1,n},max,f,2}};$$

$$\alpha_2^{r,1} = \frac{n_{2,S_{k+1,n},j,1,k}}{n_{2,S_{k+1,n},j,1,p}} \cdot \alpha_1^{r,1}, \quad (2.51)$$

де $n_{2,S_{k+1,n},1,p}$ - кількість функцій-підмножин, які будуть формувати функціонал i-ої компоненти; $i = k+1, k+2, \dots, n$; $n_{2,S_{k+1,n},1,k}$ - кількість функцій-підмножин, які не використовуються для формування центру прийняття рішень системи S; $f_{nom}(j,1)$

– перше найменше значення координати вектору $v_{\alpha'_{1,S_{k+1,n}},j}$; $n_{S_{k+1,n},max,f,2}$ - кількість задіяних на виконання функцій-підмножин при вирішенні певного завдання; $K_{f,j}$ – кількість викликів j-тої функції-підмножини; вектор $v_{\alpha'_{1,S_{k+1,n}},j}$ формується аналогічно до вектору $v_{\alpha'_{1,S_i},j}$ за тим же конструктивним правилом.

Значення $\alpha'_{2,S_{k+1,n},1}$ згідно формули (2.51) визначаємо так:

$$\alpha'_{2,S_{k+1,n},1} = \frac{\sum_{j=1}^{n_{S_{k+1,n},max}} \alpha'_{2,S_{k+1,n},j,i}}{n_{S_{k+1,n},max}}, \tag{2.52}$$

де $n_{S_{k+1,n},max}$ - кількість задіяних на виконання функцій-підмножин при вирішенні певного завдання.

Таким чином, збільшення наповнення компонент функціями-підмножинами, які не використовуються для формування центру прийняття рішень системи S, наближатиме значення рівня значущості $\alpha_2^{r,1}$ до значення $\alpha_1^{r,1}$. Значення рівня значущості $\alpha_2^{r,1}$ менше за значення $\alpha_1^{r,1}$, тому проміжок, в якому буде знаходитись значення рівня довіри до результату буде менше і, відповідно, наближатиметься до значення одиниці.

Аналогічно визначатимемо значення $\alpha'_{2,S_{k+1,n},2}$, яке враховує відносний час витрачений на пересилання проміжних результатів виконаних обчислень, через значення $\alpha'_{2,S_i,2}$ (формула (2.26)) та рівень значущості $\alpha_2^{r,2}$ так:

$$\alpha'_{2,S_{k+1,n},j,2} = 1 - \frac{\sum_{q=1}^{K_{f,j,t}} f_{nom,t}(j,q)}{\sum_{j=1}^{n_{S_{k+1,n},max}} \sum_{q=1}^{K_{f,j,t}} f_{nom,t}(j,q)} \cdot \alpha_2^{r,2};\ \alpha_2^{r,1} = \frac{n_{2,S_{k+1,n},1,k}}{n_{2,S_{k+1,n},1,p}} \cdot \alpha_1^{r,1}, \tag{2.53}$$

де $n_{2,S_{k+1,n},1,p}$ - кількість функцій-підмножин, які будуть формувати функціонал i-ої компоненти; $n_{2,S_{k+1,n},1,k}$ - кількість функцій-підмножин, які не використовуються для формування центру прийняття рішень системи S; $i = k+1, k+2, \dots, n$; $f_{nom,t}(j,q)$ – значення координати вектору $v_{\alpha'_{1,S_{k+1,n}},j,t}$, які формуються у векторі аналогічно до конструктивного правила для формування координат вектору $v_{\alpha'_{1,S_i},j,t}$; $n_{S_{k+1,n},max}$ - кількість функцій-підмножин в компоненті; $K_{f,j,t}$ – кількість викликів j-тої функції-підмножини.

Значення $\alpha'_{1,S_i,2}$ визначаємо згідно формули (2.53) так:

$$\alpha'_{2,S_{k+1,n},2} = \frac{\sum_{j=1}^{n_{S_{k+1,n},max}} \alpha'_{2,S_i,j,i}}{n_{S_{k+1,n},max}}, \quad (2.54)$$

де $n_{S_{k+1,n},max}$ - кількість функцій-підмножин в компоненті.

Значення $\alpha'_{2,S_{k+1,n},3}$, яке враховує рівень безпеки вузла в мережі визначимо аналогічно до визначення значення $\alpha'_{1,S_i,3}$. При цьому доповнимо його характеристичним показником, який відображатиме активність датчиків зловмисних ознак. Значення рівня значущості $\alpha_2^{r,3}$ корелюватимемо зі значенням $\alpha_1^{r,3}$ та визначатимемо згідно локальних рівнів значущості $\alpha_2^{r,3,w}$ $(w = 1,2,\dots,16)$. Визначення значення здійснюватимемо так:

$$\alpha'_{2,S_{k+1,n},3} = \frac{1}{16} \cdot \sum_{w=1}^{16} \alpha'_{2,S_{k+1,n},3,w};$$

$$\alpha_2^{r,3} = \frac{1}{16} \cdot \sum_{w=1}^{16} \alpha_2^{r,3,w};$$

$$\alpha_2^{r,3,w} = \frac{n_{2,S_{k+1,n},1,k}}{n_{2,S_{k+1,n},1,p}} \cdot \alpha_1^{r,3,w};\ w = 1,2,\dots,16, \quad (2.55)$$

де $n_{2,S_{k+1,n},1,p}$ - кількість функцій-підмножин, які будуть формувати функціонал i-ої компоненти; $n_{2,S_{k+1,n},1,k}$ - кількість функцій-підмножин, які не використовуються для формування центру прийняття рішень системи S; $i = k+1, k+2, \dots, n$.

Для визначення значення $\alpha'_{2,S_{k+1,n},3}$ потрібно визначити значення $\alpha'_{2,S_{k+1,n},3,16}$ характеристичного показника, який відображатиме активність датчиків зловмисних ознак. Враховуючи, що в конкретній i – ій комп'ютерній станції наявна підсистема моніторингу зловмисних подій, то кількість таких засобів підсистеми та кількість активних з них, які вказуватимуть на обробку подій на наявність зловмисних дій, впливатимуть на значення рівня безпеки в ній і, тому, з врахуванням цієї складової частини та її задання визначимо її значення в проміжку $\left[1 - \alpha_2^{r,3,16}; 1\right]$ так:

$$\alpha'_{2,S_{k+1,n},3,16} = 1 - \frac{n_{2,S_{k+1,n},1,k,16}}{n_{2,S_{k+1,n},1,p,16}} \cdot \alpha_2^{r,3,16}, \quad (2.56)$$

де $n_{2,S_{k+1,n},1,p,16}$ – загальна кількість датчиків ознак в i-ій компоненті; $n_{2,S_{k+1,n},1,k,16}$ - кількість активованих зловмисними впливами чи аномальними проявами датчиків ознак в i-ій компоненті; $\alpha_2^{r,3,16}$ – локальний рівень значущості.

Значення $\alpha'_{2,S_{k+1,n},j,4}$ враховує виконання функцій-підмножин в трьох типах компонентів (з активними наявними підсистемами центру прийняття рішень системи, з неактивними наявними підсистемами центру прийняття рішень системи, з відсутніми підсистемами центру прийняття рішень системи). Визначимо його значення з врахуванням належності проміжку $\left[1-\alpha_2^{r,4};1\right]$ так:

$$\alpha'_{2,S_{k+1,n},4}=1-\left(\frac{n_{1,S_{k+1,n},4,p,1}}{n_{1,S_{k+1,n},4,k}}+\frac{n_{1,S_{k+1,n},4,p,2}}{2\cdot n_{1,S_{k+1,n},4,k}}+\frac{n_{1,S_{k+1,n},4,p,3}}{3\cdot n_{1,S_{k+1,n},4,k}}\right)\cdot\alpha_2^{r,4};$$

$$\alpha_2^{r,4}=\frac{n_{2,S_{k+1,n},1,p}}{n_{2,S_{k+1,n},1,k}}\cdot\alpha_1^{r,4}, \qquad (2.57)$$

де $n_{1,S_{k+1,n},4,p,1}$ - кількість функцій-підмножин в i – ій компоненті взятих для її формування з активних наявних підсистемам центру прийняття рішень системи компонент системи S; $n_{1,S_{k+1,n},4,p,2}$ - кількість функцій-підмножин в i – ій компоненті взятих для її формування з неактивних наявних підсистемам центру прийняття рішень системи компонент системи S; $n_{1,S_{k+1,n},4,p,3}$ - кількість функцій-підмножин в i – ій компоненті взятих для її формування з відсутніми підсистемами центру прийняття рішень системи компонент системи S; $n_{1,S_{k+1,n},4,k}$ - загальна кількість функцій-підмножин в i – ій компоненті; $n_{1,S_{k+1,n},4,p}$ - кількість функцій-підмножин в i – ій компоненті взятих для її формування з решти компонент системи S; $\alpha_2^{r,4}$ – рівень значущості для i – ої компоненти; $i=k+1,k+2,\dots,n$.

Значення $\alpha'_{2,S_{k+1,n},5}$ враховує кількість функцій-підмножин, які приймали участь у виконанні завдання. Визначимо його так, щоб воно знаходилось в проміжку $\left[1-\alpha_2^{r,5};1\right]$, і визначатимемо його значення так:

$$\alpha'_{2,S_{k+1,n},5}=1-\frac{n_{2,S_{k+1,n},5,p}}{n_{2,S_{k+1,n},5,k}}\cdot\alpha_2^{r,5};\ \alpha_2^{r,4}=\frac{n_{2,S_{k+1,n},1,p}}{n_{2,S_{k+1,n},1,k}}\cdot\alpha_1^{r,4}, \qquad (2.58)$$

де $n_{2,S_{k+1,n},5,p}$ - кількість функцій-підмножин в i – ій компоненті, які були задіяні при виконанні поточного завдання; $n_{2,S_{k+1,n},5,k}$ - найбільша кількість функцій-підмножин, які були задіяні при виконанні поточного завдання; $\alpha_2^{r,5}$ – рівень значущості для i – ої компоненти щодо врахування кількості функцій-підмножин, які приймали участь у виконанні поточного завдання.

Таким чином, сформовано значення характеристичних показників компонент

системи S для випадку, коли функції-підмножини виконують завдання, що не належать до завдань центру прийняття рішень.

Розглянемо розподілені обчислення, які виконуються функціями-підмножинами як центру прийняття рішень системи, так і рештою функцій. Цей варіант передбачає виконання функцій-підмножин, які не відносяться до частини, що формується з функцій-підмножин центру прийняття рішень, але після їх виконання і за потреби реагування на можливу наявність ЗПЗ, звертаються для додаткових обчислень до функцій-підмножин центру прийняття рішень. В результаті будуть виконуватись функції-підмножини центру прийняття рішень та функції-підмножини, що не відносяться до формування центру прийняття рішень. Аналогічно до попередніх двох випадків введемо коефіцієнти ваг функцій-підмножин і задамо їх як функції з п’ятьма аргументами так:

$$\alpha'_{3,S_{1,n}} = f_{\alpha'_{3,S_{1,n}}}\left(\alpha'_{3,S_{1,n},1}, \alpha'_{3,S_{1,n},2}, \alpha'_{3,S_{1,n},3}, \alpha'_{3,S_{1,n},4}, \alpha'_{3,S_{1,n},5}\right), \qquad (2.59)$$

де $S_{1,n}$ – компоненти системи, в яких наявні функції-підмножини всіх типів; $i = 1, 2, \dots, n$; i – номер компоненти; $\alpha'_{3,S_{1,n},1}$ - значення, яке враховує порядок виконання функцій-підмножин в динамічних компонентах; $\alpha'_{2,S_{1,n},2}$ - значення, яке враховує відносний час витрачений на пересилання проміжних результатів виконаних обчислень; $\alpha'_{2,S_{1,n},3}$ - значення, яке враховує рівень безпеки вузла в мережі; $\alpha'_{2,S_{1,n},4}$ - значення, яке враховує виконання функцій-підмножин в трьох типах компонентів (з активними наявними підсистемами центру прийняття рішень системи, з неактивними наявними підсистемами центру прийняття рішень системи, з відсутніми підсистемами центру прийняття рішень системи); $\alpha'_{2,S_{1,n},5}$ - значення, що враховує кількість функцій-підмножин, які приймали участь у виконанні завдання.

Значення $\alpha'_{3,S_{1,n},1}, \alpha'_{3,S_{1,n},2}, \alpha'_{3,S_{1,n},3}, \alpha'_{3,S_{1,n},4}, \alpha'_{3,S_{1,n},5}$ належать проміжкам $[1 - \alpha_3^{r,w}; 1]$ $(w = 1,2, \dots ,5)$, в яких задано відповідно рівні значущості $\alpha_3^{r,w}$ $(w = 1,2, \dots ,5)$ для кожного із них. Визначення значень здійснимо з врахуванням їх кореляції із значеннями аргументів функції $f_{\alpha'_{3,S_i}}$ (формула (2.18)), бо частина характеристичних показників будуть збіжними для кожного з них. Враховуючи охоплення цими характеристичними показниками всіх функцій-підмножин та процеси, що розпочинатимуться в функціях-підмножинах, які використовуються

для виявлення зловмисних дій та аномальних проявів, і продовжуватимуться терміновими викликами функцій-підмножин, які відносяться до формування центру прийняття рішень, то складність і масштабність таких розподілених обчислень буде більшою, а також враховуючи можливе зниження значення рівня безпеки в певних компонентах. Тому, визначатимемо значення $\alpha'_{3,S_{1,n},1}, \alpha'_{3,S_{1,n},2}, \alpha'_{3,S_{1,n},3}, \alpha'_{3,S_{1,n},4}, \alpha'_{3,S_{1,n},5}$ з врахуванням значень для першого та другого випадків і формул для визначення їх значень так:

$$\alpha_3^{r,w} = \alpha_1^{r,w} + \alpha_2^{r,w}; \alpha'_{3,S_{1,n},j,w} = \alpha'_{1,S_i,j,w} + \alpha'_{2,S_{k+1,n},j,w} - 1; w = 1,2,\dots,5. \quad (2.60)$$

Значення характеристичних показників можуть бути уточнені при деталізації із залученням нових параметрів. А, також, їх кількість може бути розширена. Значення рівнів безпеки є збіжними незалежно від кількості компонентів системи S, тобто від кількості комп'ютерних станцій в корпоративній мережі. Також, всі отримані значення належать заданим для них проміжкам.

Таким чином, отримані аналітичні вирази для характеристичних показників значень рівнів безпеки компонентів є математичними моделями, які формалізують архітектуру компонент системи S згідно наявних в них функцій, їх призначення, взаємодії, місця виконання, формування центру прийняття рішень та оцінювання рівня безпеки виконуваних обчислень, тобто враховуючи специфіку призначення системи S, процеси на рівні функцій в ній та компонентах, задані математичними моделями характеристичних показників рівнів безпеки компонентів. Значення характеристичних показників рівнів безпеки компонентів задають опис оточуючого середовища корпоративної мережі для системи і будуть основою для формування рішень системи S щодо подальших кроків та виявлення ЗПЗ.

2.4. Висновки до другого розділу

Забезпечення безпеки та захисту інформації в корпоративних мережах здійснюється різними засобами різного спрямування. Їх унікальність є дуже важливою в контексті активних дій зловмисників, які через відсутність поінформованості про них матимуть складнощі при здійсненні зловмисних дій. Розроблена архітектура частково централізованих розподілених систем надає

можливість синтезу таких засобів, які створюють проблеми зловмисникам щодо визначення ними центру їх системи, принципів функціонування. В удосконаленій моделі таких систем закладена можливість динамічної зміни конфігурації системи, поділу центру прийняття рішень між різними компонентами, розподілу компонентів за функційними можливостями з наявності в них центру прийняття рішень і, тому, в ній синтезовано властивості адаптивності і самоорганізації безпосередньо в компонентах системи.

Розроблена архітектура компонент частково централізованих систем базується на отриманих аналітичних виразах, які є математичними моделями характеристичних показників значень рівнів безпеки компонентів, що формалізують архітектуру компонент системи S згідно наявних в них функцій, їх призначення, взаємодії, місця виконання, формування центру прийняття рішень та оцінювання рівня безпеки виконуваних обчислень. В характеристичних показниках враховано функційну та кібер- безпеки в корпоративних мережах.

Значення характеристичних показників рівнів безпеки компонентів частково централізованих розподілених систем та оточуючого середовища корпоративних мереж будуть базою для формування рішень щодо її подальших кроків та визначення ЗПЗ. Тому, напрямами подальшого синтезу частково централізованих розподілених систем буде використання значень характеристичних показників рівнів безпеки компонентів в методі організації їх функціонування та методах виявлення ЗПЗ.

Основні наукові результати другого розділу опубліковані в [47, 113-116].

РОЗДІЛ 3.
МЕТОДИ СИНТЕЗУ ЧАСТКОВО ЦЕНТРАЛІЗОВАНИХ РОЗПОДІЛЕНИХ СИСТЕМ ВИЯВЛЕННЯ ЗЛОВМИСНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В КОМП'ЮТЕРНИХ МЕРЕЖАХ

Розроблена архітектура частково централізованих систем, архітектура її компонент та математичні моделі характеристичних показників рівнів безпеки компонентів та оточуючого середовища корпоративної мережі є основою створення нових систем виявлення ЗПЗ в корпоративних мережах, які та функціонування яких будуть невідомі або важко зрозумілі для зловмисників.

Організація функціонування частково централізованих систем в корпоративних комп'ютерних мережах згідно розроблених архітектури та математичних моделей потребує розроблення та імплементації в них методів, які б забезпечували функціонування та реагування на події цими системами, а також дозволили б синтезувати нові математичні моделі для характеристичних показників рівнів безпеки в залежності від динамічно змінюваних та нових засобів різноманітного призначення як для функціонування комп'ютерних мереж, так і в зв'язку з появою нових програмних засобів.

Для синтезу частково централізованих розподілених систем потрібні методи, які дадуть змогу наповнити характеристичними показниками рівнів безпеки та оточуючого середовища ці системи, організації їх функціонування та методи виявлення ЗПЗ.

3.1. Метод синтезу математичних моделей рівнів безпеки

В процесі тривалого експлуатування частково централізованих розподілених систем можуть з'являтись нові технічні засоби для комп'ютерних мереж та нові програмні чи апаратно-програмні засоби різноманітного призначення для користувачів, зокрема, і засоби протидії зловмисним впливам та аномальним проявам. Ці нові засоби або інші існуючі засоби, які раніше не використовувались, можуть бути додані в наявні корпоративні мережі. В результаті таких дій параметри, які задані в частково централізованих розподілених системах, зокрема

першочергово характеристичні показники значень рівнів безпеки компонент, не будуть відповідати поточному стану корпоративної мережі. Тому, потрібно розробити метод, який би надав можливість синтезувати нові математичні моделі рівнів безпеки компонентів та втілювати їх в систему S. Також, може бути потрібною заміна певних наявних в системі математичних моделей рівнів безпеки компонентів на інші, які більш точніше враховують наявну специфіку корпоративних мереж. Наявність методу синтезу нових математичних моделей рівнів безпеки компонентів надав би змогу кваліфікованим адміністраторам корпоративної мережі змінювати налаштування в системі S в залежності від поточних та стратегічних потреб.

Нехай задамо характеристичні показники значень рівнів безпеки компонентів множиною $B = \{\beta'_1, \beta'_2, \dots, \beta'_{N_B}\}$, де β'_i - значення рівнів безпеки компонентів системи, N_B – кількість характеристичних показників, $i = 1,2, \dots, N_B$. Для кожного компонента системи S введемо множини $B_j = \{\beta'_{1,j}, \beta'_{2,j}, \dots, \beta'_{N_B,j}\}$ згідно заданої множини B, елементи якої будуть використані для обчислення значення рівня безпеки всієї системи. Значення $\beta'_{i,j}$ $(i = 1,2, \dots, N_B)$ визначатимуть рівень довіри до результатів розподілених обчислень, які здійснені в різних компонентах системи та характеризують різні показники рівнів безпеки. Введемо для значень $\beta'_{i,j}$ $(i = 1,2, \dots, N_B;\ j = 1,2, \dots, N;\ N$ – кількість компонент в системі, які встановлені в комп'ютерні станції в мережі) проміжок, в якому буде регулюватись нижня межа в залежності від параметру рівня значущості $\alpha_z^{r,i}$ $(i = 1,2, \dots, N_B;\ z = 1,2, \dots, N_z;\ N_z$ – кількість варіантів взаємодії функцій-підмножин) так: $[1 - \alpha_z^{r,i}; 1]$. За рівень значущості приймемо частку від одиниці, яка відображатиме відхилення від рівня довіри до результату розподілених обчислень внаслідок певних подій, архітектурної особливості компоненти тощо. Для двох значень з проміжку $[1 - \alpha_z^{r,i}; 1]$ приймемо за значення з більшим рівнем довіри до результатів обчислень те, яке є більшим. Кожна компонента системи сформована з певної кількості функцій-підмножин в залежності від визначеного призначення компоненти. Для визначення значень $\beta'_{i,j}$ $(i = 1,2, \dots, N_B;\ j = 1,2, \dots, N;\ N$ – кількість компонент в системі, які встановлені в комп'ютерні станції в мережі) потрібно враховувати наявність центру прийняття

рішень системи та типи обчислень, які можуть відноситись до різних груп функцій-підмножин та їх комбінацій.

Характеристичні показники значень рівнів безпеки компонентів можуть бути різними і не завжди типовими. Тому, для їх формалізації з метою оцінювання потрібно застосовувати різні варіанти.

Якщо характеристичний показник за певним критерієм чи декількома критеріями може бути сформований мінімум з двох однотипних елементів, тоді формуємо множину з цих елементів. Далі елементи впорядковуємо за їх впливом на безпеку компоненти і з них формуємо вектор, координати якого впорядковані. Після цього конструюємо функцію, в якості аргументів якої будуть координати вектору, для відображення аргументів у значення з проміжку $\left[1-\alpha_z^{r,i};1\right]$. В результаті отримані значення будуть значеннями характеристичних показників рівнів безпеки компонент. Ці значення можуть бути уточненні за певними показниками, які впливатимуть на безпеку. Такими показниками можуть бути функції-підмножини (кількість, призначення, активність, використання при виконанні завдань тощо), кількість компонент системи (активних, належних до центру прийняття рішень системи тощо). Коригування значення характеристичних показників здійснюємо введенням коефіцієнта коригування, який враховує додаткові показники, що отримуються кількісними числовими значеннями. Цей коефіцієнт коригує рівень значущості $\alpha_z^{r,i}$ таким чином, щоб отримане значення характеристичного показника належало проміжку $\left[1-\alpha_z^{r,i};1\right]$.

Якщо характеристичний показник за певним критерієм чи декількома критеріями може бути заданий кількісними значеннями, тоді його значення потрібно задати залежним від цих кількісних значень з подальшим відображенням його в проміжок $\left[1-\alpha_z^{r,i};1\right]$. Коефіцієнт коригування цього значення не задаємо, бо він буде збіжним з коефіцієнтом при рівні значущості $\alpha_z^{r,i}$.

Якщо характеристичний показник за певним критерієм чи декількома критеріями повинен бути сформований з декількох різних та різнотипних показників, тоді вважатимемо їх локальними складовими частинами та здійснюватимемо визначення значення характеристичного показника рівня безпеки компонент як середньоарифметичне або середньозважене значення серед всіх

значень локальних показників так, щоб це значення належало проміжку $[1-\alpha_z^{r,i};1]$. При цьому для всіх значень локальних показників визначаємо локальні рівні значущості, які визначатимуть як середньоарифметичні або середньозважені значення рівні значущості $\alpha_z^{r,i}$. Локальні значення характеристичних показників належатимуть проміжкам $[1-\alpha_z^{r,i,l};1]$, де $\alpha_z^{r,i,l}$ – локальні рівні значущості.

З метою уникнення випадків, коли для декількох різних показників в одній чи різних компонентах системи S або щодо певної конкретної функції-підмножини можуть бути однаковими середньоарифметичні значення, потрібно при визначеннях таких значень враховувати початкове формування динамічних компонентів системи з різних функцій-підмножин, які можуть бути в різних компонентах. Особливо такі випадки в частині визначення середньоарифметичного значення коефіцієнта для рівня значущості, який дорівнюватиме 0,5 і, відповідно, значення характеристичних показників розподілятимуться на проміжках $[1-\alpha_z^{r,i};1]$ згідно рівномірного розподілу, можуть стосуватись дискретних величин, які потрібно задати числовими значеннями. Якщо формується вектор з характеристичних показників, які задано якісними показниками елементів однієї множини, тоді сформовані числові значення таких показників на проміжку будемо розміщувати не рівномірно через однакові інтервали, а з врахуванням певної ваги, яку визначатимемо коригуючим коефіцієнтом $\gamma_{F,1}$, що враховуватиме кількість та особливість функцій-підмножин в усіх компонентах системи. Визначимо коефіцієнт впливу кількості підмножин-функцій в усіх компонентах в залежності від того, як вони формують динамічні компоненти, так:

$$\delta_{1,F}=\sum_{i=1}^{n}\frac{\sum_{j=1}^{n_{F,S_i}} n_{F,S_i,j,0}}{\sum_{j=1}^{n_{F,S_i}} n_{F,S_i,j}}, \tag{3.1}$$

де n_{F,S_i} – кількість функцій-підмножин в S_i компоненті; $n_{F,S_i,j}$ – значення показника наявності функції-підмножини в динамічній компоненті, причому як наявної безпосередньо в компоненті так і з решти компонент; $n_{F,S_i,j,0}$ – значення показника наявності функції-підмножини в динамічній компоненті не наявної безпосередньо в компоненті, а саме з решти компонент.

Визначимо, також, коефіцієнт впливу кількості підмножин-функцій в певних конкретних S_i компонентах в залежності від того, як вони формують динамічні компоненти, так:

$$\delta_{2,F,S_i} = \frac{\sum_{j=1}^{n_{F,S_i}} n_{F,S_i,j,0}}{\sum_{j=1}^{n_{F,S_i}} n_{F,S_i,j}}, \tag{3.2}$$

де n_{F,S_i} – кількість функцій-підмножин в S_i компоненті; $n_{F,S_i,j}$ – значення показника наявності функції-підмножини в динамічній компоненті, причому як наявної безпосередньо в компоненті так і з решти компонент; $n_{F,S_i,j,0}$ – значення показника наявності функції-підмножини в динамічній компоненті не наявної безпосередньо в компоненті, а саме з решти компонент.

Визначатимемо коригуючий коефіцієнт $\gamma_{F,1}$, що враховуватиме кількість та особливість функцій-підмножин в усіх компонентах системи, так:

$$\gamma_{F,1} = \left(1 - \delta_{1,F}\right)^d, \tag{3.3}$$

де d – кількість характеристичних показників у векторі.

Значення коригуючого коефіцієнту $\gamma_{F,1}$ використовуватимемо для формування розподілу характеристичних показників на відрізку $\left[1 - \alpha_z^{r,i}; 1\right]$.

Аналогічно, визначатимемо коригуючий коефіцієнт $\gamma_{F,2}$, що враховуватиме кількість та особливість функцій-підмножин в усіх компонентах системи так:

$$\gamma_{F,2} = \left(1 - \delta_{2,F}\right)^d, \tag{3.4}$$

де d – кількість характеристичних показників у векторі.

Значення коригуючого коефіцієнту $\gamma_{F,2}$ використовуватимемо для формування розподілу характеристичних показників на відрізку $\left[1 - \alpha_z^{r,i}; 1\right]$.

Якщо $\delta_{1,F}$ чи $\delta_{2,F}$ дорівнюють нулеві, тобто всі функції-підмножини знаходяться в компонентах, то отримуємо значення коригуючих коефіцієнтів такими, що дорівнюють одиниці і, відповідно, рівномірний розподіл.

Якщо наявна кореляція певних характеристичних показників між собою, тоді потрібно встановити ступінь кореляції та виразити цю взаємну залежність аналітичним виразом. В зв’язку з таким випадком певні значення характеристичних

показників можуть визначатись через аналітичні вирази певних корельованих з ним характеристичних показників.

Отримані значення характеристичних показників рівнів безпеки компонентів будуть задані аналітичними виразами. Процес отримання таких аналітичних виразів формує такі кроки методу синтезу математичних моделей рівнів безпеки компонентів системи:

1) визначити характеристичні показники значень рівнів безпеки компонентів та задати їх множиною $B = \{\beta'_1, \beta'_2, \dots, \beta'_{N_B}\}$ (N_B – кількість характеристичних показників);

2) визначити критерії та підкритерії для встановлення рівнів значущості $\alpha_z^{r,i}$ ($i = 1,2, \dots, N_B$; $z = 1,2, \dots, N_z$; N_z – кількість варіантів взаємодії функцій-підмножин), зокрема їх групування за типовими ознаками, поділ в залежності від типу компонент, варіантів взаємодії;

3) встановити згідно кроку 2 рівні значущості $\alpha_z^{r,i}$ ($i = 1,2, \dots, N_B$; $z = 1,2, \dots, N_z$; N_z – кількість варіантів взаємодії функцій-підмножин) для кожного характеристичного показника значень рівнів безпеки компонентів $B_j = \{\beta'_{1,j}, \beta'_{2,j}, \dots, \beta'_{N_B,j}\}$;

4) визначити проміжки $[1 - \alpha_z^{r,i}; 1]$ для кожного значення характеристичного показників рівнів безпеки компонентів $B_j = \{\beta'_{1,j}, \beta'_{2,j}, \dots, \beta'_{N_B,j}\}$;

5) визначити кількість функцій-підмножин в кожній з компонент системи S та згрупувати їх за критеріями з кроку 2;

6) задати кожен з характеристичних показників рівнів безпеки компоненти множиною елементів, які будуть характеризувати можливі варіанти та засоби забезпечення реалізації безпеки певного типу, і ранжувати їх введенням відповідної функції ранжування чи вектору з впорядкованими координатами;

7) якщо характеристичні показники рівнів безпеки компоненти заданою множиною елементів і ці елементи впорядковані в координатах вектору та введена функція ранжування, тоді потрібно визначити найбільше і найменше значення з координат вектору, встановити крок для унормовуваних значень з проміжку найбільшого і найменшого значення з координат вектору, задати функцію відповідності унормованих значень з врахуванням одного з коригуючих

коефіцієнтів $\gamma_{F,1}$ чи $\gamma_{F,2}$ в залежності від відношення до компонент чи компоненти в проміжок $\left[1-\alpha_z^{r,i};1\right]$ визначений на кроці 4 та задати аналітичний вираз обчислення значень характеристичних показників рівнів безпеки компонентів в різних компонентах системи S так:

$$\beta'_{i,j}=1-\mu\cdot\alpha_z^{r,i}, \tag{3.5}$$

де μ – коефіцієнт коригування, при якому більше з двох значень вказує на менший рівень безпеки компонент; $\mu\in[0,1]$; $\alpha_z^{r,i}$ – i-тий рівень значущості; $i=1,2,\dots,N_B$; $z=1,2,\dots,N_z$; N_z – кількість варіантів взаємодії функцій-підмножин; N_B – кількість характеристичних показників;

8) якщо характеристичні показники за певним критерієм чи декількома критеріями задано кількісними значеннями, тоді їх значення потрібно задати аналітичним виразом залежним від цих кількісних значень з відображенням його в проміжок $\left[1-\alpha_z^{r,i};1\right]$ за формулою (3.5);

9) якщо характеристичні показники за певним критерієм чи декількома критеріями спочатку можна задати з декількох різних та різнотипних локальних показників, тоді визначити значення характеристичних показників рівня безпеки компонент як середньоарифметичні або середньозважені значення серед всіх значень відповідних локальних показників так, щоб ці значення належали проміжкам $\left[1-\alpha_z^{r,i};1\right]$ і локальні рівні значущості визначити, також, як середньоарифметичні або середньозважені значення рівні значущості $\alpha_z^{r,i}$, так:

$$\beta'_{i,j}=\frac{1}{N_{B,i}}\cdot\sum_{w=1}^{N_{B,i}}\rho_w.\beta'_{i,j,w}. \tag{3.6}$$

де значення характеристичних показників рівнів безпеки компонент $\beta'_{i,j}$ належить проміжку $\left[1-\alpha_z^{r,i};1\right]$, $i=1,2,\dots,N_B$; N_B – кількість характеристичних показників; $j=1,2,\dots,N$; N – кількість компонент в системі, які встановлені в комп'ютерні станції в мережі; ρ_w – ваговий коефіцієнт для локальних значень $\beta'_{i,j,w}$; $w=1,2,\dots,N_{B,i}$; $N_{B,i}$ – кількість локальних показників для i – того характеристичного показника; $\sum_{w=1}^{N_{B,i}}\rho_w=N_{B,i}$; для всіх $\rho_w=1$ значення $\beta'_{i,j}$ буде обчислене як середньоарифметичне; для всіх різних ρ_w значення $\beta'_{i,j}$ буде обчислене як

середньозважене; рівень значущості $\alpha_z^{r,i}$ для задання проміжку визначається з врахування локальних рівнів значущості $\alpha_z^{r,i,l,w}$: $\alpha_z^{r,i} = \frac{1}{N_{B,i}} \cdot \sum_{w=1}^{N_{B,i}} \rho_w . \alpha_z^{r,i,l,w}$;

10) якщо наявна кореляція певних характеристичних показників між собою, тоді потрібно встановити ступінь кореляції та виразити цю взаємну залежність аналітичним виразом так:

$$\beta'_{i,j} = \sum_{u=1}^{i-1} \sigma_u \cdot \beta'_{u,j} + \sum_{u=i+1}^{N_B} \sigma_u \cdot \beta'_{u,j}, \quad (3.7)$$

де σ_u – частка від одиниці, яка виражає вагу кореляції значень $\beta'_{u,j}$ та $\beta'_{i,j}$; $u = 1,2,\dots,N_B; u \neq i$.

Таким чином, розроблено метод синтезу математичних моделей рівнів безпеки компонентів системи, який дає змогу отримувати нові аналітичні вирази для рівнів безпеки компонентів системи (формули (3.5)-(3.7)).

Синтез математичних моделей рівнів безпеки компонентів системи згідно розробленого методу дає змогу комплексного опису оточуючого середовища та процесів, які відбуватимуться в частково централізованих розподілених системах і відноситимуться до оцінювання безпеки компонент системи.

Аналіз математичних моделей, які задано аналітичними виразами в формулах (2.18)-(2.59), підтверджує їх адекватність оточуючому середовищу корпоративної мережі та процесам при застосуванні і коректність щодо граничних меж. Наприклад, в формулі (3.5) при відсутності чинників, які впливатимуть на безпеку процесів, що відбуваються в комп'ютерній станції в мережі, значення характеристичного показника $\beta'_{i,j} = 1$, тобто коефіцієнт коригування $\mu = 0$. А при $\mu = 0$ значення характеристичного показника $\beta'_{i,j} = 1 - \alpha_z^{r,i}$ відповідає нижній межі проміжку $[1 - \alpha_z^{r,i}; 1]$. Аналогічно, можна показати відповідність межам проміжку $[1 - \alpha_z^{r,i}; 1]$ для результатів кроків 8-10 розробленого методу синтезу математичних моделей рівнів безпеки компонентів системи.

Отримані значення $\beta'_{i,j}$ характеристичних показників рівнів безпеки в компонентах системи будуть використані для оцінювання результатів розподілених обчислень, отриманих з різних компонентів системи, з метою визначення ступеня довіри до них. Розроблений метод може бути застосований для дискретних та неперервних величин характеристичних показників.

3.2. Метод організації функціонування частково централізованих розподілених систем згідно принципів самоорганізації та адаптивності

3.2.1. Організація функціонування частково централізованих розподілених систем згідно принципів самоорганізації та адаптивності

Функціонування частково централізованих розподілених систем згідно принципів самоорганізації та адаптивності забезпечується не тільки організацією комунікації між їх компонентами чи виконанням певних спеціально орієнтованих завдань для виконання яких вони створені, але першочергово внутрішніми механізмами, методами та алгоритмами, які дадуть таким системам можливості вирішувати поставлені завдання без втручання користувача, самостійно приймати рішення щодо подальших кроків системи та адаптуватись в залежності під зміни зовнішнього середовища і внутрішніх подій. В частково централізованих розподілених системах потрібно синтезувати такі принципи функціонування, функційні особливості та характеристики:

1) формування системи з компонент;
2) комунікація між компонентами системи;
3) підтримка цілісності системи;
4) часткова централізація;
5) міграція центру прийняття рішень системи;
6) оцінювання стану компонент та системи;
7) оцінювання результатів розподілених обчислень в компонентах;
8) формування рішення в декількох компонентах;
9) перебудова архітектури системи;
10) визначення подальших кроків системи в поточний момент часу;
11) завершення функціонування компонентів та системи.

Здійснимо деталізацію кожного із заданих принципів функціонування та характеристик. Всі вони повинні бути синтезовані в таких системах повністю. В результаті такого синтезу система стане самоорганізованою, адаптивною та частково централізованою.

Формування системи S з компонент може бути здійснено на початку її встановлення і активації, в процесі функціонування за потреби та після увімкнення комп'ютерних станцій в мережі. Також, до системи можуть додаватись нові компоненти або вилучатись наявні. Крім того, частина комп'ютерних станцій, в які встановлені компоненти, може бути вимкненою тривалий час і, тому, система міститиме меншу частину компонент. Увімкнення комп'ютерних станцій, в яких наявні компоненти системи може відбуватись в один і той же час або в різний. Комп'ютерні станції можуть і не вимикатись, тобто бути увімкненими постійно. Ці випадки будуть впливати на формування системи S. Задамо їх в системі так, щоб її центр прийняття рішень міг враховувати ці випадки та варіації до них в процесі формування системи та функціонування, причому як активну останню подію. Варіанти формування системи задамо множиною $M_S^{var,1} = \left\{m_{S,1}^{var,1}, m_{S,2}^{var,1}, \dots, m_{S,n_{M_S^{var,1}}}^{var,1}\right\}$, де $n_{M_S^{var,1}}$ – кількість варіантів. Наприклад, такі елементи: $m_{S,1}^{var,1}$ – характеризує формування системи на її початку та активації; $m_{S,2}^{var,1}$ – характеризує формування системи в процесі функціонування за потреби; $m_{S,3}^{var,1}$ – характеризує формування системи після увімкнення комп'ютерних станцій в мережі. Із варіантів, які задано множиною $M_S^{var,1}$, в поточний момент часу може бути тільки один. Тобто, система S буде аналізувати останній варіант свого формування. Для визначення останнього варіанту формування системи введемо предикат на елементах множини $M_S^{var,1}$ так:

$$P_S^{var,1}\left(m_{S,q}^{var,1}\right) = \begin{cases} 0, m_{S,q}^{var,1} - \text{не поточний варіант} \\ q, m_{S,q}^{var,1} - \text{поточний варіант} \end{cases};\ q = 1,2, \dots, n_{M_S^{var,1}}. \quad (3.8)$$

Аналогічно, введемо множину варіацій множиною $M_S^{var,2} = \left\{m_{S,1}^{var,2}, m_{S,2}^{var,2}, \dots, m_{S,n_{M_S^{var,2}}}^{var,2}\right\}$, де $n_{M_S^{var,2}}$ – кількість варіацій. Наприклад, такі елементи: $m_{S,1}^{var,2}$ – доповнення системи новими компонентами; $m_{S,2}^{var,2}$ – вилучення компонентів з системи. Для варіацій заданих множиною $M_S^{var,2}$ випадків введемо предикат, значення якого буде відображати їх наявність чи відсутність, так;

$$P_S^{var,2}\left(m_{S,q}^{var,2}\right)=\begin{cases}0, m_{S,q}^{var,2}-\text{не поточна варіація}\\ q, m_{S,q}^{var,2}-\text{поточна варіація}\end{cases};\ q=1,2,\dots,n_{M_S^{var,2}}. \quad (3.9)$$

Аналогічно, введемо множину варіацій множиною $M_S^{var,3}=\left\{m_{S,1}^{var,3}, m_{S,2}^{var,3},\dots,m_{S,n_{M_S^{var,3}}}^{var,3}\right\}$, де $n_{M_S^{var,3}}$ – кількість варіацій. Наприклад, такі елементи: $m_{S,1}^{var,3}$ – комп'ютерні станції, в яких наявні компоненти системи, увімкнені в один і той же час; $m_{S,2}^{var,3}$ – комп'ютерні станції, в яких наявні компоненти системи, увімкнені в різний час; $m_{S,3}^{var,3}$ – комп'ютерні станції, в яких наявні компоненти системи, не вимикаються на увесь час функціонування системи. Для варіацій заданих множиною $M_S^{var,3}$ випадків введемо предикат, значення якого буде відображати їх наявність чи відсутність, так;

$$P_S^{var,3}\left(m_{S,q}^{var,3}\right)=\begin{cases}0, m_{S,q}^{var,3}-\text{не поточна варіація}\\ q, m_{S,q}^{var,3}-\text{поточна варіація}\end{cases};\ q=1,2,\dots,n_{M_S^{var,3}}. \quad (3.10)$$

Аналогічно, введемо множину варіацій множиною $M_S^{var,4}=\left\{m_{S,1}^{var,4}, m_{S,2}^{var,4},\dots,m_{S,n_{M_S^{var,4}}}^{var,4}\right\}$, де $n_{M_S^{var,4}}$ – кількість варіацій. Наприклад, такі елементи: $m_{S,1}^{var,4}$ – частина комп'ютерних станцій, в які встановлені компоненти, може бути вимкненою тривалий час, система міститиме меншу частину компонент і в цей час може відбутись поточне її формування викликане певними подіями без цих компонент; $m_{S,2}^{var,4}$ – нового формування системи не відбулось без компонент, які знаходились би у вимкнених комп'ютерних станціях. Для варіацій заданих множиною $M_S^{var,4}$ випадків введемо предикат, значення якого буде відображати їх наявність чи відсутність, так;

$$P_S^{var,4}\left(m_{S,q}^{var,4}\right)=\begin{cases}0, m_{S,q}^{var,4}-\text{не поточна варіація}\\ q, m_{S,q}^{var,4}-\text{поточна варіація}\end{cases};\ q=1,2,\dots,n_{M_S^{var,4}}. \quad (3.11)$$

Формули (3.8)-(3.11) описують етап, на якому здійснюється формування системи S, та задають його варіант та варіації. Результати обчислення предикатів формуватимуть частину вхідних даних для центру прийняття рішень системи. Після встановлення всіх компонент системи в комп'ютерні станції в мережі з врахуванням компонент з центром прийняття рішень і без нього, при першому запуску системи

компоненти з центром прийняття рішень перевіряють значення предикату для різних елементів множини $M_S^{var,1}$ та встановлюють, що всі значення дорівнюють нулеві. Тоді, система самостійно без користувача чи адміністратора розпочне первинне формування своїх компонент з наявних функцій-підмножин, а після завершення такого формування перейде до поділу компонент з центром прийняття рішень на активні і неактивні.

Для забезпечення комунікації між компонентами в системі S будемо здійснювати організацію зв'язку між компонентами не тільки з використанням стандартного надсилання повідомлень з відповідною кількістю повідомлень-підтверджень, але й з послідовним додаванням до них певних завдань, результат виконання яких відомий в компонентах, які планують надіслати основне повідомлення чи завдання, а також проведення аналізу часу, який витрачено від відправлення першого запиту на з'єднання і отримання результатів виконання тестового завдання. Взагалі вся система S буде реагуватиме цілісно на зміни в роботі її частин, в тому числі і комунікації між компонентами. Якщо всі компоненти вимикаються одночасно, тоді вони могли б фіксувати в себе певне завдання, виконання якого після наступного увімкнення вони мали б виконувати. Вимкнення комп'ютерних станцій може бути коректним і тоді така дія з фіксування однакового контрольного завдання для його використання в якості підтвердження легітимності зв'язку з відповідною компонентою могла б бути реалізована та зафіксована статично. Але може бути так, що комп'ютерна станція вимкнеться аварійно, тоді такої фіксації певного контрольного завдання не відбудеться. Таким чином, введення надмірності в організацію зв'язку між компонентами потребує врахування варіантів з увімкненими та неввімкненими комп'ютерними станціями, синхронізацією часу протягом якого компоненти є активними і встановлюють зв'язок між собою. Тому, введемо множину варіантів $M_S^{var,5} = \left\{m_{S,1}^{var,5}, m_{S,2}^{var,5}, \dots, m_{S,n_{M_S^{var,5}}}^{var,5}\right\}$, де $n_{M_S^{var,5}}$ – кількість варіантів, які виникають при введенні надмірності для організації зв'язку між компонентами. Елементи множини такі: $m_{S,1}^{var,5}$ – комп'ютерні станції, в яких наявні компоненти системи, увімкнені в один і той же час; $m_{S,2}^{var,5}$ – комп'ютерні станції, в яких наявні компоненти системи, вмикаються в різний час, причому частина може бути після певного часу

функціонування вимкнена, а певна частина після цього часу увімкнена або не вмикатись взагалі протягом тривалого певного часу. Відповідно, компоненти системи S теж бути активними тільки коли увімкнені та функціонують комп'ютерні станції. Введемо, також, множину варіантів $M_S^{var,6} = \left\{m_{S,1}^{var,6}, m_{S,2}^{var,6}, \dots, m_{S,n_{M_S^{var,6}}}^{var,6}\right\}$, де $n_{M_S^{var,6}}$ – кількість варіантів, які виникають при завершенні роботи комп'ютерних станцій, в які встановлені компоненти системи. Елементи множини такі: $m_{S,1}^{var,6}$ – комп'ютерні станції, в яких наявні компоненти системи, вимкнені коректно в один і той же час; $m_{S,2}^{var,6}$ – комп'ютерні станції, в яких наявні компоненти системи, вимкнені аварійно в один і той же час; $m_{S,3}^{var,6}$ – комп'ютерні станції, в яких наявні компоненти системи, вимкнені в різний час коректно; $m_{S,4}^{var,6}$ – комп'ютерні станції, в яких наявні компоненти системи, вимкнені в різний час частково коректно і частково аварійно. Згідно так заданих множин можна сформувати двохелементні множини, які характеризуватимуть події щодо комунікації в системі в залежності від комп'ютерних станцій так: $\{m_{S,1}^{var,5}; m_{S,1}^{var,6}\}$; $\{m_{S,1}^{var,5}; m_{S,2}^{var,6}\}$; $\{m_{S,1}^{var,5}; m_{S,3}^{var,6}\}$; $\{m_{S,1}^{var,5}; m_{S,4}^{var,6}\}$; $\{m_{S,2}^{var,5}; m_{S,1}^{var,6}\}$; $\{m_{S,2}^{var,5}; m_{S,2}^{var,6}\}$; $\{m_{S,2}^{var,5}; m_{S,3}^{var,6}\}$; $\{m_{S,2}^{var,5}; m_{S,4}^{var,6}\}$.

Для окремих комп'ютерних станцій потрібно розробити аналогічні задання множинами, оскільки згідно них буде забезпечуватись комунікація між окремими компонентами в системі S. Взагалі, в системі S встановлення зв'язку між компонентами та надсилання повідомлень буде здійснено згідно таких відношень: «один до всіх» ($m_{S,1}^{var,7}$); «всі до одного» ($m_{S,2}^{var,7}$); «один до одного» ($m_{S,3}^{var,7}$); «один до певної кількості, але не до всіх» ($m_{S,4}^{var,7}$); «певна кількість, але не всі, до одного» ($m_{S,5}^{var,7}$); «певна кількість, але не всі, до певної кількості, але не до всіх» ($m_{S,6}^{var,7}$). Задамо ці відношення множиною $M_S^{var,7} = \left\{m_{S,1}^{var,7}, m_{S,2}^{var,7}, \dots, m_{S,n_{M_S^{var,7}}}^{var,7}\right\}$, де $n_{M_S^{var,7}}$ – кількість і $n_{M_S^{var,7}} = 6$.

Для задання зв'язку окремих комп'ютерних станцій між собою введемо множину варіантів $M_S^{var,8} = \left\{m_{S,1}^{var,8}, m_{S,2}^{var,8}, \dots, m_{S,n_{M_S^{var,8}}}^{var,8}\right\}$, де $n_{M_S^{var,8}}$ – кількість варіантів, які виникають в процесі встановлення зв'язку між комп'ютерними станціями, в які встановлені компоненти системи. Елементи множини такі: $m_{S,1}^{var,8}$ –

комп'ютерна станція, в якій наявна компонента системи, увімкнена; $m_{S,2}^{var,8}$ – комп'ютерна станція, в якій наявна компонента системи, вимкнена коректно; $m_{S,3}^{var,8}$ – комп'ютерна станція, в якій наявна компонента системи, вимкнена аварійно. Згідно так заданої множини сформуємо двохелементні підмножини, які характеризуватимуть стан комп'ютерних станцій щодо їх початку та закінчення роботи так: $\{m_{S,1}^{var,8}; m_{S,2}^{var,8}\}$; $\{m_{S,1}^{var,8}; m_{S,3}^{var,8}\}$. Отже, якщо стан комп'ютерної станції, в якій наявна компонента системи S, характеризується підмножиною $\{m_{S,1}^{var,8}; m_{S,2}^{var,8}\}$, тоді повідомлення, які вона отримує та надсилає будуть вважатись центром прийняття рішень такими, що виконуються коректно. Інакше, тобто для підмножини $\{m_{S,1}^{var,8}; m_{S,3}^{var,8}\}$, центр прийняття рішень фіксує таку подію і при наступному увімкнені комп'ютерної станції опрацьовує з цією компонентою додаткову спеціальну процедуру встановлення зв'язку для поновлення цієї компоненти в системі. Крім того, при виконанні стандартної дії зв'язку між будь-якими двома компонентами системи, незалежно від типу елементу множини $M_S^{var,7}$, виконання додаткової перевірки є обов'язковим і полягає у виконанні певного завдання компоненти, яка планує встановити зв'язок і так само певного завдання від компоненти, з якою планується встановлення зв'язку.

Таким чином, встановлення зв'язку між компонентами системи в різних вузлах в мережі буде здійснено з врахуванням типів відношень, які дають змогу синтезувати часткову централізацію, та додаткової перевірки легітимності компоненти.

Корпоративна мережа підприємства може мати декілька сегментів. Компоненти системи S можуть бути встановлені в різних частинах мережі, а також віддалено в домашніх комп'ютерних станціях. В межах корпоративної мережі можуть виходити з ладу комутатори або можуть бути інші причини, які будуть спричиняти поділ системи S на дві і більше не пов'язаних підсистем. Тобто, система S в процесі функціонування може розпадатись на незв'язні частини. Тоді, кожна з частин здійснить переформування себе в зменшену систему S і буде продовжувати роботу, якщо в кожній з частин залишаться не менше двох активних компонент з центром прийняття рішень. Якщо в одній з частин відсутні активні компоненти з центром прийняття рішень та неактивні, тоді компоненти такої частини блокують

роботу комп'ютерних станцій і видають відповідне повідомлення для адміністратора. Якщо компоненти з центром прийняття рішень неактивні в момент певного аварійного або навмисного відокремлення другої частини, в якій будуть всі активні компоненти з центром прийняття рішень системи, тоді переведення їх до активного стану відбудеться після чергового сеансу зв'язку і встановлення факту відсутності активних компонент з центром прийняття рішень. Підтримка цілісності системи S в процесі її функціонування буде забезпечена процедурою періодичного обміну повідомленнями між компонентами системи згідно відношень з множини $M_S^{var,7}$, які будуть вибиратись випадковим чином. Крім цих двох випадків, які характеризують забезпечення цілісності системи S, є також випадок, який пов'язаний з синтезом в системі S часткової централізації. Якщо частина активних компонент, які містять центр прийняття рішень системи, будуть з певних причин вилучені з системи, тоді та частина, яка залишилась, розпочне процедуру формування системи з наявних компонент. Але якщо таких компонент буде менше двох, тоді всі активні компоненти, в тому числі і без функціоналу з центром прийняття рішень заблокують роботу комп'ютерних станцій і будуть видавати відповідне повідомлення адміністратору системи. Таким чином, так задана організація підтримки цілісності системи S враховує можливість синтезу в системі часткової централізації та адаптивності.

Часткову централізацію системи S задано в спроєктованій її архітектурі, зокрема формулою (2.4). Система є централізованою частково, бо всі її компоненти поділені на дві підмножини: підмножину компонент, в яких може бути центр системи; підмножину компонент, в яких відсутні функції для забезпечення функціонування центру прийняття рішень системи. Керування всією системою S відбувається з компонент, в яких знаходиться центр прийняття рішень системи. Тому, вона централізована. Часткова централізація забезпечується тим, що компоненти системи S, в яких знаходиться центр прийняття рішень системи для прийняття рішень формують пропозиції окремо в кожній з цих компонент, тобто децентралізовано, і погоджують його сумісно усі. Таким чином, система не повністю централізована.

Часткову централізацію розглядатимемо щодо компонент, в яких може міститись центр прийняття рішень системи S. Більшість з встановлених компонентів

системи в комп'ютерні станції повинна містити функціонал, який забезпечує функціонування центру прийняття рішень системи. Після завершення встановлення системи здійснюється перший запуск системи зі всіма увімкненими комп'ютерними станціями, в які встановлені компоненти системи. Всі компоненти, в яких може бути центр прийняття рішень системи, на цьому етапі функціонування системи будуть приймати участь у підготовці першого фінального рішення для визначення першого кроку системи. Це рішення буде стосуватись зменшення кількості активних компонент центру прийняття рішень переведенням частини з них до неактивного стану. Задамо множину станів, в які може переходити система S, $M_S^{st} = \left\{m_{S,1}^{st}, m_{S,2}^{st}, \dots, m_{S,n_{M_S^{st}}}^{st}\right\}$, де $n_{M_S^{st}}$ – кількість станів. Тоді, $m_{S,1}^{st}$ – стан системи, в якому оновлено активні компоненти центру прийняття рішень. Рішення про перехід до цього стану визначають активні компоненти центру прийняття рішень системи. Здійснення керування системою визначено за центром прийняття рішень системи. В ньому будуть сформовані рішення та передані вказівки до компонент для їх виконання. Формування рішення в системі буде здійснено в активних компонентах центру прийняття рішень. Якщо розглядати їх сукупно з кількістю більше однієї, тоді на архітектурному рівні вони можуть позиціонуватись як децентралізована підсистема. Тому, формування фінального рішення буде здійснено згідно рішень, які будуть отримані з активних компонент, та їх опрацювання. Завершення процесу опрацювання фінального рішення буде переходом системи до одного із станів. Передавання рішення від активних компонент центру прийняття рішень системи до визначених компонент буде здійснено згідно відношення, яке задано одним з елементів $m_{S,5}^{var,7}$ або $m_{S,6}^{var,7}$, та переведе систему до наступного стану. Таким чином, основними кроками з синтезу часткової централізації в системі є забезпечення реалізації формування компонент, в яких буде функціонувати центр прийняття рішень, формування рішень в компонентах та фінального рішення, а також опрацювання фінального рішення в частині його надсилання в компоненти, в яких воно має бути виконано. Для формування рішень відповідні компоненти отримують певні повідомлення чи результати в процесі функціонування системи.

Елементами множини станів M_S^{st} будуть задаватись поточні стани системи в цілому та її компонентів. Переходи із стану до стану задамо впорядкованими парами

$(m_{S,p}^{st}, m_{S,q}^{st})$, де p – номер поточного стану системи, q – номер наступного стану системи. Обидва стани системи обов'язково стосуватимуться і її компонент. Зокрема, перехід з поточного стану в наступний стан може охоплювати не всі компоненти системи в частині виконання певних дій для досягнення повного переходу всієї системи до наступного стану. Таким чином, задані стани множиною M_S^{st} будуть характеризувати систему в цілому, а для компонент системи вони будуть за переліком елементів множини M_S^{st} такі самі, але задаватимуть окремо стан окремих компонентів, бо компоненти можуть не перебувати одночасно в одному стані. Окремі компоненти системи можуть змінювати свій стан згідно елементів множини M_S^{st} частіше від системи. При переході системи зі стану до стану в процес може долучатись певна частина компонент, в результаті їх стани теж можуть змінюватись. Допускається, що функціонування системи можливе за наявності не менше двох компонент, в яких може бути центр функціонування системи. Таким чином, масштабування системи через її стани для мінімальної кількості компонент є допустимим. Перехід із стану до стану буде забезпечуватись певним набором функцій. За один такт система може змінити декілька станів, якщо прийме про це рішення. Система може формувати нові стани комбінацією відомих для неї станів. В певному стані система отримує поточні та вхідні данні, для опрацювання яких будуть залучатись відповідні функції. В результаті формується поле подій для опрацювання, яке задамо множиною подій $M_S^{pd} = \left\{m_{S,1}^{pd}, m_{S,2}^{pd}, \dots, m_{S,n_{M_S^{st}}}^{pd}\right\}$, де $n_{M_S^{pd}}$ – кількість подій.

При зміні кількості увімкнених комп'ютерних станцій змінюється кількість компонент в системі, зокрема і тих, в яких може бути центр прийняття рішень системи. Також, протягом певного часу функціонування системи можуть виникнути події, які вимагатимуть зміни стану системи щодо частини компонент, в яких може бути центр прийняття рішень системи. Тому, міграція центру прийняття рішень системи між певними компонентами повинна бути задана певними відповідними функціями для можливості її реалізації самою системою.

Враховуючи цільове спрямування системи S щодо виявлення ЗПЗ потрібно визначати, крім поточного стану компонент і системи, також стан щодо безпеки комп'ютерних станцій, в яких встановлені компоненти та безпосередньо їх власний

стан безпеки. Таким чином, для забезпечення належного функціонування системи S та прийняття рішень щодо її подальшого функціонування потребують врахування такі стани: стан системи; стани компонентів; стани комп'ютерних станцій. Значення цих станів визначатимуться не тільки щодо їх безпеки відносно впливів ЗПЗ, але і щодо загального завантаження ресурсів комп'ютерних станцій та навантаження виконуваними завданнями в компоненті. Інтегруємо загальні стани компонентів та комп'ютерних станцій в один показник стану компоненти системи згідно формули (2.59), згідно якої для кожної компоненти обчислюємо значення $\alpha'_{3,S_{1,n}}$. Стан системи в цілому визначимо згідно станів її компонент з врахуванням значень $\alpha'_{3,S_{1,n}}$ для всіх компонент, які активні в поточний момент в системі, так:

$$\alpha_{S,t}^{st,1} = \frac{1}{p} \cdot \sum_{q=1}^{p} \alpha'_{3,S_{1,n},q}, \tag{3.12}$$

де p – кількість активних компонент системи у ввімкнених комп'ютерних станціях; $p = 1,2, \dots, n$, n – кількість компонент в системі S; $\alpha'_{3,S_{1,n},q}$ - значення $\alpha'_{3,S_{1,n}}$ в q – ій компоненті.

В компонентах системи S будуть здійснені обчислення, результати яких передаватимуться в активні компоненти з функціонуючим центром прийняття рішень системи. В активних компонентах центру прийняття рішень, також, можуть виконуватись певні завдання і отримуватись їх результат. За певних обставин не в усіх компонентах результат виконання поставленого завдання може бути отримано і передано в задані часові інтервали. Також, в певних компонентах можуть бути сформовані результати виконання поставленого завдання відмінні від результатів, що отримані від більшості компонент із залучених до його виконання. Не всі результати виконання поставленого завдання будуть мати чіткі очікувані числові значення. Тому, для формування фінального результату і його використання при визначенні подальших кроків системи в компонентах центру прийняття рішень потрібно поділити компоненти на два класи, з яких отримано результати виконання поставленого завдання. Всі завдання, які можуть виконуватись системою S, поділені за функціями-підмножинами, що можуть їх виконувати, та типами компонент, в яких вони можуть виконуватись. Стани компонент будуть постійно змінюватись. Вони не будуть мати статичні числові значення. Значення характеристичних показників компонент системи в залежності від типів виконуваних завдань

визначені за формулами (2.18), (2.50) і (2.59). Для обчислення кожного із значень α'_{1,S_i}, $\alpha'_{2,S_{k+1,n}}$, $\alpha'_{3,S_{1,n}}$ для окремих компонент системи застосовуються функції з п'ятьма аргументами $f_{\alpha'_{1,S_i}}$, $f_{\alpha'_{2,S_{k+1,n}}}$, $f_{\alpha'_{3,S_{1,n}}}$ відповідно. Для типів завдань буде обчислюватись тільки одне із трьох значень. Але його значення буде отримуватись згідно п'яти аргументів відповідної функції. Розглянемо варіанти визначення функцій $f_{\alpha'_{1,S_i}}, f_{\alpha'_{2,S_{k+1,n}}}, f_{\alpha'_{3,S_{1,n}}}$.

Перший варіант, який можна використати для визначення значень функцій $f_{\alpha'_{1,S_i}}$, $f_{\alpha'_{2,S_{k+1,n}}}$, $f_{\alpha'_{3,S_{1,n}}}$, задамо через середньоарифметичне значення усіх п'яти аргументів так:

$$\alpha'_{1,S_i} = f_{\alpha'_{1,S_i}}\left(\alpha'_{1,S_i,1}, \alpha'_{1,S_i,2}, \alpha'_{1,S_i,3}, \alpha'_{1,S_i,4}, \alpha'_{1,S_i,5}\right) = \frac{1}{5} \cdot \sum_{q=1}^{5} \alpha'_{1,S_i,q}, \quad (3.13)$$

$$\alpha'_{2,S_{k+1,n}} = f_{\alpha'_{2,S_{k+1,n}}}\left(\alpha'_{2,S_{k+1,n},1}, \alpha'_{2,S_{k+1,n},2}, \alpha'_{2,S_{k+1,n},3}, \alpha'_{2,S_{k+1,n},4}, \alpha'_{2,S_{k+1,n},5}\right) = \frac{1}{5} \cdot \sum_{q=1}^{5} \alpha'_{2,S_{k+1,n},q}, \quad (3.14)$$

$$\alpha'_{3,S_{1,n}} = f_{\alpha'_{3,S_{1,n}}}\left(\alpha'_{3,S_{1,n},1}, \alpha'_{3,S_{1,n},2}, \alpha'_{3,S_{1,n},3}, \alpha'_{3,S_{1,n},4}, \alpha'_{3,S_{1,n},5}\right) = \frac{1}{5} \cdot \sum_{q=1}^{5} \alpha'_{3,S_{1,n},q}. \quad (3.15)$$

Після отримання значень за формулами (3.13), (3.14), (3.15) для кожної з компонент, в яку надсилалось завдання для виконання, потрібно здійснити поділ цих значень на два класи. До першого класу увійдуть ті значення, які будуть дорівнювати або знаходитись найближче на числовій осі до одиниці, а до другого класу решту. Тоді, результати виконання завдання, які отримані від компонент із значеннями з першого класу, будуть прийняті з відповідним ступенем довіри. Якщо вони будуть числовими, то серед них буде обчислено середньоарифметичне значення як фінальний результат. Якщо значення виконаного завдання будуть не числові, тоді результат виконання буде прийнято виконаним, якщо перший клас не буде порожнім. Якщо перший клас буде порожнім, тоді виконання завдання здійснюється повторно. Для формування двох класів сформуємо проміжок для значень (α'_{1,S_i}, $\alpha'_{2,S_{k+1,n}}$, $\alpha'_{3,S_{1,n}}$) з компонент так, щоб мінімальне з них було нижньою межею проміжку, а верхньою межею проміжку було число один. Так сформований проміжок буде постійно змінюватись для кожного нового завдання, оскільки нижня межа буде змінюваною. Встановимо нижню межу для першого класу як 20%

відхилення від одиниці до нижньої межі, а для другого класу, відповідно, як 80% відхилення від нижньої межі проміжку. Спільне значення двох класів віднесемо до першого класу, тоді значення другого класу будуть знаходитись в проміжку з відкритою верхньою границею. Задамо проміжок з класами так:

$$\begin{cases} \alpha'_{1,S} = min\left(\alpha'_{1,S_1}, \alpha'_{1,S_2}, \dots, \alpha'_{1,S_p}\right);\ p \le i; \\ \alpha'_{1,S} = min\left(\alpha'_{2,S_{k+1}}, \alpha'_{2,S_{k+2}}, \dots, \alpha'_{2,S_p}\right);\ p \le n; \\ \alpha'_{1,S} = min\left(\alpha'_{3,S_1}, \alpha'_{3,S_2}, \dots, \alpha'_{3,S_p}\right);\ 1 \le p \le n; \\ \left[\alpha'_{1,S}; 1\right] - \text{проміжок всіх значень;} \\ \alpha'_{2,S} = 1 - 0{,}2 \cdot \left(1 - \alpha'_{1,S}\right) - \text{граничне значення обох класів;} \\ \left[\alpha'_{2,S}; 1\right] - \text{проміжок для значень з першого класу;} \\ \left[\alpha'_{1,S}; \alpha'_{2,S}\right[- \text{проміжок для значень з другого класу.} \end{cases} \quad (3.16)$$

Таким чином, здійснена за формулою (3.16) кластеризація значень дає змогу прийняти центру прийняття рішень системи результати виконання поставленого завдання в заданих компонентах.

Але при оцінюванні результатів розподілених обчислень в компонентах за першим варіантом вага значення певного характеристичного показника нівелюється та може впливати на віднесення до певного класу. Це відбувається через те, що за формулами (3.13)-(3.15) всі доданки розглядаються як рівнозначні, без врахування їх ваги. Врахування ж ваг їх в загальному результуючому значенні ускладнено, бо ці ваги не мають встановлених значень і потребують залучення експертів для їх визначення, що буде впливати на зменшення самоорганізації системи та можливу їх точність, в зв'язку з постійними змінами станів в комп'ютерних станціях. Тому, розглянемо другий варіант для визначення значень функцій $f_{\alpha'_{1,S_i}}$, $f_{\alpha'_{2,S_{k+1,n}}}$, $f_{\alpha'_{3,S_{1,n}}}$.

Для здійснення кластеризації на два класи за другим варіантом розглядатимемо п'ятивимірний простір, в якому п'ять аргументів функцій $f_{\alpha'_{1,S_i}}$, $f_{\alpha'_{2,S_{k+1,n}}}$, $f_{\alpha'_{3,S_{1,n}}}$ задаватимуть точки. Таким чином, при отриманні значень аргументів функцій з компонент системи, в яких виконувалось завдання, буде формування з них координат точок в п'ятивимірному просторі з подальшим поділом точок на два класи. Вибір алгоритму класифікації та метрики здійснимо, виходячи із того, що цінними значеннями для системи будуть ті, які будуть найближче

знаходитись до значення, що дорівнює одиниці. Відповідно, потрібен вибір алгоритму класифікації та метрики такий, щоб формувався перший клас, в якому був би елемент (1;1;1;1;1) або кластер формувався б за його відсутності, але з покриттям області точок найближчих до нього.

Розглянемо відомі метрики та здійснимо вибір метрики для застосування при класифікації. Метрика евклідової відстані задає геометричну відстань між об'єктами в просторі. Метрика квадрату евклідової відстані характеризується наданням більшої ваги найбільш віддаленим об'єктам. Метрика манхетенської відстані призводить до зменшення впливу окремих великих відстаней. Метрика степеневої відстані застосовується, коли треба збільшити або зменшити вагу для розмірності об'єктів, які суттєво відрізняються. Її недоліком є необхідність задавати два параметри. Метрика Чебишова використовується, якщо два об'єкти відрізняються хоча б однією координатою. З проаналізованих метрик виберемо метрику Чебишова, оскільки згідно неї можна розрізнити два об'єкти відмінних однією координатою, бо решта метрик при великій кількості різних числових координат може привести для певних однакових обчислень відстані, що для побудови другого варіанту кластеризації є недопустимим. Метрика Чебишова задає відстань так:

$$\rho(x, x') = \max_{q=1,2,\dots,5}\left(\left|x_q - x'_q\right|\right), \tag{3.17}$$

де x'_q - координата центру кластера; $q = 1,2, \dots ,5$; x_q – координата точки в просторі.

Для здійснення кластеризації використаємо метод k-середніх, оскільки згідно його результатів застосування всі об'єкти будуть поділені у порівняно однорідні класи. Досягнення поділу на класи забезпечується мінімізацією суми квадратів відстаней між кожним з п'яти значень характеристичних показників, тобто аргументів функцій $f_{\alpha'_{1,S_i}}$, $f_{\alpha'_{2,S_{k+1,n}}}$, $f_{\alpha'_{3,S_{1,n}}}$ та центром кластеру, яку задамо так:

$$d_v = \sum_{i=p_1}^{p_2}\left(\max_{q=1,2,\dots,5}\left(\left|\alpha'_{w,S_i,1,q} - x'_q\right|\right)\right)^2, \tag{3.18}$$

де x'_q - координата центру кластера; $q = 1,2, \dots ,5$; для функції $f_{\alpha'_{1,S_i}}$ значення $p_1 = 1$, p_2 – кількість активних компонент з центром прийняття рішень системи; для функції $f_{\alpha'_{2,S_{k+1,n}}}$ значення $p_1 = k + 1$, p_2 – кількість активних компонент без функціоналу

для центру прийняття рішень системи і $p_2 \leq n$; для функції $f_{\alpha'_{3,S_{1,n}}}$ значення $p_1 = 1$, p_2 – кількість активних компонент і $p_2 \leq n$.

На певному кроці ітерації за центр кластера буде обиратись значення елементу, яке задано п’ятьма координатами. Встановлюємо кількісно два кластери для поділу значень. Фіксуємо отримані значення з компонент, в яких виконувалось поставлене завдання, в усіх активних компонентах, які формують центр прийняття рішень системи. Приймемо за центр першого кластеру значення задане координатами (1;1;1;1;1), а за центр другого кластеру – значення характеристичного показника, яке найбільш віддалене від точки з координатами (1;1;1;1;1). Якщо таких значень декілька, тоді приймаємо за центр кластеру останнє розглядуване значення, яке підходить. Решту значень розподіляємо між двома класами згідно формули (3.18) в залежно від відстані до двох центрів двох кластерів таким чином, щоб до класу включалось значення, до якого найменша відстань згідно метрики Чебишова (формула (3.17)). Для досягнення стійкості кластерів, тобто досягнення віднесення до кластерів одних і тих же значень, потребують уточнення центри кластерів через проведення повторних ітераційних обчислень. Для вибору наступного центру кластеру будемо знаходити середньоарифметичне значення зі всіх значень характеристичних показників, які входять до певного кластеру. Пошук таких центрів здійснюється поки не будуть залишатись в кластерах ті самі значення, які були на попередньому кроці ітерації при іншому центрі кластеру. В результаті буде досягнуто того, що дисперсія між класами буде максимізована, а між елементами – мінімізована. Для уточнення центру кластеру в активних компонентах центру прийняття рішень потрібно здійснити організацію ітераційних кроків. В подальшому на наступних кроках виконання завдань ці кластери потрібні будуть при наступних етапах виконання такого ж завдання для оцінювання розбіжності. Також, при наявності попередніх історій з виконання такого ж поставленого завдання центр прийняття рішень системи буде здійснювати усереднення значень межі класів за результатами попередніх обчислень для уникнення деградації системи, коректності оцінювання виконання завдання в системі та фіксування результату виконаного завдання. Таким чином, класифікацією згідно методу k-середніх здійснюється поділ на два класи за другим варіантом значень характеристичних показників активних компонент та однозначно визначено функції

$f_{\alpha'_{1,S_i}}$, $f_{\alpha'_{2,S_{k+1,n}}}$, $f_{\alpha'_{3,S_{1,n}}}$ як такі, що реалізують обчислення цих значень згідно метрики Чебишова.

Для визначення компонент, в яких буде виконуватись поставлене системою завдання, центр прийняття рішень системи визначає через опитування всіх компонент про їх рівень їх безпеки, який обчислюється за формулою (3.12). Потім, центром прийняття рішень за першим варіантом поділу на класи (формули (3.13)-(3.16)) визначається половина компонент, тобто в формулу (3,16) ставиться коефіцієнт 0,5 замість 0,2, в яких буде виконуватись поставлене завдання. У випадку якщо поставлене завдання вимагає негайного виконання або має статус такого, яке пов’язане з дослідженням безпеки в компонентах, то його виконують всі компоненти. До виконання поставленого завдання може залучатись менша кількість компонент, якщо їх багато, але ця кількість не може бути менше десяти компонент. Це пов’язано з необхідністю мати достатню вибірку значень, щоб визначити коректно фінальний результат. При цьому певна частина може не встигнути виконати його в поставлений час. Якщо кількість компонент в системі невелика, наприклад менше десяти, тоді всі компоненти залучаються до виконання поставленого завдання. Кожна з функцій-множин та функцій-підмножин в компонентах та системі в цілому мають пріоритети, які впливають на кількість залучених компонент для виконання поставлених завдань. Ці функції в компонентах мають чіткі зв’язки із завданнями, для яких вони призначені.

Формування центру прийняття рішень може бути здійснено кількісно від двох до всіх компонент, в яких встановлено відповідний функціонал. Рішення про кількість активних компонент центру прийняття рішень приймають з моменту старту системи всі компоненти, в яких наявний центр прийняття рішень системи. Якщо в процесі функціонування системи активні компоненти з центром прийняття рішень припиняють свою роботу, а система продовжує, тоді центр прийняття рішень долучає нові компоненти для підтримки кількості таких компонент. Для цього він переводить їх до активного стану. Рішення про кількість активних компонент приймається випадковим чином кожною активною компонентою, а потім знаходиться середньоарифметичне їх значення і проводиться відкидання його дробової частини.

В процесі функціонування системи накопичується інформація в її компонентах, в яких може бути центр прийняття рішень системи. Ця інформація необхідна для врахування при прийнятті наступних рішень щодо подальших кроків. Але не всі компоненти будуть мати однакову інформацію про пройдені стани системи, тому повинні бути введені в них механізми та функції, які дадуть змогу її актуалізувати до певного рівня. До такої інформації, яка потребує збереження для використання при визначенні подальших кроків системи та яка відноситься винятково для забезпечення функціонування системи, віднесемо: інформацію про кількість компонент в системі за час від початку її функціонування та стани їх активності чи не активності; інформація про всі виконувані в системі завдання та залучені для цього компоненти, а також рішення які були прийняті і первинні результати для їх прийняття. Для оновлення актуальної інформації в усіх компонентах центру прийняття потрібно виконувати завдання, в результаті виконання якого буде оновлено базу інформації за останні події в системі та розіслано всім компонентам центру прийняття рішень, які знаходяться в увімкнених комп'ютерних станціях. Для компонент центру прийняття рішень, які знаходяться в неувімкнених комп'ютерних станціях, оновлення такої інформації відбудеться при їх наступному увімкненні. Збереження такої інформації дасть змогу адміністратору системи проаналізувати та знайти причину зупинки системи чи комп'ютерних станцій її компонентами, приймати системою рішення щодо подальших кроків, здійснювати оптимізацію при виконанні поставлених завдань.

Перебудова архітектури системи може бути, також, необхідною у випадку виявлення аномальних подій або зловмисних проявів в комп'ютерній мережі чи станціях. В такому випадку частина компонент системи може вимкнути комп'ютерні станції і повідомити до центру прийняття рішень системи про вилучення з системи. Такі події можуть бути виявлені частково компонентами з наявним функціоналом через встановлення зв'язку між компонентами на початку роботи після увімкнення комп'ютерних станцій або при встановлені проблем із функціонуванням компоненти в певній комп'ютерній станції. Такого типу події обробляються відповідними функціями-підмножинами і центр прийняття рішень здійснює керівні дії для перебудови архітектури системи згідно елементу з множини подій M_S^{pd}.

Визначення подальших кроків системи та перехід в них в поточний момент часу залежить від подій в системі, які задано множиною M_S^{pd}, результатів обробки подій функціями компонент системи, множини варіантів кроків, які задано множиною станів M_S^{st}, результатів роботи центру прийняття рішень системи та можливості виконання визначеного переходу до наступного стану в поточний момент часу, оскільки при проведенні підготовчих заходів в системі могли відбутись зміни, та безпосереднє виконання переходу з перевіркою його повного завершення.

Події в системі оброблятимуться певними функціями. Якщо події відбуваються в комп'ютерних станціях та в мережі, то система S може їх обробляти, якщо вони видимі для її сенсорів, а може і не обробляти. Система S повинна контролювати всі об'єкти та процеси, які в подальшому можуть в цілому оцінюватись як аномальні прояви або зловмисні впливи. Для цього вона повинна мати достатню кількість сенсорів та функцій для обробки результатів. Якщо ж їх буде недостатньо, то система S може не виявити, наприклад ЗПЗ в комп'ютерних мережах. Крім того, події можуть відбуватись і в самій системі безпосередньо. Вони можуть бути викликані як зовнішніми, так і внутрішніми впливами. Але при цьому розглядатимемо всі події, як такі, що відбуваються в комп'ютерних станціях і мережі та повинні оброблятись без поділу на типи. Поділ на типи таких впливів та проявів буде використано при розробці методів їх цілеспрямованого аналізу за типами для виявлення аномальних проявів та зловмисних впливів, які викликані типами відповідних засобів. Події, які задано елементами множини M_S^{pd}, систематизовані саме через характеристики елементів. Збільшення кількості елементів множини M_S^{pd} потребуватиме збільшення кількості функцій в компонентах системи. Події, також, можуть бути задані комбінаціями елементів. Крім того, події можуть одночасно відбуватись в різних вузлах в мережі і бути видимими для компонентів системи.

Результати обробки подій функціями компонент системи будуть використані для визначення подальших кроків системи її центром прийняття рішень і, як наслідок, призведуть до появи нових подій. Взагалі система буде здійснювати постійно моніторинг подій і постійно буде їх опрацьовувати. Але не всі події будуть призводити до зміни стану чи переходу до наступного стану.

Від множини варіантів кроків, які задано множиною станів M_S^{st}, залежать можливості системи S з виконання поставлених завдань, які будуть відноситись до організації її функціонування згідно принципів самоорганізації та адаптивності. Якщо станів системи S небагато, тобто елементів множини M_S^{st}, то пар елементів, якими будуть задаватись варіанти кроків, теж буде небагато. Це може дати можливість забезпечувати належну стійкість для системи S. Але наповнення компонент функціями-множинами для розв'язування спеціалізованих завдань, а також, середовище в комп'ютерних станціях буде швидко змінюваним, тому кількість елементів множини M_S^{st} не може бути невеликим числом. В зв'язку з цим, кількість станів може бути великим числом, а тому кількість варіантів кроків і їх комбінацій буде теж достатньо великим числом і, як наслідок, описати їх всіх однозначно неможливо. Для розв'язання цього завдання потрібно задати правила за якими система буде формувати та визначати кроки для подальшого переходу до наступного стану, правила вибору варіантів з декількох сформованих кроків. При цьому повинна бути початково сформована множина станів M_S^{st}, кількість елементів якої в подальшому повинна збільшуватись за рахунок формування в системі нових станів, як комбінацій базових станів. Такі комбінації формуватимуться згідно поєднання різних станів всіх компонент системи в єдиний стан всієї системи. Протягом часу свого функціонування кожна компонента буде змінювати свій стан. Таким чином, частина можливих комбінацій базових станів доповнить множину станів M_S^{st} новими елементами. І це здійснить центр прийняття рішень системи. Визначимо такі основні стани системи S, тобто елементи множини станів M_S^{st}: $m_{S,1}^{st}$ – стан системи, в якому оновлено активні компоненти центру прийняття рішень; $m_{S,2}^{st}$ – стан системи, в якому здійснено оцінювання стану компонент та системи; $m_{S,3}^{st}$ – стан системи, в якому здійснено комунікацію між компонентами системи; $m_{S,4}^{st}$ – стан системи, в якому здійснено визначення подальших кроків системи в поточний момент часу; $m_{S,5}^{st}$ – стан системи, в якому здійснено міграцію центру прийняття рішень системи; $m_{S,6}^{st}$ – стан системи, в якому здійснено перебудову архітектури системи; $m_{S,7}^{st}$ – стан системи, в якому здійснено формування центру прийняття рішень в декількох компонентах; $m_{S,8}^{st}$ – стан системи, в якому здійснено оцінювання результатів розподілених обчислень в компонентах; $m_{S,9}^{st}$ – стан

системи, в якому здійснено завершення функціонування компоненти та інші. Для компонент ці ж стани будуть також, але якщо, наприклад, система здійснила оновлення активних компонент центру прийняття рішень, то в компонентах стани можуть бути такі: вимкнено функціонал для активізації центру прийняття рішень системи в компоненті; активовано функціонал для активізації центру прийняття рішень системи в компоненті; стан компоненти не змінився, тобто відбувся перехід до того ж стану.

Переходи зі стану до стану системи S зображено на рис. 3.1. Наприклад, виділеним відрізком на рисунку зображено перехід з стану $m_{S,7}^{st}$ до стану $m_{S,3}^{st}$ чи навпаки в залежності від координати вектору переходу у впорядкованій парі ($m_{S,7}^{st}$; $m_{S,3}^{st}$) чи ($m_{S,3}^{st}$; $m_{S,7}^{st}$).

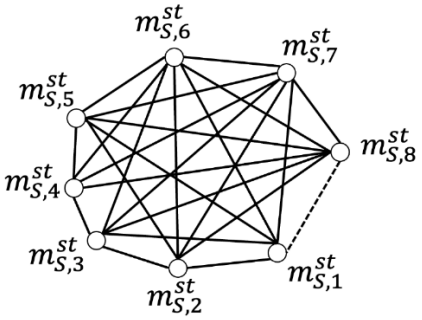


Рисунок 3.1 – Стани системи S та можливі варіанти переходів між ними

Таким чином, система S буде перебувати в одному із станів, які зображено на рис. 3.1. Деталізація впливів та засобів для зміни стану зображена на рис. 3.2 із вказанням зв'язків, якими може бути здійснено вплив.

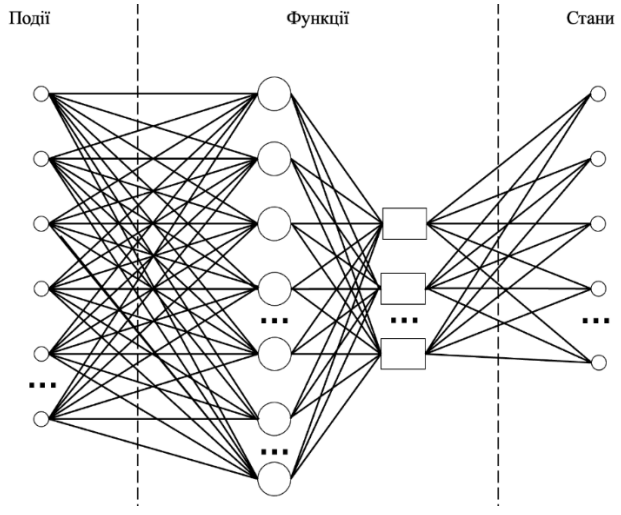


Рисунок 3.2 – Взаємозв'язок подій, функцій та станів

У зображеному зв’язку подій, функцій та станів виділено два типи функцій. До першого типу віднесено функції-множини в компонентах, які не відносяться до функцій центру прийняття рішень системи, а до другого – функції, які формують центр прийняття рішень системи. Виділені відрізки між двома типами функцій означають, що вони відносяться до компонент, в яких може бути центр прийняття рішень системи.

Зображені на рис. 3.1 та рис. 3.2 стани відносяться винятково до системи в цілому. Деталізація станів в конкретних компонентах аналогічна до зображень на рис. 3.1 та рис. 3.2. Компоненти системи можуть перебувати в різних станах одночасно, а стан системи однозначно визначатиметься станами її компонент та компонентами центру прийняття рішень. Конкретна компонента системи може бути в декількох станах одночасно, які розглядатимуться як певний стан, що сформований комбінацією базових станів. Наприклад, система S здійснює доповнення компонентами, які стали активними в результаті увімкнення комп’ютерних станцій, а компонента системи в цей час здійснює оцінювання рівня безпеки в комп’ютерній станції, тоді ці два стани в компоненті будуть поєднані в один в поточний момент часу і система зафіксує в себе стан цієї компоненти.

Для переходу зі стану до стану системи S будемо враховувати активність функцій-підмножин (аналіз матриці з формули (2.9)), значення характеристичних показників (формули (2.18), (2,50), (2.59)), варіанти формування системи згідно множини $M_S^{var,1}$ (формула (3.8)), варіації формування системи згідно множин $M_S^{var,2}$, $M_S^{var,3}$, $M_S^{var,4}$ (формули (3.9)-(3.11)), введення надмірності в організацію зв’язку згідно множин $M_S^{var,5}$ і $M_S^{var,6}$, типу відношення для встановлення зв’язку між компонентами та надсилання повідомлень згідно множини $M_S^{var,7}$, задання зв’язку окремих комп’ютерних станцій між собою згідно множин варіантів $M_S^{var,8}$, множину подій M_S^{pd}, стан системи в цілому (формула (3.12)), вибір варіантів обчислення довіри до результатів розподілених обчислень (формули (3.13)-(3.15) або згідно кластеризації (формула (3.18))) та множини станів M_S^{st}. Задамо наступний стан системи S через її поточний стан та показники компонент і системи так:

$$m_{S,p}^{st} = F_{q\to p}^{S}\begin{pmatrix} m_{S,q}^{st} \\ M_S^{st} \\ M_S^{pd} \\ M_{S,k,\psi} \\ \alpha'_{1,S_i} \\ \alpha'_{2,S_{k+1,n}} \\ \alpha'_{3,S_{1,n}} \\ M_S^{var,1} \\ M_S^{var,2} \\ M_S^{var,3} \\ M_S^{var,4} \\ M_S^{var,5} \\ M_S^{var,6} \\ M_S^{var,7} \\ M_S^{var,8} \\ f_{\alpha'_{1,S_i}} \\ f_{\alpha'_{2,S_{k+1,n}}} \\ f_{\alpha'_{3,S_{1,n}}} \\ \alpha_{S,t}^{st,1} \end{pmatrix}, \tag{3.19}$$

де $F_{q\to p}^{S}$ – функція, що визначає наступний стан системи S та задає перехід між станами.

При визначенні наступного стану системи варіантів буде стільки скільки є елементів в множині M_S^{st}. Результатом вибору може бути той самий стан, в якому система вже перебуває. Також, система може виявити стан, якого немає в множині станів. Це може відбутись у випадку, коли в певній компоненті чи декількох компонентах встановлено комбінацію декількох станів, які в множині M_S^{st} задані базовими утворюючими елементами. Тоді система доповнює цю множину станів новим елементом, що формується комбінацією певних елементів в певних компонентах чи компоненті. Але множина станів не формується в повному обсязі зі всіх комбінацій на початку, але тільки з тих які з'являться в процесі функціонування системи.

Оскільки аргументами функції $F_{q\to p}^{S}$ є данні різного типу і функцію повинні задавати правила, згідно яких буде визначатись дискретне значення, то задамо цю функцію $F_{q\to p}^{S}$ як загальне правило, яке міститиме поєднання логічними операторами «І» та «АБО» і запереченням «НЕ» у логічному виразі локальних функцій, які

віднесені до кожного аргументу. Нехай $F^{S}_{q\to p,b}$ – b-та локальна функція, де $b = 1,2,\dots,19$, аргументом якої є b – тий аргумент функції $F^{S}_{q\to p}$. Значеннями локальних функцій будуть дискретні значення $\{0\}$ та $\{1\}$, де значення $\{1\}$ означатиме виконання умов для переходу до наступного стану, а значення $\{0\}$ – не виконання таких умов. В логічному виразі, що формуватиме правило для визначення функції $F^{S}_{q\to p}$, значення локальних функцій можуть комбінуватись між собою повністю та частково, а також можуть формувати складені умови, з яких достатньо для переходу до нового стану виконання однієї з умов.

Наприклад, для переходу до стану $m^{st}_{S,1}$, в якому оновлено активні компоненти центру прийняття рішень, потрібно змінити компоненти, в яких буде центр прийняття рішень системи. Причинами зміни будуть оновлені данні таких показників та результати поточного стану показників системи. Якщо попередній стан $m^{st}_{S,2}$ системи, в якому здійснено оцінювання стану компонент та системи, і встановлено, що значення $\alpha^{st,1}_{S,t} = 0{,}23$, тобто суттєво менше порогового значення, тоді незалежно від решти показників система виконує функції необхідні для оновлення активних компонент центру прийняття рішень.

Так задані переходи зі стану до стану, також, будуть підтримувати цілісність системи і забезпечувати її стійкість. Формулою (3.19) задана система S на рівні станів, в яких вона може перебувати, та переходів між ними, що фактично задає процеси, які функціонуватимуть в ній.

Центр прийняття рішень системи згідно формули (3.19) отримує результат та встановлює можливість виконання визначеного переходу до наступного стану в поточний момент часу, оскільки при проведенні підготовчих заходів в системі могли відбутись зміни.

Виконання переходу між станами системи забезпечується функцією-підмножиною, яка здійснює перевірку його повного завершення згідно спеціально заданої в такому випадку комунікації. Якщо частина компонент системи не встигла завершити цей перехід, то в подальшому при їх активності вони відтворюють пропущені стани в своїй історії станів, поновлюють актуальні показники і переходять до поточного стану системи.

Завершення функціонування компонентів та системи може бути з подальшим їх поверненням до виконання завдань при увімкненні комп’ютерних станцій, або

при заданій команді блокування частини компонент чи системи, або вилучення компонент чи системи в цілому з вузлів в мережі.

3.2.2. Кроки методу організації функціонування частково централізованих розподілених систем згідно принципів самоорганізації та адаптивності

В результаті задамо основні загальні кроки методу організації функціонування частково централізованих розподілених систем згідно принципів самоорганізації та адаптивності.

Крок 1. Формування системи S з компонент.

1.1. Якщо система S формується після початкового встановлення всіх компонент (елемент $m_{S,1}^{var,1}$ характеристичної множини $M_S^{var,1}$, елемент $m_{S,1}^{var,3}$ множини $M_S^{var,3}$), тоді кожен з її компонент отримує інформацію про місце знаходження решти компонент в комп'ютерній мережі, фіксує таку інформацію в своїй внутрішній базі та очікує початкового запуску однієї з компонент адміністратором для подальших початкових запусків решти компонент після вказівки з неї про її початок функціонування.

1.2. Якщо комп'ютерні станції будуть увімкненими постійно (елемент $m_{S,3}^{var,1}$ характеристичної множини $M_S^{var,1}$, елемент $m_{S,3}^{var,3}$ множини $M_S^{var,3}$), тоді формування системи S з компонент буде здійснено один раз і подальші зміни (елемент $m_{S,2}^{var,1}$ характеристичної множини $M_S^{var,1}$) будуть здійснені самою системою при виникнення певних подій.

1.3. Якщо система S формується після увімкнення комп'ютерних станцій в мережі в один і той же час (елемент $m_{S,3}^{var,1}$ характеристичної множини $M_S^{var,1}$, елемент $m_{S,1}^{var,3}$ множини $M_S^{var,3}$), тоді для її подальшого функціонування всі компоненти виконують спеціальну процедуру обміну повідомленнями для початку функціонування.

1.4. Якщо комп'ютерні станції (елемент $m_{S,3}^{var,1}$ характеристичної множини $M_S^{var,1}$, елемент $m_{S,2}^{var,3}$ множини $M_S^{var,3}$), в які встановлено компоненти системи, увімкнено в різний час, тоді компоненти, які були в перших увімкнених

комп’ютерних станціях формують систему, а решта додаються в неї після виконання спеціальної процедури доповнення компонентами в динамічному режимі.

1.5. Якщо до системи S додаються нові чи вилучаються наявні компоненти (елемент $m_{S,2}^{var,1}$ характеристичної множини $M_S^{var,1}$), тоді застосовується спеціальна процедура доповнення чи вилучення компонент із наступним формуванням системи S з наявних активних компонент, які функціонують в увімкнених комп’ютерних станціях. Доповнення системи S новими компонентами або вилучення наявних компонент може бути після виконання підкроків 1.1-1.4. Спеціальна процедура доповнення та вилучення компонент передбачає участь адміністратора системи, перехід до деталізації підкроку 1.5 та подальше виконання підкроку 1.1.

1.5.1. Доповнення системи новими компонентами (елемент $m_{S,1}^{var,2}$ множини $M_S^{var,2}$, елемент $m_{S,1}^{var,3}$ множини $M_S^{var,3}$, елемент $m_{S,2}^{var,4}$ характеристичної множини $M_S^{var,4}$) здійснюємо при увімкненні всіх комп’ютерних станцій, в які встановлені компоненти системи S. Кожну компоненту системи S доповнюємо інформацією про нові компоненти, а нові компоненти доповнюємо інформацією про всі компоненти системи.

1.5.2. Вилучення компонентів з системи S (елемент $m_{S,2}^{var,2}$ множини $M_S^{var,2}$, елемент $m_{S,2}^{var,3}$ множини $M_S^{var,3}$) здійснюємо з використанням однієї комп’ютерної станції, в яку встановлено компоненту, що містить центр прийняття рішень системи. Через інтерфейс компоненти з правами доступу адміністратора надаємо вказівку про вилучення конкретної компоненти. Далі ця компонента системи надсилає повідомлення про вилучення вказаної компоненти решті компонент системи, які активні, тобто функціонують в увімкнених комп’ютерних станціях. Компоненти, які будуть не в увімкнених комп’ютерних станціях, тобто не отримають це повідомлення про вилучення конкретної компоненти, отримають це повідомлення при увімкненні комп’ютерних станціях, в яких вони встановлені, від активних компонент центру прийняття рішень системи S.

1.5.3. При вилученні єдиної компоненти в поточний момент часу (елемент $m_{S,2}^{var,3}$ множини $M_S^{var,3}$), в якій міститься центр прийняття рішень системи, потрібно увімкнути комп’ютерну станцію, в якій наявна компонента з функціоналом центру прийняття рішень, або використати пасивну в цей поточний момент часу

компоненту з центром прийняття рішень. В такому випадку пасивна компонента чи долучена компонента отримують спочатку вказівку про їх одноосібне керування системою, а потім про вилучення заданої компоненти.

1.5.4. Завершення підкроків 1.5.1-1.5.3 здійснюємо визначенням останнього варіанту формування системи згідно формули (3.9) за якою обчислюємо предикат $P_S^{var,2}\left(m_{S,q}^{var,2}\right)$ $(q = 1,2, \dots, n_{M_S^{var,2}})$ на елементах множини $M_S^{var,2}$.

1.5.5. Після виконання підкроку 1.5.4 повертаємось до підкроку 1.5.

1.6. Якщо комп'ютерні станції, в які встановлено компоненти системи S (елемент $m_{S,2}^{var,1}$ характеристичної множини $M_S^{var,1}$, елемент $m_{S,1}^{var,4}$ характеристичної множини $M_S^{var,4}$), не вмикаються тривалий час, тоді система формується з компонент, які перебувають в увімкнених комп'ютерних станціях.

1.7. Завершення кроку 1 здійснюємо визначенням останнього варіанту формування системи згідно формули (3.8) підкроків 1.2-1.4 та 1.6 за якою обчислюємо предикат $P_S^{var,1}\left(m_{S,q}^{var,1}\right)$ $(q = 1,2, \dots, n_{M_S^{var,1}})$ на елементах множини $M_S^{var,1}$, згідно формули (3.10) підкроків 1.2-1.4 та 1.6 за якою обчислюємо предикат $P_S^{var,3}\left(m_{S,q}^{var,3}\right)$ $(q = 1,2, \dots, n_{M_S^{var,3}})$ на елементах множини $M_S^{var,3}$ та згідно формули (3.11) підкроків 1.5.1 та 1.6 за якою обчислюємо предикат $P_S^{var,4}\left(m_{S,q}^{var,4}\right)$ $(q = 1,2, \dots, n_{M_S^{var,4}})$ на елементах множини $M_S^{var,4}$.

1.8. В залежності від подій, які впливатимуть винятково на формування архітектури системи S, та результатів з підкроку 1.7 здійснюємо повернення до одного з підкроків 1.2 – 1.4 чи 1.6.

Результати підкроків 1.5, 1.7 та 1.8 передаються до центру прийняття рішень системи S та опрацьовуються одним із визначених підкроків наступних кроків.

Визначення останнього варіанту формування системи згідно формули (3.8) для елементів характеристичної множини $M_S^{var,1}$ і підкроків 1.2-1.4 та 1.6 не завершує повністю крок 1, а тільки здійснює фіксування стану системи S після повного виконання одного з підкроків в певні проміжки часу, коли в системі не відбуватимуться зміни в її архітектурі. Виконання кроку 1 буде постійним і незалежним від решти кроків, оскільки архітектура системи може змінюватись

постійно і потребуватиме реакції самої системи S на такі події через виконання підкроків кроку 1.

Результати виконання підкроків кроку 1 подамо у таблицю спряження їх з відповідними елементами множин та значеннями предикатів. В результаті отримаємо інформацію про результат виконання певного підкроку та будемо його використовувати для прийняття рішень про подальші кроки системи S.

Крок 2. Встановлення і підтримка зв'язку між компонентами системи.

2.1. Якщо з комп'ютерних станцій, в яких наявні компоненти системи S, в поточний момент часу при старті системи є лише одна увімкнена комп'ютерна станція (елемент $m_{S,2}^{var,8}$ множини $M_S^{var,8}$), то компонента системи S після свого завантаження використає відношення «один до всіх» (елемент $m_{S,1}^{var,7}$ множини $M_S^{var,7}$), згідно якого здійснить розсилання повідомлення всім компонентам системи S для встановлення з ними зв'язку.

2.2. Якщо з комп'ютерних станцій, в яких наявні компоненти системи S, в поточний момент часу при поточному старті системи увімкнені всі (елемент $m_{S,1}^{var,5}$ множини $M_S^{var,5}$) і центром прийняття рішень визначено, щоб задана компонента звернулась до всіх решти компонент, то задана компонента системи S використає відношення «один до всіх» (елемент $m_{S,1}^{var,7}$ множини $M_S^{var,7}$), згідно якого здійснить розсилання повідомлення всім компонентам системи S для підтримки зв'язку з рештою.

2.3. Якщо в системі S відсутня одна компонента через неввімкнення відповідної комп'ютерної станції, тоді всі решта компонентів з певною періодичністю здійснюють звернення до нього для перевірки його наявності, щоб сформувати повну систему, тобто виконуємо відношення «всі до одного» (елемент $m_{S,2}^{var,7}$ множини $M_S^{var,7}$).

2.4. Якщо в системі S сформовано рішення щодо надсилання повідомлення для підтримки та перевірки зв'язку з заданою компонентою з певних причин, тоді всі решта компонентів здійснюють звернення до неї, тобто виконуємо відношення «всі до одного» (елемент $m_{S,2}^{var,7}$ множини $M_S^{var,7}$).

2.5. Якщо в системі S сформовано рішення щодо надсилання повідомлення із конкретної компоненти для підтримки та перевірки зв'язку з заданою компонентою

з певних причин, тоді виконуємо відношення «один до одного» (елемент $m_{S,3}^{var,7}$ множини $M_S^{var,7}$).

2.6. Якщо в системі S сформовано рішення щодо надсилання повідомлення із конкретної компоненти для підтримки та перевірки зв'язку з заданою певною кількістю компонент, але не з усіма, з певних причин, тоді виконуємо відношення «один до певної кількості, але не до всіх» (елемент $m_{S,4}^{var,7}$ множини $M_S^{var,7}$).

2.7. Якщо в системі S сформовано рішення щодо надсилання повідомлення із конкретної певної кількості визначених, але не всіх, компонент до однієї для підтримки та перевірки зв'язку з заданою компонентою з певних причин, тоді виконуємо відношення «певна кількість, але не всі, до одного» (елемент $m_{S,5}^{var,7}$ множини $M_S^{var,7}$).

2.8. Якщо в системі S сформовано рішення щодо надсилання повідомлення із конкретної певної кількості визначених, але не всіх, компонент до певної кількості для підтримки та перевірки зв'язку з ними з певних причин, тоді виконуємо відношення «певна кількість, але не всі, до певної кількості, але не до всіх» (елемент $m_{S,6}^{var,7}$ множини $M_S^{var,7}$).

2.9. Якщо компоненті надіслано повідомлення чи вказівка, а вона в поточний момент часу вимикається разом з комп'ютерною станцією, тоді вона надсилає повідомлення всім компонентам, які є активним, тобто тим, які є у ввімкнених комп'ютерних станціях, і виконуємо відношення «один до певної кількості, але не до всіх» (елемент $m_{S,4}^{var,7}$ множини $M_S^{var,7}$) і команди чи повідомлення, які їй надсилались анульовуються.

2.10. Якщо компоненті надіслано повідомлення чи вказівка, а вона в поточний момент часу вимикається аварійно разом з комп'ютерною станцією, тоді вона не надсилає повідомлення про вимкнення всім компонентам, які є активним, тобто тим, які є у ввімкнених комп'ютерних станціях, і при наступному увімкненні комп'ютерної станції компонента повідомляє всім решті активних компонент про попередню аварійну подію і встановлює з ними зв'язок, виконуючи відношення «один до певної кількості, але не до всіх» (елемент $m_{S,4}^{var,7}$ множини $M_S^{var,7}$), а команди чи повідомлення, які їй надсилались від певних компонент анульовуються.

2.11. Якщо при встановленні зв’язку між компонентами системи стандартна частина (згідно схеми «квітування» тільки для підтвердження встановлення зв’язку та активності компонент) та додаткова частина (згідно використання надмірності для додаткового підтвердження легітимності компоненти) була здійснена успішно для всіх компонентів системи в комп’ютерних станціях, які увімкнені в один і той же час (елемент $m_{S,1}^{var,5}$ множини $M_S^{var,5}$), тоді система S продовжить функціонування в штатному режимі.

2.12. Якщо при встановленні зв’язку між компонентами системи стандартна частина (згідно схеми «квітування» тільки для підтвердження встановлення зв’язку та активності компонент) та додаткова частина (згідно використання надмірності для додаткового підтвердження легітимності компоненти) була здійснена успішно для всіх компонентів системи в комп’ютерних станціях, які вмикаються в різний час, причому частина може бути після певного часу функціонування вимкнена, а певна частина після цього часу увімкнена або не вмикатись взагалі протягом тривалого певного часу (елемент $m_{S,2}^{var,5}$ множини $M_S^{var,5}$), тоді система S продовжить функціонування в штатному режимі в складі активних компонент у ввімкнених комп’ютерних станціях.

2.13. Якщо для випадків підкроків 2.11, 2.12 при встановленні зв’язку між компонентами системи стандартна частина (згідно схеми «квітування» тільки для підтвердження встановлення зв’язку та активності компонент) невиконана успішно, тоді компоненти системи, які встановили такий факт про певну компоненту, повідомляють про такий результат в центр прийняття рішень системи.

2.13.1. Якщо таке повідомлення надійшло від двох компонент, які здійснювали спробу встановлення зв’язку між собою, тоді центр прийняття рішень дає вказівку їм зробити повторно спробу встановлення зв’язку за стандартною частиною процедури, додатково ще одній компоненті дає вказівку встановити зв’язок з цими двома компонентами і про результати виконання ці три компоненти повинні повідомити центр прийняття рішень.

2.13.2. Якщо таке повідомлення надійшло від однієї з двох компонент, які здійснювали спробу встановлення зв’язку між собою, тоді центр прийняття рішень дає вказівку цій компоненті та двом іншим активним компонентам зробити спробу

встановлення зв’язку за стандартною частиною процедури і про результати виконання ці три компоненти повинні повідомити центр прийняття рішень.

2.14. Якщо було підтверджено на підкроках 2.13.1 та 2.13.2 проблеми зі встановленням зв’язку з певною компонентою за стандартною частиною процедури встановлення зв’язку, тоді така компонента центром прийняття рішень відноситься до переліку компонент, які потребують дослідження і з певною періодичністю буде тестуватись зв’язок з нею певною кількістю компонент (елемент $m_{S,2}^{var,5}$ множини $M_S^{var,5}$).

2.15. Якщо для випадків підкроків 2.11, 2.12 при встановленні зв’язку між компонентами системи стандартна частина (згідно схеми «квітування» тільки для підтвердження встановлення зв’язку та активності компонент) виконана успішно, а додаткова частина не виконана успішно, тоді компоненти системи, які встановили такий факт про певну компоненту, повідомляють про такий результат в центр прийняття рішень системи (елемент $m_{S,2}^{var,5}$ множини $M_S^{var,5}$).

2.15.1. Якщо таке повідомлення надійшло від двох компонент, які здійснювали спробу встановлення зв’язку між собою, тоді центр прийняття рішень дає вказівку їм зробити повторно спробу встановлення зв’язку за додатковою частиною процедури, додатково ще одній компоненті дає вказівку встановити зв’язок з цими двома компонентами і про результати виконання ці три компоненти повинні повідомити центр прийняття рішень.

2.15.2. Якщо таке повідомлення надійшло від однієї з двох компонент, які здійснювали спробу встановлення зв’язку між собою, тоді центр прийняття рішень дає вказівку цій компоненті та двом іншим активним компонентам зробити спробу встановлення зв’язку за додатковою частиною процедури і про результати виконання ці три компоненти повинні повідомити центр прийняття рішень.

2.16. Якщо було підтверджено на підкроках 2.13.1 та 2.13.2 проблеми зі встановленням зв’язку з певною компонентою за додатковою частиною процедури встановлення зв’язку, тоді така компонента центром прийняття рішень (елемент $m_{S,2}^{var,5}$ множини $M_S^{var,5}$) досліджується через негайне тестування зв’язку з нею певною кількістю компонент та при встановленні проблем вилучається з системи і надсилається відповідне повідомлення про таку подію адміністратору системи.

2.17. Якщо комп’ютерні станції, в яких наявні компоненти системи, вимкнені коректно в один і той же час (елемент $m_{S,1}^{var,6}$ множини $M_S^{var,6}$), тоді компоненти системи в них зберігають інформацію про коректне завершення свого функціонування і наступного разу розпочинають роботу із стандартних заданих дій.

2.18. Якщо комп’ютерні станції, в яких наявні компоненти системи, вимкнені аварійно в один і той же час (елемент $m_{S,2}^{var,6}$ множини $M_S^{var,6}$), тоді компоненти не завершили коректний вихід і при увімкненні комп’ютерних систем компоненти, які встановлені в них, будуть виконувати процедуру коректного повторного виконання не завершених попередніх завдань разом з процедурою початкового завантаження.

2.19. Якщо комп’ютерні станції, в яких наявні компоненти системи, вимкнені коректно в різний час (елемент $m_{S,3}^{var,6}$ множини $M_S^{var,6}$), тоді компоненти системи в них зберігають інформацію про коректне завершення свого функціонування і наступного разу розпочинають роботу із стандартних заданих дій з врахуванням часу вимкнення решти компонент по відношенню до певної компоненти.

2.20. Якщо комп’ютерні станції, в яких наявні компоненти системи, вимкнені в різний час частково коректно (елемент $m_{S,2}^{var,8}$ множини $M_S^{var,8}$) і частково аварійно (елемент $m_{S,4}^{var,6}$ множини $M_S^{var,6}$, елемент $m_{S,3}^{var,8}$ множини $M_S^{var,8}$), тоді для компонент, які були в комп’ютерних станціях, що вимкнулись коректно виконуємо підкрок 2.19, а для решти підкрок 2.18.

Крок 3. Забезпечення цілісності системи.

3.1. Якщо в межах корпоративної мережі відбудеться поділ системи S на дві і більше не пов’язаних підсистем через виведення з ладу на певний час обладнання, тоді кожна з частин здійснює переформування себе в зменшену систему S і буде продовжувати роботу за умови, якщо в кожній з частин залишаться не менше двох активних компонент з центром прийняття рішень.

3.1.1. Якщо в одній з частин відсутні активні компоненти з центром прийняття рішень та неактивні, тоді компоненти такої частини блокують роботу комп’ютерних станцій і видають відповідне повідомлення для адміністратора.

3.1.2. Якщо компоненти з центром прийняття рішень неактивні в момент певного аварійного або навмисного відокремлення другої частини, в якій будуть всі активні компоненти з центром прийняття рішень системи, тоді переведення їх до

активного стану відбудеться після чергового сеансу зв’язку і встановлення факту відсутності активних компонент з центром прийняття рішень.

3.2. Якщо частина активних компонент, які містять центр прийняття рішень системи, будуть з певних причин вилучені з системи, тоді та частина, яка залишилась, розпочне процедуру формування системи з наявних компонент.

3.3. Якщо наявних активних компонент з центром прийняття рішень немає, тоді наявні компоненти блокують роботу комп’ютерних станцій і видають відповідне повідомлення адміністратору.

Крок 4. Організація часткової централізації.

4.1. Формування рішення щодо кількості компонент (елемент $m_{S,1}^{st}$ множини M_S^{st}), в яких буде функціонувати центр прийняття рішень системи, визначається всіма компонентами системи, в яких наявний функціонал центру прийняття рішень, при першому старті системи. Кількість активних компонент центру прийняття рішень буде менша двох третин і більше однієї компоненти. Кожна компонента на початку старту системи генерує випадковим чином число з інтервалу від двох до двох третин кількості компонент центру і всі ці компоненти обмінюються такими числами між собою та знаходять серед цих чисел середньоарифметичне число та відкидають в ньому дробову частину.

4.2. Якщо при наступному старті системи не всі компоненти з центром прийняття рішень системи будуть активними в увімкнених комп’ютерних станціях, тоді наявні компоненти приймуть рішення щодо кількості активних компонент (елемент $m_{S,1}^{st}$ множини M_S^{st}), в яких буде центр прийняття рішень. Якщо при увімкненні комп’ютерних станцій з компонентами, в яких був на попередньому етапі функціонування активним центр прийняття рішень, то такі компоненти отримують повідомлення від центру прийняття рішень про перехід до пасивного стану їх функціоналу центру прийняття рішень.

4.3. Для вибору певних компонент центру прийняття рішень, які будуть активними, після виконання підкроку 4.1 кожна компонента випадковим чином генерує числа з діапазону від одиниці по число, що дорівнює кількості компонент з центром прийняття рішень. Після формування таких послідовностей чисел відбувається обмін результатами між усіма компонентами з центром прийняття рішень, в усіх послідовностях числа сортуються за неспаданням і після сортування

обчислюється ціла частина від середньоарифметичного значення чисел з однаковим індексом.

4.4. Для вибору певних компонент центру прийняття рішень, які будуть активними, після виконання підкроку 4.2 кожна компонента випадковим чином генерує числа з діапазону від одиниці по число, що дорівнює кількості активних компонент з центром прийняття рішень в поточний момент часу. Після формування таких послідовностей чисел відбувається обмін результатами між усіма активними компонентами з центром прийняття рішень, в усіх послідовностях числа сортуються за неспаданням і після сортування обчислюється ціла частина від середньоарифметичного значення чисел з однаковим індексом.

4.5. Якщо на підкроці 4.4 таких активних компонент буде дві, тоді вони будуть виконувати функціонал центру прийняття рішень і при появі в системі долучених компонент з функціоналом центру прийняття рішень здійснять виконання підкроку 4.4.

4.6. Якщо на підкроці 4.4 таких активних компонент буде менше двох, тоді функціонал центру прийняття рішень буде в одній компоненті і при появі в системі долучених компонент з функціоналом центру прийняття рішень здійснять виконання підкроку 4.5.

4.7. Якщо у ввімкнених комп’ютерних станціях наявні компоненти, в яких відсутні компоненти з центром прийняття рішень, тоді кожна з компонент здійснює облік подій, а система не функціонує в штатному режимі.

4.8. Завдання для системи щодо її подальших кроків або для певних компонент формуються окремо в кожній з активних компонент центру прийняття рішень і після узгодження рішення між ними таке завдання направляється для виконання.

Для підкроків кроку 4 можуть бути інші алгоритми визначення кількості компонент з центром прийняття рішень та безпосередньо компоненти центру прийняття рішень. Наприклад, можуть бути середньозважені значення, середнє гармонійне значення тощо. Також, функціонал може містити декілька алгоритмів і в поточні моменти часу всі компоненти можуть використовувати один з них.

Крок 5. Здійснення міграції центру прийняття рішень системи.

5.1. Якщо всі компоненти центру прийняття рішень активні в поточний момент часу і частина з них формує центр прийняття рішень, а решта перебувають в

пасивному стані, то періодично частина активних компонент ставатиме пасивними і навпаки пасивні ставатимуть активними. Рішення про наступний перегляд компонент, які будуть залучені до формування центру прийняття рішень буде прийнято активними в поточний момент компонентами.

5.2. Якщо стан безпеки в комп'ютерній станції за оцінюванням системи понизився і компонента в ній є активною компонентою центру прийняття рішень, тоді решта компонент системи приймають рішення про переведення цієї компоненти до пасивного стану і іншу компоненту роблять активною.

5.3. Якщо не всі компоненти центру прийняття рішень є активними в поточний момент часу через не ввімкнення їх комп'ютерних станцій, частина активних з них формує центр прийняття рішень, решта компонент з центром прийняття рішень перебувають в пасивному стані, то центр прийняття рішень доповнить кількість активних компонент за рахунок пасивних.

Крок 6. Оцінювання стану компонент та системи.

6.1. Обчислюємо загальні стани компонентів та комп'ютерних станцій значення $\alpha'_{3,S_{1,n}}$ згідно формули (2.59).

6.2. Обчислюємо стан системи в цілому за формулою (3.12).

Крок 7. Оцінювання результатів розподілених обчислень в компонентах.

7.1. Значення характеристичних показників компонент системи в залежності від типів виконуваних завдань визначаємо за формулами (2.18), (2.50) і (2.59) та надсилаємо їх всім активним компонентам центру прийняття рішень.

7.2. Якщо результати обчислень, що проведені в різних компонентах системи, будуть однакові і кожна з компонент, які приймають участь в їх опрацюванні, отримала однакові значення протягом задано проміжку часу, тоді один з отриманих результатів приймається як остаточне значення розподілених обчислень.

7.3. Якщо результати обчислень, що проведені в різних компонентах системи, не будуть однакові і кожна з компонент, які приймають участь в їх опрацюванні, отримала однакові набори значень протягом задано проміжку часу, тоді здійснюється визначення відсотку найбільшої кількості однакових значень результатів обчислень до всіх отриманих значень.

7.3.1. Якщо відсоток значень результатів обчислень дорівнює $\frac{N-1}{N} \cdot 100\%$ (N – кількість значень), тоді компоненті, в якій відмінний від решти результат буде

надіслано додаткові контрольні значення для обчислень з метою перевірки її легітимності та один з $N-1$ отриманих результатів приймається як остаточне значення розподілених обчислень. Якщо компонента, яку будуть перевіряти через відмінне значення результату від решти, відноситься до активних компонент центру прийняття рішень, тоді їй буде надіслано для обробки отриманий набір значень і решта активних компонент центру прийняття рішень будуть досліджувати її відповідь та приймати рішення щодо її подальшого функціонування в системі.

7.3.2. Якщо відсоток значень результатів обчислень менше $\frac{N-1}{N} \cdot 100\%$ (N – кількість значень) і більше 50%, тоді компонентам, в яких відмінний від решти результат буде надіслано додаткові перевіряючі значення для обчислень з метою перевірки її легітимності та один з отриманих результатів, який склав більше 50%, приймається як остаточне значення розподілених обчислень. Якщо компонента, яку будуть перевіряти через відмінне значення результату від решти, відноситься до активних компонент центру прийняття рішень, тоді їй буде надіслано для обробки отриманий набір значень і решта активних компонент центру прийняття рішень будуть досліджувати її відповідь та приймати рішення щодо її подальшого функціонування в системі.

7.3.3. Якщо відсоток значень результатів обчислень менше 50% від кількості всіх серед найбільшої кількості одного значення, тоді результати обчислень не приймаються і система розпочинає здійснювати самотестування. Після його завершення, якщо воно буде успішним, повторить виконання цього завдання або відхилить його виконання.

Значення характеристичних показників компонент системи в залежності від типів виконуваних завдань можна визначати за формулами (3.13)-(3.18) в залежності від особливостей обробки. Кластери таких значень можуть формуватись з врахуванням часових затримок при передачі результатів розподілених обчислень. Можуть бути застосовні інші формули для визначення значень характеристичних показників компонент системи.

Крок 8. Визначення компонент, в яких буде виконуватись поставлене системою завдання.

8.1. Для визначення компонент, в яких буде виконуватись поставлене системою завдання визначаємо рівень безпеки всіх компонент за формулою (3.12), за першим

варіантом поділу на класи (формули (3.13)-(3.16)) і, таким чином, визначаємо половину компонент, ставлячи в формулу (3,16) коефіцієнт 0,5 замість 0,2.

8.2. Якщо поставлене завдання вимагає негайного виконання або має статус такого, яке пов’язане з дослідженням безпеки в компонентах, то його виконують всі компоненти.

8.3. До виконання поставленого завдання за вимогою (прийнятим рішенням або за вказівкою до виконання) залучаємо меншу кількість компонент, якщо їх багато, але ця кількість не може бути менше десяти компонент.

8.4. Якщо кількість компонент в системі невелика, наприклад менше десяти, тоді всі компоненти залучаються до виконання поставленого завдання.

8.5. Обробка накопиченої інформації в компонентах центру прийняття рішень системи, формування бази прийнятих рішень в цих компонентах та надання такої бази всім компонентам центру прийняття рішень.

8.6. Прийняття рішення про виконання конкретної функції в компонентах в залежності від поточних даних в системі.

Крок 9. Перебудова архітектури системи за наявності критичних подій.

9.1. Якщо виявлено аномальні події або зловмисні прояви в комп’ютерній мережі чи станціях і про це повідомляють компоненти, тоді запускається процедура перебудови архітектури системи.

9.2. Якщо в певних компонентах системи виявлено тривале функціонування підсистем з виявлення аномальних подій або зловмисних проявів і при цьому такі компоненти повідомляють центр прийняття рішень про потребу продовжувати виконання завдання, а часові обмеження для виконання таких завдань вже пройдені, тоді системи визначається з потребою перебудови своєї архітектури без врахування цих компонент, а в них за наявності функціонал центру прийняття рішень переводиться з активного стану до пасивного.

9.3. Якщо система згідно обчисленого значення рівня безпеки перебуває в критичному стані, тоді вона вилучає зі своєї архітектури частину компонент з найбільшими значеннями критичного стану компонент і перераховує поточний стан.

9.3.1. Якщо після такої перебудови стан безпеки не є критичним, то вона продовжує функціонування.

9.3.2. Якщо після такої перебудови стан безпеки залишається критичним, то вона зупиняє функціонування і видає повідомлення адміністратору.

Крок 10. Визначення подальших кроків системи в поточний момент часу.

10.1. Визначаємо наступний стан системи S через її поточний стан та показники компонент і системи за формулою (3.19).

10.2 Якщо в системі відбулась подія з множини M_S^{pd}, то вона опрацьовується функціями компонент системи та центром прийняття рішень вибирається варіант стану з множини станів M_S^{st}, оцінюється можливість виконання визначеного переходу до наступного стану в поточний момент часу та відбувається здійснення безпосередньо переходу з перевіркою його повного завершення.

10.3. Якщо в системі відбулась подія не з множини M_S^{pd}, то вона опрацьовується функціями центру прийняття рішень.

10.3.1. Якщо вибрано варіант стану з множини станів M_S^{st}, проведено оцінювання можливості виконання визначеного переходу до наступного стану в поточний момент часу, тоді відбувається здійснення безпосередньо переходу з перевіркою його повного завершення, і доповнення множини подій цією подією.

10.3.2. Якщо вибрано варіант стану з множини станів M_S^{st}, проведено оцінювання можливості виконання визначеного переходу до наступного стану в поточний момент часу, тоді відбувається здійснення безпосередньо переходу з перевіркою його повного завершення і якщо подія залишається активною після зміни стану системи, тоді система блокує компоненти та повідомляє про проблему адміністратору.

10.4. Якщо серед елементів множини подій M_S^{pd} відсутня подія, яка виникла саме в системі і потребує опрацювання, тоді центр прийняття рішень системи повертає систему до попереднього стану і аналізує подальшу наявність цієї події.

10.4.1. Якщо так подія наявна після зміни стану системи, тоді компоненти системи блокують процеси в комп’ютерних станціях і вся система робить перехід до критичного стану безпеки, а подію додає в множину подій.

10.4.2. Якщо така подія буде відсутня після зміни стану системи до попереднього, тоді цю подію система додає в множину подій і фіксує стан системи, в якому вона зникає.

10.5. Якщо для події, яка є серед елементів множини подій M_S^{pd}, відсутні вимоги переходу до наступного стану, тоді вона опрацьовується і система залишається в поточному стані.

10.5. Якщо частина компонент системи не встигла завершити перехід у визначений стан з певних причин, то в подальшому при їх активності вони відтворюють пропущені стани в своїй історії станів, поновлюють актуальні показники і переходять до поточного стану системи.

Крок 11. Завершення функціонування компонентів та системи.

11.1. Завершення функціонування компонентів та системи в поточний момент часу з подальшим їх поверненням до виконання завдань при увімкненні комп'ютерних станцій.

11.2. Блокування частини компонент чи системи центром прийняття рішень системи.

11.3. Завершення функціонування частини компонентів в поточний момент часу з подальшим їх поверненням до виконання завдань при увімкненні комп'ютерних станцій.

11.4. Вилучення компонентів чи системи в цілому з вузлів в мережі.

Таким чином, розроблений метод організації функціонування частково централізованих розподілених систем дає змогу створювати їх згідно принципів самоорганізації та адаптивності. Часткова централізація таких розподілених систем досягається виокремленням компонент центру прийняття рішень системи, в кожній з яких приймається окремо рішення, яке в подальшому узгоджується з рештою прийнятих рішень. При цьому компоненти центру прийняття рішень функціонують за принципом децентралізації, а вся система функціонує за принципом централізації. В розробленому методі функціонування такого типу систем проведено розподіл компонент за відношенням до центру прийняття рішень, що дало змогу реалізувати часткову централізацію сумісно з принципами самоорганізації та адаптивності, які задають механізми до самостійного прийняття рішень щодо подальших кроків системою та перебудови її архітектури за потреби.

3.3. Метод виявлення worm-вірусів в комп’ютерних мережах за багатокласовою класифікацією

В комп’ютерних мережах може перебувати різноманітне ЗПЗ. Завдяки технологіям та засобам підтримки функціонування комп’ютерних мереж, крім корисного їх застосування, наявні широкі можливості їх використання зловмисниками. Наприклад, для створення бот-мереж зловмисники можуть використовувати стандартні засоби роботи з пересилання повідомлень та файлів, команди, а також можуть для досягнення своєї мети, щоб приховати свої зловмисні дії, використати мережні віруси для проникнення у вузли в мережах та встановлення в них контролю. Такими вірусами можуть бути worm-віруси. Розглянемо їх в контексті цілеспрямованого поширення і отримання контролю над комп’ютерними станціями в корпоративних мережах, а не випадкового поширення з метою нанесення шкоди користувачам комп’ютерів, які під’єднані до глобальної мережі. Шкода від таких вірусів може обмежуватись зниженням пропускної здатності. Worm-віруси на відміну від звичайних комп’ютерних вірусів, мають певні особливості. Визначальною особливістю є спрямування worm-вірусів на інфікування переважно комп’ютерів, а не файлів в них, і цільова функція спрямована саме на досягнення максимізації інфікування кількості комп’ютерів, а не файлів в них. Хоча можуть бути і такі, що додатково спрямовані на інфікування файлів в комп’ютерних станціях, в які отримали доступ. Маючи такий функціонал у worm-вірусах щодо їх поширення і спрямування саме для поширення в комп’ютерних мережах, як локальних так і глобальних, зловмисники можуть їх використати для цілеспрямованого охоплення корпоративної мережі, яка їх цікавить та, як наслідок, навколо якої можуть створити зони поширення таких worm-вірусів. Тому, захищаючи корпоративну мережу частково централізованими розподіленими системами потрібно імплементувати в них підсистеми та засоби протидії такому ЗПЗ, як worm-віруси.

Введемо множину W комп’ютерних worm-вірусів так:

$$W = \{w_1, w_2, \dots, w_{N_w}\}, \tag{3.20}$$

де w_i - i - worm-вірус; N_w – кількість відомих worm-вірусів.

Для виявлення worm-вірусів здійснимо аналіз їх будови, типів розмноження та поширення. Це дасть змогу виділити типові характеристики. За поєднанням типових характеристик здійснимо поділ елементів множини W на класи. Цей поділ дасть змогу виділити особливі характеристики у worm-вірусів певних класів, що покращить ефективність їх виявлення та дасть змогу чіткіше відокремити їх від корисних програм чи процесів.

Задамо характеристичні показники worm-вірусів множиною $M_W = \{m_{W,1}, m_{W,2}, \dots, m_{W,N_W}\}$, де N_W – кількість характеристичних показників, $m_{W,i}$ - i-ий характеристичний показник, $i = 1,2, \dots, N_W$. Деталізуємо кожен характеристичний показник з метою подальшого поєднання їх елементів для задання відповідно типу worm-вірусів.

Розглянемо перший характеристичний показник, який характеризує тип розмноження, тоді елемент $m_{W,1}$ – характеристичний показник типів розмноження worm-вірусів. Деталізуємо його так: $m_{W,1,1}$ - розмноження worm-вірусів, яке забезпечується за рахунок вразливостей програмного забезпечення; $m_{W,1,2}$ – розмноження worm-вірусів, яке забезпечується за допомогою програм для спілкування; $m_{W,1,3}$ – розмноження worm-вірусів, яке забезпечується за допомогою електронної пошти та адрес; $m_{W,1,4}$ – розмноження worm-вірусів, яке забезпечується за допомогою мережних ресурсів; $m_{W,1,5}$ – розмноження worm-вірусів, яке забезпечується за допомогою P2P мережі каналами файлообмінних пірінгових мереж. Елемент $m_{W,1,3}$ може бути поділений на два такі випадки: $m_{W,1,3,1}$ характеризує масову розсилку на всі електронні пошти; $m_{W,1,3,2}$ характеризує розсилку на визначені адреси електронної пошти. Конструктивно worm-віруси можуть поєднувати декілька $m_{W,1,j}$ $(j = 1,2, \dots,5)$ формуючи, таким чином, багатовекторність. Завдяки наявності декількох механізмів розмноження зростають можливості його поширення в комп'ютерних мережах.

Worm-вірус надходить в комп'ютерну станцію по комп'ютерній мережі в форматі виконуваного файлу і активізується в ній після його запуску. Тому, другим важливим характеристичним показником $m_{W,2}$ є структура worm-вірусів. Виділимо різні за призначенням елементи типових структур так: $m_{W,2,1}$ – експлойт (або двійковий виконуваний код) і розміщене в оперативному запам'ятовуючому

пристрої (ОЗП) корисне навантаження; $m_{W,2,2}$ - локальна частина корисного навантаження в оперативній пам'яті та завантаження решти worm-вірусу окремим файлом засобами комп'ютерної мережі; $m_{W,2,3}$ - один файл; $m_{W,2,4}$ – решта варіантів.

Елемент $m_{W,2,1}$ характеризує резидентні worm-віруси. Крім того, цей елемент може бути деталізований за ознакою відношення експлойту до певних об'єктів в комп'ютерних станціях так: $m_{W,2,1,1}$ – використання для прикладного програмного забезпечення; $m_{W,2,1,2}$ - використання для операційних систем; $m_{W,2,1,3}$ - використання для браузерів; $m_{W,2,1,4}$ – використання для сайтів; $m_{W,2,1,5}$ – використання для спеціалізованого програмного забезпечення; $m_{W,2,1,6}$ – використання для решти засобів, які використовуються в комп'ютерній станції та мають вразливості.

Елемент $m_{W,2,3}$ характеризує поштові worm-віруси.

При виборі за характеристичний показник елементу $m_{W,1}$ можна поділити всю множину worm-вірусів на такі класи: клас, в якому не міститься жодного елементу з характеристичним показником $m_{W,1,j}$ $(j = 1,2,\dots,5)$, тобто клас, в якому відсутні worm-віруси, і позначимо його K_W^0; клас K_W^j $(j = 1,2,\dots,5)$, який визначатиметься характеристичним показником $m_{W,1,j}$ $(j = 1,2,\dots,5)$, і всього таких класів буде п'ять; клас K_W^6, в який будуть віднесені елементи, для характеристики яких буде більше одного характеристичного показника $m_{W,1,j}$ $(j = 1,2,\dots,5)$. Побудова класу K_W^6 може бути здійснена системою S в процесі її функціонування при виявленні багатовекторних worm-вірусів. Для віднесення об'єкту до класу K_W^6 система S повинна встановити його мінімум в двох класах з класів K_W^j $(j = 1,2,\dots,5)$. Формування класу K_W^0 можливе за умови помилкового віднесення worm-вірусів до нього при застосуванні систем виявлення. При правильно здійсненій класифікації worm-вірусів клас K_W^0 буде порожнім, тобто $K_W^0 = \varnothing$. Наявність елементів в класі K_W^0 буде означати помилковість спрацювання відповідного детектора та системи в цілому. Таким чином, всю множину worm-вірусів поділимо на шість класів за характеристичним показником елементу $m_{W,1}$:

$$W = \bigcup_{j=1}^{6} K_W^j. \qquad (3.21)$$

Отриманий поділ множини worm-вірусів на шість класів дає змогу здійснити фіксування характерних властивостей і може бути деталізований за певними визначеними критеріями.

Аналогічно до характеристичних показників, які задані елементами $m_{W,1}$ та $m_{W,1}$, визначимо решту елементів множини M_W. Приймемо за характеристичні показники такі особливості worm-вірусів: $m_{W,3}$ - методи вибору мішені для інфікування; $m_{W,4}$ – тип носія worm-вірусу для пересилання себе у визначену ціль; $m_{W,5}$ – метод worm-вірусу для впливу на ціль, тобто метод активації; $m_{W,6}$ – тип корисного навантаження worm-вірусу для досягнення мети; $m_{W,7}$ – види цілей зловмисників, що використовують worm-віруси для досягнення мети, тобто типи спрямованості зловмисників.

Для виявлення цілі, тобто мішені, worm-віруси повинні мати функції або функцію для здійснення сканування в мережі, списки зовнішніх і внутрішніх цілей, підготовлені списки цілей, передані через певний час списки цілей, пасивного моніторингу. Також, у worm-віруси можуть бути закладені комбінації функцій різного призначення щодо виявлення цілі. Тому, характеристичний показник, який задано елементом $m_{W,3}$, можемо деталізувати так: $m_{W,3,1}$ - здійснення сканування у внутрішній мережі; $m_{W,3,2}$ – пасивний моніторинг; $m_{W,3,3}$ – отримання списку цілей ззовні корпоративної мережі; $m_{W,3,4}$ - здійснення сканування у зовнішній мережі; $m_{W,3,3}$ – отримання списку цілей безпосередньо з певного вузла корпоративної мережі; $m_{W,3,4}$ - автоматичне спрямування на вказану ціль без сканування. Деталізуємо елемент $m_{W,3,1}$ так: $m_{W,3,1,1}$ – послідовна обробка впорядкованих адрес для виявлення вразливих хостів; $m_{W,3,1,2}$ – випадкова обробка випадкових адрес для виявлення вразливих хостів. Пасивний моніторинг передбачає очікування певної поведінки користувача або з’єднання з worm-вірусом ззовні.

Деталізуємо тип носія worm-вірусу для пересилання себе у визначену ціль так: $m_{W,4,1}$ – самостійне перенесення; $m_{W,4,2}$ – використання вторинного каналу зв’язку для завершення інфікування; $m_{W,4,3}$ – worm-вірус є частиною звичайного повідомлення.

Для впливу на ціль, тобто метод активації worm-вірусу заданий елементом $m_{W,5}$, деталізуємо так: $m_{W,5,1}$ - скануючі worm-віруси; $m_{W,5,2}$ - автономні worm-

віруси; $m_{W,5,3}$ - worm-віруси, які потребують активації за таймером; $m_{W,5,4}$ - worm-віруси, які потребують активації користувачем; $m_{W,5,5}$ – застосування розподіленої координації при скануванні.

Використання методу поширення worm-вірусів, який базується на елементі $m_{W,5,1}$, формує в мережі аномальні прояви, що призводить до дослідження поведінки таких проявів відповідними засобами. В результаті класи worm-вірусів, для яких характерним є показник, який задано елементом $m_{W,5,1}$, можуть швидко виявлятись методами, що базуються на виявленні аномалій.

Деталізуємо тип корисного навантаження worm-вірусу для досягнення мети так: $m_{W,6,1}$ – відсутнє корисне навантаження; $m_{W,6,2}$ – надання відкритого доступу до інфікованого комп'ютера; $m_{W,6,3}$ – надання доступу для спамерів до електронної пошти; $m_{W,6,4}$ – підтримка функцій забезпечення фішингових атак; $m_{W,6,5}$ – підтримка зв'язку з зловмисником, якщо worm-вірус розроблявся для цілеспрямованих атак; $m_{W,6,6}$ – збір даних з комп'ютерної станції; $m_{W,6,7}$ – надання віддаленого доступу до комп'ютерної станції.

Деталізуємо тип корисного навантаження worm-вірусу для досягнення мети так: $m_{W,7,1}$ – підтримка атак, зокрема розподілених з узгодженою комунікацією між worm-вірусами; $m_{W,7,2}$ – пошкодження даних в комп'ютерній станції; $m_{W,7,3}$ – захист worm-вірусу в комп'ютерній станції від спроби користувача видалити його, тобто активізація зловмисного функціоналу лише у випадку спроби його видалити; $m_{W,7,4}$ – пошкодження апаратного забезпечення певного типу в комп'ютерній станції, яке має вразливості, зокрема і перепрограмування або знищення програм ініціалізації.

Елементи множини M_W, які відповідно задають характерні показники worm-вірусів можуть бути деталізовані іншими особливостями. Всі елементи множини M_W сформовані у worm-вірусах певними функціями. Задамо множину функцій-множин $F_{M_W,1} = \left\{f_{m_{W,1}}, f_{m_{W,2}}, \dots, f_{m_{W N_{W,}}}\right\}$, де N_W – кількість характеристичних показників та, відповідно, функцій-множин. Кожну з функцій-множин задамо функціями-підмножинами, які формуватимуть їх. Функції-підмножини будуть задавати та відповідатимуть безпосередньо функціям, які зловмисники закладатимуть в функціонал worm-вірусів. Ці функції підмножини можуть

повторюватись в різних функціях-множинах. Тому, поодинокі їх використання для ідентифікації worm-вірусів не можуть бути використані. Ці функції-підмножини потрібно поєднувати між собою для виконання певних завершених дій і ці дії повинні відповідати характеристичним показникам, що визначені елементами множини M_W. Функції-підмножини задамо відповідною множиною $F_{M_W,2} = \left\{f_{m_{W,2,1}}, f_{m_{W,2,2}}, \dots, f_{m_{W,2N_{W,2}}}\right\}$, де $N_{W,2}$ – кількість функцій-підмножин.

Для виявлення чи ідентифікації worm-вірусів побудуємо ознакове поле. До ознакового поля включимо поведінкові сигнатури worm-вірусів, аналітичні вирази характеристик згідно поведінкових сигнатур, шаблони атак та відбитки, які можуть бути отримані з приманок для worm-вірусів, а також зміни в оточуючому середовищі, тобто в корпоративній мережі.

Сформуємо поведінкові сигнатури з функцій-підмножин і задамо їх векторами так:

$$v_z^{f_{m_{W,h}}} = \left(v_{z,1}^{f_{m_{W,h}}}, v_{z,2}^{f_{m_{W,h}}}, \dots, v_{z,N_{v_z}^{f_{m_{W,h}}}}^{f_{m_{W,h}}}\right), \tag{3.22}$$

де $v_z^{f_{m_{W,h}}}$ - вектор для h-тої функції $f_{m_{W,h}}$ з множини $F_{M_W,1}$; $h = 1,2,\dots,N_W$; N_W – кількість функцій-множин; $N_{v_z^{f_{m_{W,h}}}}$- кількість функцій-підмножин у векторі $v_z^{f_{m_{W,h}}}$; z - тий вектор $v_z^{f_{m_{W,h}}}$ для h-тої функції $f_{m_{W,h}}$; $z = 1,2,\dots,N_{z,h}$; $N_{z,h}$ - кількість векторів $v_z^{f_{m_{W,h}}}$ для h-тої функції $f_{m_{W,h}}$.

Введемо аналітичні вирази для обчислення ваги функцій-підмножин (координат вектору $v_z^{f_{m_{W,h}}}$), які переважно використовують зловмисники при створенні worm-вірусів. Введемо функцію $F_{v_z^{f_{m_{W,h}}}}$, значенням якої буде вага функції-підмножини з множини $F_{M_W,2}$, так:

$$F_{v_z^{f_{m_{W,h}}}}: v_{z,i}^{f_{m_{W,h}}} \to [0;1], \tag{3.23}$$

де $i = 1,2,\dots,N_{v_z^{f_{m_{W,h}}}}$; $N_{v_z^{f_{m_{W,h}}}}$- кількість функцій-підмножин у векторі $v_z^{f_{m_{W,h}}}$.

Тоді, для кожного вектору $v_z^{f_{m_{W,h}}}$ (формула (3.22)) отримуємо за формулою (3.23) значення ваги всіх функцій-підмножин, які в нього входять як його компоненти. Обчислюючи ці ваги, сформуємо матрицю ваг E згідно формул (3.22) і (3.23), елементи якої визначимо так:

$$E_{v_z^{f_{m_{W,h}}}} = \left(F_{v_z^{f_{m_{W,h}}}}(v_{z,1}^{f_{m_{W,h}}}), F_{v_z^{f_{m_{W,h}}}}(v_{z,2}^{f_{m_{W,h}}}), \dots, F_{v_z^{f_{m_{W,h}}}}(v_{z,N_{v_z^{f_{m_{W,h}}}}}^{f_{m_{W,h}}}) \right), \quad (3.24)$$

де $v_z^{f_{m_{W,h}}}$ - вектор для h-тої функції $f_{m_{W,h}}$ з множини $F_{M_W,1}$; $h = 1,2,\dots,N_W$; N_W – кількість функцій-множин; $N_{v_z^{f_{m_{W,h}}}}$ - кількість функцій-підмножин у векторі $v_z^{f_{m_{W,h}}}$; z - тий вектор $v_z^{f_{m_{W,h}}}$ для h-тої функції $f_{m_{W,h}}$; $z = 1,2,\dots,N_{z,h}$; $N_{z,h}$ - кількість векторів $v_z^{f_{m_{W,h}}}$ для h-тої функції $f_{m_{W,h}}$; $i = 1,2,\dots,N_{v_z^{f_{m_{W,h}}}}$; $N_{v_z^{f_{m_{W,h}}}}$ - кількість функцій-підмножин у векторі $v_z^{f_{m_{W,h}}}$.

Тоді, ваги певного вектору за координатами значень із функцій-підмножин будуть визначені так:

$$W_{v_z^{f_{m_{W,h}}}} = \sum_{i=1}^{N_{v_z^{f_{m_{W,h}}}}} F_{v_z^{f_{m_{W,h}}}}(v_{z,i}^{f_{m_{W,h}}}), \quad (3.25)$$

де $i = 1,2,\dots,N_{v_z^{f_{m_{W,h}}}}$; $N_{v_z^{f_{m_{W,h}}}}$ - кількість функцій-підмножин у векторі $v_z^{f_{m_{W,h}}}$.

Графічне подання виразу з формули (3.25) зображено на рис. 3.3.

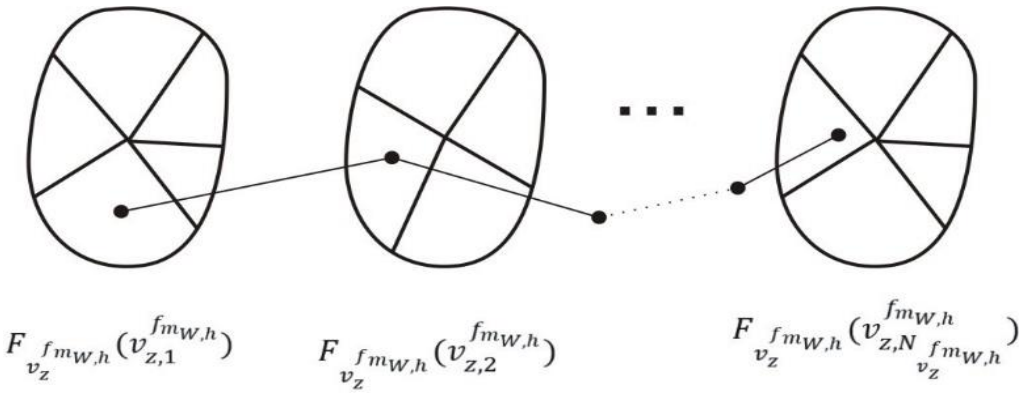


Рисунок 3.3 – Вага вектору $W_{v_z^{f_{m_{W,h}}}}$ за координатами

Повторюваність викликів функцій-підмножин, для яких задані ваги через координати вектору формулою (3.23), буде збільшувати ймовірність саме

зловмисного прояву порівняно з одиничними викликами. Тому, вага для такого вектору буде збільшуватись за рахунок введення коефіцієнтів повторюваності функцій-підмножин. Тоді, визначення значення ваги певного вектору з формули (3.25) модифікуємо так:

$$W_{v_z^{f_{m_{W,h}}}} = \sum_{i=1}^{N_{v_z}^{f_{m_{W,h}}}} (\alpha_{z,i}^{f_{m_{W,h}}} F_{v_z^{f_{m_{W,h}}}}(v_{z,i}^{f_{m_{W,h}}})); \quad (3.26)$$

$$\text{якщо } v_{z,i}^{f_{m_{W,h}}} = v_{z,i+1}^{f_{m_{W,h}}}, \text{ то } \alpha_{z,i}^{f_{m_{W,h}}} = 2\alpha_{z,i}^{f_{m_{W,h}}},$$

де $\alpha_{z,i}^{f_{m_{W,h}}}$ - ваговий коефіцієнт; $\alpha_{z,i}^{f_{m_{W,h}}} = 1$.

Також, можна значно збільшити значення вагового коефіцієнту $\alpha_{z,i}^{f_{m_{W,h}}}$ для тих координат вектору, які будуть повторюватись більше, ніж два рази.

В послідовності координат вектору $v_z^{f_{m_{W,h}}}$ (формула (3.22)) можуть повторюватись послідовності по два, три і більше координат. Для такого випадку застосуємо метод N-грам і визначимо вагу з врахуванням таких повторів послідовностей:

$$W_{v_z^{f_{m_{W,h}}}} = \sum_{i=1}^{N_{v_z}^{f_{m_{W,h}}}} (\alpha_{z,i}^{f_{m_{W,h}}} F_{v_z^{f_{m_{W,h}}}}(v_{z,i}^{f_{m_{W,h}}})); \quad (3.27)$$

якщо $v_{z,i}^{f_{m_{W,h}}} = v_{z,p}^{f_{m_{W,h}}}; v_{z,i}^{f_{m_{W,h}}} = v_{z,p+1}^{f_{m_{W,h}}}$, то $\alpha_{z,i+1}^{f_{m_{W,h}}} = 2\alpha_{z,i+1}^{f_{m_{W,h}}}; \alpha_{z,p+1}^{f_{m_{W,h}}} = 2\alpha_{z,p+1}^{f_{m_{W,h}}},$

де $\alpha_{z,i}^{f_{m_{W,h}}}, \alpha_{z,p+1}^{f_{m_{W,h}}}$ - вагові коефіцієнти; $\alpha_{z,i}^{f_{m_{W,h}}} = 1; \alpha_{z,p+1}^{f_{m_{W,h}}} = 1$.

Тобто, не тільки для 2-грам, але і для більшої кількості коефіцієнти, починаючи з третього повторення будуть збільшуватись згідно формули (3.27). Хоча, за потреби їх можна зменшувати чи збільшувати.

Таким чином, отримуємо, крім показників фактичних викликів функцій, також, числові значення, що характеризують послідовності викликів від функціонуючих процесів.

В основу методу виявлення worm-вірусів, крім значень показників безпеки компонент, що описують оточуюче середовище в корпоративній мережі, та значення, які визначенні за формулами (3.22)-(3.27), візьмемо поділ їх на класи за характеристиками, які визначені елементами множини M_W, згідно стратегії, яку

задано для першого характеристичного показника, що задано формулою (3.21). Аналогічно, решта показників можуть бути використані для побудови класів за ними. Тоді, можна кожен показник досліджувати і відносити досліджуваний об’єкт до певного класу. Після опрацювання всіх показників буде сформовано для досліджуваного об’єкту його задання в багатьох класах за різними показниками. Тобто, буде здійснено відносно нього багатокласову класифікацію для багатьох характеристичних показників окремо. Аналогічно, до визначення значення ваг за формулами (3.26) і (3.27) здійснюємо призначення ваг певним класам за певним одним показником і для об’єкту його вагу. Цю вагу використаємо для його порівняння з вагами інших об’єктів. Якщо об’єкт віднесено до більше, ніж одного класу за певним характеристичним показником, тоді вагу обчислюємо з врахуванням його присутності в усіх класах.

Метод виявлення worm-вірусів згідно класифікації за типовими характеристиками враховує поділ на класи та їх ознакове поле, яке включає поведінкові сигнатури worm-вірусів, аналітичні вирази характеристик згідно поведінкових сигнатур, шаблони атак та відбитки, які можуть бути отримані з приманок для worm-вірусів, значення ваг досліджуваних процесів, значення ваг багатокласової класифікації для багатьох характеристичних показників, а також зміни в оточуючому середовищі, тобто в корпоративній мережі. Метод імплементовано в архітектуру частково централізованих розподілених систем. Тому, в ньому передбачено опрацювання, також, стану функційної та кібербезпеки в корпоративній мережі, які визначені аналітичними виразами при описі оточуючого середовища.

Суть та основні кроки методу:

1) отримання інформації з сенсору щодо успішної / неуспішної спроби ззовні завантажити файл в оперативний запам’ятовуючий пристрій та створення і запуск процесу;

2) збір інформації щодо функціонування процесу з п. 1);

3) оновлення інформації щодо поточного стану частково централізованої розподіленої системи;

4) формування сигнатури процесу з функцій-підмножин множини $F_{M_W,2}$;

5) формування вектору для виконуваних в процесі функцій-підмножин за формулою (3.22);

6) формування шаблону атаки, якщо вона відбувається;

7) аналіз вмісту приманок та формування відбитку-шаблону, як результату;

8) формування пакету відомостей про процес в комп’ютерній станції в корпоративній мережі;

9) виконання кроку 7 (оцінювання результатів розподілених обчислень в компонентах) методу організації функціонування частково централізованих розподілених систем;

10) виконання кроку 8 (визначення компонент, в яких буде виконуватись поставлене системою завдання) методу організації функціонування частково централізованих розподілених систем;

11) класифікація процесу до класів worm-вірусів або до класу підозрілих процесів за різними характеристичними показниками і, відповідно, до багатьох класів;

12) налаштування коефіцієнтів, визначення співвіднесеності важливості функцій-підмножин між собою, обчислення значень ваг за формулами (3.23)-(3.27);

13) встановлення факту щодо наявності worm-вірусу;

14) виконання кроку 9 (перебудова архітектури системи за наявності критичних подій) методу організації функціонування частково централізованих розподілених систем.

Таким чином, розроблено метод виявлення worm-вірусів, суть якого у здійсненні поділу їх на класи за спільними ознаками і визначеними критеріями за багатьма класами та з врахуванням імплементації його в архітектуру частково централізованих розподілених систем для залучення компонент системи до процесу прийняття рішення щодо віднесення worm-вірусу до певного класу.

3.4. Висновки до третього розділу

Забезпечення організації функціонування частково централізованих розподілених систем виявлення ЗПЗ в комп’ютерних мережах реалізовано згідно трьох розроблених методів.

Розроблений метод синтезу математичних моделей рівнів безпеки компонентів системи дає змогу отримувати нові аналітичні вирази для комплексного опису об’єктів та процесів, які відбуватимуться в частково централізованих розподілених системах і відноситимуться до оцінювання безпеки компонент системи. Він може бути застосований для дискретних та неперервних величин характеристичних показників. Отримані згідно них значення характеристичних показників рівнів безпеки в компонентах системи будуть використані для оцінювання результатів розподілених обчислень, отриманих з різних компонентів системи, з метою визначення ступеня довіри до них.

Метод організації функціонування частково централізованих розподілених систем дає змогу створювати такі системи. В ньому для функціонування такого типу систем проведено розподіл компонент за відношенням до центру прийняття рішень, що дало змогу реалізувати часткову централізацію сумісно з принципами самоорганізації та адаптивності, які задають механізми до самостійного прийняття рішень щодо подальших кроків системою та перебудови її архітектури за потреби.

Розроблено метод виявлення worm-вірусів за багатокласовою класифікацією із здійсненням поділу їх на класи за спільними ознаками і визначеними критеріями за багатьма класами та з врахуванням імплементації його в архітектуру частково централізованих розподілених систем для залучення компонент системи до прийняття рішення щодо віднесення worm-вірусу до певного класу та врахування результатів виявлення рештою компонент.

Таким чином, частково централізовані розподілені системи виявлення ЗПЗ можна створювати з використанням трьох розроблених методів та наповнювати їх спеціалізованим функціоналом.

Основні наукові результати третього розділу опубліковані в [38, 47, 71, 109, 111, 112, 114-116]..

РОЗДІЛ 4.
МЕТОДИКА ВИЗНАЧЕННЯ ЕФЕКТИВНОСТІ ТА ЕКСПЕРИМЕНТИ З ЧАСТКОВО ЦЕНТРАЛІЗОВАНОЮ РОЗПОДІЛЕНОЮ СИСТЕМОЮ ВИЯВЛЕННЯ ЗЛОВМИСНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В КОМП'ЮТЕРНИХ МЕРЕЖАХ

4.1. Методика визначення ефективності функціонування частково централізованих розподілених систем

Для дослідження ефективності функціонування частково централізованих розподілених систем виявлення ЗПЗ в комп'ютерних мережах розглянемо її суттєві характеристики в контексті вимог до таких систем. Подамо ефективність системи S за трьома згрупованими характеристиками. До першої групи віднесемо ускладнення розуміння таких систем зловмисниками, прийняття самостійних рішень щодо подальших кроків та здійснення гнучкої перебудови архітектури при зміні середовища функціонування. Тобто, перша група характеристик буде сформована щодо аналізу результатів, які планувалось досягнути синтезуючи розподілену систему згідно принципів часткової централізації, самоорганізації та адаптивності. До другої групи віднесемо характеристики щодо розподілених систем в контексті дослідження і аналізу властивостей саме систем, як стійкість та деградація. Тобто, наскільки такі системи здатні функціонувати тривалий час та виконувати поставлені завдання. До третьої групи віднесемо показники достовірності виявлення різних типів ЗПЗ різними методами, які імплементовані в архітектуру системи S і будуть відображати ефективність таких систем в частині достовірності виявлення ЗПЗ. Таким чином, ефективність функціонування частково централізованих розподілених систем виявлення ЗПЗ будемо розглядати як комплексну характеристику і задамо її так:

$$E = \sum_{i=1}^{3} \rho_i \cdot E_i, \tag{4.1}$$

де E_i – значення ефективності i-тої групи показників; ρ_i. - унормоване значення ваги показника; $\sum_{i=1}^{3} \rho_i = 1$; $i = 1, 2, 3$.

Показник першої групи E_1 розділимо на три складових частини так: E_{11} - ускладнення розуміння таких систем зловмисниками; E_{12} - прийняття самостійних

рішень щодо подальших кроків; E_{13} - здійснення гнучкої перебудови архітектури при зміні середовища функціонування.

Визначимо показник E_{11} з врахуванням порівняння кількості варіантів центру прийняття рішень для різної архітектури систем включно з варіантами формування динамічних компонент системи так:

$$E_{11} = 1 - \frac{1}{k_{E_{11}}}, \tag{4.2}$$

де $k_{E_{11}}$ - кількість варіантів центру прийняття рішень для різної архітектури систем включно з варіантами формування динамічних компонент системи.

Значення E_{11} належить проміжку $[0; 1]$ та із збільшенням кількості компонент буде прямувати до одиниці, оскільки кількість можливих варіантів зростатиме.

Визначимо показник E_{12} з врахуванням кількості самостійно прийнятих рішень центром прийняття рішень до кількості рішень порівняно з усіма рішеннями, які потрібно було прийняти, так:

$$E_{12} = \frac{k_1^{E_{12}}}{k_1^{E_{12}} + k_2^{E_{12}} + k_3^{E_{12}}}, \tag{4.3}$$

де $k_1^{E_{12}}$ - кількість самостійно прийнятих рішень центром системи; $k_2^{E_{12}}$ - кількість прийнятих рішень адміністратором; $k_3^{E_{12}}$ - кількість рішень, які не було прийнято, але які зафіксовані в системі і потребували розгляду.

Фіксування всіх випадків відбувається в системі. Значення E_{12} належить проміжку $[0; 1]$ та із зростанням кількості рішень, які прийнято самостійно центром прийняття рішень ефективність системи зростає, тобто значення E_{12} прямує до одиниці.

Визначимо показник E_{13} з врахуванням кількості здійснених перебудов архітектури системи відносно до кількості визначених перебудов центром прийняття рішень системи так:

$$E_{13} = \frac{k_1^{E_{13}}}{k_1^{E_{13}} + k_2^{E_{13}}}, \tag{4.4}$$

де $k_1^{E_{13}}$ - кількість здійснених перебудов архітектури системи; $k_2^{E_{13}}$ - кількість визначених перебудов центром прийняття рішень системи, які не відбулись.

Тоді, визначимо показник ефективності функціонування для першої групи так:

$$E_1 = \frac{1}{3} \cdot \sum_{i=1}^{3} E_{1i}. \quad (4.5)$$

Визначення ефективності функціонування розподілених систем для другої групи характеристик стосується стійкості та деградації таких систем.

Розглянемо визначення ступеня стійкості системи S в процесі її функціонування з врахуванням специфіки виконуваних нею завдань. Стійкість системи S будемо досліджувати в контексті її можливості продовжувати своє функціонування і виконання поставлених завдань в умовах змін в середовищі функціонування, які зумовлені внутрішніми процесами самої системи та зовнішніми процесами, що можуть бути викликані різними причинами, зокрема ЗПЗ, з мінімальною зміною чи втратою її функційності. Стани системи S, в яких вона буде функціонувати за відсутності впливів на неї ззовні і впливів, що будуть пов'язані з надійністю функціонування комп'ютерних станцій в корпоративній мережі, в які встановлені компоненти системи і які впливатимуть на внутрішні процеси в ній, включаючи і встановлення надійного зв'язку між ним, віднесемо до стану рівноваги всієї системи. Решту станів системи S віднесемо до нестійкого стану системи. Серед станів рівноваги виділимо стани часткової рівноваги, до яких віднесемо ті, в яких система S активуватиме методи виявлення ЗПЗ в комп'ютерній мережі. Тоді, фактично система може бути в трьох станах, переходи між якими і стани можна задати повним графом. Стан рівноваги системи зумовлений відсутністю впливів на неї збурюючих чинників. Стан часткової рівноваги зумовлений відсутністю впливів на неї збурюючих чинників і, при цьому, активізацією підсистеми для виявлення ЗПЗ. Стійкість системи S будемо характеризувати її здатністю повертатись до стану рівноваги після завершення перебування в стані часткової рівноваги або в нестійкому стані. Систему, в якій внаслідок впливу чинників відбувається віддалення від стану рівноваги або стану часткової рівноваги і, при цьому, вона тривалий час перебуває в нестійкому стані та не може перейти до інших станів, вважатимемо нестійкою.

Розглянемо умови стійкості системи S. Для кожного початкового значення, яке буде оброблятись, система повинна формувати результат, що не буде залишати її або в стані рівноваги або в стані часткової рівноваги. Якщо ж протягом певного часу система S не отримує вхідних значень, тоді вона не формує жодних рішень. Так задані умови стійкості системи збіжні із заданими умовами її функціонування в

комп’ютерних мережах згідно методу організації функціонування частково централізованих розподілених систем. Тоді, такі умови будемо вважати такими, що відповідають внутрішнім принципам функціонування системи і їх дотримання та аналіз можуть бути основою для дослідження стійкості системи в частині її стабільного функціонування. Показник стабільності буде встановлено для конкретної характеристики впливу. Для системи *S* характерною буде динамічна стійкість, яка відображає здатність до відновлення початкового стану після впливу чинників. Система *S* через наявність різних станів компонентів буде мати велику кількість варіантів в компонентах, тому розглядатимемо її як нелінійну динамічну систему.

Систему *S* будемо розглядати як самоорганізовану дискретну систему, оскільки вона перебуватиме в станах в залежності від станів її компонент. Перехід між станами з врахуванням часу буде характеризуватись точками розриву першого роду. Наприклад, зобразимо на рис. 4.1 графік залежності станів компоненти та певного проміжку часу її функціонування.

На графіку (рис. 4.1) стани позначено вершинами, а переходи між ними дугами. Переходи між вершинами 1 і 2, 2 і 3, 4 і 5, 6 і 7, 8 і 9, 10 і 11 є фактично зображенням перебування компоненти в поточному стані протягом певного часу, а саме зображення перебування в тому самому стані тільки їх виокремлення у вершини пов’язане з періодичним за часом обліком стану компоненти, що відображено часовою шкалою. Точки розриву першого роду на графіку зображено переходами між вершинами 3 і 4, 5 і 6, 7 і 8, 9 і 10. Зобразимо на рис. 4.2 графік функції з врахуванням точок розриву часовою діаграмою.

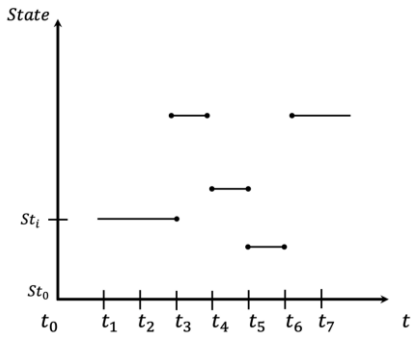


Рисунок 4.1 - Графік залежності станів компоненти від часу

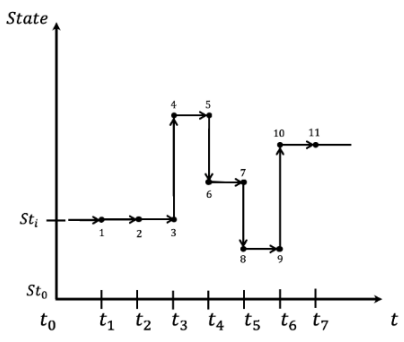


Рисунок 4.2 - Графік часової діаграми компоненти, що перебуває в певних станах

Отже, наявність точок розриву протягом певного часу функціонування компоненти системи підтверджує її дискретність. І, оскільки, ця дискретність викликана не тільки періодами часу, а саме перебуванням компонент в різних станах в залежності від часу, які є елементами множини і їх не можна задати неперервно, то система S буде дискретною за рівнем і за часом. За часом система буде дискретна, бо її функціонування в комп'ютерних станціях відбуватиметься на рівні виконання процесів. Зобразимо, наприклад, фрагмент функціонування системи через квантування за рівнем і часом на рис. 4.3.

Таким чином, система S є дискретною і для визначення рівня її стабільності розглянемо значення характеристичних показників, які дають змогу отримувати значення стану компонентів. Ці значення є аргументами функцій в формулах (2.18), (2.50), (2.59) і для обчислення їх значень визначено відповідні аналітичні вирази. Для наочного аналізу використання значень характеристичних показників при дослідженні стійкості системи S розглянемо спочатку два показники, а в подальшому отриманий результат масштабуємо до всіх 15 (16), для яких визначені аналітичні вирази.

Використаємо для дослідження стабільності системи S узагальнені характеристичні показники значень рівнів безпеки компонентів, які задано множиною $B = \{\beta'_1, \beta'_2, \dots, \beta'_{N_B}\}$, де β'_i - значення рівнів безпеки компонентів системи, N_B – кількість характеристичних показників, $i = 1,2, \dots, N_B$. Для кожного компонента системи S введемо підмножини $B_j = \{\beta'_{1,j}, \beta'_{2,j}, \dots, \beta'_{N_B,j}\}$ згідно заданої множини B, елементи якої будуть використані для обчислення значення рівня безпеки всієї системи. Значення $\beta'_{i,j}$ $(i = 1,2, \dots, N_B)$ визначатимуть рівень довіри до результатів розподілених обчислень, які здійснені в різних компонентах системи та характеризують різні показники рівнів безпеки. Введемо для значень $\beta'_{i,j}$ $(i = 1,2, \dots, N_B;\ j = 1,2, \dots, N;\ N$ – кількість компонент в системі, які встановлені в комп'ютерні станції в мережі) проміжок, в якому буде регулюватись нижня межа в залежності від параметру рівня значущості $\alpha_z^{r,i}$ $(i = 1,2, \dots, N_B;\ z = 1,2, \dots, N_z;\ N_z$ – кількість варіантів взаємодії функцій-підмножин) так: $[1 - \alpha_z^{r,i}; 1]$. За рівень значущості приймемо частку від одиниці, яка відображатиме відхилення від рівня довіри до результату розподілених обчислень внаслідок певних подій, архітектурної

особливості компоненти тощо. Тоді, якщо розглянути два характеристичних показники однієї компоненти, наприклад $\beta'_{1,j}, \beta'_{2,j}$ для j – ої компоненти, то результати обчислень можна відобразити на координантій площині двома точками. Якщо компонента системи функціонує стабільно, то значення точок міститимуться в прямокутнику, що задаватиметься по осі абсцис відрізком $\left[1-\alpha_z^{r,1};1\right]$ і по осі ординат відрізком $\left[1-\alpha_z^{r,2};1\right]$. Зображення прямокутника подано на рис. 4.4.

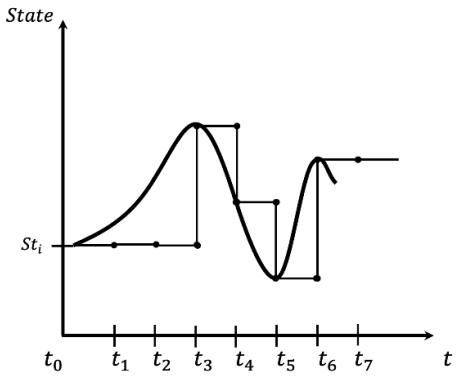


Рисунок 4.3 - Графік квантування за рівнем і часом

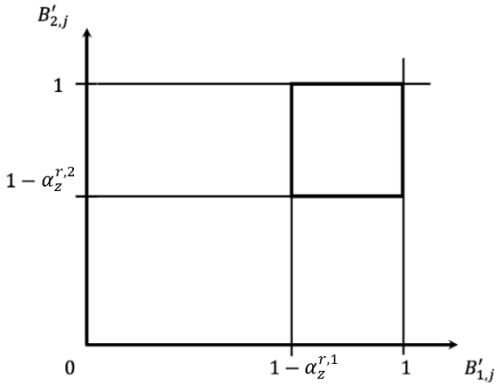


Рисунок 4.4 - Область значень двох характеристичних показників

Ці ж значення характеристичних показників можуть бути задані на графіку залежності їх значень від часу. Наприклад, на рис. 4.5 задано два графіки для двох значень характеристичних показників.

Результати задані на графіках при обчисленні їх середніх значень можуть відповідати тим же характеристикам, що і для станів компонент, як вже зображено на рис. 4.1-4.3. Тобто, таке подання значень характеристичних показників підтверджує дискретність системи і дає змогу визначити стабільність системи від значень характеристичних показників.

Задамо функціонування системи S її структурною схемою на рис. 4.6.

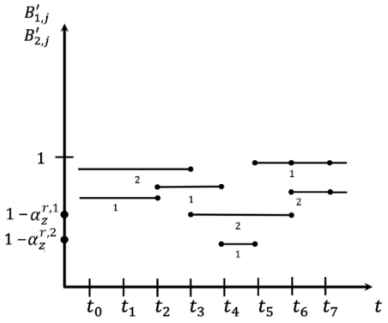


Рисунок 4.5 - Графіки залежності двох значень характеристичних показників від часу

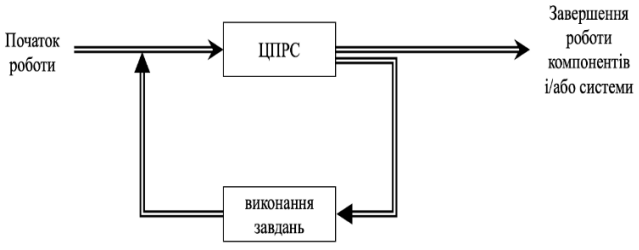


Рисунок 4.6 - Структурна схема системи S

На рис. 4.7 зображено уточнену структурну схему системи S. Позначення елементів матриці B означає множину характеристичних показників значень рівнів безпеки компонентів, яка постійно оновлюється через певні інтервали часу. Позначення ЦПРС на рис. 4.6 і рис. 4.7 означає центр прийняття рішень системи S. Визначення елементів множини B здійснюється лінійно, тому в цьому блоці ця частина може бути виражена лінійними функціями, але друга частина блоку ЦПРС є нелінійною. Другий блок відображає функціонування системи для виконання спеціалізованих завдань і є нелінійним. Задамо елементи множини B координатами вектору. В результаті отримаємо простір стану з різними векторами і їх значеннями.

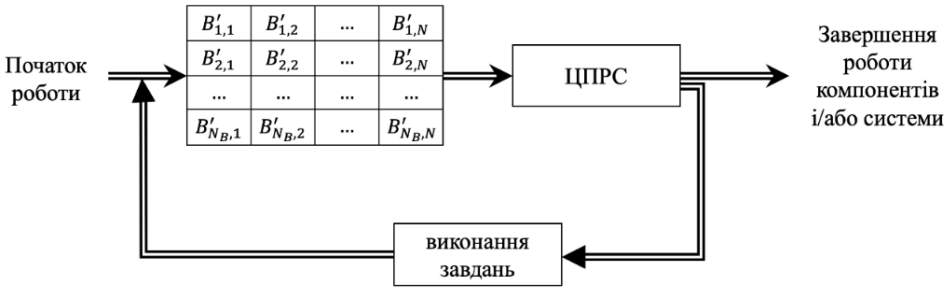


Рисунок 4.7 - Уточнена структурна схема системи S

Задамо функцію $W_{S,c}^1$ для опису блоку центру прийняття рішень системи так:

$$W_{S,c}^1 = \sqrt{\sum_{i=1}^{N_B} \sum_{j=1}^{N} {\beta'_{i,j}}^2}, \tag{4.6}$$

де N_B – кількість характеристичних показників, $i = 1,2,\dots,N_B$; $\beta'_{i,j}$ $(i = 1,2,\dots,N_B)$ значення, які визначатимуть рівень довіри до результатів розподілених обчислень, що здійснені в різних компонентах системи та характеризують різні показники рівнів безпеки; $j = 1,2,\dots,N$; N – кількість компонент в системі, які встановлені в комп'ютерні станції в мережі.

Значення функції $W_{S,c}^1$ буде відрізком, довжина якого не перевищуватиме значення $\sqrt{N_B \cdot N}$, і характеризуватиме стан системи коли увімкнені всі комп'ютерні станції та активні в них всі компоненти системи S. Нижньою межею значення функції $W_{S,c}^1$ буде $\sqrt{N \cdot \sum_{i=1}^{N_B} \left(1 - \alpha_z^{r,i}\right)^2}$. Геометричною інтерпретацією функції $W_{S,c}^1$ буде точка в $N_B \cdot N$ – просторі з кількість координат $N_B \cdot N$. Тому, стабільність системи S буде залежати від значення функції $W_{S,c}^1$. Якщо значення буде

перевищуватиме значення $\sqrt{N_B \cdot N}$, то система перейде до стану нерівноваги і буде вилучати компоненти зі своєї архітектури, в яких найбільший вплив на значення функції $W_{S,c}^1$. Після таких вилучень компонент, система S повернеться до стану рівноваги і пробуватиме знову поетапно доповнювати компоненти. Якщо значення частини компонент дорівнюватимуть нулеві через їх відсутність в системі (вимкнені комп'ютерні станції), то значення функції $W_{S,c}^1$ обчислюється для наявних компонент і, тоді, точка задається в просторі меншого розміру, ніж $N_B \cdot N$. При цьому обчислене значення функції $W_{S,c}^1$ буде знаходитись так само в тому ж проміжку. Якщо значення функції $W_{S,c}^1$ буде менше за число $\sqrt{N \cdot \sum_{i=1}^{N_B}\left(1-\alpha_z^{r,i}\right)^2}$, тоді система теж перейде до стану нерівноваги, але цей стан буде викликано значеннями характеристичних показників станів безпеки компонент, при обчисленні яких було отримано значення, які не належать хоча б одному з проміжків $\left[1-\alpha_z^{r,i};1\right]$, але яке настільки менше $1-\alpha_z^{r,i}$, що вплинуло на загальний результуючий показник, тому цю або ці компоненти система вилучає. Але може бути так, що це значення не вплине на загальний показник, хоча воно буде менше заданого значення, тоді система буде залишатись в стані рівноваги і вирішуватиме питання щодо аналізу значення з компоненти та за потреби її вилучення з системи.

Ступінь стійкості системи S в процесі її функціонування, враховуючи специфіку виконуваних завдань будемо визначати коефіцієнтом $k_{W_{S,c}^1}$ згідно значення функції $W_{S,c}^1$, визначеного за формулою (4.6), так:

$$k_{W_{S,c}^1} = \frac{\sqrt{\sum_{i=1}^{N_B}\sum_{j=1}^{N}\beta_{i,j}'^{\,2}}}{\sqrt{N_B \cdot N}}. \qquad (4.7)$$

Тоді, система S при значенні $\frac{\sqrt{\sum_{i=1}^{N_B}\left(1-\alpha_z^{r,i}\right)^2}}{\sqrt{N_B}} \le k_{W_{S,c}^1} \le 1$, коли значення всіх $\alpha_z^{r,i}$ є найбільш допустимими, буде перебувати в стані рівноваги і значення $k_{W_{S,c}^1}$ з цього проміжку буде критерієм стабільності для цієї системи. Коли частина компонентів системи S не буде активна через вимкнені комп'ютерні станції, то це теж впливатиме на стабільність її роботи і, відповідно, значення коефіцієнту буде меншим, бо враховуватиме потребу в усіх компонентах системи.

Ступінь деградації системи S в процесі її функціонування та деградації її компонентів будемо розглядати в контексті втрати частини компонентів системою і, як наслідок, або виведення частини компонентів безповоротно зі складу системи, або зниження працездатності системи через втрату частини компонентів, або неправильне їх функціонування.

Ступінь деградації системи корелює зі ступенем стійкості. Але стійкість системи відображає можливості продовжувати своє функціонування і виконання поставлених завдань в умовах змін в середовищі функціонування з мінімальною зміною чи втратою її функційності, а деградація – можливості виконувати поставлені завдання після повної або часткової втрати функційності компонент та наближення чи перехід, як до стану відмови, так і до стану повної зупинки. Таким чином, спільним для обох характеристик системи є можливість продовження виконання поставлених завдань. Відмінним в цьому є для стійкості – ймовірність продовження роботи, а для деградації – ймовірність наближення до стану відмови.

Ступінь деградації системи в цілому буде залежати від кількості компонент в системі, часу функціонування системи та її компонент, а також від подій, що буде опрацьовувати система і впливу на середовище, в якому буде функціонувати система і її компоненти. Врахуємо при визначенні коефіцієнту ступеня деградації системи кількість компонент в системі, час функціонування системи та компонент і значення рівнів безпеки компонент та системи в поточний момент часу. Тоді, задамо коефіцієнт деградації системи так:

$$k_{S,t}^{d} = 1 - \left(\frac{n}{k}\right)^{-\frac{\frac{\sum_{r=1}^{i}\alpha'_{1,S_i}}{k_1} + \frac{\sum_{r=i+1}^{n}\alpha'_{2,S_{k+1,n}}}{k_2}}{\alpha_{S,t}^{st,1}} \cdot t}, \qquad (4.8)$$

де $\alpha_{S,t}^{st,1}$ – значення рівня безпеки системи, яке обчислюється за формулою (3.12); k – кількість активних компонент в системі; k_1 – кількість активних компонент з центром прийняття рішень в системі; k_2 – кількість активних компонент без центру прийняття рішень в системі; α'_{1,S_i}, $\alpha'_{2,S_{k+1,n}}$ - значення рівнів безпеки компонент системи, які обчислюються за формулами (3.13), (3.14); $k = k_1 + k_2$; n – кількість компонент в системі; t – час функціонування системи.

При великій кількості компонент в системі втрата частини з них не буде суттєво впливати на ступінь деградації системи, бо архітектура системи на рівні компонент

є централізованою і, тому, виконання завдань може бути направлено решті активних компонент. Якщо ж показник деградації стосується безпосередньо саме певної компоненти і вона перебуває на етапі вилучення самою системою, то це не впливає суттєво на ступінь деградації всієї системи.

Якщо всі компоненти системи функціонують в штатному режимі, тоді $k = n$ і коефіцієнт деградації $k_{S,t}^{d} = 0$. Якщо $k < n$, тоді значення коефіцієнту деградації буде відмінним від нуля та буде вказувати на ступінь деградації системи.

Таким чином, визначено основні фактори, що впливають на систему, та згідно них розроблено аналітичні вирази для обчислення ступенів стійкості та деградації системи S в процесі її функціонування з врахуванням специфіки виконуваних завдань. Показник ефективності для другої групи визначимо згідно значень коефіцієнтів стійкості та деградації з формул (4.7) та (4.8) так:

$$E_2 = \frac{1}{2} \cdot \left(k_{W_{S,c}^{1}} - k_{S,t}^{d} + 1 \right). \tag{4.9}$$

Для третьої групи визначимо показник E_3 з врахуванням достовірності виявлення ЗПЗ різними методами, які імплементовані в архітектуру системи так:

$$E_3 = \frac{1}{k_1^{E_3}} \sum_{i=1}^{k_1^{E_3}} D_i^{E_3}, \tag{4.10}$$

де $D_i^{E_3}$ – достовірність виявлення ЗПЗ i – того типу i – тим методом; $i = 1, 2, \dots, k_1^{E_3}$; $k_1^{E_3}$ – кількість типів ЗПЗ та методів їх виявлення.

В формулі (4.10) прийнято, що для певного типу ЗПЗ застосовується один метод виявлення, який імплементовано в архітектурі системи S. Допускається, що цей метод може в певних випадках забезпечувати виявлення іншого типу ЗПЗ при збіжності певних характерних ознак із тим типом ЗПЗ, для якого він використовується. Також, для певного типу ЗПЗ може бути розроблено декілька різних методів виявлення, але приймемо в формулі (4.10), що в такому випадку їх імплементація в архітектуру системи S здійснюється як одного методу з розширеними варіантами. Крім того, деякі з розроблених методів можуть бути призначені для виявлення декількох типів ЗПЗ, тоді їх застосування будемо розглядати окремо до кожного з поділених типів ЗПЗ. Оскільки для проведення експериментальних досліджень з метою виявлення ЗПЗ в системі S може бути реалізовано один або декілька методів, тобто для одного або декількох типів ЗПЗ, то

при визначенні показника ефективності за формулою (4.1) унормоване значення ваги показника ρ_3 може бути однаковим з показниками ρ_1та ρ_2 або кратно меншим відповідно до кількості типів ЗПЗ, що буде відображати ефективність функціонування системи в частині саме достовірності виявлення порівняно з існуючими системами.

Таким чином, розроблена методика визначення ефективності частково централізованих розподілених систем виявлення ЗПЗ враховує її особливості з ускладнення розуміння таких систем зловмисниками, прийняття самостійних рішень щодо подальших кроків, здійснення гнучкої перебудови архітектури при зміні середовища функціонування, стійкості та деградації в процесі функціонування, а також виявлення ЗПЗ. Розроблена методика орієнтована на дослідження саме ефективності функціонування частково централізованих систем виявлення ЗПЗ, що є комплексним показником.

4.2. Постановка експериментів та результати експериментальних досліджень з частково централізованою системою виявлення зловмисного програмного забезпечення

Варіант реалізації частково централізованої розподіленої системи розроблено і її детальний опис подано в [107] та фрагментом програмного коду в Додатку В. З розробленою системою S було проведено експериментальні дослідження щодо ефективності функціонування таких систем.

Здійснимо постановку першого експерименту для обчислення показника першої групи E_1 (формула (4.5)). Спочатку проаналізуємо вирази для обчислення показника E_{11}. За формулами (2.13) та (2.14) обчислюємо значення k_{Q_1} кількості можливих варіантів розміщення центру прийняття рішень системи, тобто кількість компонент зі статичними функціями, а значення k_{Q_2} - кількість варіантів формування динамічних компонент системи. Також, для порівняння розглянемо застосування формули (2.13) для випадку централізованої архітектури з одним центром ($k = m = 1$), для децентралізованої архітектури ($k = m = n$), для частково централізованої архітектури з фіксованою кількістю компонент з функціоналом центру прийняття рішень ($k = m$). Порівняння кількості варіантів при різній

архітектурі розподілених систем подано в табл. 4.1.

Таблиця 4.1

Значення кількості варіантів для різної архітектури розподілених систем

№ з/п	Тип архітектури	Кількість компонент	Формула	Значення кількості варіантів
1	Централізована (один центр)	$k = m = 1$	$\sum_{m=1}^{1} \frac{n!}{1! \cdot (n-1)!}$	n
2	Децентралізована	$k = m = n$	$\sum_{m=n}^{n} \frac{n!}{n! \cdot (n-n)!}$	1
3	Часткова централізація (фіксована стала кількість компонент центру)	$k = m$	$\sum_{m=k}^{k} \frac{n!}{m! \cdot (n-m)!}$	$\frac{n!}{m! \cdot (n-m)!}$
4	Часткова централізація (статичні компоненти)	$2 \le m \le k < n$	$\sum_{m=2}^{k} \frac{n!}{m! \cdot (n-m)!}$	k_{Q_1}
5	Часткова централізація (динамічні компоненти)	$2 \le m \le k < n$	$\left(n_{S_k,max} + 1\right) \cdot \sum_{m=2}^{k} \frac{n!}{m! \cdot (n-m)!}$	$\left(n_{S_k,max} + 1\right) \cdot k_{Q_1}$

Позначення: n-кількість всіх компонент; k-кількість компонент з центром; m – кількість активних компонент з центром; k_{Q_1} - кількість можливих варіантів розміщення центру прийняття рішень системи (формула (2.13)); $n_{S_k,max}$ – кількість функцій-підмножин.

Розглянемо проведення експерименту щодо показника E_{11}. Кількість комп'ютерних станцій, в які були встановлені компоненти розробленої системи S дорівнює п'ятдесят, тобто $n = 50$. Кількість компонент з функціоналом центру прийняття рішень $k = 25$, кількість функцій-підмножин $n_{S_k,max} = 7$. Розрахункові результати при умовах експерименту для всіх випадків з табл. 4.1 подано в табл. 4.2.

Експеримент було проведено протягом 240 год. Кожна подія при функціонуванні системи зберігалась у файл-логу з поділом на інтервали. Межами інтервалів були завершені дії з перебудови центру системи. Результати експериментальних досліджень подано в табл. Г.1.

Таблиця 4.2

Розрахункові результати для різної архітектури розподілених систем

№ з/п	Тип архітектури	n	Значення кількості варіантів	Значення показника E_{11}
1	Централізована (один центр)	50	50	0.98
		25	25	0,96
		10	10	0,9
		5	5	0,8
		4	4	0,75
2	Децентралізована	50	1	0
3	Часткова централізація (фіксована стала кількість компонент центру)	50	$\frac{50!}{25! \cdot 25!} =$ =126410606437752	0.999999999999992
4	Часткова централізація (статичні компоненти)	50	$\sum_{m=2}^{25} \frac{n!}{m! \cdot (n-m)!} =$ =626155256640137	0.999999999999998
5	Часткова централізація (динамічні компоненти)	50	$5.0092420531211 \cdot 10^{15}$	1.00000000000000

Розрахункові мінімальні значення кількості варіантів $k_{Q_1 min} = 1225$ та показника $E_{11,min}$= 0.999183673469388, максимальні значення кількості варіантів $k_{Q_1 max} = 126410606437752$ та показника $E_{11,max} = 0.999999999999992$ (у випадку часткової централізації із фіксованою кількістю компонент центру). Отримані значення в межах інтервалу з мінімально та максимально допустимих значень, тому результати експериментальних досліджень коректні. Середньоарифметичні значення $E_{11,s} = 0.999999913464257$ для статичних та значення $E_{11,s} = 0.999999913464257$ для динамічних компонент відображають ефективність функціонування системи S в частині встановлення кількості можливих варіантів розміщення центру прийняття рішень системи. Порівняння цих отриманих значень з розрахунковими підтверджує покращення ефективності функціонування розподілених систем в частині ускладнення розуміння таких систем зловмисниками в процесі пошуку компонент з центром за рахунок реалізації часткової централізації.

Протягом 240 год. функціонування системи S було отримано, також, значення

необхідні для обчислення показника E_{12}. Результати експериментальних досліджень подано в табл. Г.1 кількістю рішень, які зафіксовані системою. Значення показника E_{12} за результатами експерименту обчислено за формулою (4.3). Він відображає синтезований в архітектурі системи S принцип самоорганізації. Тоді, $E_{12} = \frac{837}{837+2+7} = 0.98936$, що підтверджує достатній рівень самоорганізації системи S.

Для аналізу синтезованого в архітектурі системи S принципу адаптивності при проведенні експерименту було досліджено і зафіксовано здійснення гнучкої перебудови архітектури при зміні середовища функціонування. Дані про кількість здійснених перебудов в частині переміщення центру між компонентами, яку зафіксовано системою, подано в табл. Г.1. Тоді, за формулою (4.4) отримуємо значення показника $E_{13} = \frac{423}{423+6} = 0.98601$.

Значення обчисленого за формулою (4.5) показника ефективності функціонування для першої групи $E_1 = 0,99179$. При цьому, ускладнення визначення компонент з центром системи для розробленої моделі систем в середньому становить 4-5% порівняно з відомими варіантами архітектури і фактично для варіанту з використанням динамічних компонент значення досягає 100%. Значення показників самоорганізації та адаптивності системи становить більше 98% і порівняно з системами, в яких ці принципи синтезовані окремо (або самоорганізація або адаптивність), має більше значення самостійно прийнятих рішень приблизно на 50%.

Таким чином, результати проведеного експерименту для першої групи показників підтверджують покращення ефективності функціонування системи S порівняно з іншими типами архітектури розподілених систем.

Розглянемо другу групу показників для обчислення показника E_2 (формула (4.9)). Здійснимо дослідження таких показників системи S: ступінь стійкості в процесі функціонування, враховуючи специфіку виконуваних завдань; ступінь деградації в процесі функціонування системи та її компонентів.

Для встановлення значення коефіцієнту стійкості при різних навантаженнях на систему і при різній архітектурі системи в частині кількості її компонент здійснимо постановку і проведення другого експерименту [47]. В ньому проведемо чотири серії експериментальних досліджень з розробленою системою. Данні, які отримані

в певний момент часу функціонування системи S, були зафіксовані за таких умов: архітектура системи була сформована зі всіх 100 компонентів; підсистеми, які забезпечують виявлення ЗПЗ, не активізувались в системі за відсутності таких проявів. Тобто, за таких початкових встановлено, що система повинна функціонувати стабільно. Результати подано в п’ятнадцяти таблицях в Додатку Д (табл. Д.1-Д.15, Додаток Д). Для кожного характеристичного показника у відповідному рядку і стовпці подане його значення для конкретного компонента. Для проведення цього експерименту рівні значущості характеристичних показників було встановлено в залежності від їх важливості так:

1) $\alpha'_{1,S_i,1} = 0{,}01$, $\alpha'_{1,S_i,3} = 0{,}01$, $\alpha'_{2,S_{k+1,n},2} = 0{,}01$, $\alpha'_{2,S_{k+1,n},3} = 0{,}01$, $\alpha'_{2,S_{k+1,n},4} = 0{,}01$, $\alpha'_{2,S_{k+1,n},5} = 0{,}01$, $\alpha'_{3,S_{1,n},3} = 0{,}01$;

2) $\alpha'_{1,S_i,2} = 0{,}02$, $\alpha'_{3,S_{1,n},2} = 0{,}02$;

3) $\alpha'_{1,S_i,4} = 0{,}05$, $\alpha'_{1,S_i,5} = 0{,}05$, $\alpha'_{2,S_{k+1,n},1} = 0{,}05$, $\alpha'_{3,S_{1,n},1} = 0{,}05$, $\alpha'_{3,S_{1,n},4} = 0{,}05$, $\alpha'_{3,S_{1,n},5} = 0{,}05$.

За формулами (4.6) і (4.7) знаходимо значення функції $W^1_{S,c}$ та значення коефіцієнта стійкості системи $k_{W^1_{S,c}}$ і значення нижньої межі проміжку для коефіцієнту стійкості $\frac{\sqrt{\sum_{i=1}^{15}\left(1-\alpha_z^{r,i}\right)^2}}{\sqrt{15}}$. Отримані значення $W^1_{S,c} = 38.214301635550662$, $k_{W^1_{S,c}} = 0.98668902547623$ і числове значення нижньої межі, що дорівнює 0.972848052541266, підтверджують перебування системи в стабільному стані.

Для проведення другої серії експерименту вимкнемо 30 комп’ютерних станцій. Отримуємо так само 15 таблиць, але в кожній з них буде мінімум тридцять нульових значень. Здійснюємо обчислення значень $W^1_{S,c} = 31.96599695772904$, $k_{W^1_{S,c}} = 0.825358492414677$ і числового значення нижньої межі, яке дорівнює 0.813943077452799. Результати (табл. Д.16-Д.30, Додаток Д) підтверджують перебування системи в стабільному стані.

Третю серію експерименту проведемо при вимкнених 40 комп’ютерних станцій. Надамо додатково навантаження на певні показники і отримаємо сім числових значень більше одиниці. Проведено обчислення значень $W^1_{S,c} =$

29.58811566557844, $k_{W_{S,c}^1} = 0.763961861456294$ і числового значення нижньої межі, яке дорівнює 0.753564861176528. Особливість результатів (табл. Д.31-Д.45 Додаток Д) цієї серії експерименту відображає стабільність системи на нижчому рівні порівняно з попередніми випадками через втрату значної кількості компонент.

Аналогічно було проведено четверту серію експерименту, результати якої подано в табл. Д.46-Д.60 (Додаток Д). Значення $W_{S,c}^1 = 29.58811566557844$, $k_{W_{S,c}^1} = 0.987074336891810$. Значення нижньої межі дорівнює 0.972848052541266. Особливість результатів цієї серії експерименту в тому, що в частини компонент системи певні показники перевищили гранично допустиме значення, що дорівнює одиниці, але система при цьому зберегла стабільність на високому рівні.

Таким чином, розроблена розподілена система S згідно принципів часткової централізації, самоорганізації та адаптивності, які реалізовані в її архітектурі, в результаті проведених експериментальних досліджень з різною кількістю її компонентів та значеннями характеристичних показників перебувала в стабільному стані.

Проведемо постановку третього експерименту [38] для визначення ступеня деградації системи. Значення рівнів значущості характеристичних показників використаємо такі ж, як і для попереднього експерименту. Розглянемо систему S на рівні її компонент. Від кожної з компонент може бути отримано до центру прийняття рішень системи два показники: значення рівня безпеки компоненти; виконання поставленого завдання. Тоді, можливі чотири випадки:

1) значення рівня безпеки компоненти відповідає допустимому значенню і поставлене завдання виконано правильно;

2) значення рівня безпеки компоненти відповідає допустимому значенню, але поставлене завдання виконано неправильно або не виконано в заданий час;

3) значення рівня безпеки компоненти не відповідає допустимому значенню і поставлене завдання виконано неправильно або не виконано в заданий час;

4) значення рівня безпеки компоненти не відповідає допустимому значенню, але поставлене завдання виконано правильно.

В системі сформовано файл-лог результатів роботи напротязі певного часу з фіксуванням результатів щодо значень рівнів безпеки компонент та результатів виконання поставлених завдань. Після певного часу функціонування системи

опрацюємо результати із файлу-логу. Експеримент [38] було проведено десятьма серіями протягом десяти днів. Результати експерименту подано таблицями значень рівнів безпеки і результатів виконання поставленого завдання в Додатку Е. В результаті проведеного експерименту було проведено класифікацію компонент системи на предмет виконання ними поставлених завдань. Це дало змогу центру прийняття рішень системи визначити її стан через рівні безпеки компонент та результат виконання поставлених завдань. Кількість компонент, в яких рівень безпеки перебував в заданих межах і в них було правильно виконано поставлене завдання, є більшим 91%. Середньоарифметичне значення серед значень всіх десяти серій експерименту дорівнює 95,29%. Крім того, при певних серіях експерименту значення було близьким або дорівнювало 100%. Таким чином, середньоарифметичне значення ступеня деградації системи для проведених серій експериментів $k_{S,t}^{d} = 0{,}0471$. Це вказує на те, що так розроблена архітектура системи забезпечує низький рівень її деградації. Результати експерименту підтверджують можливість виконання поставленого завдання системою незалежно від компонент, в яких високий ступінь безпеки і від яких отримано негативний результат, та компонент з низьким рівнем безпеки, від яких теж отримано негативний результат. Переважна більшість компонент системи виконали поставлене завдання правильно і оцінили результати виконання від всіх компонент для видачі узгодженого рішення.

Показник ефективності функціонування для другої групи $E_2 = \frac{1}{2} \cdot (0.9871 - 0{,}0471 + 1) = 0{,}97$. Значення показника ефективності функціонування підтверджує цілісність системи S та, відповідно, можливість практичної реалізації розробленої архітектури частково централізованих розподілених систем.

Для визначення показника E_3 для третьої групи проведено четвертий експеримент [47] з системою S щодо достовірності виявлення ЗПЗ. Як об’єкти дослідження було розглянуто worm-віруси. Для проведення експериментальних досліджень спочатку здійснено конструювання п’яти класів worm-вірусів по чотири екземпляри. Для цього використано конструктивні елементи формування штучних worm-вірусів без зловмисного навантаження та з функціоналом, який повідомлятиме на екран про позитивний результат інфікування комп’ютерної станції і продовження розмноження в комп’ютерній мережі. При цьому на екран,

також, буде видаватись інформація про час завершення повної процедури інфікування комп'ютерної станції.

В усіх комп'ютерних станціях встановлено ОС Windows і всі комп'ютерні станції мають однакове конфігурування. Кількість комп'ютерних станцій, в які встановлено компоненти системи S, дорівнювало ста, а кількість комп'ютерних станцій, в яких не встановлено компоненти системи S, дорівнювало десяти. Кількість сегментів, на які поділено корпоративну мережу, дорівнювало п'яти. Корпоративна мережа містила два сервери.

Проведення експерименту [47] з системою S для перевірки достовірності виявлення worm-вірусів імплементованим в неї методом здійснено з врахуванням шести типів джерел їх поширення. Ці джерела було розглянуто в контексті шістьох можливих варіантів. Під час проведення експериментів корпоративна мережа та додаткові десять комп'ютерних станцій, які не належать їй, були від'єднані від мережі Internet. Але ці десять комп'ютерних станцій були під'єднані до корпоративної мережі, як частина вузлів глобальної мережі.

В контурі корпоративної мережі для першого варіанту вибрано довільним чином комп'ютерну станцію, яка не знаходилась в демілітаризованій зоні. Крім того, в цій комп'ютерній станції встановлено компоненту системи S. В цій комп'ютерній станції було активовано штучний worm-вірус.

З набору з 20 штучних worm-вірусів, з яких по чотири worm-віруси відносяться до кожного з п'яти класів, для проведення експерименту використано лише один і для нього проведено повний експеримент в декілька серій. Кількість серій експериментів, яку було проведено з кожним штучним worm-вірусом, дорівнювала трьом. Таким чином, сукупно для всіх штучних worm-вірусів проведено 360 серій експериментів (12 серій експериментів · 5 класів worm-вірусів · 6 варіантів джерел = 360 серій експериментів).

Другий варіант експерименту полягав в тому, що комп'ютерною станцією, в якій буде активовано штучний worm-вірус, була комп'ютерна станція в корпоративній мережі, але без встановленої компоненти системи. Третій варіант аналогічний другому варіанту за винятком того, що комп'ютерна станція, в якій було активовано штучний worm-вірус, була ззовні корпоративної мережі. В результаті для цих трьох варіантів було використано відношення «один до всіх»,

тобто об’єктами для поширення штучного worm-вірусу були усі комп’ютерні станції корпоративної мережі.

В наступних трьох варіантах використано відношення «всі до всіх», тобто об’єктами для поширення штучного worm-вірусу були десять комп’ютерних станцій корпоративної мережі. Здійснимо поділ на варіанти детальніше. Четвертий варіант експерименту полягав в активізації одночасно одного і того ж штучного worm-вірусу в десяти комп’ютерних станцій корпоративної мережі, в кожній з яких встановлено компоненти системи S. П’ятий варіант експерименту полягав в активуванні одночасно одного і того ж штучного worm-вірусу з десяти комп’ютерних станцій корпоративної мережі, в яких не встановлено компоненти системи S. Шостий варіант експерименту полягав в активуванні одночасно одного і того ж штучного worm-вірусу з десяти комп’ютерних станцій, які не належать корпоративній мережі. При активізації одного і того ж штучного worm-вірусу в одній чи в десяти комп’ютерних станцій може бути так, що він не зможе отримати контроль в ній чи в частині з них, і ця подія, також, може бути і враховується в постановці експерименту.

Тривалість експерименту для одного з п’яти типів worm-вірусів однієї серії та одного варіанту становила дві години. Враховуючи відомий досвід поширення реальних worm-вірусів, який було проаналізовано, тривалість їх масового поширення була відносно невеликою в часі і поширення більше одного worm-вірусу одночасно не зафіксовано. Тому, при проведенні експериментів в кожній серії було використано тільки один екземпляр штучного worm-вірусу і кількість серій для кожного з двадцяти екземплярів штучних worm-вірусів встановлено кількісно так, що дорівнювало трьом. Аналіз та обробку результатів інфікування було здійснено винятково з врахуванням тих комп’ютерних станцій, в яких встановлено компоненти системи одного і того ж штучного worm-вірусу, тобто зі ста комп’ютерних станцій. Worm-вірус певного типу може в комп’ютерній станції інфікувати її тільки один раз і про результат інфікування він видає повідомлення на екран. Якщо він в подальшому інфікує файли напротязі певного часу, то все-таки він вже, отримавши контроль, повідомляє про свою присутність. Таким чином, при проведенні експериментів комп’ютерна станція за результатами кожної з їх серій буде інфікована або не інфікована в результаті складності конфігурування

комп'ютерної станції, наявності системи S тощо. І, тому, цим забезпечується повне поле подій для певної комп'ютерної станції щодо результату прояву worm-вірусу. Тобто, одна комп'ютерна станція або інфікована ним або ні, що є необхідною умовою забезпечення коректності експерименту. Але в комп'ютерних станціях встановлені компоненти системи S для виявлення worm-вірусів, які можуть підтверджувати виявлення, результат якого буде збіжним з результатом повідомлення самим worm-вірусом, або може бути розбіжним, якщо worm-вірус підтверджує інфікування, а система S не підтверджує, або навпаки. В останньому випадку система S може правильно виявити worm-вірус або інший worm-вірус.

При встановленні результатів експерименту було розглянуто чотири варіанти подій, які відбулись, і розподілено їх так: тип worm-вірусу встановлено правильно і, відповідно, його віднесено до одного з класів K_W^j $(j = 1,2,\dots,5)$, для якого проводились дослідження; заповнено клас $K_W^{0,j}$ $(j = 1,2,\dots,5)$, тобто був пропущений worm-вірус в комп'ютерних станціях системою S, але відповідний штучний worm-вірус проінформував своїм корисним функціоналом про успішне інфікування комп'ютерної станції; система S віднесла до відповідного класу worm-вірусу об'єкт, який таким не був та, відповідно, не проінформував своїм корисним функціоналом про успішне інфікування комп'ютерної станції, і, тоді, виділено його додатковим класом $K_W^{j,p}$ $(j = 1,2,\dots,5)$; інфікування комп'ютерної станції не відбулось і компонента та система S це підтвердили та позначено клас для цього варіанту як $K_W^{j,Y}$ $(j = 1,2,\dots,5)$. Результати серій та етапів експерименту зведені за типами worm-вірусів та варіантами джерел задано в табл. 4.3 [47, 71, 109].

Таблиця 4.3.

Результати четвертого експерименту

Класи worm-вірусів	Серії експерименту												Разом
	Екземпляри класу												
	1			2			3			4			
	1	2	3	4	5	6	7	8	9	10	11	12	
$K_W^{j,p}$	44	58	59	55	54	50	64	63	64	42	58	54	665
K_W^j	1351	1355	1177	1257	1223	1244	1312	1336	1162	1406	1281	1194	15298
$K_W^{0,j}$	44	20	32	76	45	67	37	98	67	48	17	93	644
$K_W^{j,Y}$	1561	1567	1732	1612	1678	1639	1587	1503	1707	1504	1644	1659	19393

Достовірність виявлення worm-вірусів $D_1^{E_3} = 0{,}957142$ (95,7142%). Тоді, значення показника $E_3 = 0{,}957142$. Ефективність функціонування частково централізованих розподілених систем виявлення ЗПЗ:

$$E = \frac{1}{3} \cdot (0{,}99179 + 0{,}97000 + 0{,}95714) = 0{,}97298, \qquad (4.11)$$

де в формулі (4.1): $\sum_{i=1}^{3} \rho_i = 1$; $i = 1, 2, 3$, $\rho_1 = \rho_2 = \rho_3 = \frac{1}{3}$.

Таким чином, в результаті здійснення постановки експериментів та їх проведення було отримано результати, які підтверджують покращення ефективності функціонування частково централізованої розподіленої системи, можливість її застосування до виявлення worm-вірусів, а також підтверджують її стійкість згідно визначених ступенів стійкості та деградації системи.

4.4. Висновки до четвертого розділу

Для визначення ефективності частково централізованих систем виявлення ЗПЗ розроблена методика, яка враховує особливості таких систем з ускладнення їх розуміння зловмисниками, прийняття самостійних рішень щодо подальших кроків, здійснення гнучкої перебудови архітектури при зміні середовища функціонування, стійкості та деградації в процесі функціонування, а також виявлення ЗПЗ. Розроблена методика орієнтована на дослідження саме ефективності функціонування частково централізованих систем виявлення ЗПЗ, що є комплексним показником.

Розроблено частково централізовану розподілену систему виявлення worm-вірусів та проведено з нею експериментальні дослідження щодо встановлення ефективності функціонування та впроваджено її у виробництво. Результати щодо дослідження ефективності функціонування, зокрема з ускладнення розуміння таких систем зловмисниками, прийняття самостійних рішень системою щодо подальших кроків, здійснення гнучкої перебудови архітектури системи при зміні середовища функціонування, її стійкості та деградації в процесі експлуатації є достатніми для практичної реалізації та експлуатації таких систем.

Основні наукові результати четвертого розділу опубліковані в [38, 47, 71, 107, 109, 111, 114-116].

ВИСНОВКИ

У результаті виконання дисертаційного дослідження було розв’язано актуальну науково-прикладну задачу покращення ефективності функціонування розподілених систем з частковою централізацією, самоорганізацією та адаптивністю для виявлення ЗПЗ в комп’ютерних мережах, а також розроблено відповідні засоби. У роботі отримано наступні наукові та практичні результати.

1. Проведено аналіз методів синтезу архітектури розподілених систем, моделей показників оточуючого середовища для розподілених систем в корпоративних мережах, методів організації функціонування розподілених систем та методів виявлення ЗПЗ, зокрема worm-вірусів.

2. Розроблено формальних опис середовища функціонування розподілених систем через математичні моделі характеристичних показників, які враховуються при визначенні рівнів безпеки компонентів частково централізованих розподілених систем та при формуванні рішень щодо їх подальших кроків і виявлення ЗПЗ.

3. Розроблено метод синтезу математичних моделей рівнів безпеки компонентів системи для отримання нових аналітичних виразів комплексного опису оточуючого середовища корпоративних мереж і процесів, які відбуватимуться в частково централізованих розподілених системах.

4. Удосконалено модель частково централізованих розподілених систем, в якій враховано вимоги щодо можливості таких систем до динамічної зміни конфігурації, поділу центру прийняття рішень, розподілу компонентів центру прийняття рішень, з синтезованими в ній властивостями адаптивності і самоорганізації, реалізація яких здійснена безпосередньо в компонентах системи, в основному в тих з них, в яких буде знаходитись центр прийняття рішень системи після його переміщень, та приймає рішення згідно розроблених математичних моделей характеристичних показників значень рівнів безпеки компонентів, що задають опис середовища функціонування в комп’ютерній мережі. Модель дала змогу створювати системи виявлення ЗПЗ, функціонування яких ускладнює розуміння їх зловмисниками, що покращило їх стійкість до зловмисних дій.

5. Розроблено метод організації функціонування частково централізованих розподілених систем, в якому для функціонування такого типу систем проведено

розподіл компонент за відношенням до центру прийняття рішень для реалізації часткової централізації сумісно з принципами самоорганізації та адаптивності. Організація функціонування систем згідно такого методу задала механізми до ускладнення розуміння їх функціонування зловмисниками, самостійного прийняття рішень щодо подальших кроків системою та перебудови її архітектури за потреби і стала основою для наповнення системи методами виявлення ЗПЗ. Значення показника ефективності функціонування систем з частковою централізацією, самоорганізацією та адаптивністю є більшим за 99%. При цьому, ускладнення визначення компонент з центром системи для розробленої моделі систем в середньому становить 4-5% порівняно з відомими варіантами архітектури і фактично для варіанту з використанням динамічних компонент значення досягає 100%. Значення показників самоорганізації та адаптивності системи становить більше 98% і порівняно з системами, в яких ці принципи синтезовані окремо (або самоорганізація або адаптивність), має більше значення самостійно прийнятих рішень приблизно на 50%.

6. Розроблено метод виявлення worm-вірусів з використанням поділу їх на класи за спільними ознаками і визначеними критеріями за багатьма класами і з врахуванням імплементації його в архітектуру частково централізованих розподілених систем для прийняття рішення щодо віднесення worm-вірусу до певного класу, що дало змогу забезпечити достовірність виявлення worm-вірусів більше 95%.

7. Розроблено частково централізовану розподілену систему виявлення worm-вірусів, проведено з нею експериментальні дослідження щодо встановлення ефективності функціонування, стійкості, деградації, достовірності виявлення worm-вірусів та впроваджено її у виробництво. Значення показника стійкості так синтезованих систем є більшим, ніж 98%. Різні граничні варіанти станів безпеки компонентів системи не вплинули суттєво на її перебування в стабільному стані.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ahmet Ercan Topcu, Aimen Mukhtar Rmis. Analysis and evaluation of the riak cluster environment in distributed databases. *Computer Standards & Interfaces*. 2020. Vol. 72. ISSN 0920-5489. https://doi.org/10.1016/j.csi.2020.103452

2. AhnLab Endpoint Security. *AhnLab*. URL: https://global.ahnlab.com/site/about/aboutAhnlab.do. (date of access: 11.01.2024).

3. ASAX (University of Namur, Belgium). *University of Namur*. URL: www.ja.net/CERT/Software/asax/ (date of access: 11.10.2021).

4. Autonomous Agents for Intrusion Detection. *Purdue*. URL: https://www.cerias.purdue.edu/site/about/history/coast/projects/aafid.php (date of access: 21.01.2024).

5. Awwama Emad, Kadi Mohammad, Krayem Said, Lazar Ivo, Rihawi Ahmad. Using formal methods in distributed system design. *MATEC Web Conf*. 125 02033. 2017. Vol. 125. Pp. 1-4. DOI: 10.1051/matecconf/201712502033

6. Bakir C., HAKKOYMAZ V., Banu D. İ. R. İ., GÜÇLÜ M. Comparisons on intrusion detection and prevention systems in distributed databases. *Balkan Journal of Electrical and Computer Engineering*. 2017. Vol. 7(4). Pp. 446-455.

7. Balasubramaniyan J.S., Garcia-Fernandez J.O., Isacoff D., Spafford E., Zamboni D. An architecture for intrusion detection using autonomous agents. *CSAC*. 1999. P. 13 - 24. 10.1109/CSAC.1998.738563.

8. Barracuda. *Barracuda*. URL: https://www.barracuda.com/company# (date of access: 11.01.2024).

9. Battiston F., Cencetti G., Iacopini I. et al. Networks beyond pairwise interactions: structure and dynamics. *Physics Reports*. 2020. Vol. 874. Pp. 1–92, https://doi.org/10.1016/j.physrep.2020.05.004

10. Bellman K., Landauer C., Dutt N. et al. Self-aware cyber-physical systems. *ACM Trans. Cyber-Phys. Syst*. 2020. Vol. 4. https://doi.org/10.1145/3375716

11. Bitdefender Endpoint Security (Ultra). *Bitdefender*. URL: https://www.bitdefender.com/business/solutions/cyber-resilience.html. (date of access: 11.01.2024).

12. Brtis J.S., McEvilley M.A., Pennock M.J. Resilience Requirements Patterns. *INCOSE Int. Symp*. 2021. Vol. 31. Pp. 570–584. https://doi.org/10.1002/j.2334-5837.2021.00855.x

13. Botta A., De Donato W., Persico V., Pescap A. Integration of Cloud Computing and Internet of Things: A Survey, *Future Generation Computer Systems.* 2016. Vol. 56. Pp. 684-700. https://doi.org/10.1016/j.future.2015.09.021

14. Cigdem B., Veli H. A New Scalable and Expandable Access Control Model for Distributed Database Systems in Data Security. *Department of Computer Engineering Yildiz Technical University*. 2020, [online] URL: https://doi.org/10.1155/2020/8875069

15. Check Point Endpoint Security. *Check Point.* URL: https://www.checkpoint.com/ru/solutions/endpoint-security/ (date of access: 21.01.2024).

16. COAST (Computer Operations, Audit, and Security Technology). *Purdue.* URL: https://www.cerias.purdue.edu/site/about/history/coast/ (date of access: 21.01.2024).

17. Cointe N., Bonnet G., Boissier O. Ethics-based cooperation in multi-agent systems. *Advances in Social Simulation, Springer, Cham, Manhattan, NY, USA*. 2020. https://doi.org/10.1007/978-3-030-34127-5_10

18. Cruciani E., Mimun H.A., Quattropani M. et al. Phase transition of the k-majority dynamics in biased communication models. *Distrib. Comput.* 2023. Vol. 36. Pp. 107-135. https://doi.org/10.1007/s00446-023-00444-2

19. Czaja L. Distributed Systems—Objectives, Features, Applications. *Introduction to Distributed Computer Systems. Lecture Notes in Networks and Systems*. Springer, *Cham*. 2018. Vol. 27. Pp. 49-64. https://doi.org/10.1007/978-3-319-72023-4_2

20. Darabseh A. Freris N. M. A software-defined architecture for control of IoT cyberphysical systems. *Cluster Computing*. 2019. Vol. 22. No. 4. Pp. 1107–1122. https://doi.org/10.1007/s10586-018-02889-8

21. Darabseh A., Freris N. A software defined architecture for cyberphysical systems. *4th IEEE International Conference on Software Defined Systems (SDS)*. 2017. Pp. 54–60. DOI:10.1109/SDS.2017.7939141

22. Desmedt Y. Trojan Horses, Computer Viruses, and Worms. In: van Tilborg, H.C.A., Jajodia, S. (eds) *Encyclopedia of Cryptography and Security*. Springer, Boston, MA. 2011. Pp. 1319–1320. https://doi.org/10.1007/978-1-4419-5906-5_331

23. Edge C., Barker W., Hunter B., Sullivan G. Malware Security: Combating Viruses, Worms, and Root Kits. In: *Enterprise Mac Security*. Apress. 2010. Pp. 213–232. https://doi.org/10.1007/978-1-4302-2731-1_8

24. ESET Endpoint Security. *ESET*. URL: https://www.eset.com/ua/business/enterprise/ (date of access: 10.01.2024).

25. Esterle I. Deep learning in multiagent systems. *Deep Learning for Robot Perception and Cognition.* 2022. Pp. 435-460. 10.1016/B978-0-32-385787-1.00022-1

26. Fischer O., Oshman R. A distributed algorithm for directed minimum-weight spanning tree. *Distrib. Comput*. 2023. Vol. 36. Pp. 57–87. https://doi.org/10.1007/s00446-021-00398-3

27. Foremost Open-Source Intrusion Prevention System. *SNORT*. URL: https://www.snort.org/ (date of access: 12.01.2024).

28. Gasior J. and Seredynsky F. Decentralized job scheduling in the cloud based on a spatially generalized prisoner's dilemma game. *Int. J.APl. Math. Comput. Sci.* 2015. Vol. 25. No. 4. Pp. 737-751. https://intapi.sciendo.com/pdf/10.1515/amcs-2015-0053

29. Gen Digital Inc. *AVAST*. URL: https://www.avast.ua/about#pc. (date of access: 21.01.2024).

30. Götte, T., Hinnenthal, K., Scheideler, C. et al. Time-optimal construction of overlay networks. *Distrib. Comput.* 2023. Pp. 1-35. https://doi.org/10.1007/s00446-023-00442-4

31. Han K., Kokot G., Tovkach O., Glatz A., Aranson I. S., Snezhko A. Emergence of self-organized multivortex states in flocks of active rollers. *Proceedings of the National Academy of Sciences*. 2020. Vol. 117. No. 18. Pp. 9706–9711. https://doi.org/10.1073/pnas.2000061117

32. Herakovič N., Zupan H., Pipan M., Protner J., Šimic M. Distributed manufacturing systems with digital agents. *Journal of Mechanical Engineering*. 2019. Vol. 65. Pp. 650–657. https://doi.org/10.5545/sv-jme.2019.6331

33. Hossein Ashtari. What Is Network Behavior Analysis? Definition, Importance, and Best Practices. Network behavior analysis solutions collect and analyze enterprise network data to identify unusual activity and counter security threats. URL: https://www.spiceworks.com/tech/networking/articles/network-behavior-analysis/(date of access: 12.04.2023).

34. Hu V. C., Kuhn D. R., Ferraiolo D. F. Access Control for Emerging Distributed Systems. *Computer*. 2018. Vol. 51. No. 10. Pp. 100-103. doi: 10.1109/MC.2018.3971347

35. Hu Z., Dychka I., Onai M., Zhykin Y. Payment Protocol for Payment Channel Networks. *International Journal of Computer Network and Information Security*. 2019. Vol. 11 (6). Pp. 22-28. https://www.mecs-press.org/ijcnis/ijcnis-v11-n6/IJCNIS-V11-N6-3.pdf doi: 10.5815/ijcnis.2019.06.03

36. Hu Z., Mukhin V., Kornaga Y. *et al.* The Analytical Model for Distributed Computer System Parameters Control Based on Multi-factoring Estimations. *J Netw Syst Manage* 27. 2019. Pp. 351–365. https://doi.org/10.1007/s10922-018-9468-x

37. Katahira K., Chen Y., Akiyama E. Self-organized Speculation Game for the spontaneous emergence of financial stylized facts. *Physica A: Statistical Mechanics and Its Applications*. 2021. Vol. 582. https://doi.org/10.1016/j.physa.2021.126227

38. Kashtalian A., Lysenko S., Savenko B., Sochor T., Kysil T. Principle and method of deception systems synthesizing for malware and computer attacks detection. *Radioelectronic and Computer Systems.* 2023. Vol. *0*(4). Pp. 112-151. DOI: https://doi.org/10.32620/reks.2023.4.10

39. Kinouchi O., Pazzini R., Copelli M. Mechanisms of self-organized quasicriticality in neuronal network models. *Frontiers in Physiology*. 2020. Vol. 8. https://doi.org/10.3389/fphy.2020.583213

40. Kharchenko V., Ponochovnyi Y., Ivanchenko O., Fesenko H., Illiashenko O. Combining Markov and Semi-Markov Modelling for Assessing Availability and Cybersecurity of Cloud and IoT Systems. *Cryptography*. 2022. Vol. 6. No. 44. https://doi.org/10.3390/cryptography6030044

41. Khoroshko V., Khokhlachova Y., Vyshnevska N. Choice of indicators for forecasting cyber protection of computer systems. *Ukrainian Scientific Journal of Information Security*. 2023. Vol. 29. No. 1. C. 41-47.

42. Khoroshko V., Khokhlachova Y., Vyshnevska N. Decomposition of computer network technology in their design. *Ukrainian Scientific Journal of Information Security*. 2023. Vol. 29. Issue 3. Pp. 130-137. DOI 10.18372/2225-5036.29.18072

43. Laycraft K. C. Decision-making as a self-organizing process. *Ann. Cogn. Sci.* 2019. Vol. 3. No. 1. Pp. 86–99. https://doi.org/10.1016/j.physrep.2020.05.004

44. Letychevskyi O., Peschanenko V. Applying Algebraic Virtual Machine to Cybersecurity Tasks. *Proceedings of the IEEE 9th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*. IEEE, Hammamet, Tunisia, 2022. Pp. 161-169. DOI: 10.1109/SETIT54465.2022.9875895

45. Li Y., Jiang Y. Self-organization based service discovery approach considering intermediary utility. *Proceedings of the 2016 IEEE International Conference on Web Services (ICWS)*. IEEE, San Francisco, CA, USA. June 2016. Pp. 308–315. DOI: 10.1109/ICWS.2016.47

46. 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*. 2019. Vol. 1039. PP. 127-143, ISSN: 1865-0929.

47. Lysenko S., Savenko B. Distributed Discrete Malware Detection Systems Based on Partial Centralization and Self-Organization. *International Journal of Computing*. 2023. Vol. 22. Pp. 117-139. DOI: https://doi.org/10.47839/ijc.22.2.3082

48. Lysenko S., Savenko O., Bobrovnikova K., Kryshchuk A. Self-adaptive system for the corporate area network resilience in the presence of botnet cyberattacks. *Communications in Computer and Information Science*. 2018. Vol. 860. Pp. 385-401.

49. Maarten van Steen, Andrew S. Tanenbaum. Distributed Systems. *Distributed Systems*. Third edition. Preliminary version 3.01pre. 2017. ISBN: 978-90-815406-2-9

50. Malwarebytes Endpoint Security. *Malwarebytes.* URL: https://www.malwarebytes.com (date of access: 20.01.2024).

51. Martynyuk O., Sugak A., Martynyuk D., Drozd O. Evolutionary Network Model of Testing of the Distributed Information Systems. *Proceedings of the 2017 IEEE 9th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS*. 2017. Vol. 2. Pp. 888-893.

52. Misik S., Cela A., Bradac Z. Distributed Systems - A brief review of theory and practice, *IFAC-PapersOnLine*. 2016. Vol. 49. Issue 25. Pp. 318-323. ISSN 2405-8963. https://doi.org/10.1016/j.ifacol.2016.12.057

53. Moskalenko V., Kharchenko V., Moskalenko A., Kuzikov B. Resilience and Resilient Systems of Artificial Intelligence: Taxonomy, Models and Methods. *Algorithms*. 2023, Vol. 16, P. 165. https://doi.org/10.3390/a16030165

54. Mukhin V., Kornaga Y., Bondarenko V., Zavgorodnii V., Herasymenko O., Sholokhov O. Mathematical Model for Heterogeneous Databases Parameters Estimation in Distributed Systems with Dynamic Structure. *2020 IEEE 2nd International Conference on Advanced Trends in Information Theory (ATIT) /* Kyiv, Ukraine. 2020. Pp. 158-161. doi: 10.1109/ATIT50783.2020.9349331

55. Mukhin V., Kornaga Y., Zavgorodnii V., Bazaka Y., Zavgorodnya A., Mukhin O. Method of Data Processing System Synthesis for Heterogeneous Distributed Databases Based on Network-Centric Control. *2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Dortmund, Germany*. 2023. Pp. 607-612. doi: 10.1109/IDAACS58523.2023.10348667

56. Mukhin V., Zavgorodnii V., Nikitin V., Kornaga Y., Fartushnyi I., Stepanov A. Method of Determining the Required Number of Database Nodes in a Distributed Data Processing System. *2021 IEEE 3rd International Conference on Advanced Trends in Information Theory (ATIT), Kyiv, Ukraine*. 2021. Pp. 88-92. doi: 10.1109/ATIT54053.2021.9678569

57. Murthy J.K. A Functional Decomposition of Virus and Worm Programs. In: Qing, S., Gollmann, D., Zhou, J. (eds) Information and Communications Security. ICICS 2003. *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg. 2003. Vol. 2836. Pp. 405-414. https://doi.org/10.1007/978-3-540-39927-8_37

58. NetSTAT (University of California at Santa Barbara). *University of California*. URL: www.es.ucsb.edu/~kemm/netstat.html (date of access: 21.10.2021).

59. Network Admission Control. *Cisco.* URL: https://www.cisco.com/c/en/us/products/security/what-is-network-access-control-nac.html#~how-nac-works (date of access: 20.01.2024).

60. Network Intrusion Detection System. URL: https://www.sciencedirect.com/topics/computer-science/network-based-intrusion-detection-system. (date of access: 11.01.2024).

61. Neuer M. Cognitive perception and self-organization for digital twins in cyber-physical steel production systems. *Proceedings of the Industry 4.0 and Steelmaking Webinar of Steel Times International*. Future Steel Forum, Prague, Czech Republic, June 2020.

https://www.researchgate.net/publication/342503882_Cognitive_perception_and_self-organization_for_digital_twins_in_cyber-physical_steel_production_systems

62. Ngo F.T., Agarwal A., Govindu R., MacDonald C. Malicious Software Threats. In: *The Palgrave Handbook of International Cybercrime and Cyberdeviance*. Palgrave Macmillan, Cham. 2019. Pp. 1-22. https://doi.org/10.1007/978-3-319-90307-1_35-1

63. Östberg P. -O., Elmroth E. Decentralized Prioritization-Based Management Systems for Distributed Computing. *2013 IEEE 9th International Conference on e-Science, Beijing, China*. 2013. Pp. 228-237. doi: 10.1109/eScience.2013.44

64. Ozturk A. OSSEC-HIDS Capabilities, Architecture and plans. *Presentation at the 5th Linux and Free Software Festival, Ankara, Turkey*, 2006.

65. Palo Alto Networks. *Palo Alto*. URL: https://www.paloaltonetworks.com/cyberpedia/what-is-an-intrusion-detection-system-ids. (date of access: 11.01.2024).

66. Pandurangan G., Robinson P., Scquizzato M. A time- and message-optimal distributed algorithm for minimum spanning trees. *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing* (STOC). 2017. Pp. 743–756.

67. Pentland B. T., Liu P., Kremser W., Haerem T. The dynamics of drift in digitized processes. *MIS Quarterly*. 2020. Vol. 44. No. 1. Pp. 19–47. DOI: 10.25300/MISQ/2020/14458

68. Pester A., Sulema Y., Dychka I., Sulema O. Temporal Multimodal Data-Processing Algorithms Based on Algebraic System of Aggregates. *Algorithms*. 2023. Vol. 16. P. 186. https://doi.org/10.3390/a16040186

69. Pham V.H., Dacier M., Urvoy-Keller G., En-Najjary T. The Quest for Multi-headed Worms. In: Zamboni, D. (eds) Detection of Intrusions and Malware, and Vulnerability Assessment. DIMVA 2008. *Lecture Notes in Computer Science.* Springer, Berlin, Heidelberg. 2008. Vol. 5137. Pp. 247–266. https://doi.org/10.1007/978-3-540-70542-0_13

70. Prelude (Yoann Vandoorselaere, Laurent Oudot). *Prelude*. URL: www.prelude-ids.org/ (date of access: 21.10.2021).

71. Savenko B., Kashtalian A., Lysenko S., Savenko O. Malware Detection By Distributed Systems with Partial Centralization. *2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and*

Applications (IDAACS), Dortmund, Germany, 2023 / Pp. 265-270. DOI: https://doi.org/10.1109/IDAACS58523.2023.10348773

72. Savenko B., Kashtalian A., Sochor T., Nicheporuk A. Self-organized distributed anomaly detection system in computer systems based on the principal components method. *Intelligent Information Technologies & Systems of Information Security (IntelITSIS-2022)* : The 3th International Workshop, CEUR-Workshop Proceedings, Khmelnytskyi, 23-25 March 2022 / Khmelnytskyi National University, 2022. Vol. 3156. Pp. 589-600. URL: https://web.archive.org/web/20220619004420id_/http://ceur-ws.org/Vol-3156/paper25.pdf

73. Savenko O., Lysenko S. The Technique for Computer Systems Trojan Diagnosis in the Monitor Mode. *Proceedings of the 6-th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Prague (Czech Republic)*, 2011 / Pp. 770-774.

74. Security information portal Virus Bulletin, threat landscape. *Virus Bulletin*. URL: https://www.virusbulletin.com/ (date of access: 18.01.2024).

75. Shaojie W., Qiming L. Analysis of a Mathematical Model for Worm Virus Propagation. Advances in Information Security and Its Application. ISA 2009. *Communications in Computer and Information Science*. Springer, Berlin, Heidelberg. 2009. Vol. 36. Pp. 78-84. https://doi.org/10.1007/978-3-642-02633-1_10

76. Sheikh A. Trojans, Backdoors, Viruses, and Worms. In: Certified Ethical Hacker (CEH) *Preparation Guide*. Apress, Berkeley, CA. 2021. 217 p. https://doi.org/10.1007/978-1-4842-7258-9_5

77. Sophos Endpoint Protection with EDR, XDR, MDR. *Sophos*. URL: https://www.sophos.com/en-us/products/endpoint-antivirus. (date of access: 21.01.2024).

78. STAT (University of California at Santa Barbara). *University of California*. URL: www.es.ucsb.edu/ (date of access: 21.10.2021).

79. Symantec Endpoint Protection, *Symantec*. URL: https://techdocs.broadcom.com/de/de/symantec-security-software/endpoint-security-and-management/endpoint-protection/all/what-is-v45096464-d43e1648.html (date of access: 20.01.2024).

80. Systems, S.-Organisation and Information, *An Interdisciplinary Perspective*, Routledge, Taylor & Francis Group, Oxfordshire, UK, 2018.

https://www.routledge.com/Systems-Self-Organisation-and-Information-An-Interdisciplinary-Perspective/Alfredo-Pickering-Gudwin/p/book/9781138609938

81. Tadmor E. Mathematical aspects of self-organized dynamics: consensus, emergence of leaders, and social hydrodynamics, *SIAM News*. 2015. Vol. 48. No, 9. https://www.math.umd.edu/~tadmor/pub/flocking+consensus/SIAM%20News%2048(9)%207pp%20Tadmor%20self-organized%20dynamics.pdf

82. Tkachov V., Kovalenko A., Kharchenko V., Hunko M. Hvozdetska K. Cellular Technology Based Overlay Networks for the Secure Control of Intelligent Mobile Objects: Models and Numerical Study. 2022. 10.1007/978-3-031-20834-8_3

83. TrendMicro. URL: https://www.trendmicro.com/vinfo/us/security/news/ botnets (date of access: 10.01.2024)

84. The Independent IT-Security Institute. *AV-TEST*. URL: https://www.av-test.org/en/ (date of access:18.01.2024)

85. Voulgaris S., Dobson M., Steen M. Decentralized Network-level Synchronization in Mobile Ad Hoc Networks. *ACM Transactions on Sensor Networks*. 2016. Vol. 12. Issue 1. No. 5. Pp. 1-42. https://doi.org/10.1145/2880223

86. Wang W., Li D., Luo W., Kang Y., Wang L. Anthropomorphic diagnosis of runtime hidden behaviors in OpenMP multi-threaded applications. *Journal of Parallel and Distributed Computing*. 2023. Vol. 177. Pp. 17-27. ISSN 0743-7315, https://doi.org/10.1016/j.jpdc.2023.02.012

87. What is a Wireless Intrusion Prevention System (WIPS)? Wi-Fi Security That's No Longer Up in the Air. URL: https://www.justfirewalls.com/what-is-a-wireless-intrusion-prevention-system/ (date of access: 12.04.2023).

88. Wu K. Nan Q. Information characteristics, processes, and mechanisms of self-organization evolution. *Complexity*. 2019. Vol. 2019. https://doi.org/10.1155/2019/5603685

89. Zashcholkin, K., Drozd, O., Sulima, Y., Ivanova, O., & Perebeinos, I. DETECTION METHOD OF THE PROBABLE INTEGRITY VIOLATION AREAS IN FPGA-BASED SAFETY-CRITICAL SYSTEMS. *International Journal of Computing*. 2020. No. 19(2), Pp. 282-289. https://doi.org/10.47839/ijc.19.2.1772

90. Zashcholkin K. The detection method of probable areas of hardware Trojans location in FPGA-based components of safety-critical systems / K. Zashcholkin, O.

Drozd, *Proceedings of 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies, DESSERT, 2018* / Pp. 212-217.

91. Zeek. URL: https:// https://zeek.org (date of access: 26.01.2024)

92. Zillya Антивірус. URL: https://zillya.ua/index.php?q=worm (дата звернення: 27.01.2024).

93. Zuzcak M., Sochor T. Behavioral Analysis of Bot Activity in Infected Systems Using Honeypots. *Proceedings of the 24-st International Conference on Computer Networks*. Springer (Cham). 2017. Vol. 718. Pp. 118-133.

94. Валід, А., Поночовний, Ю., Харченко, В., & Узун, Д. (2020). RESEARCH OF THE MARKOV AVAILABILITY MODEL OF THE PHYSICAL SECURITY SYSTEM WITH DEGRADATION CAYSED BY ATTACKS AND HARDWARE FAILURES. *Radioelectronic and Computer Systems, 0*(1), 37-43. doi:https://doi.org/10.32620/reks.2020.1.04

95. Вірус вразив тисячі комп'ютерів. *Gazeta.ua*. 02 листопада 2023. URL: https://gazeta.ua/articles/edu-and-science/_virus-vraziv-tisyachi-kompyuteriv/867493 (дата звернення: 27.01.2024)

96. Видалити не можна залишити: путінські хакери народили небезпечний вірус. *Znaj.ua*. URL: https://znaj.ua/world/177015-vidaliti-ne-mozhna-zalishiti-putinski-hakeri-narodili-nebezpechniy-virus (дата звернення 27.01.2024).

97. ДСТУ ISO/IEC 2382-18:2005 Інформаційні технології. Словник термінів. Частина 18. Розподілене оброблення даних.

98. Дудикевич В. Б., Микитин Г. В., Ребець А. І. Квінтесенція інформаційної безпеки кіберфізичної системи. *Вісник Національного університету «Львівська політехніка». Інформаційні системи та мережі*. Львів: Видавництво Львівської політехніки. 2018. № 887. С. 58–68.

99. Дичка І. А., Сулема Є. С. Модель подання мультимодальних даних для комплексного опису об’єктів спостереження. *Вісник Вінницького політехнічного інституту*. 2020. № 1. С. 53–60. https://doi.org/10.31649/1997-9266-2020-148-1-53-60

100. Кльоц Ю., Петляк Н. Виявлення аномального трафіку у загальнодоступних комп’ютерних мережах. *MEASURING AND COMPUTING DEVICES IN TECHNOLOGICAL PROCESSES*. 2022. С. 79–86. https://doi.org/10.31891/2219-9365-2022-71-3-9

101. Корченко, А.О. Методи ідентифікації аномальних станів для систем виявлення вторгнень: автореф. дис. … д-ра техн. наук : 05.13.21. Київ, 2019. 40 с.

102. Крищук А. Ф. Мультиагентна інформаційна технологія діагностування комп'ютерних систем на наявність бот-мереж у корпоративних мережах : автореф. дис. ... канд. техн. наук : 05.13.06. Тернопіль. 2015. 20 с. http://library.wunu.edu.ua/libsearch/DocDescription?doc_id=315623

103. Лисенко С. М. Адаптивна інформаційна технологія діагностування комп'ютерних систем на наявність троянських програм: автореф. дис. ... канд. техн. наук : 05.13.06. Тернопіль. 2010. 20 с.

104. Лисенко С. М. Методологічні основи та інформаційна технологія забезпечення резильєнтності комп’ютерних систем в умовах кіберзагроз: дис. … д-ра техн. наук : 05.13.06. Львів. 2020. 409 с.

105. Лукова-Чуйко Н.В. Методологічні основи забезпечення функціональної стійкості розподілених інформаційних систем до кібернетичних загроз: автореферат дис. … д-ра техн. наук: 05.13.06. Київ. 2018. с. 40.

106. Поночовний Ю., Харченко В. DEPENDABILITY ASSURANCE METHODOLOGY OF INFORMATION AND CONTROL SYSTEMS USING MULTIPURPOSE SERVICE STRATEGIES. *Radioelectronic and Computer Systems*. 2020. Vol. 0(3). Pp. 43-58. doi:https://doi.org/10.32620/reks.2020.3.05

107. Савенко Б. О. А. с. 124840, Україна. Комп'ютерна програма «Проміжне програмне забезпечення частково централізованих розподілених систем виявлення зловмисного програмного забезпечення в комп’ютерних мережах». Дата реєстрації 07.03.2024.

108. Савенко Б., Каштальян А. Метод визначення ефективності розподіленої системи виявлення аномальних проявів. *Computer Systems and Information Technologies*. 2022. №2. С. 14–22. https://doi.org/10.31891/csit-2022-2-2

109. Савенко Б., Каштальян А., Петляк Н. Розподілені системи виявлення worm-вірусів. *2023 ITSec: Безпека інформаційних технологій:* матеріали XII Міжнар. наук.-техн. конф., м. Ужгород, 2-4 трав. 2023 р. / НАУ, м. Київ, 2023. С. 37-39. http://bit.nau.edu.ua/wp-content/uploads/2023/05/2023-ITSec_zbirnyk-1.pdf

110. Савенко Б., Каштальян А. Удосконалення методу централізованого виявлення розподілених аномалій за алгоритмом пошуку головних компонент.

MEASURING AND COMPUTING DEVICES IN TECHNOLOGICAL PROCESSES. 2021. № 2. С. 46–56. https://doi.org/10.31891/2219-9365-2021-68-2-6

111. Савенко Б. О. Метод виявлення worm-вірусів згідно багатокласової класифікації. *Вісник Хмельницького національного університету. Серія: Технічні науки*. 2024. № 1 (331). С. 18-28. DOI: https://doi.org/10.31891/2307-5732-2024-331-2

112. Савенко Б. О. Метод синтезу математичних моделей рівнів безпеки для частково централізованих розподілених систем виявлення зловмисного програмного забезпечення. *Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки.* 2023. № 3. Ч. 1. С. 217-227. DOI: https://doi.org/10.32782/2663-5941/2023.3.1/34

113. Савенко Б. О. Модель архітектури частково розподілених систем та їх компонентів в комп'ютерних мережах. *2023 Інформаційні технології та інженерія*: Тези доп. Всеукраїнської науково-практичної конференції молодих вчених, аспірантів і студентів, м. Миколаїв, 7–10 лютого 2023 р. / ЧНУ імені Петра Могили, м. Миколаїв, 2023. С. 81-82. https://dspace.chmnu.edu.ua/jspui/handle/123456789/875

114. Савенко Б. О. Розподілена частково централізована система виявлення зловмисного програмного забезпечення в комп'ютерних мережах. *Актуальні проблеми комп'ютерних наук АПКН-2022* : матеріали XIV всеукр. наук.-практ. конф., м. Хмельницький, 18-19 лист. 2022 р. / Хмельницький національний університет. Хмельницький, 2022. С. 251–253. URL: https://kn.khmnu.edu.ua/wp-content/uploads/sites/18/apkn2022_corpuspaper.pdf

115. Савенко Б. О. Розподілені системи виявлення зловмисного програмного забезпечення. *2022 International Conference on Innovative Solutions in Software Engineering (ICISSE-2022)* : Conference Proceedings, Ivano-Frankivsk, Ukraine, November 29-30, 2022 / Kuz M., Kozenko M. eds. Ivano-Frankivsk, VSPNU, 2022. Pp. 22–25. URL: https://kit.pnu.edu.ua/wp-content/uploads/sites/70/2023/01/2022_International_Conference_on_Innovative_Solutions_in_Software.pdf

116. Савенко Б. О. Самоорганізована розподілена система виявлення зловмисного програмного забезпечення в комп'ютерних мережах. *Математичне та програмне забезпечення інтелектуальних систем (МПЗІС-2022)* : тези доповідей XX міжнар. наук.-практ. конф., м. Дніпро, 23-25 лист. 2022 р. / під заг. ред. О.М.

Кісельової. Дніпро, ДНУ, 2022. С. 172–173. URL: http://mpzis.dnu.dp.ua/wp-content/uploads/2022/12/MPZIS-2022-1.pdf

117. Савченко А.С. Методи розподіленого управління корпоративними комп'ютерними мережами: дис. … д-ра техн. наук : 05.13.06. Київ. 2021. 341 с. https://er.nau.edu.ua/handle/NAU/48951?mode=full

118. Стецюк М. В. Методи та засоби забезпечення відмовостійкості та живучості спеціалізованих інформаційних технологій в умовах впливів зловмисного програмного забезпечення : дис. … доктора філософії : 123. Хмельницький. 2022. 249 с. https://nauka.khmnu.edu.ua/wp-content/uploads/dysertacziya-1.pdf

119. США ліквідували шкідливе ПЗ Snake, за допомогою якого Росія 20 років шпигувала у країнах НАТО. *Politico (zn.ua).* URL: https://zn.ua/ukr/usa/ssha-likviduvali-shkidlive-prohramne-zabezpechennja-snake-za-dopomohoju-jakoho-rosija-20-rokiv-shpihuvala-v-krajinakh-nato.html (дата звернення: 26.01.2024)

120. Терейковський І. А., Корченко О. Г., Погорелов В. В.. Методи розпізнавання кібератак: розпізнавання комп'ютерних вірусів.- Київ. – Навчальний посібник. КПІ ім. Ігоря Сікорського. – 2022. – 127 с. https://ela.kpi.ua/server/api/core/bitstreams/8b220575-458d-4a9f-b615-efaae753a6ee/content

121. Хакери зламали захист "Київстару" через обліковий запис одного зі співробітників – гендиректор. *Українська правда. 13 грудня 2023.* URL: https://www.epravda.com.ua/news/2023/12/13/707676/ (дата звернення: 19.01.2024).

ДОДАТКИ

ДОДАТОК А.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

Наукові праці, в яких опубліковані основні наукові результати дисертації

1. Lysenko S., Savenko B. Distributed Discrete Malware Detection Systems Based on Partial Centralization and Self-Organization. *International Journal of Computing*. 2023. Vol. 22. Pp. 117-139. DOI: https://doi.org/10.47839/ijc.22.2.3082

2. Kashtalian A., Lysenko S., Savenko B., Sochor T., Kysil T. Principle and method of deception systems synthesizing for malware and computer attacks detection. *Radioelectronic and Computer Systems*. 2023. Vol. *0*(4). Pp. 112-151. DOI: https://doi.org/10.32620/reks.2023.4.10

3. Савенко Б. О. Метод синтезу математичних моделей рівнів безпеки для частково централізованих розподілених систем виявлення зловмисного програмного забезпечення. *Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки*. 2023. № 3. Ч. 1. С. 217-227. DOI: https://doi.org/10.32782/2663-5941/2023.3.1/34

4. Савенко Б. О. Метод виявлення worm-вірусів згідно багатокласової класифікації. *Вісник Хмельницького національного університету. Серія: Технічні науки*. 2024. № 1 (331). С. 18-28. DOI: https://doi.org/10.31891/2307-5732-2024-331-2

Праці, які засвідчують апробацію матеріалів дисертації

5. Савенко Б. О. Розподілена частково централізована система виявлення зловмисного програмного забезпечення в комп'ютерних мережах. *Актуальні проблеми комп'ютерних наук АПКН-2022* : матеріали XIV всеукр. наук.-практ. конф., м. Хмельницький, 18-19 лист. 2022 р. / Хмельницький національний університет. Хмельницький, 2022. С. 251–253. URL: https://kn.khmnu.edu.ua/wp-content/uploads/sites/18/apkn2022_corpuspaper.pdf

6. Савенко Б. О. Самоорганізована розподілена система виявлення зловмисного програмного забезпечення в комп'ютерних мережах. *Математичне та програмне забезпечення інтелектуальних систем (МПЗІС-2022)* : тези доповідей XX міжнар. наук.-практ. конф., м. Дніпро, 23-25 лист. 2022 р. / під заг. ред. О.М. Кісельової.

Дніпро, ДНУ, 2022. С. 172–173. URL: http://mpzis.dnu.dp.ua/wp-content/uploads/2022/12/MPZIS-2022-1.pdf

7. Савенко Б. О. Розподілені системи виявлення зловмисного програмного забезпечення. *2022 International Conference on Innovative Solutions in Software Engineering (ICISSE-2022)* : Conference Proceedings, Ivano-Frankivsk, Ukraine, November 29-30, 2022 / Kuz M., Kozenko M. eds. Ivano-Frankivsk, VSPNU, 2022. Pp. 22–25. URL: https://kit.pnu.edu.ua/wp-content/uploads/sites/70/2023/01/2022_International_Conference_on_Innovative_Solutions_in_Software.pdf

8. Савенко Б. О. Модель архітектури частково розподілених систем та їх компонентів в комп'ютерних мережах. *2023 Інформаційні технології та інженерія*: Тези доп. Всеукраїнської науково-практичної конференції молодих вчених, аспірантів і студентів, м. Миколаїв, 7–10 лютого 2023 р. / ЧНУ імені Петра Могили, м. Миколаїв, 2023. С. 81-82. https://dspace.chmnu.edu.ua/jspui/handle/123456789/875

9. Савенко Б., Каштальян А., Петляк Н. Розподілені системи виявлення worm-вірусів. *2023 ITSec: Безпека інформаційних технологій:* матеріали XII Міжнар. наук.-техн. конф., м. Ужгород, 2-4 трав. 2023 р. / НАУ, м. Київ, 2023. С. 37-39. http://bit.nau.edu.ua/wp-content/uploads/2023/05/2023-ITSec_zbirnyk-1.pdf

10. Savenko B., Kashtalian A., Lysenko S., Savenko O. Malware Detection By Distributed Systems with Partial Centralization. 2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Dortmund, Germany, 2023 / Pp. 265-270. DOI: https://doi.org/10.1109/IDAACS58523.2023.10348773

Публікації, які додатково відображають наукові результати дисертації

11. Савенко Б. О. А. с. 124840, Україна. Комп'ютерна програма «Проміжне програмне забезпечення частково централізованих розподілених систем виявлення зловмисного програмного забезпечення в комп'ютерних мережах». Дата реєстрації 07.03.2024.

ДОДАТОК Б.

АКТИ ВПРОВАДЖЕННЯ

ЗАТВЕРДЖУЮ
В.о. директора ТОВ «Новатор»
_____ Олександр ЯНОВИЦЬКИЙ
«02» лютого 2024 р.

АКТ
про впровадження результатів дисертаційної роботи
аспіранта кафедри комп'ютерної інженерії та інформаційних систем
Хмельницького національного університету Савенка Богдана Олеговича
«Метод та частково централізовані системи виявлення зловмисного програмного забезпечення в комп'ютерних мережах»

Результати дисертаційної роботи аспіранта кафедри комп'ютерної інженерії та інформаційних систем Хмельницького національного університету Савенка Б. О. впроваджені на Товаристві з обмеженою відповідальністю «Новатор» у відділі автоматизованих систем управління.

В процесі розроблення і впровадження частково централізованої системи виявлення зловмисного програмного забезпечення в комп'ютерних мережах були використані на ТОВ «Новатор» такі результати, які одержані Савенко Б. О. особисто:

1) модель архітектури частково централізованих розподілених систем, в якій враховано вимоги щодо можливості систем до динамічної зміни конфігурації, поділу центру прийняття рішень, розподілу компонентів центру прийняття рішень з синтезованими в ній властивостями адаптивності і самоорганізації, реалізація яких здійснена безпосередньо в компонентах системи, в основному в тих з них, в яких буде знаходитись центр прийняття рішень системи та приймає рішення згідно розроблених математичних моделей характеристичних показників значень рівнів безпеки компонентів, що задають опис середовища функціонування в комп'ютерній мережі і така модель дала змогу створювати згідно неї системи виявлення зловмисного програмного забезпечення, функціонування яких ускладнює розуміння їх зловмисниками, що покращує їх стійкість до зловмисних дій;

2) метод організації функціонування частково централізованих розподілених систем, в якому для функціонування такого типу систем проведено розподіл компонент за відношенням до центру прийняття рішень для реалізації часткової централізації сумісно з принципами самоорганізації та адаптивності, що дало змогу задати механізми до самостійного прийняття рішень щодо подальших кроків системою, перебудови її архітектури за потреби і наповнення системи методами виявлення ЗПЗ.

За результатами виконаних досліджень Савенко Б. О. було розроблено модель та метод, алгоритми та частково централізовану розподілену систему покращення ефективності функціонування розподілених систем з частковою централізацією для виявлення зловмисного програмного забезпечення в

комп'ютерних мережах та виявлення зловмисного програмного забезпечення з їх використанням за рахунок синтезу їх архітектури таким чином, щоб принципи функціонування таких систем ускладнювали зловмисниками їх розуміння.

Отриманні результати дозволили покращити ефективність функціонування розподілених систем з частковою централізацією для виявлення зловмисного програмного забезпечення в комп'ютерних мережах. Результати роботи використано у відділі автоматизованих систем управління для розробки розподілених систем з комбінованою архітектурою в частині централізації.

Цей акт не є підставою для фінансових розрахунків.

В.о. начальника відділу
автоматизованих систем управління Денис АНДРІЄВСЬКИЙ

«Затверджую»
Генеральний директор ПП «Нолт Технолоджис»
Мельниченко О.В.
лютого 2024 р.

АКТ
про впровадження результатів дисертаційної роботи
аспіранта кафедри комп'ютерної інженерії та інформаційних систем
Хмельницького національного університету Савенка Богдана Олеговича
«Метод та частково централізовані системи виявлення зловмисного програмного забезпечення в комп'ютерних мережах»

Результати дисертаційної роботи аспіранта кафедри комп'ютерної інженерії та інформаційних систем Хмельницького національного університету Савенка Б. О. впроваджені на ПП «Нолт Технолоджис».

При розробленні частково централізованої системи виявлення зловмисного програмного забезпечення в комп'ютерних мережах, включаючи підсистему виявлення worm-вірусів, були використані на ТОВ «Нолт Тех Солюшенс» такі результати, які одержані Савенко Б. О. особисто:

1) метод синтезу математичних моделей рівнів безпеки компонентів системи для отримання нових математичних моделей рівнів безпеки компонентів системи для комплексного опису оточуючого середовища корпоративних мереж і процесів, які відбуватимуться в частково централізованих розподілених системах, що дало змогу узгодити між собою характеристичні показники, які задані дискретними та неперевними величинами, та для формування нових характеристик аналітичними виразами з врахуванням їх при визначенні рівнів безпеки в компонентах і системах в цілому;

2) метод виявлення worm-вірусів, в якому здійснено поділ їх на класи за спільними ознаками і визначеними критеріями згідно класифікації об'єктів за багатьма класами і він імплементується в архітектуру частково централізованих розподілених систем для отримання цілісного сенсору та прийняття рішення щодо віднесення worm-вірусу до певного класу, що покращило достовірність виявлення, зокрема за рахунок приховування принципів функціонування системи.

Виконані дослідження включають методи та частково централізовану розподілену систему покращення ефективності функціонування розподілених систем з частковою централізацією для виявлення зловмисного програмного забезпечення в комп'ютерних мережах та виявлення зловмисного програмного забезпечення з їх використанням за рахунок синтезу їх архітектури таким чином,

щоб принципи функціонування таких систем ускладнювали зловмисниками їх розуміння.

Отриманні результати покращують ефективність функціонування розподілених систем з частковою централізацією для виявлення worm-вірусів в комп'ютерних мережах порівняно із системами з відомими зловмисникам архітектурами приблизно на 8-11%.

Цей акт не є підставою для фінансових розрахунків.

Директор піп

«Затверджую»
Директор ТОВ «ІТТ»
Сумогук Т.С.
«26» лютого 2024 р.

АКТ
про впровадження результатів дисертаційної роботи
аспіранта кафедри комп'ютерної інженерії та інформаційних систем
Хмельницького національного університету Савенка Богдана Олеговича
«Метод та частково централізовані системи виявлення зловмисного програмного забезпечення в комп'ютерних мережах»

Результати дисертаційної роботи Савенка Б. О. аспіранта кафедри комп'ютерної інженерії та інформаційних систем Хмельницького національного університету впроваджені на ТОВ «ІТТ».

При розробленні розподіленої системи виявлення зловмисного програмного забезпечення в комп'ютерних мережах були використані на ТОВ «ІТТ» такі результати, які одержані Савенко Б. О. особисто:

1) модель архітектури частково централізованих розподілених систем, в якій враховано вимоги щодо можливості систем до динамічної зміни конфігурації, поділу центру прийняття рішень, розподілу компонентів центру прийняття рішень з синтезованими в ній властивостями адаптивності і самоорганізації, реалізація яких здійснена безпосередньо в компонентах системи, в основному в тих з них, в яких буде знаходитись центр прийняття рішень системи та приймає рішення згідно розроблених математичних моделей характеристичних показників значень рівнів безпеки компонентів, що задають опис середовища функціонування в комп'ютерній мережі і така модель дала змогу створювати згідно неї системи виявлення зловмисного програмного забезпечення, функціонування яких ускладнює розуміння їх зловмисниками, що покращує їх стійкість до зловмисних дій;

2) метод синтезу математичних моделей рівнів безпеки компонентів системи для отримання нових математичних моделей рівнів безпеки компонентів системи для комплексного опису оточуючого середовища корпоративних мереж і процесів, які відбуватимуться в частково централізованих розподілених системах, що дало змогу узгодити між собою характеристичні показники, які задані дискретними та неперевними величинами, та для формування нових характеристик аналітичними виразами з врахуванням їх при визначенні рівнів безпеки в компонентах і системах в цілому;

3) метод організації функціонування частково централізованих розподілених систем, в якому для функціонування такого типу систем проведено розподіл компонент за відношенням до центру прийняття рішень для реалізації часткової централізації сумісно з принципами самоорганізації та адаптивності, що дало змогу задати механізми до самостійного прийняття рішень щодо подальших кроків системою, перебудови її архітектури за потреби і наповнення системи методами виявлення ЗПЗ;

Виконані дослідження включають модель архітектури, метод організації функціонування, метод синтезу математичних моделей та частково централізовану розподілену систему покращення ефективності функціонування розподілених систем з частковою централізацією для виявлення зловмисного програмного забезпечення в комп'ютерних мережах за рахунок синтезу їх архітектури таким чином, щоб принципи функціонування таких систем ускладнювали зловмисниками їх розуміння.

Отриманні результати дозволили покращити ефективність функціонування розподілених систем з частковою централізацією для виявлення worm-вірусів в комп'ютерних мережах порівняно із системами з відомими зловмисникам архітектурами приблизно на 9-11%.

Цей акт не є підставою для фінансових розрахунків.

Директор В. С. Сімогук

«Затверджую»
Проректор з науково-педагогічної роботи
Віктор ЛОПАТОВСЬКИЙ
«26» 02 2024 р.

АКТ

про впровадження в навчальний процес Хмельницького національного університету результатів дисертаційної роботи аспіранта кафедри комп'ютерної інженерії та інформаційних систем Савенка Богдана Олеговича «Метод та частково централізовані системи виявлення зловмисного програмного забезпечення в комп'ютерних мережах»

Ми, комісія в складі: завідувача кафедри комп'ютерної інженерії та інформаційних систем, д.т.н., професора Говорущенко Т. О. (голова комісії), доцента кафедри комп'ютерної інженерії та інформаційних систем, к.т.н., доцента Нічепорука А. О., доцента кафедри комп'ютерної інженерії та інформаційних систем, к.т.н., доцента Капустян М. В., склали акт про те, що результати дисертаційної роботи Савенка Б. О. впроваджені та використовуються в освітньому процесі Хмельницького національного університету при викладанні дисциплін на кафедрі комп'ютерної інженерії та інформаційних систем для спеціальностей 123 Комп'ютерна інженерія, 126 Інформаційні системи та технології, зокрема в освітніх компонентах «Теорія і проєктування комп'ютерних та кіберфізичних систем і мереж», «Теорія і технології проєктування спеціалізованих операційних систем», «Методи розв'язування наукових задач комп'ютерної інженерії», «Технології та методи забезпечення надійності та безпеки інформаційних систем та технологій».

При викладанні цих дисциплін викладачами кафедр використовувалися наступні матеріали досліджень, отримані Савенком Б. О. особисто:

1) модель архітектури частково централізованих розподілених систем, в якій враховано вимоги щодо можливості систем до динамічної зміни конфігурації, поділу центру прийняття рішень, розподілу компонентів центру прийняття рішень з синтезованими в ній властивостями адаптивності і самоорганізації, реалізація яких здійснена безпосередньо в компонентах системи, в основному в тих з них, в яких буде знаходитись центр прийняття рішень системи та приймає рішення згідно розроблених математичних моделей характеристичних показників значень рівнів безпеки компонентів, що задають опис середовища функціонування в комп'ютерній мережі і така модель дала змогу створювати згідно неї системи виявлення зловмисного програмного забезпечення, функціонування яких

ускладнює розуміння їх зловмисниками, що покращує їх стійкість до зловмисних дій;

2) метод синтезу математичних моделей рівнів безпеки компонентів системи для отримання нових математичних моделей рівнів безпеки компонентів системи для комплексного опису оточуючого середовища корпоративних мереж і процесів, які відбуватимуться в частково централізованих розподілених системах, що дало змогу узгодити між собою характеристичні показники, які задані дискретними та неперевними величинами, та для формування нових характеристик аналітичними виразами з врахуванням їх при визначенні рівнів безпеки в компонентах і системах в цілому;

3) метод організації функціонування частково централізованих розподілених систем, в якому для функціонування такого типу систем проведено розподіл компонент за відношенням до центра прийняття рішень для реалізації часткової централізації сумісно з принципами самоорганізації та адаптивності, що дало змогу задати механізми до самостійного прийняття рішень щодо подальших кроків системою, перебудови її архітектури за потреби і наповнення системи методами виявлення ЗПЗ;

4) метод виявлення worm-вірусів, в якому здійснено поділ їх на класи за спільними ознаками і визначеними критеріями згідно класифікації об'єктів за багатьма класами і він імплементується в архітектуру частково централізованих розподілених систем для отримання цілісного сенсору та прийняття рішення щодо віднесення worm-вірусу до певного класу, що покращило достовірність виявлення, зокрема за рахунок приховування принципів функціонування системи.

5) розроблена частково централізована розподілена система виявлення worm-вірусів.

Отримані матеріали досліджень дозволили розробити лабораторні практикуми з використанням моделі архітектури та методу створення розподілених систем з частковою централізацією, методу виявлення worm-вірусів згідно із багатокласовою класифікацією.

Говорущенко Т. О.

Нічепорук А. О.

Капустян М. В.

ДОДАТОК В.

ЛІСТИНГ ПРОГРАМНОГО КОДУ (ФРАГМЕНТ)

Database.h

```
#pragma once
#include "Structs.h"
#include "Logger.h"
#include <fstream>
#include <random>

using namespace std;

namespace db {
	class Database
	{
	public:
		Database();
		~Database();

	public:
		void appendDatabase(string filename, vector<UserAddressDto>* addresses,
unique_ptr<loggernamespace::Logger>& logger);
		void createDatabase(vector<UserAddressDto>* addresses,
unique_ptr<loggernamespace::Logger>& logger);
		bool fileExists(const string& filename);
		long long generateId();
		vector<UserAddressDto> getAddresses(bool* isDecisionMakerCenter, string
hostname, unique_ptr<loggernamespace::Logger>& logger);
	};
}
```

Database.cpp

```
#include "Database.h"

namespace db {
	Database::Database()
	{
	}

	Database::~Database()
	{
	}

	bool Database::fileExists(const string& filename)
	{
		ifstream file(filename);
		return file.good();
	}

	long long Database::generateId()
	{
		std::random_device rd;
		std::mt19937_64 generator(rd());
		std::uniform_int_distribution<long long> distribution;
		return distribution(generator) % (10000000 + 1);
	}
```

```
	void Database::createDatabase(vector<UserAddressDto>* addresses,
unique_ptr<loggernamespace::Logger>& logger)
	{
		fstream file("database.dat", ios::in | ios::out | ios::app);

		if (!file)
		{
			logger->addLog("Failed opening the database file.");
			exit(EXIT_FAILURE);
		}
		for (const auto& address : *addresses) {
			file << address.id << " " << address.pcAddress << " " <<
address.isDecisionMakerCenter << '\n';
		}
		file.close();
	}

	vector<UserAddressDto> Database::getAddresses(bool* isDecisionMakerCenter, string
hostname, unique_ptr<loggernamespace::Logger>& logger)
	{
		vector<UserAddressDto> usersArray;
		UserAddressDto readenUser;
		fstream file("database.dat", ios::in | ios::out | ios::app);
		if (!file) {
			logger->addLog("Failed opening the database file.");
			exit(EXIT_FAILURE);
		}

		while (file) {
			file >> readenUser.id >> readenUser.pcAddress >>
readenUser.isDecisionMakerCenter;
			if (!file)
			{
				break;
			}

			if (readenUser.pcAddress != hostname)
			{
				usersArray.push_back(readenUser);
				continue;
			}

			*isDecisionMakerCenter = readenUser.isDecisionMakerCenter;
		}

		file.close();

		return usersArray;
	}

	void Database::appendDatabase(string filename, vector<UserAddressDto>* addresses,
unique_ptr<loggernamespace::Logger>& logger)
	{
		ofstream outputFile(filename, std::ios::trunc);
		if (!outputFile) {
			logger->addLog("Failed opening the database file.");
			return;
		}

		for (const auto& address : *addresses) {
			outputFile << address.id << ' ' << address.pcAddress << ' ' <<
address.isDecisionMakerCenter << '\n';
		}
		outputFile.close();
	}
}
```

Logger.h

```
#pragma once

#include <ctime>
#include <chrono>
#include <filesystem>
#include <fstream>
#include <iostream>
#include <string>

#define BUFFER_SIZE 26

namespace loggernamespace
{
	class Logger
	{
	private:
		const std::string directoryName = "Logs";
		const std::string filePath{ directoryName + "/log.txt" };

		std::ofstream file{};

		bool isFileOpened{};
		std::string address{};

		void createFileDirectory();
		void openLoggingFile();

	public:
		Logger();
		~Logger();

		void addLog(std::string message);
		void initialize();
		bool isInitialized();
		void setAddress(std::string address);
	};
}
```

Logger.cpp

```
#include "Logger.h"

namespace loggernamespace
{
	Logger::Logger()
	{
	}

	Logger::~Logger()
	{
		if (file.is_open())
		{
			file.close();
		}
	}

	void Logger::addLog(std::string message)
	{
		if (!isFileOpened)
		{
			std::cerr << "File is not opened" << std::endl;

			return;
```

```
		}

		char currentDateTime[BUFFER_SIZE]{};

		std::time_t currentTime =
std::chrono::system_clock::to_time_t(std::chrono::system_clock::now());

		errno_t error{ ctime_s(currentDateTime, BUFFER_SIZE, &currentTime) };

		if (error)
		{
			std::cerr << "Error occured: " << error << std::endl;
			return;
		}

		currentDateTime[BUFFER_SIZE - 2] = '\0';

		if (address.size())
		{
			file << address << ": "
				<< currentDateTime
				<< " "
				<< message << std::endl;
		}
		else
		{
			file << currentDateTime
				<< " "
				<< message << std::endl;
		}

		if (file.bad())
		{
			std::cerr << "Can't write log" << std::endl;
		}
	}

	void Logger::initialize()
	{
		createFileDirectory();
		openLoggingFile();
	}

	bool Logger::isInitialized()
	{
		return !isFileOpened;
	}

	void Logger::setAddress(std::string address)
	{
		this->address = address;
	}

	void Logger::createFileDirectory()
	{
		std::filesystem::path folderPath(directoryName);

		if (std::filesystem::exists(folderPath))
		{
			return;
		}

		if (!std::filesystem::create_directory(folderPath))
		{
			std::cerr << "Folder was not created." << std::endl;
			return;
		}
```

```
			std::cerr << "Folder created." << std::endl;
	}

	void Logger::openLoggingFile()
	{
			file.open(filePath, std::ios::app | std::ios::out);
			if (!file.is_open())
			{
				std::cerr << "Can't create/open logging file" << std::endl;
				return;
			}

			isFileOpened = true;
	}
}
```

Structs.h

```
#pragma once
#include <string>
#include <iostream>
#include <vector>

using namespace std;

struct ActivityDto
{
	bool isActive{};
	bool isDecisionMakerActive{}; //makes sense only if isDecisionMakerCenter = true
};

struct DataItemDto
{
	char nodeName[100]{};
	int a;
	int b;
};

struct UserAddressDto
{
	long long id{ 0 };
	string pcAddress;
	bool isDecisionMakerCenter{};
	ActivityDto activity{}; //makes sense only if isDecisionMakerCenter = true
};
```

Source.cpp

```
#include <iostream>

#include <WinSock2.h>

#include <ws2tcpip.h>

#include <vector>

#include <string>

#include <map>

#include <Windows.h>
```

```
#include <fstream>
#include <limits>
#include <stdio.h>
#include "Logger.h"
#include "Structs.h"
#include "Database.h"

#pragma comment (lib, "Ws2_32.lib")

constexpr auto DEFAULT_PORT = "27015";
constexpr auto DEFAULT_BUFFER_LENGTH = 512;
constexpr auto WAIT_MSEC_BY_DECISIONMAKER = 10000;

using namespace std;

static unique_ptr<loggernamespace::Logger> logger{};
static unique_ptr<db::Database> database{};
static int decisionMakersCount{};
static bool exitThread{ false };
static map<string, UserAddressDto> nodes{};
static bool isDecisionMakerCenter{};
static bool isDecisionMakerCenterActive{};

SOCKET connectToNode(string nodeName, PCSTR port)
{
	struct addrinfo* address = nullptr, * ptr = nullptr, hints{};

	hints.ai_family = AF_INET;
	hints.ai_protocol = IPPROTO_TCP;
	hints.ai_socktype = SOCK_STREAM;
	hints.ai_flags = 0;
	hints.ai_addr = NULL;
```

```
hints.ai_canonname = NULL;
hints.ai_next = NULL;

int response = getaddrinfo(nodeName.c_str(), port, &hints, &address);
if (response != 0)
{
	logger->addLog("getaddrinfo failed: " + to_string(response) + " Name: " + nodeName);

	return 0;
}

int optval{ 1 };

SOCKET nodeSocket{};

addrinfo* addressIterator{};

for (addressIterator = address; addressIterator != NULL; addressIterator = addressIterator->ai_next)
{
	nodeSocket = socket(address->ai_family, address->ai_socktype, address->ai_protocol);

	if (nodeSocket == INVALID_SOCKET)
	{
		continue;
	}

	if (connect(nodeSocket, addressIterator->ai_addr, addressIterator->ai_addrlen) == -1)
	{
		closesocket(nodeSocket);
		continue;
	}
```

```
            break;
        }

        if (addressIterator == NULL)
        {
            return 0;
        }

        freeaddrinfo(addressIterator);

        return nodeSocket;
}

SOCKET generateSocket(PCSTR port, const char* hostname)
{
        struct addrinfo* address = nullptr, * ptr = nullptr, hints{};

        hints.ai_family = AF_INET;
        hints.ai_protocol = IPPROTO_TCP;
        hints.ai_socktype = SOCK_STREAM;
        hints.ai_flags = AI_PASSIVE;
        hints.ai_addr = NULL;
        hints.ai_canonname = NULL;
        hints.ai_next = NULL;

        int response = getaddrinfo(hostname, port, &hints, &address);
        if (response != 0)
        {
            logger->addLog("getaddrinfo failed: " + to_string(response));

            return -1;
        }
```

```
	SOCKET listenSocket{ INVALID_SOCKET };

	int optval{ 1 };

	for (auto addressIterator = address; addressIterator != NULL; addressIterator = addressIterator->ai_next)
	{
		listenSocket = socket(address->ai_family, address->ai_socktype, address->ai_protocol);

		if (listenSocket == INVALID_SOCKET)
		{
			continue;
		}

		if (setsockopt(listenSocket, SOL_SOCKET, SO_REUSEADDR, (char*)&optval, sizeof(int)) == -1)
		{
			logger->addLog("setsockopt() error");
			closesocket(listenSocket);
			return -1;
		}

		response = bind(listenSocket, address->ai_addr, (int)address->ai_addrlen);

		if (response == SOCKET_ERROR)
		{
			logger->addLog("bind failed with error: " + to_string(WSAGetLastError()));

			freeaddrinfo(address);
			closesocket(listenSocket);

			return -1;
```

```
            }
        }

        IN_ADDR* addr{};

        char ipstr[INET_ADDRSTRLEN]{};

        struct sockaddr_in* ipv4 = (struct sockaddr_in*)address->ai_addr;
        addr = &(ipv4->sin_addr);

        inet_ntop(address->ai_family, addr, ipstr, sizeof ipstr);

        if (ipstr == NULL)
        {
            logger->addLog("inet_ntop error: " + to_string(WSAGetLastError()));
        }

        logger->setAddress(ipstr);

        freeaddrinfo(address);

        if (listen(listenSocket, SOMAXCONN) == SOCKET_ERROR)
        {
            logger->addLog("Listen failed with error: " + to_string(WSAGetLastError()));
            closesocket(listenSocket);
            return -1;
        }

        return listenSocket;
}

SOCKET establishConnection(SOCKET socket)
```

```
{
	SOCKET ClientSocket{ INVALID_SOCKET };

	ClientSocket = accept(socket, NULL, NULL);

	if (ClientSocket == INVALID_SOCKET)
	{
		logger->addLog("Accept failed: " + to_string(WSAGetLastError()));
		closesocket(socket);
		return -1;
	}

	return ClientSocket;
}

void sendMessage(SOCKET socket, const char* sendBuf, int size)
{
	int iResult{ send(socket, sendBuf, size, 0) };
	if (iResult == SOCKET_ERROR)
	{
		logger->addLog("send failed: " + to_string(WSAGetLastError()));
		closesocket(socket);
		return;
	}
}

long long generateNum(int min, int max) {
	std::random_device rd;
	std::mt19937_64 generator(rd());
	std::uniform_int_distribution<long long> distribution(min, max);
	return distribution(generator);
}
```

```
DWORD WINAPI connectNewNode(LPVOID socketParam)
{
	SOCKET listenSocket{ *(SOCKET*)socketParam };
	int activity{};
	DataItemDto data{};
	fd_set rfds{};
	timeval time{};
	time.tv_sec = 1;

	while (!exitThread)
	{
		FD_ZERO(&rfds);
		FD_SET(listenSocket, &rfds);

		activity = select(listenSocket + 1, &rfds, NULL, NULL, &time);

		if ((activity <= 0) && (errno != EINTR))
		{
			continue;
		}

		if (!FD_ISSET(listenSocket, &rfds))
		{
			continue;
		}

		SOCKET clientSocket{ establishConnection(listenSocket) };

		logger->addLog("Connecting to listenSocket");

		if (clientSocket == -1)
```

```
{
	logger->addLog("Can't connect to client");
	continue;
}

char messageChunk[DEFAULT_BUFFER_LENGTH]{};

int size = recv(clientSocket, messageChunk, DEFAULT_BUFFER_LENGTH - 1, 0);

if (size == sizeof(DataItemDto))
{
	memcpy(&data, messageChunk, sizeof(DataItemDto));

	int response{ data.a + data.b };

	if (nodes[data.nodeName].activity.isActive)
	{
		nodes[data.nodeName].activity.isActive = false;
		closesocket(clientSocket);
		continue;
	}

	sendMessage(clientSocket, (char*)&response, sizeof(int));
	sendMessage(clientSocket, (char*)&isDecisionMakerCenterActive, sizeof(bool));

	nodes[data.nodeName].activity.isActive = true;

	if (isDecisionMakerCenter && isDecisionMakerCenterActive)
	{
		nodes[data.nodeName].activity.isDecisionMakerActive = generateNum(0,
decisionMakersCount) >= decisionMakersCount / 2;
```

```
                                sendMessage(clientSocket,
(char*)&nodes[data.nodeName].activity.isDecisionMakerActive, sizeof(bool));


                                logger->addLog("ConnectNewNode make center active:  " +
to_string(nodes[data.nodeName].activity.isDecisionMakerActive));
                        }
                }
                else if (size == sizeof(bool))
                {
                        memcpy(&isDecisionMakerCenterActive, messageChunk, sizeof(bool));


                        logger->addLog("Is decision maker active: " + to_string(isDecisionMakerCenterActive));
                }


                closesocket(clientSocket);
        }


        return 0;
}


DWORD WINAPI decisionMaker(LPVOID socketParam)
{
        while (!exitThread) {
                Sleep(WAIT_MSEC_BY_DECISIONMAKER);


                if (!isDecisionMakerCenterActive)
                {
                        continue;
                }


                for (auto iter{ nodes.begin() }; iter != nodes.end(); iter++)
                {
                        if (!iter->second.activity.isActive || !iter->second.isDecisionMakerCenter)
```

```
                {
                        continue;
                }

                SOCKET nodeSocket{ connectToNode(iter->second.pcAddress, DEFAULT_PORT) };

                bool sendData = generateNum(0, decisionMakersCount) >= decisionMakersCount / 2;

                sendMessage(nodeSocket, (char*)&sendData, sizeof(bool));

                logger->addLog("DecisionMaker of " + iter->second.pcAddress + " is " +
to_string(sendData));
            }
        }
        return 0;
}

vector<UserAddressDto> getNodesData(string hostname)
{
        vector<UserAddressDto> nodeNames{};

        nodeNames = database->getAddresses(&isDecisionMakerCenter, hostname, logger);

        return nodeNames;
}

int main()
{
        logger = unique_ptr<loggernamespace::Logger>(new loggernamespace::Logger());

        logger->initialize();
```

```
	if (logger->isInitialized())
	{
		cerr << "Can't create logger" << endl;

		return 1;
	}

	database = unique_ptr<db::Database>(new db::Database());

	WSADATA wsaData{};
	int response{ WSAStartup(MAKEWORD(2, 2), &wsaData) };
	if (response != 0)
	{
		logger->addLog("WSAStartup failed with error: " + to_string(response));

		return 1;
	}

	char hostname[100]{};

	if (gethostname(hostname, sizeof(hostname)) == -1)
	{
		logger->addLog("Cannot get hostname.");

		WSACleanup();

		exit(1);
	};

#pragma region database

	int idToDelete{};
```

```
	string addressToAdd{};
	vector<UserAddressDto> addresses{};

	//Computer names
	addresses.push_back({ database->generateId(),"Computer1", true });
	addresses.push_back({ database->generateId(),"Computer2", false });
	addresses.push_back({ database->generateId(),"Computer13", true });
	addresses.push_back({ database->generateId(),"Computer7", true });
	addresses.push_back({ database->generateId(),"Computer12", false });
	addresses.push_back({ database->generateId(),"Computer5", false });

	if (database->fileExists("database.dat"))
	{
		logger->addLog("Database file already exists.");
	}
	else
	{
		database->createDatabase(&addresses, logger);
	}

	vector<UserAddressDto> nodesData{ getNodesData(hostname) };

	for (size_t i = 0; i < nodesData.size(); i++)
	{
		nodes[nodesData[i].pcAddress] = nodesData[i];
	}

#pragma endregion

	SOCKET listenSocket{ generateSocket(DEFAULT_PORT, hostname) };

	if (listenSocket == -1)
```

```
{
	WSACleanup();

	return 1;
}

string smt;

bool ifTheOnlyDecisionMaker{ true };

for (size_t i{}; i < nodesData.size(); i++)
{
	logger->addLog("Connecting to:  " + nodesData[i].pcAddress);

	SOCKET nodeSocket{ connectToNode(nodesData[i].pcAddress, DEFAULT_PORT) };

	if (!nodeSocket)
	{
		continue;
	}
	else if (nodesData[i].isDecisionMakerCenter && isDecisionMakerCenter) {
		ifTheOnlyDecisionMaker = false;
	}

	logger->addLog("Connected to " + nodesData[i].pcAddress + " succsessfully");

	DataItemDto sendData{};

	memcpy(sendData.nodeName, hostname, sizeof(sendData.nodeName));

	sendData.a = 10;
	sendData.b = 65;
```

```
sendMessage(nodeSocket, (char*)&sendData, sizeof(DataItemDto));

logger->addLog("Sending message: " +
        to_string(sendData.a) + " " +
        to_string(sendData.b) + " to " +
        nodesData[i].pcAddress);

int receiveData{};
bool receiveIsDecisionMakerNodeActive{};
bool receiveIsDecisionMakerActive{};

if (recv(nodeSocket, (char*)&receiveData, sizeof(int), NULL) <= 0)
{
        logger->addLog("Can`t receive data from: " + nodesData[i].pcAddress);
        closesocket(nodeSocket);
        continue;
}

if (receiveData != sendData.a + sendData.b)
{
        logger->addLog("Invalid response from: " + nodesData[i].pcAddress);
        closesocket(nodeSocket);
        continue;
}

nodes[nodesData[i].pcAddress].activity.isActive = true;

logger->addLog("Message was received");

if (nodesData[i].isDecisionMakerCenter && isDecisionMakerCenter)
{
```

```
                decisionMakersCount++;

        if (recv(nodeSocket, (char*)&receiveIsDecisionMakerNodeActive, sizeof(bool), NULL) <= 0)
                {
                        logger->addLog("Can`t receive data from: " + nodesData[i].pcAddress);
                        closesocket(nodeSocket);
                        continue;
                }

                if (receiveIsDecisionMakerNodeActive)
                {
                if (recv(nodeSocket, (char*)&receiveIsDecisionMakerActive, sizeof(bool), NULL) <= 0)
                        {
                                logger->addLog("Can`t receive data from: " + nodesData[i].pcAddress);
                                closesocket(nodeSocket);
                                continue;
                        }

                        isDecisionMakerCenterActive = receiveIsDecisionMakerActive;
                }
        }

        closesocket(nodeSocket);
}

if (ifTheOnlyDecisionMaker && isDecisionMakerCenter)
{
        isDecisionMakerCenterActive = true;
}

HANDLE connectNewNodeThread
{
```

```
        CreateThread(
                NULL,
                0,
                connectNewNode,
                (LPVOID)&listenSocket,
                0,
                0
        )
};

if (connectNewNodeThread == NULL)
{
        logger->addLog("Can't create thread connectNewNodeThread");

        closesocket(listenSocket);

        WSACleanup();

        return 1;
}
HANDLE decisionMakerCenterThread{};

if (isDecisionMakerCenter)
{
        decisionMakerCenterThread =
                CreateThread(
                        NULL,
                        0,
                        decisionMaker,
                        0,
                        0,
                        0
```

```
                );

        if (decisionMakerCenterThread == NULL)
        {
                logger->addLog("Can't create thread decisionMakerCenterThread");

                exitThread = true;

                WaitForSingleObject(connectNewNodeThread, INFINITE);

                closesocket(listenSocket);

                WSACleanup();

                return 1;
        }

        logger->addLog("Decision maker thread started");
}
getchar();

for (size_t i{}; i < nodesData.size(); i++)
{
        DataItemDto dataToDisconnect{};

        memcpy(dataToDisconnect.nodeName, hostname, sizeof(dataToDisconnect.nodeName));

        if (nodes[nodesData[i].pcAddress].activity.isActive)
        {
                SOCKET nodeSocket{ connectToNode(nodesData[i].pcAddress, DEFAULT_PORT) };

                if (!nodeSocket)
```

```
                {
                        continue;
                }

                logger->addLog("Connected to " + nodesData[i].pcAddress + " succsessfully");

                sendMessage(nodeSocket, (char*)&dataToDisconnect, sizeof(DataItemDto));

                closesocket(nodeSocket);
        }
    }
    exitThread = true;

    WaitForSingleObject(connectNewNodeThread, INFINITE);

    if (isDecisionMakerCenter)
    {
        WaitForSingleObject(decisionMakerCenterThread, INFINITE);
    }

    closesocket(listenSocket);

    WSACleanup();

    return 0;
}
```

ДОДАТОК Г.

РЕЗУЛЬТАТИ ПЕРШОГО ЕКСПЕРИМЕНТУ

Таблиця Г.1

Дата	Номер зміни стану системи	m	Номери компонент з центром системи	$k_1^{E_1}$	$k_2^{E_1}$	$k_3^{E_1}$	k_{Q_1}	E_{11}	k_{Q_2}	E_{11}
10-Dec-2023 00:00:00	0	14	{8, 9, 15, 20, 22, 23, 25, 27, 28, 29, 30, 32, 40, 46}	2	0	0	937845656300	0,999999999998934	7502765250400	0,999999999999867
10-Dec-2023 00:00:00	1	14	{1, 7, 8, 11, 13, 23, 24, 25, 28, 37, 41, 45, 49, 50}	1	0	0	937845656300	0,999999999998934	7502765250400	0,999999999999867
10-Dec-2023 00:00:00	2	14	{1, 2, 11, 13, 15, 16, 19, 21, 22, 28, 37, 47, 49, 50}	2	0	0	937845656300	0,999999999998934	7502765250400	0,999999999999867
10-Dec-2023 01:00:00	3	14	{2, 8, 9, 10, 12, 13, 18, 27, 32, 35, 38, 39, 47, 48}	1	0	0	937845656300	0,999999999998934	7502765250400	0,999999999999867
10-Dec-2023 01:00:00	4	23	{1, 5, 6, 8, 16, 17, 18, 20, 22, 23, 27, 28, 29, 32, 33, 35, 37, 40, 42, 43, 46, 48, 49}	2	0	0	108043253365600	0,999999999999991	864346026924800	0,999999999999999
10-Dec-2023 01:00:00	5	23	{2, 6, 9, 10, 13, 15, 17, 18, 19, 22, 25, 27, 30, 31, 33, 36, 37, 40, 41, 42, 44, 47, 48}	2	0	0	108043253365600	0,999999999999991	864346026924800	0,999999999999999
10-Dec-2023 02:00:00	6	23	{2, 3, 4, 5, 11, 13, 14, 15, 16, 19, 22, 23, 26, 28, 29, 32, 33, 34, 36, 40, 41, 42, 43}	3	0	0	108043253365600	0,999999999999991	864346026924800	0,999999999999999
10-Dec-2023 02:00:00	7	23	{1, 2, 3, 7, 9, 10, 15, 17, 18, 25, 26, 27, 28, 29, 30, 31, 35, 38, 40, 42, 43, 45, 48}	3	0	0	108043253365600	0,999999999999991	864346026924800	0,999999999999999
10-Dec-2023 02:00:00	8	24	{1, 2, 3, 5, 10, 11, 14, 17, 19, 20, 23, 24, 28, 30, 34, 36, 38, 39, 40, 41, 42, 43, 46, 47}	3	0	0	121548660036300	0,999999999999992	972389280290400	0,999999999999999
10-Dec-2023 02:00:00	9	24	{3, 4, 7, 8, 10, 12, 13, 14, 17, 19, 24, 25, 27, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46}	2	0	0	121548660036300	0,999999999999992	972389280290400	0,999999999999999
10-Dec-2023 03:00:00	10	24	{1, 3, 5, 7, 11, 14, 15, 17, 18, 19, 22, 23, 24, 31, 35, 37, 38, 41, 42, 43, 44, 47, 48, 50}	3	0	0	121548660036300	0,999999999999992	972389280290400	0,999999999999999
10-Dec-2023 03:00:00	11	24	{7, 8, 9, 11, 12, 14, 16, 18, 20, 21, 22, 23, 24, 25, 28, 29, 34, 35, 38, 39, 42, 44, 45, 49}	2	0	0	121548660036300	0,999999999999992	972389280290400	0,999999999999999
10-Dec-2023 03:00:00	12	24	{1, 2, 3, 4, 6, 7, 8, 10, 12, 13, 15, 26, 27, 28, 32, 34, 35, 36, 38, 39, 43, 45, 47, 49}	2	0	0	121548660036300	0,999999999999992	972389280290400	0,999999999999999
10-Dec-2023 03:00:00	13	24	{3, 5, 11, 14, 18, 19, 20, 22, 24, 25, 26, 28, 29, 30, 31, 32, 34, 38, 39, 40, 41, 44, 46, 50}	1	0	0	121548660036300	0,999999999999992	972389280290400	0,999999999999999
10-Dec-2023 04:00:00	14	24	{1, 3, 4, 5, 6, 7, 11, 16, 17, 20, 22, 26, 27, 28, 31, 33, 37, 38, 43, 45, 46, 47, 48, 49}	1	0	0	121548660036300	0,999999999999992	972389280290400	0,999999999999999
10-Dec-2023 04:00:00	15	24	{1, 4, 6, 7, 8, 15, 18, 20, 24, 26, 27, 29, 30, 33, 34, 37, 39, 41, 42, 45, 46, 47, 49, 50}	1	0	0	121548660036300	0,999999999999992	972389280290400	0,999999999999999
10-Dec-2023 05:00:00	16	24	{1, 5, 7, 8, 9, 13, 17, 19, 20, 21, 22, 25, 27, 30, 32, 33, 37, 38, 39, 40, 41, 43, 46, 48}	3	0	0	121548660036300	0,999999999999992	972389280290400	0,999999999999999
10-Dec-2023 05:00:00	17	18	{2, 9, 11, 12, 13, 14, 15, 16, 19, 28, 38, 39, 40, 41, 42, 46, 48, 50}	1	0	0	18053528883775	0,999999999999945	144428231070200	0,999999999999993
10-Dec-2023 05:00:00	18	18	{1, 7, 9, 13, 14, 16, 19, 22, 24, 25, 28, 33, 35, 37, 41, 44, 47, 48}	3	0	0	18053528883775	0,999999999999945	144428231070200	0,999999999999993
10-Dec-2023 05:00:00	19	18	{2, 3, 8, 13, 16, 17, 18, 21, 22, 23, 28, 30, 31, 32, 33, 34, 44, 50}	1	0	0	18053528883775	0,999999999999945	144428231070200	0,999999999999993
10-Dec-2023 06:00:00	20	18	{1, 3, 7, 10, 11, 14, 21, 23, 25, 28, 29, 34, 35, 36, 41, 43, 44, 45}	3	0	0	18053528883775	0,999999999999945	144428231070200	0,999999999999993
10-Dec-2023 06:00:00	21	18	{1, 3, 5, 6, 7, 8, 9, 10, 11, 15, 25, 27, 30, 31, 32, 43, 49, 50}	2	0	0	18053528883775	0,999999999999945	144428231070200	0,999999999999993
10-Dec-2023 06:00:00	22	3	{27, 32, 36}	1	0	0	19600	0,999948979591837	156800	0,999993622448980
10-Dec-2023 07:00:00	23	3	{11, 38, 50}	3	0	0	19600	0,999948979591837	156800	0,999993622448980
10-Dec-2023 07:00:00	24	3	{19, 42, 44}	5	0	0	19600	0,999948979591837	156800	0,999993622448980
10-Dec-2023 07:00:00	25	3	{16, 21, 30}	1	0	0	19600	0,999948979591837	156800	0,999993622448980
10-Dec-2023 07:00:00	26	3	{29, 45, 46}	1	0	0	19600	0,999948979591837	156800	0,999993622448980
10-Dec-2023 08:00:00	27	7	{2, 17, 22, 24, 32, 42, 43}	2	0	0	99884400	0,999999989988427	799075200	0,999999998748553
10-Dec-2023 08:00:00	28	7	{3, 9, 10, 16, 17, 19, 25}	2	0	0	99884400	0,999999989988427	799075200	0,999999998748553
10-Dec-2023 08:00:00	29	22	{2, 3, 4, 9, 12, 13, 14, 17, 19, 20, 23, 25, 27, 29, 30, 31, 33, 34, 37, 45, 46, 49}	2	0	0	88749815264600	0,999999999999989	709998522116800	0,999999999999999
10-Dec-2023 09:00:00	30	22	{6, 7, 9, 11, 12, 13, 20, 22, 23, 24, 25, 26, 27, 29, 31, 33, 34, 37, 39, 40, 48, 50}	3	0	1	88749815264600	0,999999999999989	709998522116800	0,999999999999999
10-Dec-2023 09:00:00	31	22	{3, 9, 11, 12, 15, 17, 18, 21, 27, 28, 29, 31, 34, 36, 37, 38, 39, 44, 45, 47, 48, 49}	2	0	0	88749815264600	0,999999999999989	709998522116800	0,999999999999999
10-Dec-2023 10:00:00	32	22	{2, 3, 4, 5, 6, 9, 10, 13, 14, 16, 18, 19, 21, 25, 27, 33, 36, 37, 39, 40, 49, 50}	2	0	0	88749815264600	0,999999999999989	709998522116800	0,999999999999999
10-Dec-2023 10:00:00	33	22	{8, 9, 11, 12, 14, 15, 18, 23, 25, 26, 33, 36, 37, 38, 39, 40, 45, 46, 47, 48, 49, 50}	2	0	0	88749815264600	0,999999999999989	709998522116800	0,999999999999999
10-Dec-2023 10:00:00	34	20	{2, 3, 4, 5, 8, 9, 10, 16, 18, 26, 30, 31, 34, 37, 38, 40, 43, 46, 48, 49}	3	0	0	47129212243960	0,999999999999979	377033697951680	0,999999999999997
10-Dec-2023 11:00:00	35	20	{1, 3, 8, 9, 10, 11, 14, 18, 20, 21, 24, 29, 35, 36, 38, 39, 42, 43, 46, 50}	1	0	0	47129212243960	0,999999999999979	377033697951680	0,999999999999997
10-Dec-2023 11:00:00	36	20	{1, 3, 7, 15, 16, 18, 19, 21, 22, 28, 30, 31, 37, 38, 42, 45, 46, 47, 49, 50}	2	0	0	47129212243960	0,999999999999979	377033697951680	0,999999999999997
10-Dec-2023 12:00:00	37	2	{4, 47}	2	0	0	1225	0,999183673469388	9800	0,999897959183674
10-Dec-2023 12:00:00	38	2	{25, 38}	2	0	0	1225	0,999183673469388	9800	0,999897959183674
10-Dec-2023 13:00:00	39	2	{38, 41}	1	0	0	1225	0,999183673469388	9800	0,999897959183674
10-Dec-2023 13:00:00	40	2	{8, 23}	2	0	0	1225	0,999183673469388	9800	0,999897959183674
10-Dec-2023 14:00:00	41	2	{31, 46}	2	0	0	1225	0,999183673469388	9800	0,999897959183674
10-Dec-2023 14:00:00	42	11	{2, 7, 18, 28, 31, 39, 40, 42, 44, 45, 48}	1	0	0	37353738800	0,999999999973229	298829910400	0,999999999996654
10-Dec-2023 14:00:00	43	11	{4, 8, 13, 16, 17, 19, 27, 30, 42, 46, 50}	3	0	0	37353738800	0,999999999973229	298829910400	0,999999999996654
10-Dec-2023 14:00:00	44	11	{12, 13, 14, 17, 22, 32, 35, 37, 38, 43, 49}	1	0	0	37353738800	0,999999999973229	298829910400	0,999999999996654
10-Dec-2023 15:00:00	45	11	{11, 13, 17, 21, 23, 27, 31, 32, 40, 41, 42}	1	0	0	37353738800	0,999999999973229	298829910400	0,999999999996654
10-Dec-2023 15:00:00	46	11	{1, 3, 12, 17, 26, 28, 35, 38, 40, 43, 50}	2	0	0	37353738800	0,999999999973229	298829910400	0,999999999996654
10-Dec-2023 16:00:00	47	2	{11, 28}	1	0	0	1225	0,999183673469388	9800	0,999897959183674
10-Dec-2023 16:00:00	48	2	{11, 16}	3	0	0	1225	0,999183673469388	9800	0,999897959183674
10-Dec-2023 16:00:00	49	2	{5, 37}	2	0	0	1225	0,999183673469388	9800	0,999897959183674
10-Dec-2023 17:00:00	50	2	{27, 38}	2	0	0	1225	0,999183673469388	9800	0,999897959183674
10-Dec-2023 17:00:00	51	2	{17, 41}	2	0	0	1225	0,999183673469388	9800	0,999897959183674
10-Dec-2023 17:00:00	52	2	{28, 47}	2	0	0	1225	0,999183673469388	9800	0,999897959183674
10-Dec-2023 17:00:00	53	2	{18, 45}	1	0	0	1225	0,999183673469388	9800	0,999897959183674
10-Dec-2023 18:00:00	54	12	{2, 6, 15, 17, 18, 28, 30, 32, 35, 37, 38, 48}	1	0	0	121399651100	0,999999999991763	971197208800	0,999999999998970
10-Dec-2023 18:00:00	55	12	{3, 7, 11, 12, 13, 16, 25, 33, 38, 40, 44, 47}	3	0	0	121399651100	0,999999999991763	971197208800	0,999999999998970
10-Dec-2023 18:00:00	56	12	{3, 9, 10, 15, 17, 18, 25, 28, 38, 39, 45, 48}	1	0	0	121399651100	0,999999999991763	971197208800	0,999999999998970
10-Dec-2023 19:00:00	57	12	{1, 11, 13, 22, 24, 29, 32, 33, 35, 37, 42, 47}	2	0	0	121399651100	0,999999999991763	971197208800	0,999999999998970
10-Dec-2023 19:00:00	58	20	{1, 3, 4, 9, 18, 21, 22, 23, 26, 29, 30, 32, 35, 36, 41, 44, 45, 48, 49, 50}	5	0	0	47129212243960	0,999999999999979	377033697951680	0,999999999999997
10-Dec-2023 20:00:00	59	20	{5, 8, 10, 12, 13, 15, 19, 20, 23, 30, 32, 33, 39, 42, 43, 45, 46, 47, 49, 50}	2	0	0	47129212243960	0,999999999999979	377033697951680	0,999999999999997
10-Dec-2023 20:00:00	60	20	{1, 3, 6, 8, 15, 16, 18, 22, 23, 24, 27, 28, 33, 36, 39, 44, 45, 46, 47, 49}	3	0	0	47129212243960	0,999999999999979	377033697951680	0,999999999999997
10-Dec-2023 20:00:00	61	20	{2, 4, 5, 7, 11, 16, 17, 23, 25, 26, 28, 30, 33, 38, 41, 42, 44, 47, 48, 50}	3	0	0	47129212243960	0,999999999999979	377033697951680	0,999999999999997
10-Dec-2023 21:00:00	62	20	{2, 5, 6, 8, 9, 12, 14, 15, 19, 20, 22, 23, 27, 31, 32, 37, 38, 43, 46, 48}	2	0	0	47129212243960	0,999999999999979	377033697951680	0,999999999999997
10-Dec-2023 21:00:00	63	20	{4, 6, 7, 12, 15, 18, 19, 20, 21, 24, 26, 31, 32, 33, 40, 41, 45, 46, 47, 49}	3	0	0	47129212243960	0,999999999999979	377033697951680	0,999999999999997
10-Dec-2023 21:00:00	64	20	{1, 2, 3, 4, 11, 12, 15, 16, 19, 20, 24, 25, 26, 30, 33, 35, 36, 37, 48, 49}	2	0	0	47129212243960	0,999999999999979	377033697951680	0,999999999999997
10-Dec-2023 21:00:00	65	2	{15, 37}	3	0	0	1225	0,999183673469388	9800	0,999897959183674
10-Dec-2023 22:00:00	66	2	{21, 28}	2	0	0	1225	0,999183673469388	9800	0,999897959183674
10-Dec-2023 22:00:00	67	2	{14, 37}	2	0	0	1225	0,999183673469388	9800	0,999897959183674
10-Dec-2023 23:00:00	68	2	{36, 45}	2	0	0	1225	0,999183673469388	9800	0,999897959183674
10-Dec-2023 23:00:00	69	19	{2, 3, 10, 11, 12, 14, 17, 19, 23, 25, 27, 33, 34, 39, 44, 45, 46, 47, 48}	2	0	0	30405943383200	0,999999999999967	243247547065600	0,999999999999996
10-Dec-2023 23:00:00	70	19	{1, 2, 3, 4, 7, 8, 9, 14, 15, 17, 20, 21, 34, 37, 38, 42, 43, 45, 48}	2	0	1	30405943383200	0,999999999999967	243247547065600	0,999999999999996
11-Dec-2023 00:00:00	71	5	{6, 10, 30, 39, 41}	1	0	0	2118760	0,999999528025827	16950080	0,999999941003228
11-Dec-2023 00:00:00	72	5	{1, 25, 26, 28, 44}	2	0	0	2118760	0,999999528025827	16950080	0,999999941003228

Продовження таблиці Г.1

11-Dec-2023 00:00:00	73	5	{4, 19, 34, 39, 42}	2	0	0	2118760	0,9999995528025827	16950080	0,999999941003228
11-Dec-2023 01:00:00	74	5	{9, 17, 18, 39, 42}	3	0	0	2118760	0,9999995528025827	16950080	0,999999941003228
11-Dec-2023 01:00:00	75	16	{1, 2, 6, 8, 10, 11, 13, 15, 17, 18, 19, 20, 28, 35, 44, 50}	2	0	0	4923689695575	0,9999999999999797	39389517564600	0,9999999999999975
11-Dec-2023 01:00:00	76	16	{2, 3, 5, 7, 9, 15, 17, 18, 20, 21, 26, 34, 38, 39, 45, 48}	2	0	0	4923689695575	0,9999999999999797	39389517564600	0,9999999999999975
11-Dec-2023 01:00:00	77	16	{3, 7, 8, 12, 13, 17, 21, 23, 28, 30, 34, 37, 42, 47, 48, 49}	3	0	0	4923689695575	0,9999999999999797	39389517564600	0,9999999999999975
11-Dec-2023 02:00:00	78	19	{1, 2, 8, 9, 10, 12, 14, 17, 19, 20, 22, 28, 30, 32, 34, 37, 38, 46, 50}	2	0	0	30405943383200	0,9999999999999967	243247547065600	0,9999999999999996
11-Dec-2023 02:00:00	79	19	{5, 6, 14, 18, 19, 20, 23, 25, 26, 27, 33, 34, 35, 38, 42, 43, 45, 47, 48}	2	0	0	30405943383200	0,9999999999999967	243247547065600	0,9999999999999996
11-Dec-2023 03:00:00	80	24	{1, 4, 10, 13, 14, 18, 20, 22, 23, 25, 29, 30, 31, 32, 33, 34, 37, 38, 40, 41, 45, 47, 48, 50}	2	0	0	121548660036300	0,9999999999999992	972389280290400	0,9999999999999999
11-Dec-2023 03:00:00	81	24	{2, 3, 5, 6, 7, 10, 12, 13, 14, 15, 21, 23, 26, 31, 32, 35, 37, 38, 39, 41, 46, 48, 49, 50}	2	0	0	121548660036300	0,9999999999999992	972389280290400	0,9999999999999999
11-Dec-2023 03:00:00	82	20	{5, 7, 11, 13, 15, 16, 18, 19, 24, 26, 33, 34, 36, 39, 40, 43, 45, 46, 47, 49}	2	0	0	47129212243960	0,9999999999999979	377033697951680	0,9999999999999997
11-Dec-2023 03:00:00	83	20	{2, 3, 5, 10, 12, 13, 15, 19, 20, 25, 26, 28, 34, 37, 41, 42, 45, 46, 48, 49}	3	0	0	47129212243960	0,9999999999999979	377033697951680	0,9999999999999997
11-Dec-2023 04:00:00	84	8	{4, 6, 16, 21, 25, 42, 48, 49}	2	0	0	536878650	0,9999999998137382	4295029200	0,9999999999767173
11-Dec-2023 04:00:00	85	8	{3, 4, 7, 18, 21, 29, 34, 44}	2	0	0	536878650	0,9999999998137382	4295029200	0,9999999999767173
11-Dec-2023 04:00:00	86	15	{4, 7, 10, 12, 13, 15, 19, 23, 26, 34, 35, 39, 46, 48, 49}	3	0	0	2250829575120	0,9999999999999556	18006636600960	0,9999999999999945
11-Dec-2023 04:00:00	87	15	{2, 6, 7, 9, 10, 18, 35, 36, 42, 44, 46, 47, 48, 49, 50}	1	0	0	2250829575120	0,9999999999999556	18006636600960	0,9999999999999945
11-Dec-2023 05:00:00	88	15	{2, 4, 15, 25, 26, 27, 30, 34, 36, 38, 39, 41, 43, 46, 47}	3	0	0	2250829575120	0,9999999999999556	18006636600960	0,9999999999999945
11-Dec-2023 05:00:00	89	15	{2, 3, 4, 13, 14, 20, 25, 28, 30, 33, 36, 39, 44, 47, 50}	2	0	0	2250829575120	0,9999999999999556	18006636600960	0,9999999999999945
11-Dec-2023 06:00:00	90	15	{1, 2, 7, 8, 10, 13, 21, 22, 30, 34, 35, 36, 44, 47, 48}	3	0	0	2250829575120	0,9999999999999556	18006636600960	0,9999999999999945
11-Dec-2023 06:00:00	91	15	{3, 7, 11, 14, 21, 23, 25, 26, 30, 31, 36, 40, 44, 46, 50}	3	0	0	2250829575120	0,9999999999999556	18006636600960	0,9999999999999945
11-Dec-2023 06:00:00	92	17	{2, 5, 7, 8, 10, 11, 12, 16, 17, 20, 22, 29, 31, 45, 48, 49, 50}	1	0	1	9847379391150	0,9999999999999898	78779035129200	0,9999999999999987
11-Dec-2023 07:00:00	93	17	{2, 4, 14, 16, 19, 20, 28, 29, 30, 32, 34, 39, 40, 42, 45, 47, 48}	2	0	0	9847379391150	0,9999999999999898	78779035129200	0,9999999999999987
11-Dec-2023 07:00:00	94	17	{1, 4, 6, 11, 13, 16, 17, 18, 19, 24, 27, 28, 41, 44, 46, 48, 49}	2	0	0	9847379391150	0,9999999999999898	78779035129200	0,9999999999999987
11-Dec-2023 07:00:00	95	18	{3, 5, 9, 11, 14, 18, 19, 21, 22, 23, 29, 31, 33, 42, 45, 46, 48, 50}	2	0	0	18053528883775	0,9999999999999945	144428231070200	0,9999999999999993
11-Dec-2023 07:00:00	96	18	{1, 7, 9, 10, 14, 15, 20, 24, 25, 28, 29, 32, 39, 40, 43, 44, 46, 49}	2	0	0	18053528883775	0,9999999999999945	144428231070200	0,9999999999999993
11-Dec-2023 08:00:00	97	18	{3, 8, 9, 11, 12, 16, 17, 20, 21, 27, 32, 33, 34, 38, 39, 42, 43, 48}	1	0	0	18053528883775	0,9999999999999945	144428231070200	0,9999999999999993
11-Dec-2023 08:00:00	98	18	{1, 2, 4, 5, 7, 13, 16, 19, 22, 31, 32, 34, 36, 37, 39, 44, 46, 49}	2	0	0	18053528883775	0,9999999999999945	144428231070200	0,9999999999999993
11-Dec-2023 09:00:00	99	18	{7, 11, 15, 18, 20, 21, 22, 23, 27, 29, 30, 31, 34, 35, 41, 42, 47, 48}	2	0	0	18053528883775	0,9999999999999945	144428231070200	0,9999999999999993
11-Dec-2023 09:00:00	100	24	{2, 3, 5, 6, 7, 10, 11, 14, 18, 19, 22, 27, 28, 29, 31, 33, 34, 36, 39, 42, 43, 45, 48, 50}	2	0	0	121548660036300	0,9999999999999992	972389280290400	0,9999999999999999
11-Dec-2023 09:00:00	101	24	{3, 6, 11, 13, 14, 16, 18, 20, 23, 25, 28, 32, 33, 35, 36, 37, 39, 40, 41, 42, 47, 48, 49, 50}	2	0	0	121548660036300	0,9999999999999992	972389280290400	0,9999999999999999
11-Dec-2023 10:00:00	102	24	{1, 2, 4, 5, 7, 8, 12, 13, 15, 18, 20, 24, 25, 26, 27, 28, 31, 32, 37, 39, 40, 44, 48, 49}	3	0	0	121548660036300	0,9999999999999992	972389280290400	0,9999999999999999
11-Dec-2023 10:00:00	103	24	{1, 3, 6, 10, 12, 14, 15, 17, 19, 21, 22, 23, 25, 27, 31, 32, 33, 34, 36, 40, 43, 45, 47, 50}	1	0	0	121548660036300	0,9999999999999992	972389280290400	0,9999999999999999
11-Dec-2023 11:00:00	104	24	{1, 3, 4, 5, 6, 11, 12, 15, 17, 22, 24, 25, 28, 29, 31, 37, 38, 40, 42, 43, 44, 45, 46, 48}	2	0	0	121548660036300	0,9999999999999992	972389280290400	0,9999999999999999
11-Dec-2023 11:00:00	105	19	{1, 3, 4, 9, 12, 19, 20, 23, 24, 28, 30, 32, 36, 38, 39, 40, 41, 42, 49}	3	0	0	30405943383200	0,9999999999999967	243247547065600	0,9999999999999996
11-Dec-2023 12:00:00	106	19	{4, 8, 9, 10, 11, 12, 14, 16, 19, 21, 28, 29, 30, 33, 35, 41, 42, 44, 50}	2	0	0	30405943383200	0,9999999999999967	243247547065600	0,9999999999999996
11-Dec-2023 12:00:00	107	4	{5, 12, 19, 50}	3	0	0	230300	0,9999995657837603	1842400	0,9999994572297 00
11-Dec-2023 12:00:00	108	4	{24, 26, 28, 47}	2	0	0	230300	0,9999995657837603	1842400	0,9999994572297 00
11-Dec-2023 13:00:00	109	12	{1, 12, 13, 14, 17, 21, 23, 31, 32, 37, 45, 49}	2	0	0	121399651100	0,9999999999991763	971197208800	0,9999999999989 70
11-Dec-2023 13:00:00	110	12	{4, 7, 13, 14, 15, 23, 25, 27, 29, 34, 47, 48}	2	0	0	121399651100	0,9999999999991763	971197208800	0,9999999999989 70
11-Dec-2023 14:00:00	111	12	{10, 16, 18, 26, 28, 30, 31, 36, 41, 46, 47, 48}	3	0	0	121399651100	0,9999999999991763	971197208800	0,9999999999989 70
11-Dec-2023 14:00:00	112	12	{2, 9, 12, 14, 16, 24, 28, 31, 36, 38, 47, 49}	1	0	0	121399651100	0,9999999999991763	971197208800	0,9999999999989 70
11-Dec-2023 14:00:00	113	12	{2, 8, 10, 11, 13, 15, 24, 29, 30, 38, 45, 47}	1	0	0	121399651100	0,9999999999991763	971197208800	0,9999999999989 70
11-Dec-2023 14:00:00	114	17	{3, 4, 10, 11, 13, 18, 19, 20, 31, 32, 35, 38, 40, 41, 43, 44, 46}	2	0	0	9847379391150	0,9999999999999898	78779035129200	0,9999999999999987
11-Dec-2023 15:00:00	115	17	{2, 8, 10, 13, 19, 21, 22, 23, 25, 27, 32, 34, 35, 39, 41, 42, 46}	2	0	0	9847379391150	0,9999999999999898	78779035129200	0,9999999999999987
11-Dec-2023 15:00:00	116	17	{4, 9, 13, 15, 17, 20, 22, 24, 29, 30, 34, 41, 42, 45, 46, 47, 49}	1	0	0	9847379391150	0,9999999999999898	78779035129200	0,9999999999999987
11-Dec-2023 15:00:00	117	17	{2, 3, 4, 7, 10, 11, 14, 18, 24, 25, 30, 32, 37, 47, 48, 49, 50}	3	0	0	9847379391150	0,9999999999999898	78779035129200	0,9999999999999987
11-Dec-2023 15:00:00	118	10	{2, 9, 11, 13, 14, 17, 20, 36, 49, 50}	3	0	0	10272278170	0,9999999999902651	82178225360	0,9999999999987831
11-Dec-2023 16:00:00	119	10	{6, 18, 25, 27, 28, 38, 42, 45, 47, 50}	2	0	0	10272278170	0,9999999999902651	82178225360	0,9999999999987831
11-Dec-2023 16:00:00	120	10	{4, 5, 10, 11, 18, 31, 33, 43, 46, 50}	2	0	0	10272278170	0,9999999999902651	82178225360	0,9999999999987831
11-Dec-2023 16:00:00	121	10	{24, 26, 27, 28, 30, 36, 37, 39, 45, 48}	2	0	0	10272278170	0,9999999999902651	82178225360	0,9999999999987831
11-Dec-2023 17:00:00	122	10	{4, 10, 12, 17, 22, 29, 31, 34, 36, 47}	2	0	0	10272278170	0,9999999999902651	82178225360	0,9999999999987831
11-Dec-2023 17:00:00	123	10	{8, 16, 19, 21, 22, 23, 28, 30, 31, 37}	1	0	0	10272278170	0,9999999999902651	82178225360	0,9999999999987831
11-Dec-2023 18:00:00	124	11	{9, 17, 19, 22, 25, 33, 38, 41, 45, 46, 48}	2	0	0	37353738800	0,9999999999973229	298829910400	0,9999999999966654
11-Dec-2023 18:00:00	125	11	{2, 5, 8, 9, 12, 23, 30, 38, 44, 45, 50}	2	0	1	37353738800	0,9999999999973229	298829910400	0,9999999999966654
11-Dec-2023 19:00:00	126	11	{1, 3, 6, 13, 21, 25, 26, 36, 39, 44, 49}	1	0	0	37353738800	0,9999999999973229	298829910400	0,9999999999966654
11-Dec-2023 19:00:00	127	11	{7, 17, 22, 26, 27, 29, 35, 37, 38, 47, 48}	1	0	0	37353738800	0,9999999999973229	298829910400	0,9999999999966654
11-Dec-2023 19:00:00	128	11	{4, 8, 9, 15, 20, 24, 27, 31, 37, 40, 43}	3	0	0	37353738800	0,9999999999973229	298829910400	0,9999999999966654
11-Dec-2023 20:00:00	129	21	{3, 4, 5, 7, 8, 9, 16, 22, 28, 29, 32, 34, 38, 39, 40, 41, 46, 47, 48, 49, 50}	2	0	0	67327446062800	0,9999999999999985	538619568502400	0,9999999999999998
11-Dec-2023 20:00:00	130	21	{1, 6, 10, 11, 15, 16, 21, 22, 25, 26, 27, 30, 33, 34, 38, 41, 43, 45, 46, 47, 50}	2	0	0	67327446062800	0,9999999999999985	538619568502400	0,9999999999999998
11-Dec-2023 20:00:00	131	21	{6, 7, 10, 11, 14, 15, 18, 19, 20, 25, 26, 28, 29, 36, 38, 39, 41, 46, 48, 49, 50}	1	0	0	67327446062800	0,9999999999999985	538619568502400	0,9999999999999998
11-Dec-2023 20:00:00	132	21	{3, 4, 7, 8, 10, 11, 15, 17, 22, 26, 30, 32, 34, 35, 38, 39, 40, 44, 46, 47, 50}	1	0	0	67327446062800	0,9999999999999985	538619568502400	0,9999999999999998
11-Dec-2023 21:00:00	133	21	{3, 5, 7, 8, 10, 15, 16, 18, 19, 22, 23, 25, 26, 27, 31, 32, 34, 37, 44, 47, 49}	3	0	0	67327446062800	0,9999999999999985	538619568502400	0,9999999999999998
11-Dec-2023 21:00:00	134	21	{8, 10, 11, 12, 13, 17, 18, 20, 21, 22, 25, 27, 28, 29, 35, 36, 39, 41, 42, 45, 49}	2	0	0	67327446062800	0,9999999999999985	538619568502400	0,9999999999999998
11-Dec-2023 21:00:00	135	16	{5, 7, 8, 9, 15, 17, 18, 23, 24, 30, 33, 40, 42, 43, 45, 46}	1	0	0	4923689695575	0,9999999999999797	39389517564600	0,9999999999999975
11-Dec-2023 21:00:00	136	16	{3, 6, 10, 12, 14, 16, 18, 19, 20, 26, 30, 31, 35, 37, 43, 44}	2	0	0	4923689695575	0,9999999999999797	39389517564600	0,9999999999999975
11-Dec-2023 22:00:00	137	14	{4, 16, 18, 20, 21, 24, 29, 31, 32, 38, 39, 41, 48, 50}	2	0	0	937845656300	0,9999999999998934	7502765250400	0,9999999999999867
11-Dec-2023 22:00:00	138	14	{1, 2, 5, 6, 9, 11, 12, 14, 18, 19, 27, 34, 48, 50}	3	0	0	937845656300	0,9999999999998934	7502765250400	0,9999999999999867
11-Dec-2023 22:00:00	139	14	{3, 4, 5, 8, 10, 11, 16, 18, 26, 29, 37, 39, 42, 50}	1	0	0	937845656300	0,9999999999998934	7502765250400	0,9999999999999867
11-Dec-2023 22:00:00	140	14	{5, 12, 13, 14, 15, 16, 22, 32, 40, 44, 45, 47, 48, 50}	2	0	0	937845656300	0,9999999999998934	7502765250400	0,9999999999999867
11-Dec-2023 23:00:00	141	14	{1, 2, 4, 12, 14, 15, 16, 17, 20, 26, 27, 29, 38, 48}	2	0	0	937845656300	0,9999999999998934	7502765250400	0,9999999999999867
11-Dec-2023 23:00:00	142	8	{3, 16, 22, 25, 32, 34, 35, 40}	2	0	0	536878650	0,9999999998137382	4295029200	0,9999999999767173
11-Dec-2023 23:00:00	143	8	{21, 22, 24, 27, 31, 38, 45, 47}	1	0	0	536878650	0,9999999998137382	4295029200	0,9999999999767173
11-Dec-2023 23:00:00	144	8	{5, 9, 23, 26, 32, 36, 38, 47}	3	0	0	536878650	0,9999999998137382	4295029200	0,9999999999767173
12-Dec-2023 00:00:00	145	8	{1, 15, 23, 26, 27, 33, 40, 47}	1	0	0	536878650	0,9999999998137382	4295029200	0,9999999999767173
12-Dec-2023 00:00:00	146	8	{5, 6, 7, 25, 31, 35, 47, 49}	2	0	0	536878650	0,9999999998137382	4295029200	0,9999999999767173
12-Dec-2023 01:00:00	147	8	{3, 4, 10, 15, 20, 39, 41, 44}	2	0	0	536878650	0,9999999998137382	4295029200	0,9999999999767173
12-Dec-2023 01:00:00	148	8	{1, 6, 11, 23, 29, 47, 48, 50}	2	0	0	536878650	0,9999999998137382	4295029200	0,9999999999767173
12-Dec-2023 02:00:00	149	4	{1, 5, 10, 18}	2	0	0	230300	0,9999995657837603	1842400	0,9999994572297 00
12-Dec-2023 02:00:00	150	4	{15, 24, 31, 35}	2	0	0	230300	0,9999995657837603	1842400	0,9999994572297 00
12-Dec-2023 03:00:00	151	4	{13, 21, 25, 42}	1	0	0	230300	0,9999995657837603	1842400	0,9999994572297 00
12-Dec-2023 03:00:00	152	17	{2, 11, 12, 14, 15, 16, 17, 18, 25, 27, 29, 33, 40, 45, 46, 47, 49}	3	1	0	9847379391150	0,9999999999999898	78779035129200	0,9999999999999987
12-Dec-2023 04:00:00	153	17	{2, 3, 6, 9, 11, 12, 16, 18, 26, 29, 35, 37, 40, 42, 43, 45, 49}	3	0	0	9847379391150	0,9999999999999898	78779035129200	0,9999999999999987
12-Dec-2023 04:00:00	154	17	{1, 5, 6, 13, 15, 16, 21, 22, 25, 28, 32, 33, 37, 45, 47, 49, 50}	1	0	0	9847379391150	0,9999999999999898	78779035129200	0,9999999999999987
12-Dec-2023 04:00:00	155	17	{3, 4, 5, 6, 9, 18, 22, 25, 29, 33, 34, 38, 39, 42, 45, 47, 48}	2	0	0	9847379391150	0,9999999999999898	78779035129200	0,9999999999999987
12-Dec-2023 04:00:00	156	17	{1, 2, 5, 7, 9, 14, 16, 19, 21, 26, 28, 32, 36, 38, 39, 48, 50}	1	0	0	9847379391150	0,9999999999999898	78779035129200	0,9999999999999987
12-Dec-2023 05:00:00	157	17	{4, 8, 9, 11, 12, 16, 17, 19, 21, 26, 27, 31, 37, 38, 43, 47, 50}	3	0	0	9847379391150	0,9999999999999898	78779035129200	0,9999999999999987
12-Dec-2023 05:00:00	158	17	{2, 4, 5, 6, 11, 16, 19, 23, 30, 31, 37, 38, 40, 43, 44, 46, 49}	3	0	0	9847379391150	0,9999999999999898	78779035129200	0,9999999999999987
12-Dec-2023 06:00:00	159	17	{1, 3, 4, 6, 8, 14, 16, 19, 21, 24, 26, 34, 36, 40, 43, 44, 50}	2	0	0	9847379391150	0,9999999999999898	78779035129200	0,9999999999999987
12-Dec-2023 06:00:00	160	17	{5, 6, 9, 12, 15, 16, 18, 23, 25, 27, 31, 33, 41, 45, 47, 49, 50}	3	0	0	9847379391150	0,9999999999999898	78779035129200	0,9999999999999987
12-Dec-2023 07:00:00	161	17	{2, 3, 9, 10, 11, 17, 19, 20, 24, 27, 28, 34, 35, 39, 41, 45, 48}	2	0	0	9847379391150	0,9999999999999898	78779035129200	0,9999999999999987

Продовження таблиці Г.1

12-Dec-2023 07:00:00	162	17	{4, 10, 12, 15, 18, 24, 26, 32, 34, 37, 38, 39, 40, 43, 45, 48, 50}	2	0	0	9847379391150	0,9999999999999898	78779035129200	0,9999999999999987
12-Dec-2023 07:00:00	163	17	{4, 5, 7, 8, 11, 18, 24, 29, 30, 33, 36, 38, 40, 41, 46, 47, 48}	1	0	0	9847379391150	0,9999999999999898	78779035129200	0,9999999999999987
12-Dec-2023 08:00:00	164	17	{3, 5, 7, 13, 14, 15, 16, 19, 23, 25, 29, 33, 37, 40, 41, 45, 50}	3	0	0	9847379391150	0,9999999999999898	78779035129200	0,9999999999999987
12-Dec-2023 08:00:00	165	17	{1, 2, 6, 8, 10, 11, 18, 20, 21, 23, 25, 28, 30, 37, 39, 48, 50}	3	0	0	9847379391150	0,9999999999999898	78779035129200	0,9999999999999987
12-Dec-2023 09:00:00	166	17	{3, 10, 12, 14, 16, 18, 22, 24, 25, 28, 38, 39, 40, 42, 43, 47, 49}	3	0	0	9847379391150	0,9999999999999898	78779035129200	0,9999999999999987
12-Dec-2023 09:00:00	167	19	{1, 4, 7, 10, 11, 15, 16, 17, 18, 19, 20, 22, 27, 33, 38, 39, 43, 44, 49}	2	0	0	30405943383200	0,9999999999999967	243247547065600	0,9999999999999996
12-Dec-2023 10:00:00	168	19	{1, 6, 7, 10, 12, 19, 20, 21, 22, 26, 28, 33, 34, 37, 40, 42, 43, 48, 49}	2	0	1	30405943383200	0,9999999999999967	243247547065600	0,9999999999999996
12-Dec-2023 10:00:00	169	19	{4, 9, 10, 11, 12, 15, 18, 19, 23, 25, 26, 30, 32, 36, 38, 41, 42, 45, 50}	2	0	0	30405943383200	0,9999999999999967	243247547065600	0,9999999999999996
12-Dec-2023 11:00:00	170	19	{3, 5, 6, 8, 9, 11, 12, 13, 21, 22, 25, 28, 30, 31, 33, 35, 39, 47, 50}	2	0	0	30405943383200	0,9999999999999967	243247547065600	0,9999999999999996
12-Dec-2023 11:00:00	171	25	{1, 3, 8, 15, 16, 17, 18, 20, 22, 25, 26, 27, 28, 29, 30, 31, 34, 35, 36, 39, 41, 44, 45, 48, 50}	1	0	0	126410606437752	0,9999999999999992	1011284851502020	0,9999999999999999
12-Dec-2023 12:00:00	172	25	{1, 2, 4, 6, 8, 11, 12, 14, 17, 21, 22, 25, 27, 31, 33, 36, 40, 41, 42, 43, 45, 46, 47, 49, 50}	2	0	0	126410606437752	0,9999999999999992	1011284851502020	0,9999999999999999
12-Dec-2023 12:00:00	173	25	{3, 5, 6, 7, 10, 12, 13, 14, 15, 17, 18, 21, 27, 28, 30, 34, 35, 36, 37, 39, 45, 46, 47, 48, 49}	2	0	0	126410606437752	0,9999999999999992	1011284851502020	0,9999999999999999
12-Dec-2023 12:00:00	174	25	{1, 2, 3, 6, 9, 11, 13, 14, 15, 16, 17, 23, 25, 27, 28, 29, 31, 34, 35, 38, 42, 44, 45, 46, 48}	1	0	0	126410606437752	0,9999999999999992	1011284851502020	0,9999999999999999
12-Dec-2023 12:00:00	175	4	{28, 38, 39, 42}	1	0	0	230300	0,999995657837603	1842400	0,9999994572297 00
12-Dec-2023 13:00:00	176	4	{3, 9, 45, 49}	2	0	0	230300	0,999995657837603	1842400	0,9999994572297 00
12-Dec-2023 13:00:00	177	3	{1, 13, 33}	1	0	0	19600	0,999948979591837	156800	0,9999936222448980
12-Dec-2023 14:00:00	178	12	{6, 8, 9, 15, 16, 24, 27, 29, 31, 40, 46, 47}	2	0	0	121399651100	0,999999999991763	971197208800	0,9999999999989 70
12-Dec-2023 14:00:00	179	12	{1, 2, 5, 13, 19, 27, 28, 29, 31, 33, 43, 44}	2	0	0	121399651100	0,999999999991763	971197208800	0,9999999999989 70
12-Dec-2023 14:00:00	180	12	{3, 13, 14, 17, 18, 20, 21, 26, 35, 37, 39, 42}	3	0	0	121399651100	0,999999999991763	971197208800	0,9999999999989 70
12-Dec-2023 14:00:00	181	20	{5, 6, 11, 12, 16, 17, 18, 19, 21, 22, 23, 26, 30, 31, 35, 36, 39, 42, 46, 47}	2	0	0	47129212243960	0,9999999999999979	377033697951680	0,9999999999999997
12-Dec-2023 15:00:00	182	20	{1, 5, 7, 8, 10, 11, 12, 20, 29, 30, 32, 34, 35, 36, 38, 41, 44, 46, 48, 50}	2	0	0	47129212243960	0,9999999999999979	377033697951680	0,9999999999999997
12-Dec-2023 15:00:00	183	20	{3, 5, 7, 8, 9, 13, 15, 17, 18, 25, 27, 32, 33, 35, 37, 38, 43, 46, 47, 49}	3	0	0	47129212243960	0,9999999999999979	377033697951680	0,9999999999999997
12-Dec-2023 15:00:00	184	20	{5, 8, 10, 13, 14, 15, 16, 17, 20, 21, 22, 26, 30, 32, 36, 37, 38, 39, 40, 43}	2	0	0	47129212243960	0,9999999999999979	377033697951680	0,9999999999999997
12-Dec-2023 16:00:00	185	20	{2, 3, 4, 5, 6, 14, 15, 16, 18, 20, 21, 22, 26, 30, 33, 35, 36, 39, 42, 46}	2	0	0	47129212243960	0,9999999999999979	377033697951680	0,9999999999999997
12-Dec-2023 16:00:00	186	20	{1, 5, 9, 10, 12, 13, 17, 18, 20, 23, 24, 27, 33, 34, 35, 38, 41, 42, 45, 49}	1	0	0	47129212243960	0,9999999999999979	377033697951680	0,9999999999999997
12-Dec-2023 16:00:00	187	20	{1, 4, 5, 6, 11, 16, 17, 23, 27, 29, 31, 32, 33, 35, 36, 41, 42, 43, 48, 49}	2	0	0	47129212243960	0,9999999999999979	377033697951680	0,9999999999999997
12-Dec-2023 16:00:00	188	20	{1, 2, 4, 7, 9, 10, 12, 13, 15, 18, 19, 20, 21, 32, 33, 34, 40, 42, 46, 48}	1	0	0	47129212243960	0,9999999999999979	377033697951680	0,9999999999999997
12-Dec-2023 17:00:00	189	25	{1, 2, 3, 5, 6, 7, 10, 12, 13, 14, 15, 22, 24, 26, 27, 28, 29, 31, 32, 34, 35, 40, 42, 43, 50}	1	0	0	126410606437752	0,9999999999999992	1011284851502020	0,9999999999999999
12-Dec-2023 17:00:00	190	25	{2, 3, 7, 10, 11, 12, 13, 14, 17, 19, 21, 22, 23, 29, 30, 32, 33, 34, 36, 37, 38, 40, 43, 45, 49}	3	0	0	126410606437752	0,9999999999999992	1011284851502020	0,9999999999999999
12-Dec-2023 18:00:00	191	25	{2, 5, 8, 9, 11, 12, 17, 18, 20, 21, 23, 24, 25, 26, 27, 29, 30, 34, 35, 36, 37, 43, 44, 45, 49}	3	0	0	126410606437752	0,9999999999999992	1011284851502020	0,9999999999999999
12-Dec-2023 18:00:00	192	25	{1, 2, 4, 5, 6, 8, 9, 10, 12, 13, 14, 15, 16, 17, 20, 27, 28, 29, 35, 36, 38, 39, 41, 44, 47}	3	0	0	126410606437752	0,9999999999999992	1011284851502020	0,9999999999999999
12-Dec-2023 18:00:00	193	25	{4, 5, 9, 10, 13, 14, 15, 17, 19, 20, 22, 26, 28, 29, 32, 35, 37, 38, 40, 42, 43, 44, 48, 49, 50}	1	0	0	126410606437752	0,9999999999999992	1011284851502020	0,9999999999999999
12-Dec-2023 19:00:00	194	25	{1, 4, 6, 8, 9, 10, 11, 16, 17, 18, 19, 21, 22, 26, 27, 28, 30, 36, 38, 40, 41, 44, 45, 47, 50}	1	0	0	126410606437752	0,9999999999999992	1011284851502020	0,9999999999999999
12-Dec-2023 19:00:00	195	25	{1, 3, 4, 6, 7, 8, 9, 11, 12, 17, 19, 22, 23, 27, 28, 29, 31, 32, 36, 38, 40, 41, 46, 47, 50}	2	0	0	126410606437752	0,9999999999999992	1011284851502020	0,9999999999999999
12-Dec-2023 20:00:00	196	5	{5, 21, 26, 27, 30}	2	0	0	2118760	0,999999528025827	16950080	0,9999999410032 28
12-Dec-2023 20:00:00	197	5	{1, 3, 6, 12, 15}	2	0	0	2118760	0,999999528025827	16950080	0,9999999410032 28
12-Dec-2023 20:00:00	198	4	{17, 30, 39, 47}	2	0	0	230300	0,999995657837603	1842400	0,9999994572297 00
12-Dec-2023 21:00:00	199	4	{12, 27, 33, 35}	2	0	0	230300	0,999995657837603	1842400	0,9999994572297 00
12-Dec-2023 21:00:00	200	4	{29, 31, 43, 45}	3	0	0	230300	0,999995657837603	1842400	0,9999994572297 00
12-Dec-2023 22:00:00	201	4	{14, 22, 30, 46}	2	0	0	230300	0,999995657837603	1842400	0,9999994572297 00
12-Dec-2023 22:00:00	202	4	{3, 15, 31, 47}	2	0	0	230300	0,999995657837603	1842400	0,9999994572297 00
12-Dec-2023 22:00:00	203	14	{1, 3, 4, 13, 15, 16, 22, 30, 32, 35, 36, 37, 40, 48}	1	0	0	937845656300	0,999999999998934	7502765250400	0,9999999999998 67
12-Dec-2023 23:00:00	204	14	{1, 3, 4, 8, 16, 17, 34, 36, 37, 38, 39, 42, 43, 47}	2	0	0	937845656300	0,999999999998934	7502765250400	0,9999999999998 67
12-Dec-2023 23:00:00	205	14	{1, 7, 18, 19, 21, 29, 31, 34, 36, 41, 42, 43, 46, 48}	3	0	0	937845656300	0,999999999998934	7502765250400	0,9999999999998 67
13-Dec-2023 00:00:00	206	14	{2, 5, 6, 7, 8, 11, 14, 15, 21, 30, 37, 45, 49, 50}	3	0	0	937845656300	0,999999999998934	7502765250400	0,9999999999998 67
13-Dec-2023 00:00:00	207	13	{5, 6, 12, 14, 21, 24, 26, 27, 29, 34, 36, 43, 50}	2	0	0	354860518600	0,999999999997182	2838884148800	0,9999999999996 48
13-Dec-2023 00:00:00	208	13	{1, 8, 10, 13, 19, 21, 23, 24, 33, 37, 41, 48, 49}	1	0	0	354860518600	0,999999999997182	2838884148800	0,9999999999996 48
13-Dec-2023 00:00:00	209	13	{7, 13, 14, 16, 17, 25, 29, 33, 36, 40, 41, 44, 49}	3	0	0	354860518600	0,999999999997182	2838884148800	0,9999999999996 48
13-Dec-2023 01:00:00	210	13	{9, 12, 13, 14, 19, 21, 23, 26, 31, 32, 35, 45, 46}	1	0	0	354860518600	0,999999999997182	2838884148800	0,9999999999996 48
13-Dec-2023 01:00:00	211	13	{5, 10, 14, 15, 28, 29, 32, 40, 41, 44, 46, 48, 50}	2	0	0	354860518600	0,999999999997182	2838884148800	0,9999999999996 48
13-Dec-2023 01:00:00	212	13	{1, 5, 9, 14, 18, 20, 22, 28, 35, 42, 47, 48, 49}	1	0	0	354860518600	0,999999999997182	2838884148800	0,9999999999996 48
13-Dec-2023 02:00:00	213	13	{2, 3, 6, 8, 22, 26, 28, 30, 35, 36, 40, 46, 47}	2	0	1	354860518600	0,999999999997182	2838884148800	0,9999999999996 48
13-Dec-2023 02:00:00	214	18	{2, 3, 5, 7, 8, 15, 18, 22, 23, 25, 26, 29, 38, 39, 40, 43, 45, 47}	1	0	0	18053528883775	0,999999999999945	144428231070200	0,9999999999999 93
13-Dec-2023 02:00:00	215	18	{2, 3, 8, 10, 11, 12, 14, 19, 28, 33, 34, 36, 39, 41, 43, 44, 47, 49}	3	0	0	18053528883775	0,999999999999945	144428231070200	0,9999999999999 93
13-Dec-2023 03:00:00	216	18	{1, 9, 11, 12, 13, 14, 15, 17, 18, 20, 23, 24, 30, 33, 42, 45, 46, 48}	1	0	0	18053528883775	0,999999999999945	144428231070200	0,9999999999999 93
13-Dec-2023 03:00:00	217	14	{1, 3, 7, 10, 23, 25, 27, 30, 31, 34, 36, 37, 39, 45}	1	0	0	937845656300	0,999999999998934	7502765250400	0,9999999999998 67
13-Dec-2023 03:00:00	218	14	{1, 9, 10, 12, 19, 22, 23, 25, 26, 28, 29, 39, 41, 45}	2	0	0	937845656300	0,999999999998934	7502765250400	0,9999999999998 67
13-Dec-2023 03:00:00	219	14	{2, 4, 5, 6, 11, 12, 14, 15, 18, 23, 27, 28, 33, 36}	2	0	0	937845656300	0,999999999998934	7502765250400	0,9999999999998 67
13-Dec-2023 04:00:00	220	14	{4, 6, 8, 15, 19, 27, 28, 31, 32, 33, 38, 40, 44, 47}	1	0	0	937845656300	0,999999999998934	7502765250400	0,9999999999998 67
13-Dec-2023 04:00:00	221	14	{3, 8, 18, 19, 20, 23, 27, 29, 30, 33, 34, 37, 39, 41}	3	0	0	937845656300	0,999999999998934	7502765250400	0,9999999999998 67
13-Dec-2023 05:00:00	222	14	{5, 7, 14, 16, 23, 27, 29, 32, 33, 34, 36, 41, 42, 45}	1	0	0	937845656300	0,999999999998934	7502765250400	0,9999999999998 67
13-Dec-2023 05:00:00	223	5	{14, 37, 40, 45, 48}	2	0	0	2118760	0,999999528025827	16950080	0,9999999410032 28
13-Dec-2023 05:00:00	224	5	{19, 27, 28, 33, 39}	2	0	0	2118760	0,999999528025827	16950080	0,9999999410032 28
13-Dec-2023 06:00:00	225	5	{23, 38, 40, 41, 49}	2	0	0	2118760	0,999999528025827	16950080	0,9999999410032 28
13-Dec-2023 06:00:00	226	8	{2, 10, 12, 17, 24, 30, 38, 44}	1	0	0	536878650	0,999999998137382	4295029200	0,9999999997671 73
13-Dec-2023 07:00:00	227	8	{9, 11, 12, 22, 32, 34, 40, 43}	2	0	0	536878650	0,999999998137382	4295029200	0,9999999997671 73
13-Dec-2023 07:00:00	228	8	{4, 9, 10, 14, 16, 29, 39, 50}	1	0	0	536878650	0,999999998137382	4295029200	0,9999999997671 73
13-Dec-2023 07:00:00	229	8	{2, 3, 20, 23, 28, 37, 46, 50}	2	0	0	536878650	0,999999998137382	4295029200	0,9999999997671 73
13-Dec-2023 07:00:00	230	8	{13, 15, 17, 26, 32, 37, 41, 44}	2	0	0	536878650	0,999999998137382	4295029200	0,9999999997671 73
13-Dec-2023 08:00:00	231	20	{2, 3, 7, 10, 14, 15, 17, 21, 27, 28, 29, 34, 35, 37, 39, 41, 44, 46, 48, 49}	3	0	0	47129212243960	0,9999999999999979	377033697951680	0,9999999999999997
13-Dec-2023 08:00:00	232	12	{11, 13, 15, 25, 34, 36, 37, 38, 41, 43, 45, 48}	3	0	0	121399651100	0,999999999991763	971197208800	0,9999999999989 70
13-Dec-2023 09:00:00	233	12	{2, 8, 9, 20, 22, 24, 26, 32, 38, 43, 47, 49}	3	0	0	121399651100	0,999999999991763	971197208800	0,9999999999989 70
13-Dec-2023 09:00:00	234	10	{3, 8, 26, 27, 28, 29, 32, 40, 41, 50}	2	0	0	10272278170	0,999999999902651	82178225360	0,9999999998878 31
13-Dec-2023 10:00:00	235	10	{2, 7, 9, 18, 24, 32, 36, 41, 42, 50}	2	0	0	10272278170	0,999999999902651	82178225360	0,9999999998878 31
13-Dec-2023 10:00:00	236	10	{2, 12, 14, 17, 21, 29, 33, 40, 41, 42}	2	0	0	10272278170	0,999999999902651	82178225360	0,9999999998878 31
13-Dec-2023 11:00:00	237	10	{1, 15, 16, 18, 24, 26, 28, 34, 44, 48}	1	0	0	10272278170	0,999999999902651	82178225360	0,9999999998878 31
13-Dec-2023 11:00:00	238	10	{5, 8, 13, 18, 22, 31, 34, 40, 48, 50}	2	0	0	10272278170	0,999999999902651	82178225360	0,9999999998878 31
13-Dec-2023 11:00:00	239	11	{5, 13, 14, 15, 28, 29, 30, 39, 41, 46, 47}	2	0	0	37353738800	0,999999999973229	298829910400	0,9999999999966 54
13-Dec-2023 12:00:00	240	11	{9, 10, 17, 20, 23, 26, 35, 38, 42, 43, 50}	2	0	0	37353738800	0,999999999973229	298829910400	0,9999999999966 54
13-Dec-2023 12:00:00	241	11	{3, 5, 16, 18, 22, 27, 33, 39, 43, 45, 47}	2	0	0	37353738800	0,999999999973229	298829910400	0,9999999999966 54
13-Dec-2023 12:00:00	242	11	{3, 6, 12, 13, 16, 18, 20, 24, 37, 40, 44}	1	0	0	37353738800	0,999999999973229	298829910400	0,9999999999966 54
13-Dec-2023 12:00:00	243	11	{2, 6, 7, 18, 28, 31, 32, 34, 36, 41, 46}	2	0	0	37353738800	0,999999999973229	298829910400	0,9999999999966 54
13-Dec-2023 13:00:00	244	11	{3, 4, 11, 12, 14, 18, 28, 29, 40, 47, 49}	3	0	0	37353738800	0,999999999973229	298829910400	0,9999999999966 54
13-Dec-2023 13:00:00	245	18	{3, 5, 9, 12, 13, 17, 18, 21, 24, 25, 30, 36, 37, 40, 42, 45, 47, 49}	3	0	0	18053528883775	0,999999999999945	144428231070200	0,9999999999999 93
13-Dec-2023 13:00:00	246	18	{3, 5, 6, 8, 9, 12, 15, 16, 18, 23, 26, 33, 34, 41, 43, 46, 48, 49}	1	0	0	18053528883775	0,999999999999945	144428231070200	0,9999999999999 93
13-Dec-2023 14:00:00	247	18	{1, 4, 6, 7, 9, 13, 16, 17, 19, 20, 24, 26, 29, 31, 34, 39, 41, 43}	2	0	0	18053528883775	0,999999999999945	144428231070200	0,9999999999999 93

Продовження таблиці Г.1

13-Dec-2023 14:00:00	248	18	{1, 3, 6, 8, 11, 13, 15, 16, 17, 18, 25, 34, 41, 42, 43, 46, 47, 49}	2	0	0	18053528883775	0,999999999999945	144428231070200	0,999999999999993
13-Dec-2023 14:00:00	249	18	{1, 3, 4, 13, 21, 24, 26, 27, 29, 31, 36, 37, 38, 45, 47, 48, 49, 50}	1	0	0	18053528883775	0,999999999999945	144428231070200	0,999999999999993
13-Dec-2023 14:00:00	250	2	{1, 38}	2	0	0	1225	0,999183673469388	9800	0,999897959183674
13-Dec-2023 15:00:00	251	2	{35, 43}	3	0	0	1225	0,999183673469388	9800	0,999897959183674
13-Dec-2023 15:00:00	252	2	{30, 31}	2	0	0	1225	0,999183673469388	9800	0,999897959183674
13-Dec-2023 15:00:00	253	2	{6, 11}	2	0	0	1225	0,999183673469388	9800	0,999897959183674
13-Dec-2023 15:00:00	254	12	{1, 2, 9, 14, 16, 22, 23, 25, 27, 37, 41, 42}	3	0	0	121399651100	0,999999999991763	971197208800	0,999999999998970
13-Dec-2023 16:00:00	255	12	{4, 7, 8, 16, 18, 22, 24, 25, 27, 39, 46, 49}	2	0	0	121399651100	0,999999999991763	971197208800	0,999999999998970
13-Dec-2023 16:00:00	256	12	{2, 8, 9, 15, 23, 24, 26, 27, 34, 41, 45, 47}	3	0	0	121399651100	0,999999999991763	971197208800	0,999999999998970
13-Dec-2023 16:00:00	257	6	{6, 13, 14, 24, 32, 44}	2	0	0	15890700	0,999999937070110	127125600	0,999999992133764
13-Dec-2023 17:00:00	258	6	{15, 21, 26, 38, 42, 45}	1	0	0	15890700	0,999999937070110	127125600	0,999999992133764
13-Dec-2023 17:00:00	259	12	{3, 5, 6, 13, 14, 16, 24, 28, 33, 35, 43, 48}	3	0	0	121399651100	0,999999999991763	971197208800	0,999999999998970
13-Dec-2023 17:00:00	260	12	{4, 8, 9, 12, 15, 21, 24, 25, 27, 30, 38, 40}	2	0	0	121399651100	0,999999999991763	971197208800	0,999999999998970
13-Dec-2023 18:00:00	261	12	{1, 4, 6, 13, 15, 20, 28, 39, 44, 46, 48, 50}	1	0	0	121399651100	0,999999999991763	971197208800	0,999999999998970
13-Dec-2023 18:00:00	262	12	{1, 3, 4, 9, 17, 30, 32, 40, 41, 45, 46, 49}	1	0	0	121399651100	0,999999999991763	971197208800	0,999999999998970
13-Dec-2023 18:00:00	263	12	{2, 6, 8, 13, 16, 29, 31, 42, 45, 46, 47, 48}	1	0	0	121399651100	0,999999999991763	971197208800	0,999999999998970
13-Dec-2023 18:00:00	264	12	{1, 3, 5, 15, 20, 22, 32, 34, 36, 37, 40, 49}	1	0	0	121399651100	0,999999999991763	971197208800	0,999999999998970
13-Dec-2023 19:00:00	265	11	{1, 14, 19, 27, 31, 33, 34, 40, 41, 47, 49}	2	0	0	37353738800	0,999999999973229	298829910400	0,999999999996654
13-Dec-2023 19:00:00	266	11	{2, 7, 8, 14, 16, 18, 21, 24, 28, 35, 41}	2	0	0	37353738800	0,999999999973229	298829910400	0,999999999996654
13-Dec-2023 19:00:00	267	11	{4, 8, 13, 16, 22, 24, 32, 38, 40, 45, 49}	3	0	0	37353738800	0,999999999973229	298829910400	0,999999999996654
13-Dec-2023 20:00:00	268	11	{2, 9, 10, 12, 17, 28, 35, 36, 44, 46, 47}	2	0	0	37353738800	0,999999999973229	298829910400	0,999999999996654
13-Dec-2023 20:00:00	269	11	{1, 8, 11, 14, 19, 21, 23, 24, 36, 39, 41}	2	0	0	37353738800	0,999999999973229	298829910400	0,999999999996654
13-Dec-2023 21:00:00	270	11	{8, 10, 19, 29, 37, 39, 40, 43, 45, 48, 50}	1	0	0	37353738800	0,999999999973229	298829910400	0,999999999996654
13-Dec-2023 21:00:00	271	11	{2, 9, 13, 15, 17, 22, 26, 27, 29, 39, 47}	2	0	0	37353738800	0,999999999973229	298829910400	0,999999999996654
13-Dec-2023 21:00:00	272	11	{4, 14, 15, 16, 26, 27, 30, 31, 34, 36, 39}	2	0	0	37353738800	0,999999999973229	298829910400	0,999999999996654
13-Dec-2023 22:00:00	273	11	{2, 5, 11, 15, 21, 26, 30, 31, 37, 39, 40}	2	0	0	37353738800	0,999999999973229	298829910400	0,999999999996654
13-Dec-2023 22:00:00	274	11	{8, 13, 14, 22, 23, 27, 32, 34, 41, 47, 48}	2	0	0	37353738800	0,999999999973229	298829910400	0,999999999996654
13-Dec-2023 22:00:00	275	11	{5, 13, 15, 16, 23, 25, 30, 34, 37, 47, 49}	1	0	0	37353738800	0,999999999973229	298829910400	0,999999999996654
13-Dec-2023 22:00:00	276	11	{4, 5, 22, 23, 28, 33, 36, 41, 47, 48, 49}	2	0	0	37353738800	0,999999999973229	298829910400	0,999999999996654
13-Dec-2023 23:00:00	277	11	{7, 10, 14, 15, 19, 21, 26, 30, 31, 33, 34}	3	0	0	37353738800	0,999999999973229	298829910400	0,999999999996654
13-Dec-2023 23:00:00	278	20	{1, 7, 10, 11, 14, 16, 20, 22, 28, 29, 31, 32, 33, 34, 36, 39, 40, 42, 44, 50}	2	0	0	47129212243960	0,999999999999979	377033697951680	0,999999999999997
14-Dec-2023 00:00:00	279	20	{3, 4, 6, 12, 13, 14, 18, 21, 22, 23, 25, 29, 30, 31, 34, 36, 46, 47, 48, 49}	1	0	0	47129212243960	0,999999999999979	377033697951680	0,999999999999997
14-Dec-2023 01:00:00	280	20	{5, 7, 9, 10, 14, 16, 18, 19, 20, 21, 22, 26, 28, 29, 34, 35, 39, 42, 45, 46}	2	0	0	47129212243960	0,999999999999979	377033697951680	0,999999999999997
14-Dec-2023 02:00:00	281	20	{1, 2, 3, 5, 7, 8, 10, 13, 14, 18, 23, 30, 31, 36, 37, 40, 41, 48, 49, 50}	2	0	0	47129212243960	0,999999999999979	377033697951680	0,999999999999997
14-Dec-2023 03:00:00	282	20	{3, 7, 9, 10, 11, 15, 16, 19, 20, 22, 24, 25, 29, 32, 37, 40, 43, 44, 45, 47}	2	0	0	47129212243960	0,999999999999979	377033697951680	0,999999999999997
14-Dec-2023 04:00:00	283	20	{8, 15, 16, 18, 19, 20, 26, 30, 31, 32, 33, 34, 35, 36, 38, 39, 41, 42, 44, 50}	2	0	0	47129212243960	0,999999999999979	377033697951680	0,999999999999997
14-Dec-2023 05:00:00	284	7	{2, 5, 16, 21, 28, 30, 39}	2	0	0	99884400	0,999999989988427	799075200	0,999999998748553
14-Dec-2023 06:00:00	285	7	{6, 10, 18, 26, 28, 30, 38}	2	0	0	99884400	0,999999989988427	799075200	0,999999998748553
14-Dec-2023 07:00:00	286	7	{4, 12, 16, 20, 28, 34, 36}	3	0	0	99884400	0,999999989988427	799075200	0,999999998748553
14-Dec-2023 08:00:00	287	24	{1, 2, 6, 9, 13, 18, 19, 20, 22, 23, 24, 28, 32, 36, 37, 38, 39, 40, 41, 42, 44, 46, 47, 48}	2	0	0	121548660036300	0,999999999999992	972389280290400	0,999999999999999
14-Dec-2023 09:00:00	288	24	{8, 10, 11, 12, 13, 16, 21, 23, 24, 25, 28, 30, 31, 32, 33, 34, 38, 39, 40, 43, 44, 47, 48, 50}	1	0	0	121548660036300	0,999999999999992	972389280290400	0,999999999999999
14-Dec-2023 10:00:00	289	24	{1, 2, 6, 7, 8, 9, 10, 13, 15, 16, 22, 24, 28, 29, 31, 32, 33, 36, 39, 41, 44, 47, 49, 50}	3	0	0	121548660036300	0,999999999999992	972389280290400	0,999999999999999
14-Dec-2023 11:00:00	290	24	{1, 2, 3, 4, 8, 13, 15, 16, 17, 20, 21, 22, 23, 25, 27, 30, 31, 34, 39, 40, 43, 46, 48, 50}	2	0	0	121548660036300	0,999999999999992	972389280290400	0,999999999999999
14-Dec-2023 12:00:00	291	24	{5, 6, 8, 11, 12, 14, 15, 18, 19, 20, 21, 24, 25, 26, 29, 31, 33, 35, 36, 38, 41, 42, 45, 48}	2	0	0	121548660036300	0,999999999999992	972389280290400	0,999999999999999
14-Dec-2023 13:00:00	292	24	{1, 2, 11, 13, 14, 16, 17, 18, 20, 22, 24, 27, 29, 32, 35, 36, 37, 41, 42, 43, 44, 45, 46, 47}	3	0	0	121548660036300	0,999999999999992	972389280290400	0,999999999999999
14-Dec-2023 14:00:00	293	18	{6, 9, 10, 12, 13, 15, 17, 18, 19, 21, 31, 32, 35, 37, 41, 46, 48, 49}	3	0	0	18053528883775	0,999999999999945	144428231070200	0,999999999999993
14-Dec-2023 15:00:00	294	18	{7, 8, 12, 14, 17, 21, 27, 28, 29, 30, 37, 40, 41, 45, 47, 48, 49, 50}	3	0	0	18053528883775	0,999999999999945	144428231070200	0,999999999999993
14-Dec-2023 16:00:00	295	18	{3, 4, 8, 11, 13, 16, 17, 21, 25, 32, 33, 34, 36, 38, 39, 40, 48, 50}	2	0	0	18053528883775	0,999999999999945	144428231070200	0,999999999999993
14-Dec-2023 17:00:00	296	22	{1, 2, 6, 8, 10, 16, 18, 19, 22, 23, 25, 26, 27, 30, 32, 35, 38, 40, 45, 46, 47, 48}	3	0	0	88749815264600	0,999999999999989	709998522116800	0,999999999999999
14-Dec-2023 18:00:00	297	22	{1, 2, 3, 7, 10, 11, 12, 13, 15, 16, 17, 19, 23, 31, 32, 34, 35, 38, 39, 45, 47, 48}	2	0	0	88749815264600	0,999999999999989	709998522116800	0,999999999999999
14-Dec-2023 19:00:00	298	22	{2, 4, 10, 11, 14, 17, 19, 20, 25, 28, 29, 32, 34, 35, 37, 40, 41, 43, 44, 46, 47, 50}	2	0	0	88749815264600	0,999999999999989	709998522116800	0,999999999999999
14-Dec-2023 20:00:00	299	22	{2, 5, 7, 12, 14, 16, 19, 20, 23, 25, 27, 29, 31, 33, 35, 36, 40, 42, 43, 44, 45, 49}	2	0	0	88749815264600	0,999999999999989	709998522116800	0,999999999999999
14-Dec-2023 21:00:00	300	6	{1, 27, 35, 36, 42, 49}	1	0	0	15890700	0,999999937070110	127125600	0,999999992133764
14-Dec-2023 22:00:00	301	6	{5, 6, 10, 25, 28, 45}	1	0	0	15890700	0,999999937070110	127125600	0,999999992133764
14-Dec-2023 23:00:00	302	6	{9, 16, 33, 38, 47, 50}	1	0	0	15890700	0,999999937070110	127125600	0,999999992133764
15-Dec-2023 00:00:00	303	6	{9, 23, 26, 34, 35, 48}	2	0	0	15890700	0,999999937070110	127125600	0,999999992133764
15-Dec-2023 01:00:00	304	6	{10, 17, 25, 26, 38, 42}	1	0	0	15890700	0,999999937070110	127125600	0,999999992133764
15-Dec-2023 02:00:00	305	6	{1, 10, 24, 26, 31, 36}	2	0	0	15890700	0,999999937070110	127125600	0,999999992133764
15-Dec-2023 03:00:00	306	25	{1, 4, 7, 8, 10, 11, 12, 13, 16, 17, 18, 20, 28, 29, 30, 32, 33, 34, 36, 37, 40, 41, 43, 44, 47}	3	0	0	126410606437752	0,999999999999992	1011284851502020	0,999999999999999
15-Dec-2023 04:00:00	307	25	{1, 2, 3, 4, 6, 8, 10, 11, 13, 16, 21, 23, 24, 25, 26, 29, 33, 34, 36, 40, 43, 45, 46, 48, 50}	3	0	0	126410606437752	0,999999999999992	1011284851502020	0,999999999999999
15-Dec-2023 05:00:00	308	25	{2, 3, 4, 5, 9, 10, 12, 15, 18, 22, 23, 25, 26, 27, 33, 34, 35, 37, 39, 40, 42, 46, 47, 49, 50}	1	0	0	126410606437752	0,999999999999992	1011284851502020	0,999999999999999
15-Dec-2023 06:00:00	309	25	{1, 6, 7, 10, 11, 12, 14, 15, 16, 20, 21, 22, 24, 27, 28, 29, 30, 31, 32, 37, 39, 44, 47, 49, 50}	2	0	0	126410606437752	0,999999999999992	1011284851502020	0,999999999999999
15-Dec-2023 07:00:00	310	25	{1, 2, 4, 5, 8, 10, 13, 14, 15, 19, 21, 27, 28, 29, 30, 32, 33, 35, 37, 38, 40, 41, 43, 44, 49}	1	0	0	126410606437752	0,999999999999992	1011284851502020	0,999999999999999
15-Dec-2023 08:00:00	311	25	{1, 3, 5, 8, 9, 11, 12, 14, 15, 16, 17, 18, 23, 28, 33, 34, 35, 36, 37, 40, 41, 42, 43, 48, 49}	3	0	1	126410606437752	0,999999999999992	1011284851502020	0,999999999999999
15-Dec-2023 09:00:00	312	25	{1, 2, 6, 7, 8, 13, 18, 19, 20, 21, 24, 25, 26, 27, 28, 29, 31, 32, 35, 36, 37, 38, 42, 48, 49}	2	0	0	126410606437752	0,999999999999992	1011284851502020	0,999999999999999
15-Dec-2023 10:00:00	313	23	{2, 3, 4, 5, 9, 12, 13, 14, 21, 23, 24, 25, 27, 28, 29, 30, 33, 39, 40, 42, 46, 47, 49}	2	0	0	108043253365600	0,999999999999991	864346026924800	0,999999999999999
15-Dec-2023 11:00:00	314	23	{5, 12, 14, 15, 16, 17, 22, 24, 25, 26, 28, 31, 32, 35, 36, 37, 38, 39, 40, 42, 43, 45, 50}	3	0	0	108043253365600	0,999999999999991	864346026924800	0,999999999999999
15-Dec-2023 12:00:00	315	23	{1, 2, 3, 4, 6, 7, 10, 11, 12, 15, 20, 22, 29, 30, 37, 39, 41, 42, 44, 47, 48, 49, 50}	2	0	0	108043253365600	0,999999999999991	864346026924800	0,999999999999999
15-Dec-2023 13:00:00	316	23	{2, 4, 5, 6, 9, 10, 14, 17, 18, 23, 24, 26, 31, 33, 34, 35, 37, 38, 39, 40, 45, 46, 50}	1	0	0	108043253365600	0,999999999999991	864346026924800	0,999999999999999
15-Dec-2023 14:00:00	317	5	{3, 5, 10, 30, 41}	2	0	0	2118760	0,999999528025827	16950080	0,999999941003228
15-Dec-2023 15:00:00	318	5	{8, 30, 34, 38, 39}	1	1	0	2118760	0,999999528025827	16950080	0,999999941003228
15-Dec-2023 16:00:00	319	5	{9, 14, 21, 38, 44}	1	0	0	2118760	0,999999528025827	16950080	0,999999941003228
15-Dec-2023 17:00:00	320	5	{17, 18, 22, 30, 44}	2	0	0	2118760	0,999999528025827	16950080	0,999999941003228
15-Dec-2023 18:00:00	321	5	{12, 15, 44, 45, 50}	1	0	0	2118760	0,999999528025827	16950080	0,999999941003228
15-Dec-2023 19:00:00	322	11	{4, 5, 6, 7, 18, 21, 22, 26, 28, 39, 44}	2	0	0	37353738800	0,999999999973229	298829910400	0,999999999996654
15-Dec-2023 20:00:00	323	11	{1, 8, 14, 15, 21, 22, 25, 30, 35, 47, 48}	1	0	0	37353738800	0,999999999973229	298829910400	0,999999999996654
15-Dec-2023 21:00:00	324	21	{1, 3, 5, 8, 9, 11, 13, 15, 16, 20, 21, 27, 28, 29, 30, 34, 39, 40, 41, 42, 44}	1	0	0	67327446062800	0,999999999999985	538619568502400	0,999999999999998
15-Dec-2023 22:00:00	325	21	{1, 2, 4, 5, 6, 10, 11, 18, 20, 21, 22, 23, 26, 31, 34, 36, 38, 39, 40, 48, 49}	1	0	0	67327446062800	0,999999999999985	538619568502400	0,999999999999998
15-Dec-2023 23:00:00	326	21	{1, 2, 6, 8, 9, 18, 19, 20, 21, 22, 28, 29, 30, 31, 36, 40, 44, 45, 48, 49, 50}	1	0	0	67327446062800	0,999999999999985	538619568502400	0,999999999999998
16-Dec-2023 00:00:00	327	21	{2, 4, 6, 7, 14, 18, 19, 22, 26, 27, 28, 29, 30, 31, 32, 33, 36, 37, 45, 46, 47}	2	0	0	67327446062800	0,999999999999985	538619568502400	0,999999999999998
16-Dec-2023 01:00:00	328	21	{1, 4, 8, 10, 14, 15, 16, 17, 20, 22, 23, 24, 28, 29, 33, 36, 37, 42, 43, 48, 50}	1	0	0	67327446062800	0,999999999999985	538619568502400	0,999999999999998
16-Dec-2023 02:00:00	329	21	{1, 5, 8, 10, 15, 18, 19, 22, 24, 25, 27, 30, 32, 34, 36, 40, 41, 43, 44, 45, 49}	2	0	0	67327446062800	0,999999999999985	538619568502400	0,999999999999998

Продовження таблиці Г.1

16-Dec-2023 03:00:00	330	9	{1, 9, 10, 21, 26, 28, 31, 46, 49}	2	0	0	2505433700	0,9999999999600868	20043469600	0,999999999950109
16-Dec-2023 04:00:00	331	9	{4, 6, 13, 17, 21, 25, 30, 47, 50}	2	0	0	2505433700	0,9999999999600868	20043469600	0,999999999950109
16-Dec-2023 05:00:00	332	9	{3, 4, 15, 16, 23, 24, 32, 34, 43}	3	0	0	2505433700	0,9999999999600868	20043469600	0,999999999950109
16-Dec-2023 06:00:00	333	9	{8, 11, 12, 18, 25, 29, 33, 41, 44}	1	0	0	2505433700	0,9999999999600868	20043469600	0,999999999950109
16-Dec-2023 07:00:00	334	9	{1, 8, 11, 12, 14, 15, 27, 31, 44}	3	0	0	2505433700	0,9999999999600868	20043469600	0,999999999950109
16-Dec-2023 08:00:00	335	9	{6, 7, 11, 15, 25, 29, 46, 49, 50}	3	0	0	2505433700	0,9999999999600868	20043469600	0,999999999950109
16-Dec-2023 09:00:00	336	4	{24, 27, 35, 40}	2	0	0	230300	0,999995657837603	1842400	0,999999457229700
16-Dec-2023 10:00:00	337	4	{14, 23, 45, 47}	1	0	0	230300	0,999995657837603	1842400	0,999999457229700
16-Dec-2023 11:00:00	338	4	{11, 25, 43, 47}	2	0	0	230300	0,999995657837603	1842400	0,999999457229700
16-Dec-2023 12:00:00	339	4	{6, 7, 22, 28}	3	0	0	230300	0,999995657837603	1842400	0,999999457229700
16-Dec-2023 13:00:00	340	5	{19, 22, 34, 37, 43}	2	0	0	2118760	0,999999528025827	16950080	0,999999941003228
16-Dec-2023 14:00:00	341	13	{6, 16, 19, 28, 29, 31, 35, 39, 40, 41, 43, 45, 47}	1	0	0	354860518600	0,999999999997182	2838884148800	0,999999999999648
16-Dec-2023 15:00:00	342	13	{8, 15, 18, 19, 20, 22, 25, 26, 27, 32, 33, 34, 39}	2	0	0	354860518600	0,999999999997182	2838884148800	0,999999999999648
16-Dec-2023 16:00:00	343	13	{1, 2, 11, 13, 16, 18, 20, 30, 35, 37, 39, 41, 49}	1	0	0	354860518600	0,999999999997182	2838884148800	0,999999999999648
16-Dec-2023 17:00:00	344	13	{10, 19, 22, 24, 25, 27, 37, 38, 39, 41, 43, 44, 50}	2	0	0	354860518600	0,999999999997182	2838884148800	0,999999999999648
16-Dec-2023 18:00:00	345	13	{3, 4, 7, 11, 15, 16, 18, 29, 30, 33, 34, 36, 43}	3	0	0	354860518600	0,999999999997182	2838884148800	0,999999999999648
16-Dec-2023 19:00:00	346	13	{2, 5, 18, 20, 23, 26, 28, 29, 31, 33, 35, 42, 44}	2	0	0	354860518600	0,999999999997182	2838884148800	0,999999999999648
16-Dec-2023 20:00:00	347	13	{4, 7, 11, 14, 15, 19, 23, 29, 32, 43, 47, 49, 50}	1	0	0	354860518600	0,999999999997182	2838884148800	0,999999999999648
16-Dec-2023 21:00:00	348	3	{17, 27, 41}	2	0	0	19600	0,9999489795918370	156800	0,9999936224489800
16-Dec-2023 22:00:00	349	3	{10, 11, 26}	2	0	0	19600	0,9999489795918370	156800	0,9999936224489800
16-Dec-2023 23:00:00	350	15	{1, 4, 7, 10, 15, 18, 20, 22, 29, 32, 33, 36, 37, 42, 44}	1	0	0	2250829575120	0,999999999999556	18006636600960	0,999999999999945
17-Dec-2023 00:00:00	351	15	{5, 7, 8, 9, 20, 23, 24, 30, 31, 32, 34, 35, 39, 41, 47}	3	0	0	2250829575120	0,999999999999556	18006636600960	0,999999999999945
17-Dec-2023 01:00:00	352	15	{2, 3, 5, 10, 13, 18, 30, 31, 35, 39, 40, 47, 48, 49, 50}	2	0	0	2250829575120	0,999999999999556	18006636600960	0,999999999999945
17-Dec-2023 02:00:00	353	15	{2, 4, 5, 8, 9, 14, 21, 24, 28, 29, 32, 42, 43, 45, 50}	2	0	0	2250829575120	0,999999999999556	18006636600960	0,999999999999945
17-Dec-2023 03:00:00	354	15	{4, 6, 13, 14, 17, 18, 20, 21, 22, 23, 31, 35, 41, 45, 48}	2	0	0	2250829575120	0,999999999999556	18006636600960	0,999999999999945
17-Dec-2023 04:00:00	355	15	{2, 5, 8, 10, 11, 16, 17, 22, 24, 28, 35, 38, 39, 45, 47}	2	0	0	2250829575120	0,999999999999556	18006636600960	0,999999999999945
17-Dec-2023 05:00:00	356	18	{1, 3, 4, 12, 17, 18, 21, 23, 24, 25, 29, 32, 33, 36, 43, 48, 49, 50}	2	0	0	18053528883775	0,999999999999945	144428231070200	0,999999999999993
17-Dec-2023 06:00:00	357	18	{1, 5, 6, 11, 12, 17, 21, 22, 23, 28, 31, 32, 34, 37, 43, 45, 47, 49}	2	0	0	18053528883775	0,999999999999945	144428231070200	0,999999999999993
17-Dec-2023 07:00:00	358	18	{1, 4, 7, 10, 12, 15, 22, 24, 25, 31, 33, 36, 39, 40, 42, 45, 48, 50}	2	0	0	18053528883775	0,999999999999945	144428231070200	0,999999999999993
17-Dec-2023 08:00:00	359	18	{1, 3, 7, 10, 14, 16, 18, 19, 24, 27, 28, 29, 30, 33, 37, 39, 42, 44}	2	0	0	18053528883775	0,999999999999945	144428231070200	0,999999999999993
17-Dec-2023 09:00:00	360	18	{1, 3, 4, 6, 10, 18, 23, 24, 25, 27, 28, 37, 38, 39, 41, 44, 45, 46}	1	0	0	18053528883775	0,999999999999945	144428231070200	0,999999999999993
17-Dec-2023 10:00:00	361	18	{1, 5, 7, 8, 10, 17, 20, 23, 24, 25, 31, 33, 35, 37, 44, 47, 48, 50}	2	0	0	18053528883775	0,999999999999945	144428231070200	0,999999999999993
17-Dec-2023 11:00:00	362	18	{3, 4, 5, 11, 20, 24, 30, 31, 32, 33, 37, 39, 42, 44, 45, 48, 49, 50}	3	0	0	18053528883775	0,999999999999945	144428231070200	0,999999999999993
17-Dec-2023 12:00:00	363	21	{2, 9, 14, 16, 17, 18, 20, 22, 24, 25, 31, 34, 36, 38, 40, 43, 45, 46, 47, 48, 50}	2	0	0	67327446062800	0,999999999999985	538619568502400	0,999999999999998
17-Dec-2023 13:00:00	364	21	{2, 4, 9, 10, 11, 16, 17, 19, 21, 23, 24, 33, 35, 37, 39, 42, 43, 44, 47, 48, 50}	2	0	0	67327446062800	0,999999999999985	538619568502400	0,999999999999998
17-Dec-2023 14:00:00	365	21	{6, 7, 11, 14, 21, 24, 25, 28, 29, 31, 32, 34, 38, 40, 41, 43, 45, 46, 47, 48, 49}	3	0	0	67327446062800	0,999999999999985	538619568502400	0,999999999999998
17-Dec-2023 15:00:00	366	21	{2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 16, 19, 22, 26, 28, 29, 33, 40, 41, 49}	1	0	0	67327446062800	0,999999999999985	538619568502400	0,999999999999998
17-Dec-2023 16:00:00	367	21	{4, 5, 9, 10, 13, 16, 19, 20, 23, 26, 28, 30, 32, 34, 35, 36, 40, 41, 42, 44, 46}	1	0	0	67327446062800	0,999999999999985	538619568502400	0,999999999999998
17-Dec-2023 17:00:00	368	25	{1, 4, 5, 9, 11, 15, 18, 20, 21, 22, 23, 25, 27, 31, 32, 35, 36, 37, 38, 39, 40, 42, 44, 47, 50}	2	0	0	126410606437752	0,999999999999992	1011284851502020	0,999999999999999
17-Dec-2023 18:00:00	369	25	{2, 5, 7, 10, 11, 12, 13, 14, 15, 16, 17, 21, 23, 29, 32, 33, 34, 37, 38, 41, 42, 43, 47, 48, 49}	2	0	0	126410606437752	0,999999999999992	1011284851502020	0,999999999999999
17-Dec-2023 19:00:00	370	25	{1, 5, 6, 8, 10, 11, 12, 15, 16, 17, 21, 22, 27, 28, 29, 30, 31, 32, 34, 35, 36, 37, 41, 44, 46}	2	0	0	126410606437752	0,999999999999992	1011284851502020	0,999999999999999
17-Dec-2023 20:00:00	371	25	{2, 3, 4, 5, 10, 12, 14, 17, 19, 20, 21, 22, 25, 26, 27, 29, 32, 33, 35, 39, 41, 42, 46, 49, 50}	2	0	0	126410606437752	0,999999999999992	1011284851502020	0,999999999999999
17-Dec-2023 21:00:00	372	25	{1, 5, 8, 9, 15, 17, 19, 20, 22, 23, 24, 25, 26, 28, 31, 32, 33, 39, 40, 42, 43, 45, 46, 47, 48}	2	0	0	126410606437752	0,999999999999992	1011284851502020	0,999999999999999
17-Dec-2023 22:00:00	373	25	{3, 5, 9, 10, 12, 13, 15, 16, 17, 18, 22, 23, 24, 29, 30, 31, 32, 34, 39, 40, 43, 47, 48, 49, 50}	1	0	0	126410606437752	0,999999999999992	1011284851502020	0,999999999999999
17-Dec-2023 23:00:00	374	25	{[1, 10, 11, 12, 15, 16, 18, 20, 22, 24, 25, 27, 29, 30, 31, 32, 33, 38, 39, 41, 43, 44, 45, 48, 50}	1	0	0	126410606437752	0,999999999999992	1011284851502020	0,999999999999999
18-Dec-2023 00:00:00	375	22	{3, 10, 12, 13, 21, 24, 25, 29, 30, 32, 33, 34, 35, 36, 39, 41, 42, 44, 45, 46, 49, 50}	2	0	0	88749815264600	0,999999999999989	709998522116800	0,999999999999999
18-Dec-2023 01:00:00	376	25	{3, 4, 5, 7, 9, 12, 13, 14, 16, 18, 19, 23, 25, 26, 27, 29, 31, 33, 34, 40, 42, 43, 44, 45, 49}	2	0	0	126410606437752	0,999999999999992	1011284851502020	0,999999999999999
18-Dec-2023 02:00:00	377	25	{1, 2, 5, 6, 7, 9, 10, 14, 15, 16, 17, 18, 20, 24, 25, 27, 29, 30, 31, 35, 36, 38, 44, 45, 49}	3	0	0	126410606437752	0,999999999999992	1011284851502020	0,999999999999999
18-Dec-2023 03:00:00	378	25	{1, 4, 7, 11, 12, 14, 18, 22, 23, 24, 25, 26, 27, 29, 30, 32, 35, 39, 41, 42, 43, 45, 48, 49, 50}	3	0	0	126410606437752	0,999999999999992	1011284851502020	0,999999999999999
18-Dec-2023 04:00:00	379	25	{1, 3, 5, 6, 8, 11, 12, 14, 15, 16, 18, 25, 30, 32, 33, 36, 38, 39, 40, 41, 43, 44, 46, 48, 49}	2	0	0	126410606437752	0,999999999999992	1011284851502020	0,999999999999999
18-Dec-2023 05:00:00	380	25	{3, 4, 5, 10, 11, 13, 14, 16, 21, 22, 24, 26, 27, 28, 29, 30, 32, 33, 35, 38, 39, 42, 44, 48, 49}	3	0	0	126410606437752	0,999999999999992	1011284851502020	0,999999999999999
18-Dec-2023 06:00:00	381	13	{5, 10, 13, 24, 29, 34, 37, 40, 41, 46, 47, 48, 49}	2	0	0	354860518600	0,999999999997182	2838884148800	0,999999999999648
18-Dec-2023 07:00:00	382	13	{4, 8, 11, 12, 20, 22, 28, 31, 33, 44, 46, 48, 49}	2	0	0	3548605186000	0,999999999997182	28388841488000	0,999999999999648
18-Dec-2023 08:00:00	383	13	{4, 9, 10, 17, 19, 20, 23, 26, 32, 36, 40, 43, 45}	3	0	0	354860518600	0,999999999997182	2838884148800	0,999999999999648
18-Dec-2023 09:00:00	384	13	{3, 7, 17, 20, 21, 28, 29, 30, 34, 35, 38, 47, 50}	3	0	0	354860518600	0,999999999997182	2838884148800	0,999999999999648
18-Dec-2023 10:00:00	385	7	{8, 15, 18, 29, 33, 46, 49}	2	0	0	99884400	0,999999989888427	799075200	0,999999998748553
18-Dec-2023 11:00:00	386	7	{5, 7, 15, 26, 27, 33, 46}	1	0	0	99884400	0,999999989888427	799075200	0,999999998748553
18-Dec-2023 12:00:00	387	7	{3, 21, 36, 38, 42, 43, 48}	2	0	0	99884400	0,999999989888427	799075200	0,999999998748553
18-Dec-2023 13:00:00	388	7	{7, 12, 15, 18, 31, 39, 50}	2	0	0	99884400	0,999999989888427	799075200	0,999999998748553
18-Dec-2023 14:00:00	389	7	{2, 12, 21, 25, 30, 34, 49}	2	0	0	99884400	0,999999989888427	799075200	0,999999998748553
18-Dec-2023 15:00:00	390	7	{9, 22, 30, 34, 38, 39, 50}	1	0	0	99884400	0,999999989888427	799075200	0,999999998748553
18-Dec-2023 16:00:00	391	23	{1, 2, 4, 5, 7, 8, 9, 11, 12, 14, 19, 21, 22, 24, 27, 31, 32, 33, 35, 38, 39, 41, 50}	2	0	0	108043253365600	0,999999999999991	864346026924800	0,999999999999999
18-Dec-2023 17:00:00	392	23	{4, 6, 7, 8, 9, 17, 18, 22, 23, 25, 26, 27, 28, 29, 30, 33, 34, 36, 37, 38, 45, 49, 50}	2	0	0	108043253365600	0,999999999999991	864346026924800	0,999999999999999
18-Dec-2023 18:00:00	393	23	{1, 3, 4, 7, 8, 11, 12, 15, 20, 21, 22, 24, 26, 30, 32, 34, 35, 36, 38, 39, 43, 46, 50}	1	0	0	108043253365600	0,999999999999991	864346026924800	0,999999999999999
18-Dec-2023 19:00:00	394	23	{2, 4, 5, 8, 11, 12, 13, 19, 23, 25, 26, 27, 31, 33, 35, 36, 37, 40, 44, 45, 46, 47, 50}	1	0	0	108043253365600	0,999999999999991	864346026924800	0,999999999999999
18-Dec-2023 20:00:00	395	22	{1, 8, 11, 12, 13, 14, 15, 17, 18, 19, 23, 24, 26, 29, 35, 36, 37, 38, 41, 42, 43, 45}	2	0	0	88749815264600	0,999999999999989	709998522116800	0,999999999999999
18-Dec-2023 21:00:00	396	22	{2, 5, 9, 10, 11, 13, 15, 16, 17, 18, 19, 20, 22, 26, 27, 31, 33, 34, 39, 45, 46, 50}	2	0	0	88749815264600	0,999999999999989	709998522116800	0,999999999999999
18-Dec-2023 22:00:00	397	22	{2, 3, 10, 11, 13, 18, 20, 21, 23, 24, 26, 28, 29, 30, 32, 34, 35, 40, 44, 46, 47, 50}	2	0	0	88749815264600	0,999999999999989	709998522116800	0,999999999999999
18-Dec-2023 23:00:00	398	22	{2, 3, 7, 10, 11, 15, 17, 19, 20, 21, 22, 24, 26, 27, 30, 33, 35, 36, 38, 43, 47, 50}	3	0	0	88749815264600	0,999999999999989	709998522116800	0,999999999999999
19-Dec-2023 00:00:00	399	22	{1, 3, 4, 5, 6, 8, 10, 12, 14, 15, 18, 19, 20, 25, 26, 27, 28, 30, 35, 40, 41, 44}	2	0	0	88749815264600	0,999999999999989	709998522116800	0,999999999999999
19-Dec-2023 01:00:00	400	16	{1, 3, 6, 7, 11, 17, 20, 22, 24, 27, 35, 41, 42, 45, 47, 50}	1	0	0	4923689695575	0,999999999999797	39389517564600	0,999999999999975
19-Dec-2023 02:00:00	401	16	{3, 6, 11, 12, 13, 20, 27, 28, 29, 30, 32, 34, 37, 39, 41, 43}	1	0	0	4923689695575	0,999999999999797	39389517564600	0,999999999999975
19-Dec-2023 03:00:00	402	16	{4, 8, 9, 10, 14, 17, 20, 23, 24, 29, 33, 34, 36, 37, 40, 44}	3	0	0	4923689695575	0,999999999999797	39389517564600	0,999999999999975
19-Dec-2023 04:00:00	403	16	{2, 4, 5, 8, 14, 16, 21, 23, 29, 30, 33, 34, 36, 44, 49, 50}	2	0	0	4923689695575	0,999999999999797	39389517564600	0,999999999999975
19-Dec-2023 05:00:00	404	16	{5, 9, 11, 12, 20, 22, 23, 27, 29, 32, 34, 39, 43, 45, 47, 50}	2	0	0	4923689695575	0,999999999999797	39389517564600	0,999999999999975
19-Dec-2023 06:00:00	405	16	{1, 6, 8, 11, 15, 19, 21, 31, 32, 33, 36, 38, 42, 45, 46, 47}	1	0	0	4923689695575	0,999999999999797	39389517564600	0,999999999999975
19-Dec-2023 07:00:00	406	17	{2, 11, 13, 14, 15, 17, 21, 22, 23, 27, 29, 33, 36, 38, 41, 43, 46}	2	0	0	9847379391150	0,999999999999898	78779035129200	0,999999999999987
19-Dec-2023 08:00:00	407	17	{1, 3, 4, 5, 17, 23, 28, 31, 32, 34, 35, 40, 43, 46, 48, 49, 50}	2	0	0	9847379391150	0,999999999999898	78779035129200	0,999999999999987
19-Dec-2023 09:00:00	408	17	{2, 4, 5, 8, 15, 16, 20, 26, 27, 28, 36, 38, 39, 40, 43, 44, 48}	2	0	0	9847379391150	0,999999999999898	78779035129200	0,999999999999987
19-Dec-2023 10:00:00	409	19	{7, 8, 12, 16, 19, 21, 22, 23, 25, 28, 34, 37, 40, 41, 43, 44, 45, 46, 48}	3	0	0	30405943383200	0,999999999999967	243247547065600	0,999999999999996
19-Dec-2023 11:00:00	410	19	{1, 2, 4, 6, 9, 11, 12, 15, 18, 19, 23, 25, 32, 36, 39, 40, 43, 44, 50}	3	0	0	30405943383200	0,999999999999967	243247547065600	0,999999999999996

Кінець таблиці Г.1

19-Dec-2023 12:00:00	411	19	{3, 7, 10, 12, 16, 21, 22, 24, 25, 26, 27, 30, 34, 38, 40, 42, 46, 47, 49}	2	0	0	30405943383200	0,999999999999967	243247547065600	0,999999999999996
19-Dec-2023 13:00:00	412	19	{5, 8, 9, 13, 14, 15, 19, 21, 24, 25, 28, 31, 33, 35, 36, 41, 43, 46, 48}	1	0	0	30405943383200	0,999999999999967	243247547065600	0,999999999999996
19-Dec-2023 14:00:00	413	19	{4, 7, 8, 9, 12, 14, 16, 19, 22, 26, 27, 29, 30, 32, 34, 36, 40, 41, 44}	3	0	0	30405943383200	0,999999999999967	243247547065600	0,999999999999996
19-Dec-2023 15:00:00	414	25	{1, 2, 4, 6, 7, 12, 14, 15, 16, 18, 19, 20, 22, 24, 27, 28, 32, 34, 35, 36, 42, 46, 47, 48, 50}	2	0	0	126410606437752	0,999999999999992	1011284851502020	0,999999999999999
19-Dec-2023 16:00:00	415	25	{2, 6, 8, 11, 16, 17, 19, 20, 21, 23, 25, 26, 27, 28, 30, 31, 34, 35, 42, 43, 44, 45, 46, 48, 49}	3	0	0	126410606437752	0,999999999999992	1011284851502020	0,999999999999999
19-Dec-2023 17:00:00	416	25	{4, 5, 6, 7, 10, 11, 12, 14, 15, 16, 20, 21, 22, 23, 28, 29, 30, 33, 39, 40, 41, 42, 43, 44, 47}	2	0	0	126410606437752	0,999999999999992	1011284851502020	0,999999999999999
19-Dec-2023 18:00:00	417	25	{5, 7, 8, 9, 11, 13, 17, 18, 21, 24, 26, 27, 28, 29, 31, 32, 33, 34, 36, 39, 40, 41, 43, 46, 49}	2	0	0	126410606437752	0,999999999999992	1011284851502020	0,999999999999999
19-Dec-2023 19:00:00	418	25	{2, 3, 5, 8, 9, 10, 11, 12, 15, 17, 19, 20, 21, 24, 25, 30, 31, 34, 39, 40, 43, 44, 46, 48, 49}	2	0	0	126410606437752	0,999999999999992	1011284851502020	0,999999999999999
19-Dec-2023 20:00:00	419	25	{2, 4, 5, 6, 7, 8, 9, 11, 14, 15, 16, 17, 18, 19, 21, 31, 32, 35, 37, 38, 42, 43, 44, 48, 49}	1	0	0	126410606437752	0,999999999999992	1011284851502020	0,999999999999999
19-Dec-2023 21:00:00	420	25	{2, 5, 6, 7, 8, 12, 15, 16, 17, 18, 19, 21, 22, 23, 24, 29, 30, 31, 32, 33, 40, 43, 46, 49, 50}	2	0	0	126410606437752	0,999999999999992	1011284851502020	0,999999999999999
19-Dec-2023 22:00:00	421	15	{4, 8, 9, 11, 14, 15, 16, 17, 19, 23, 25, 29, 42, 48, 49}	2	0	0	2250829575120	0,999999999999556	18006636600960	0,999999999999945
19-Dec-2023 23:00:00	422	15	{2, 3, 4, 7, 8, 12, 15, 16, 25, 26, 28, 38, 40, 42, 50}	2	0	0	2250829575120	0,999999999999556	18006636600960	0,999999999999945
	Середньоарифметичне значення:							0.999960255573753		0.999995031946719
	Разом:			837	2	7				

ДОДАТОК Д.

РЕЗУЛЬТАТИ ДРУГОГО ЕКСПЕРИМЕНТУ

Таблиця Д.1

0.99629835	0.99188046	0.99702435	0.99796461	0.99609335	0.99245893	0.99337480	0.99016114	0.99093823	0.99869130
0.99127971	0.99610240	0.99673396	0.99139706	0.99524079	0.99855517	0.99616196	0.99919160	0.99290693	0.99646384
0.99884376	0.99208823	0.99584674	0.99095240	0.99542571	0.99805587	0.99727197	0.99976129	0.99107158	0.99538165
0.99717399	0.99465947	0.99034180	0.99059290	0.99789529	0.99007928	0.99869421	0.99889753	0.99697712	0.99026160
0.99159506	0.99966975	0.99368467	0.99516478	0.99152451	0.99964577	0.99541518	0.99613382	0.99827354	0.99458512
0.99386656	0.99322069	0.99236188	0.99834983	0.99838324	0.99996914	0.99633476	0.99876396	0.99718962	0.99801354
0.99846788	0.99858617	0.99583249	0.99063355	0.99388201	0.99039927	0.99957290	0.99935884	0.99895657	0.99648004
0.99703162	0.99580602	0.99516015	0.99389551	0.99906672	0.99731140	0.99258204	0.99464933	0.99992183	0.99351481
0.99040765	0.99565227	0.99253365	0.99172699	0.99954234	0.99202168	0.99720784	0.99798874	0.99879138	0.99389242
0.99533281	0.99513042	0.99895677	0.99503525	0.99882953	0.99098508	0.99958907	0.99176258	0.99589172	0.99367150

Таблиця Д.2

0.95857211	0.95943472	0.95734371	0.97421062	0.99734269	0.98088778	0.95575127	0.95441665	0.99140100	0.95849652
0.96957264	0.95373744	0.97243681	0.99536487	0.98365926	0.98953094	0.97831294	0.96310724	0.96862286	0.98741131
0.97868211	0.97320589	0.97295334	0.96968229	0.95462613	0.99332360	0.96734429	0.99506802	0.95130308	0.95037746
0.96770585	0.99131691	0.97709453	0.95270963	0.96351017	0.96431322	0.97835395	0.96988860	0.95703769	0.98040221
0.98162629	0.96859350	0.97075853	0.98698000	0.96267306	0.99923913	0.96477883	0.96771924	0.95287079	0.97518268
0.95880731	0.97355998	0.95122205	0.99005049	0.96181518	0.96362093	0.97031084	0.98798565	0.95226100	0.95885737
0.97651851	0.95253062	0.95929352	0.95405599	0.97930412	0.95051897	0.96796303	0.97406292	0.95606477	0.99575885
0.97105545	0.95414055	0.95266162	0.96047715	0.99175171	0.96044119	0.96711502	0.95058255	0.97906668	0.96798892
0.96816157	0.97094533	0.96873721	0.97824083	0.97471323	0.99095234	0.98958330	0.95136950	0.98606181	0.99663960
0.96675417	0.97520642	0.95572414	0.96390797	0.97851475	0.96477501	0.95251468	0.98246774	0.99016540	0.98127454

Таблиця Д.3

0.99999093	0.96724059	0.99424883	0.96150047	0.99608122	0.97300035	0.99629991	0.99892508	0.96007864	0.95469625
0.95194968	0.97309081	0.98910783	0.96475454	0.95831965	0.99444638	0.97066817	0.98629128	0.98565358	0.96971602
0.99918118	0.95850453	0.95886390	0.97039774	0.96752833	0.99296366	0.95513243	0.98138010	0.95482093	0.98625103
0.98130322	0.98832208	0.98077261	0.98075871	0.98346567	0.98413988	0.98983759	0.96747193	0.95756839	0.97808048
0.99944840	0.96976698	0.98785312	0.96967803	0.95688230	0.98544113	0.99444077	0.99507408	0.99945744	0.95136201
0.98184182	0.95169871	0.95728375	0.99187514	0.96935299	0.95455620	0.96267464	0.97516140	0.98013741	0.97140021
0.99383016	0.96301318	0.95964030	0.97657995	0.97103430	0.98248649	0.99302220	0.96502925	0.99574978	0.95433084
0.96865396	0.99838124	0.97828821	0.98881970	0.97283001	0.99806636	0.99205727	0.95616767	0.97036552	0.98974432
0.98180394	0.95614450	0.98158845	0.96633066	0.95145019	0.99312467	0.97301687	0.95807852	0.99605316	0.99078938
0.95215196	0.98691038	0.97102072	0.98208678	0.96491250	0.99096498	0.95466158	0.97346904	0.95816712	0.95487569

Таблиця Д.4

0.98444526	0.99083653	0.99629553	0.99036105	0.98699996	0.98602945	0.99135818	0.99382574	0.99620900	0.98601279
0.98176117	0.98897940	0.98985864	0.99938962	0.99129137	0.98022858	0.99260214	0.98572981	0.99984425	0.99946691
0.98918095	0.98962289	0.99834515	0.99094717	0.99755657	0.99557391	0.98030448	0.99087408	0.98739628	0.98277042
0.98481010	0.98497187	0.99816809	0.98030348	0.99498926	0.98813828	0.99271160	0.99857870	0.98526632	0.98626322
0.98196668	0.99306762	0.99231755	0.98169633	0.99981921	0.98788430	0.98725050	0.98750926	0.99780064	0.98788800
0.99947510	0.99003841	0.99620372	0.99367257	0.99454292	0.98190240	0.98404156	0.98289040	0.99223342	0.98975966
0.98576366	0.98027201	0.99928946	0.98417822	0.99922783	0.98843500	0.99020737	0.99321311	0.99229338	0.99208311
0.98321839	0.99747657	0.98801582	0.98818415	0.99417127	0.99337381	0.98944100	0.98890637	0.99960213	0.98407171
0.98444450	0.99054584	0.99759674	0.98090107	0.98364094	0.99539641	0.99293815	0.99587467	0.98673304	0.98757523
0.98658482	0.99604290	0.99566339	0.98055972	0.98402969	0.98432455	0.98933633	0.99876308	0.99960283	0.98136999

Таблиця Д.5

0.99977795	0.99127206	0.99404346	0.99957813	0.99846604	0.99214719	0.99998851	0.99626434	0.99986197	0.99491588
0.99088302	0.99925802	0.99506318	0.99194042	0.99123186	0.99999482	0.99471988	0.99273547	0.99198221	0.99181277
0.99277962	0.99771557	0.99643729	0.99644764	0.99061992	0.99811126	0.99055281	0.99512324	0.99790790	0.99728807
0.99360840	0.99496910	0.99330168	0.99203830	0.99520991	0.99678616	0.99156568	0.99775899	0.99871988	0.99878435
0.99421467	0.99764152	0.99852179	0.99176684	0.99763517	0.99872994	0.99403364	0.99631628	0.99589890	0.99331909
0.99073791	0.99456021	0.99339875	0.99386780	0.99765693	0.99988282	0.99543672	0.99153638	0.99944000	0.99282557
0.99823297	0.99608966	0.99035487	0.99962440	0.99040984	0.99401426	0.99951643	0.99306363	0.99587689	0.99462734
0.99217010	0.99625165	0.99478371	0.99030287	0.99485458	0.99127753	0.99852862	0.99975417	0.99091759	0.99279480
0.99571337	0.99684983	0.99714278	0.99960417	0.99125629	0.99916343	0.99309858	0.99407809	0.99784569	0.99703365
0.99220238	0.99015185	0.99010242	0.99000791	0.99990391	0.99010929	0.99840191	0.99547201	0.99424009	0.99940560

Таблиця Д.6

0.99632147	0.99146400	0.98905930	0.98066238	0.98732891	0.99227343	0.99045557	0.99637315	0.99603245	0.99088553
0.99708728	0.98180115	0.99804479	0.99620412	0.98135417	0.99369127	0.98752892	0.98833623	0.99103361	0.99021763
0.99749279	0.98596775	0.99851540	0.99777541	0.99246945	0.98511768	0.98549710	0.99474864	0.98700877	0.98245606
0.99391649	0.99628771	0.98409813	0.99371569	0.99908961	0.99112387	0.99725680	0.98638617	0.99616664	0.99942125
0.98064556	0.98187211	0.98493940	0.99512703	0.98821320	0.98843151	0.99200593	0.98268714	0.99904433	0.98242920
0.98953326	0.99840760	0.98790294	0.99495884	0.99909713	0.98578059	0.99493953	0.98165884	0.98326508	0.98179880
0.98650027	0.98067853	0.98981116	0.98353393	0.99887018	0.98909114	0.99428157	0.98050112	0.98113711	0.98789774
0.98857980	0.98668593	0.98766074	0.99301460	0.99771117	0.99350231	0.98421467	0.99049359	0.99164345	0.99696197
0.99723727	0.99242607	0.99760174	0.98137646	0.99189058	0.99063078	0.98633424	0.98842401	0.98294800	0.98596069
0.98037862	0.99869055	0.98155406	0.98684379	0.99862568	0.99347298	0.99372801	0.99625040	0.98379929	0.99024631

Таблиця Д.7

0.99069136	0.99599402	0.99918865	0.99133470	0.99124909	0.99222146	0.99797080	0.99409499	0.99888669	0.99964403
0.99215703	0.99198847	0.99995433	0.99471854	0.99278274	0.99213332	0.99527738	0.99635936	0.99199100	0.99232044
0.99279904	0.99405005	0.99618300	0.99396580	0.99469451	0.99104104	0.99715958	0.99732935	0.99600296	0.99043812
0.99993013	0.99514934	0.99638711	0.99285867	0.99749268	0.99305084	0.99858329	0.99608053	0.99228995	0.99192762
0.99855737	0.99135978	0.99604338	0.99235649	0.99593955	0.99189760	0.99121415	0.99466779	0.99557348	0.99911799
0.99177529	0.99954243	0.99195855	0.99858452	0.99415313	0.99847752	0.99015060	0.99755417	0.99054835	0.99790245
0.99545883	0.99031310	0.99842184	0.99069563	0.99980917	0.99342758	0.99981038	0.99873558	0.99511356	0.99613053
0.99921644	0.99959178	0.99248127	0.99058002	0.99446984	0.99957866	0.99537809	0.99506052	0.99079730	0.99656251
0.99469838	0.99895786	0.99467348	0.99215110	0.99846179	0.99784924	0.99296881	0.99135667	0.99407900	0.99194699
0.99508551	0.99925494	0.99687541	0.99632420	0.99826305	0.99523043	0.99447100	0.99521912	0.99573874	0.99062757

Таблиця Д.8

0.99616241	0.99393319	0.99499498	0.99086267	0.99344767	0.99705027	0.99337091	0.99974124	0.99612285	0.99031739
0.99296921	0.99648385	0.99813508	0.99524311	0.99966888	0.99666341	0.99343468	0.99909537	0.99930875	0.99997569
0.99198415	0.99261265	0.99881839	0.99667235	0.99469686	0.99551611	0.99559922	0.99178744	0.99593995	0.99982683
0.99972409	0.99304877	0.99696350	0.99273931	0.99194882	0.99921976	0.99976563	0.99992460	0.99219793	0.99803199
0.99935967	0.99655018	0.99005663	0.99950425	0.99158999	0.99317362	0.99236177	0.99262015	0.99814187	0.99573069
0.99078575	0.99782104	0.99961009	0.99058395	0.99120499	0.99473216	0.99446561	0.99628011	0.99178104	0.99497012
0.99964699	0.99480482	0.99658685	0.99382283	0.99178067	0.99168703	0.99479149	0.99799480	0.99163893	0.99835357
0.99205237	0.99068229	0.99795119	0.99502220	0.99408020	0.99669118	0.99571574	0.99714843	0.99421102	0.99920509
0.99302143	0.99666637	0.99425478	0.99189749	0.99366713	0.99972203	0.99445364	0.99698395	0.99789884	0.99900965
0.99664734	0.99419453	0.99258231	0.99701481	0.99306354	0.99846947	0.99228966	0.99849399	0.99738303	0.99258931

Таблиця Д.9

0.99742892	0.99482554	0.99439544	0.99928613	0.99741997	0.99811935	0.99796059	0.99033417	0.99232651	0.99978301
0.99537534	0.99912132	0.99975289	0.99528742	0.99143575	0.99786203	0.99852002	0.99746644	0.99271830	0.99436225
0.99560014	0.99910764	0.99309735	0.99115635	0.99061534	0.99092754	0.99365138	0.99572343	0.99075113	0.99826570
0.99026086	0.99498354	0.99036485	0.99848489	0.99433103	0.99022549	0.99394037	0.99723858	0.99034532	0.99348400
0.99745310	0.99995231	0.99744609	0.99744673	0.99232743	0.99551804	0.99191170	0.99029112	0.99323964	0.99204137
0.99996057	0.99097288	0.99054477	0.99586642	0.99173161	0.99220379	0.99068761	0.99106400	0.99114504	0.99812380
0.99917497	0.99353930	0.99912612	0.99560616	0.99004388	0.99993624	0.99394919	0.99735583	0.99889368	0.99796993
0.99685354	0.99690982	0.99468276	0.99067766	0.99252160	0.99468197	0.99555403	0.99132581	0.99787960	0.99426470
0.99908063	0.99725182	0.99583520	0.99588447	0.99742840	0.99290204	0.99115710	0.99192170	0.99512065	0.99965375
0.99004141	0.99090653	0.99507675	0.99044717	0.99096945	0.99563147	0.99688005	0.99241600	0.99257440	0.99548846

Таблиця Д.10

0.95314337	0.98227487	0.97919851	0.99468562	0.96642187	0.97292312	0.96007282	0.98587985	0.97179049	0.95644810
0.95652154	0.97997779	0.97548140	0.99561079	0.97049770	0.98285444	0.97037411	0.99868536	0.99210460	0.98515154
0.98977083	0.96735537	0.96162134	0.98420958	0.99567182	0.95024788	0.98361055	0.96373721	0.97532730	0.99577349
0.96210760	0.95412790	0.95573952	0.99956534	0.99098678	0.98990925	0.95079654	0.98944963	0.99920664	0.96454192
0.98001375	0.98771593	0.99314007	0.96066731	0.96481447	0.98186188	0.98825325	0.98963624	0.99623923	0.96649900
0.98212403	0.99609100	0.97314551	0.96509359	0.99827246	0.95075279	0.99062161	0.98202242	0.97656592	0.96425008
0.98337040	0.97484797	0.99864810	0.95851059	0.99734646	0.97320722	0.95808076	0.98432424	0.98542167	0.99469361
0.97703950	0.97513181	0.99445088	0.97302379	0.97923067	0.97681874	0.96222519	0.99872944	0.96714123	0.97443165
0.96618257	0.95129577	0.96808206	0.97201584	0.98521044	0.98150441	0.98377243	0.98068840	0.95021407	0.98997743
0.97811864	0.95056361	0.98460560	0.95297770	0.95930283	0.97089737	0.95405096	0.96080842	0.97016704	0.95817573

Таблиця Д.11

0.99027137	0.99215344	0.99710254	0.99658338	0.99813446	0.99396860	0.99632949	0.99790073	0.99165707	0.99490724
0.99226791	0.99351911	0.99124888	0.99889060	0.99756007	0.99492850	0.99943941	0.99977104	0.99851521	0.99124064
0.99688880	0.99130220	0.99427546	0.99874654	0.99236962	0.99188685	0.99130534	0.99882713	0.99381762	0.99620774
0.99299865	0.99386404	0.99596996	0.99136903	0.99291594	0.99386376	0.99525875	0.99458629	0.99056814	0.99838015
0.99869081	0.99699732	0.99368232	0.99498077	0.99966349	0.99050332	0.99301467	0.99105999	0.99067524	0.99415656
0.99906579	0.99085799	0.99174178	0.99006040	0.99555962	0.99450046	0.99493598	0.99434212	0.99468207	0.99631499
0.99977905	0.99840816	0.99889243	0.99513997	0.99507142	0.99478879	0.99008491	0.99337219	0.99759130	0.99553193
0.99506296	0.99017693	0.99317617	0.99763460	0.99168269	0.99596303	0.99039512	0.99076808	0.99106813	0.99542258
0.99730657	0.99209645	0.99938498	0.99816096	0.99052939	0.99628075	0.99536024	0.99346579	0.99732193	0.99596645
0.99859691	0.99993007	0.99319955	0.99203948	0.99063735	0.99606133	0.99474709	0.99439025	0.99544161	0.99187410

Таблиця Д.12

0.95846007	0.99839977	0.97713870	0.95796138	0.99027189	0.98290404	0.95720646	0.98072931	0.99605484	0.96672851
0.95715368	0.96808475	0.95350375	0.96104208	0.99171939	0.96258479	0.96165683	0.97418355	0.95315976	0.97705807
0.96185651	0.95419833	0.97839114	0.97628363	0.96586828	0.96796803	0.99403397	0.97904176	0.99640874	0.96662088
0.97421100	0.98103557	0.97126929	0.95942554	0.99891008	0.99599237	0.98262267	0.95349573	0.96625887	0.99869598
0.97642583	0.96986413	0.98888925	0.99822003	0.97968530	0.95436492	0.99865210	0.96020125	0.99529032	0.98580436
0.98129289	0.96993239	0.98429399	0.97120306	0.96795250	0.95263920	0.98761370	0.95700919	0.96453946	0.97379901
0.95347814	0.99048188	0.95506814	0.98238129	0.99600191	0.99697154	0.98813413	0.99634991	0.95456952	0.98293974
0.97083700	0.97203271	0.95542418	0.96261770	0.98475311	0.98206585	0.96697553	0.97787758	0.98273432	0.99742332
0.99805906	0.95273896	0.99421073	0.97840704	0.98681386	0.99453139	0.96419214	0.97675749	0.95589404	0.99560938
0.98267885	0.99887543	0.95237715	0.98090204	0.96362349	0.96066965	0.97850822	0.96041896	0.99747523	0.97291737

Таблиця Д.13

0.98351813	0.97818551	0.98841076	0.95788523	0.96627347	0.99253887	0.95275349	0.99264007	0.98278565	0.98050097
0.95131329	0.97328600	0.98741240	0.99381868	0.98638718	0.97893719	0.99259615	0.99679747	0.97333488	0.95702760
0.98761614	0.99628820	0.97779841	0.97136073	0.95068375	0.98738018	0.99819317	0.99657302	0.99145153	0.95494268
0.97445783	0.98273189	0.95913857	0.99129620	0.97668063	0.99445741	0.97714981	0.97317507	0.97104193	0.97756495
0.98684064	0.96687096	0.99484723	0.99901779	0.95490947	0.98323878	0.99898483	0.99983460	0.97919597	0.99888587
0.98504195	0.95646905	0.98548704	0.95907072	0.96069638	0.96332100	0.97707902	0.95251847	0.97173145	0.96209124
0.98254059	0.98928384	0.98273040	0.97576489	0.97645991	0.96559692	0.97603058	0.96128139	0.96153119	0.98141721
0.99464596	0.99407001	0.99701362	0.99184014	0.96159725	0.97765787	0.97361876	0.98953943	0.99017842	0.97789541
0.95287510	0.95394374	0.98016811	0.96861060	0.99156905	0.99011857	0.97655073	0.97590074	0.99259715	0.99922798
0.97288461	0.97574558	0.96145681	0.99806176	0.98240856	0.98699276	0.99786241	0.96894925	0.98642928	0.95670291

Таблиця Д.14

0.99662055	0.99026217	0.99558836	0.99153741	0.99644973	0.99429710	0.99609731	0.99477446	0.99359045	0.99528371
0.99572329	0.99603649	0.99988742	0.99363727	0.99066646	0.99112551	0.99029732	0.99630840	0.99349540	0.99581964
0.99934260	0.99010084	0.99816549	0.99752028	0.99879874	0.99074359	0.99122051	0.99640477	0.99655551	0.99901043
0.99471167	0.99917406	0.99192394	0.99172436	0.99538532	0.99756889	0.99056542	0.99739721	0.99148705	0.99667324
0.99992723	0.99987410	0.99206641	0.99167291	0.99358740	0.99507232	0.99403899	0.99690206	0.99399931	0.99715254
0.99165959	0.99342288	0.99773318	0.99956129	0.99365602	0.99955079	0.99963465	0.99156507	0.99633457	0.99586553
0.99940713	0.99244442	0.99708820	0.99722257	0.99777733	0.99511519	0.99756681	0.99488873	0.99418036	0.99432905
0.99616786	0.99933060	0.99785263	0.99247183	0.99816992	0.99784936	0.99283288	0.99339542	0.99048939	0.99511818
0.99705619	0.99540210	0.99598125	0.99129061	0.99013767	0.99972778	0.99277645	0.99518413	0.99437530	0.99995726
0.99649179	0.99247623	0.99394407	0.99168364	0.99446916	0.99453137	0.99797670	0.99810599	0.99247396	0.99975299

Таблиця Д.15

0.96601411	0.99473051	0.97840713	0.98232152	0.97288431	0.98535906	0.98358658	0.98718009	0.97357615	0.97373470
0.97354944	0.98395770	0.99550447	0.97643406	0.97299384	0.97478995	0.98991398	0.95279715	0.97646317	0.99646873
0.99073009	0.96701501	0.95039182	0.99596932	0.95572328	0.95923736	0.95846313	0.95637359	0.97364446	0.99483460
0.95687991	0.97242323	0.99529750	0.96386853	0.96803712	0.95816706	0.99995576	0.99038960	0.98346208	0.98056108
0.97798883	0.95863675	0.99108915	0.99825054	0.97122684	0.99085758	0.98636847	0.98602158	0.96562875	0.99574100
0.97256006	0.98915903	0.98444597	0.97670912	0.97504150	0.97644816	0.95522566	0.95150566	0.99171602	0.96190069
0.99281665	0.95314829	0.97506294	0.96843453	0.95104069	0.97516968	0.95207428	0.96749459	0.97660239	0.95394781
0.98616901	0.96897488	0.98391857	0.98795418	0.95847221	0.97398240	0.98664386	0.99941435	0.96042091	0.98366589
0.96207107	0.98574833	0.98910490	0.95927926	0.99038879	0.99409094	0.99032548	0.98296049	0.98770225	0.95635473
0.99367809	0.95874936	0.96931898	0.97758487	0.97466046	0.99084796	0.99178268	0.97029629	0.95010766	0.99224102

Таблиця Д.16

0.99626031	0.99947944	0.99658965	0.99739382	0.99023334	0.99865149	0.99162400	0.00000000	0.99192966	0.99039304
0.99500065	0.99668330	0.00000000	0.99235228	0.99332103	0.00000000	0.99971394	0.99835605	0.00000000	0.00000000
0.99384078	0.99365771	0.99131969	0.99012383	0.99092449	0.99549028	0.99039760	0.99702484	0.99024516	0.00000000
0.99357841	0.99250409	0.99174380	0.99839539	0.00000000	0.00000000	0.00000000	0.99530803	0.00000000	0.99100088
0.99959159	0.99715552	0.99163964	0.99560855	0.99641200	0.00000000	0.00000000	0.99604817	0.99098509	0.99640953
0.99283090	0.00000000	0.99266214	0.99595087	0.99718584	0.00000000	0.00000000	0.99851616	0.00000000	0.99313488
0.00000000	0.00000000	0.99526590	0.99445034	0.99529083	0.99114823	0.99646790	0.99210660	0.00000000	0.99010625
0.99109224	0.99438318	0.99329350	0.00000000	0.00000000	0.99806776	0.99683814	0.99017169	0.00000000	0.00000000
0.00000000	0.00000000	0.00000000	0.99401979	0.00000000	0.99335122	0.99710098	0.99763073	0.99112283	0.99122099
0.00000000	0.99573798	0.99937827	0.99219094	0.99756720	0.00000000	0.99477496	0.00000000	0.99863494	0.99077138

Таблиця Д.17

0.95268438	0.95826389	0.96687505	0.97136697	0.96558530	0.99758979	0.95379437	0.00000000	0.96140316	0.95670560
0.96656238	0.95801040	0.00000000	0.96178048	0.96434582	0.00000000	0.96857719	0.99415729	0.00000000	0.00000000
0.96226106	0.99974702	0.96210495	0.97537683	0.95864788	0.97417960	0.96481803	0.99451814	0.98134800	0.00000000
0.98673379	0.96601219	0.95008272	0.97255774	0.00000000	0.00000000	0.00000000	0.96455394	0.00000000	0.98984812
0.98020762	0.99260410	0.96187052	0.96910827	0.97143161	0.00000000	0.00000000	0.95061882	0.97246687	0.98749543
0.99907505	0.00000000	0.96037228	0.98417126	0.95066771	0.00000000	0.00000000	0.97479218	0.00000000	0.95345362
0.00000000	0.00000000	0.99340963	0.98156208	0.97712984	0.99239967	0.95945498	0.96885026	0.00000000	0.98087497
0.97162305	0.95788758	0.98740058	0.00000000	0.00000000	0.99186860	0.96259606	0.96728383	0.00000000	0.00000000
0.00000000	0.00000000	0.00000000	0.98230203	0.00000000	0.95977359	0.97239320	0.99173636	0.95338641	0.96616039
0.00000000	0.95568964	0.96680305	0.95957359	0.96073329	0.00000000	0.97640892	0.00000000	0.95984338	0.99804623

Таблиця Д.18

0.95833374	0.95945089	0.99114499	0.98704237	0.95147929	0.96472844	0.96874997	0.00000000	0.96314133	0.97233766
0.98490788	0.98697760	0.00000000	0.95055629	0.95003218	0.00000000	0.97160031	0.96433710	0.00000000	0.00000000
0.95500480	0.96316830	0.97794321	0.96012625	0.95341529	0.98315529	0.95938827	0.95733833	0.95000618	0.00000000
0.95867547	0.95892926	0.99596103	0.95354796	0.00000000	0.00000000	0.00000000	0.96196417	0.00000000	0.98237668
0.99106818	0.99695915	0.98855146	0.99764530	0.95531185	0.00000000	0.00000000	0.98197966	0.97303361	0.98616634
0.99271976	0.00000000	0.98702960	0.99940309	0.95215271	0.00000000	0.00000000	0.98176217	0.00000000	0.98862672
0.00000000	0.00000000	0.97952204	0.96322204	0.99565397	0.98012515	0.99728952	0.99962232	0.00000000	0.98116071
0.95472927	0.98135810	0.97952318	0.00000000	0.00000000	0.99720153	0.99473682	0.98028223	0.00000000	0.00000000
0.00000000	0.00000000	0.00000000	0.99505496	0.00000000	0.99310546	0.97481342	0.98710506	0.96631637	0.95287839
0.00000000	0.99180456	0.97634256	0.95816284	0.96944018	0.00000000	0.96210226	0.00000000	0.99477077	0.98011585

Таблиця Д.19

0.99282674	0.98735096	0.99227409	0.98166409	0.99244321	0.99457810	0.99785055	0.00000000	0.99475437	0.98161798
0.99826654	0.99820888	0.00000000	0.99126229	0.98959512	0.00000000	0.98708437	0.99761227	0.00000000	0.00000000
0.99278053	0.99776652	0.98279929	0.98060391	0.98907014	0.98408725	0.99148545	0.99945411	0.98399676	0.00000000
0.98624255	0.98569664	0.99042534	0.99803051	0.00000000	0.00000000	0.00000000	0.99818256	0.00000000	0.98224328
0.98276266	0.99620980	0.98313951	0.99726091	0.99208750	0.00000000	0.00000000	0.99527501	0.99210880	0.98042790
0.99108815	0.00000000	0.99398575	0.98432456	0.98143410	0.00000000	0.00000000	0.99297262	0.00000000	0.99287670
0.00000000	0.00000000	0.99985392	0.99591197	0.98442064	0.98537277	0.99701713	0.98944519	0.00000000	0.99725649
0.99516823	0.98894629	0.99751883	0.00000000	0.00000000	0.98747307	0.98763128	0.99438422	0.00000000	0.00000000
0.00000000	0.00000000	0.00000000	0.99865458	0.00000000	0.99644784	0.99244471	0.99232393	0.98109447	0.98984591
0.00000000	0.98996287	0.99536104	0.99323292	0.98573536	0.00000000	0.99184841	0.00000000	0.99931812	0.98160902

Таблиця Д.20

0.99339110	0.99313823	0.99108547	0.99325895	0.99970772	0.99969053	0.99689740	0.00000000	0.99745843	0.99741631
0.99946371	0.99197841	0.00000000	0.99983028	0.99624762	0.00000000	0.99399383	0.99492360	0.00000000	0.00000000
0.99657549	0.99726984	0.99171994	0.99423775	0.99960367	0.99211164	0.99028419	0.99213085	0.99095715	0.00000000
0.99640877	0.99502747	0.99562236	0.99651222	0.00000000	0.00000000	0.00000000	0.99338598	0.00000000	0.99969564
0.99845442	0.99425494	0.99080739	0.99337133	0.99781038	0.00000000	0.00000000	0.99970098	0.99617683	0.99096879
0.99960573	0.00000000	0.99860735	0.99827768	0.99112277	0.00000000	0.00000000	0.99585512	0.00000000	0.99871377
0.00000000	0.00000000	0.99320062	0.99347714	0.99479827	0.99043073	0.99680786	0.99650699	0.00000000	0.99914118
0.99251962	0.99069010	0.99152790	0.00000000	0.00000000	0.99500128	0.99926877	0.99785300	0.00000000	0.00000000
0.00000000	0.00000000	0.00000000	0.99620574	0.00000000	0.99295134	0.99315399	0.99169525	0.99980997	0.99585653
0.00000000	0.99882557	0.99707483	0.99673717	0.99967286	0.00000000	0.99522617	0.00000000	0.99614596	0.99910903

Таблиця Д.21

.99941525	0.99630138	0.99926780	0.99522834	0.98515607	0.99283894	0.99750249	0.00000000	0.99386416	0.99902210
0.99137383	0.98962944	0.00000000	0.98775158	0.98410579	0.00000000	0.99451609	0.98507645	0.00000000	0.00000000
0.99457358	0.98313355	0.99428581	0.99377383	0.98437521	0.99847298	0.98682834	0.99062425	0.98326596	0.00000000
0.99891363	0.99080199	0.98940573	0.98593705	0.00000000	0.00000000	0.00000000	0.98447033	0.00000000	0.98421766
0.99951000	0.98257367	0.98871309	0.99543022	0.98930941	0.00000000	0.00000000	0.99395871	0.98599924	0.99171188
0.98718296	0.00000000	0.98794928	0.99514527	0.99966121	0.00000000	0.00000000	0.98774557	0.00000000	0.99663453
0.00000000	0.00000000	0.98034959	0.98245225	0.99709994	0.99554145	0.98455990	0.98193939	0.00000000	0.99371959
0.99112478	0.99574412	0.98658492	0.00000000	0.00000000	0.98527068	0.98475185	0.98694467	0.00000000	0.00000000
0.00000000	0.00000000	0.00000000	0.98861355	0.00000000	0.98510766	0.99756888	0.99063231	0.99515130	0.99356391
0.00000000	0.98293435	0.98229105	0.98516279	0.98435794	0.00000000	0.99721013	0.00000000	0.99903205	0.98536382

Таблиця Д.22

0.99839044	0.99246390	0.99274474	0.99313058	0.99445372	0.99604706	0.99542824	0.00000000	0.99176657	0.99133015
0.99474947	0.99807811	0.00000000	0.99622839	0.99453649	0.00000000	0.99369910	0.99100551	0.00000000	0.00000000
0.99847291	0.99244923	0.99026436	0.99645900	0.99848710	0.99397813	0.99930631	0.99263907	0.99282259	0.00000000
0.99481023	0.99811515	0.99935489	0.99043598	0.00000000	0.00000000	0.00000000	0.99487612	0.00000000	0.99358308
0.99209757	0.99364569	0.99931795	0.99604823	0.99383611	0.00000000	0.00000000	0.99963975	0.99257955	0.99373581
0.99524273	0.00000000	0.99292175	0.99563528	0.99178704	0.00000000	0.00000000	0.99237506	0.00000000	0.99825035
0.00000000	0.00000000	0.99882999	0.99922922	0.99878841	0.99356725	0.99486784	0.99317394	0.00000000	0.99084120
0.99250627	0.99636642	0.99380003	0.00000000	0.00000000	0.99814874	0.99698213	0.99858595	0.00000000	0.00000000
0.00000000	0.00000000	0.00000000	0.99776653	0.00000000	0.99440971	0.99906173	0.99327480	0.99444617	0.99479527
0.00000000	0.99015069	0.99070638	0.99697661	0.99338622	0.00000000	0.99068206	0.00000000	0.99636855	0.99950328

Таблиця Д.23

0.99243968	0.99181112	0.99286629	0.99819610	0.99252602	0.99581172	0.99786146	0.00000000	0.99994099	0.99396779
0.99289038	0.99298276	0.00000000	0.99108995	0.99257709	0.00000000	0.99309763	0.99247096	0.00000000	0.00000000
0.99619646	0.99739707	0.99114531	0.99664958	0.99068539	0.99845480	0.99160272	0.99760198	0.99252817	0.00000000
0.99873040	0.99758482	0.99061421	0.99816549	0.00000000	0.00000000	0.00000000	0.99711850	0.00000000	0.99349229
0.99233303	0.99698491	0.99707525	0.99591509	0.99941607	0.00000000	0.00000000	0.99103454	0.99809632	0.99602616
0.99640444	0.00000000	0.99849267	0.99367319	0.99557253	0.00000000	0.00000000	0.99147809	0.00000000	0.99153372
0.00000000	0.00000000	0.99333289	0.99070438	0.99048131	0.99446870	0.99077741	0.99771910	0.00000000	0.99181197
0.99538445	0.99884644	0.99754118	0.00000000	0.00000000	0.99106309	0.99629814	0.99957533	0.00000000	0.00000000
0.00000000	0.00000000	0.00000000	0.99724349	0.00000000	0.99946996	0.99623083	0.99474885	0.99640473	0.99895638
0.00000000	0.99017132	0.99004503	0.99438397	0.99704407	0.00000000	0.99903101	0.00000000	0.99190798	0.99710510

Таблиця Д.24

0.99763946	0.99202781	0.99931814	0.99397017	0.99152846	0.99701730	0.99599990	0.00000000	0.99346716	0.99892164
0.99643386	0.99984340	0.00000000	0.99743720	0.99928047	0.00000000	0.99499658	0.99427184	0.00000000	0.00000000
0.99994270	0.99284190	0.99116435	0.99774120	0.99728727	0.99477069	0.99094439	0.99391667	0.99024442	0.00000000
0.99558652	0.99428509	0.99396076	0.99629720	0.00000000	0.00000000	0.00000000	0.99010495	0.00000000	0.99993325
0.99908119	0.99603121	0.99348405	0.99513340	0.99345209	0.00000000	0.00000000	0.99538243	0.99895938	0.99973799
0.99230950	0.00000000	0.99316230	0.99012561	0.99784423	0.00000000	0.00000000	0.99744656	0.00000000	0.99255833
0.00000000	0.00000000	0.99840761	0.99746102	0.99437133	0.99441201	0.99898287	0.99609911	0.00000000	0.99048310
0.99573093	0.99769113	0.99480834	0.00000000	0.00000000	0.99483011	0.99036122	0.99273858	0.00000000	0.00000000
0.00000000	0.00000000	0.00000000	0.99610575	0.00000000	0.99901456	0.99337655	0.99974510	0.99112658	0.99516440
0.00000000	0.99996334	0.99010298	0.99972125	0.99959872	0.00000000	0.99412854	0.00000000	0.99449856	0.99748569

Таблица Д.25

0.99138446	0.99421808	0.96824654	0.96501052	0.99102056	0.95757047	0.99938447	0.00000000	0.98584944	0.95444055
0.96783532	0.96940579	0.00000000	0.99948905	0.95124199	0.00000000	0.96528231	0.96099244	0.00000000	0.00000000
0.98893803	0.95277088	0.99485882	0.95981909	0.99418465	0.96618875	0.96142215	0.97251654	0.95623358	0.00000000
0.95051502	0.97771865	0.99407163	0.99037083	0.00000000	0.00000000	0.00000000	0.96780789	0.00000000	0.97695651
0.96454541	0.99717017	0.96744930	0.99439136	0.96188480	0.00000000	0.00000000	0.96452189	0.97993771	0.98370418
0.99098408	0.00000000	0.95420457	0.98430410	0.97524144	0.00000000	0.00000000	0.98516075	0.00000000	0.95037332
0.00000000	0.00000000	0.96537083	0.95323918	0.99573290	0.95119500	0.99026797	0.99099929	0.00000000	0.99425037
0.97942824	0.95064413	0.98725398	0.00000000	0.00000000	0.97787861	0.99244815	0.98891466	0.00000000	0.00000000
0.00000000	0.00000000	0.00000000	0.97671995	0.00000000	0.95304584	0.96488351	0.96891760	0.96139778	0.95230280
0.00000000	0.99382996	0.98680814	0.97696926	0.96811261	0.00000000	0.96557024	0.00000000	0.98165506	0.99361048

Таблица Д.26

0.99389804	0.99166120	0.99947637	0.99483528	0.99282102	0.99832949	0.99406072	0.00000000	0.99609386	0.99915339
0.99647877	0.99709377	0.00000000	0.99675297	0.99564254	0.00000000	0.99445626	0.99673677	0.00000000	0.00000000
0.99720992	0.99358179	0.99815661	0.99891059	0.99739864	0.99025235	0.99426082	0.99359647	0.99868262	0.00000000
0.99687705	0.99442208	0.99263950	0.99732066	0.00000000	0.00000000	0.00000000	0.99093858	0.00000000	0.99554867
0.99680000	0.99238668	0.99936738	0.99700301	0.99421392	0.00000000	0.00000000	0.99340835	0.99143147	0.99618833
0.99216251	0.00000000	0.99950557	0.99577759	0.99853472	0.00000000	0.00000000	0.99168921	0.00000000	0.99075370
0.00000000	0.00000000	0.99486856	0.99389718	0.99818798	0.99561565	0.99204502	0.99404399	0.00000000	0.99782938
0.99311574	0.99702331	0.99915075	0.00000000	0.00000000	0.99325452	0.99934315	0.99199795	0.00000000	0.00000000
0.00000000	0.00000000	0.00000000	0.99995092	0.00000000	0.99278133	0.99802763	0.99758165	0.99321498	0.99385884
0.00000000	0.99265108	0.99753559	0.99434777	0.99519002	0.00000000	0.99393772	0.00000000	0.99115985	0.99678286

Таблица Д.27

0.96603048	0.97306215	0.97297744	0.98412742	0.98994000	0.96119914	0.96195224	0.00000000	0.97362507	0.96520527
0.99705494	0.99012121	0.00000000	0.95586046	0.96221300	0.00000000	0.98722064	0.97589639	0.00000000	0.00000000
0.96083932	0.96041337	0.98427577	0.99868664	0.97201380	0.99100397	0.97629195	0.98218684	0.97816713	0.00000000
0.95971671	0.98879197	0.98205324	0.99047776	0.00000000	0.00000000	0.00000000	0.97459390	0.00000000	0.95717453
0.96419608	0.95310774	0.98064095	0.96057012	0.95703630	0.00000000	0.00000000	0.98800041	0.99815131	0.95996943
0.97649126	0.00000000	0.97172671	0.98052831	0.96082244	0.00000000	0.00000000	0.97081499	0.00000000	0.96427631
0.00000000	0.00000000	0.96634213	0.96345320	0.98112263	0.98905928	0.95655914	0.99925102	0.00000000	0.97915544
0.95495769	0.96326195	0.97480487	0.00000000	0.00000000	0.99215067	0.96822740	0.97547331	0.00000000	0.00000000
0.00000000	0.00000000	0.00000000	0.98130310	0.00000000	0.95031401	0.95064443	0.98061995	0.95292573	0.96209406
0.00000000	0.95139215	0.96102739	0.96644769	0.96616301	0.00000000	0.96482479	0.00000000	0.96933816	0.98808069

Таблица Д.28

0.99314152	0.97953139	0.96508916	0.96819590	0.96721855	0.95152995	0.99034189	0.00000000	0.96743295	0.95109296
0.97702981	0.95418703	0.00000000	0.97970756	0.97692594	0.00000000	0.97391117	0.98476978	0.00000000	0.00000000
0.99770716	0.99233408	0.99654449	0.96683280	0.97433501	0.95717638	0.96053817	0.97470971	0.99017235	0.00000000
0.96932349	0.96421854	0.99820200	0.95295128	0.00000000	0.00000000	0.00000000	0.99989523	0.00000000	0.97629111
0.95483439	0.98631284	0.98602863	0.98044503	0.96969413	0.00000000	0.00000000	0.96707442	0.99536263	0.97591829
0.99144745	0.00000000	0.99862814	0.98250927	0.98412394	0.00000000	0.00000000	0.97508842	0.00000000	0.95762923
0.00000000	0.00000000	0.97399413	0.99240293	0.99201292	0.96958368	0.97205861	0.95857566	0.00000000	0.98130109
0.99841133	0.98185112	0.99343315	0.00000000	0.00000000	0.96080012	0.97946588	0.98337145	0.00000000	0.00000000
0.00000000	0.00000000	0.00000000	0.99223972	0.00000000	0.99072113	0.96819596	0.97505280	0.97561307	0.99400281
0.00000000	0.98526339	0.98723495	0.95355313	0.95933197	0.00000000	0.99228660	0.00000000	0.95285487	0.98476645

Таблица Д.29

0.99413913	0.99116318	0.99995283	0.99426862	0.99735061	0.99706366	0.99212112	0.00000000	0.99450152	0.99817858
0.99214875	0.99360681	0.00000000	0.99649475	0.99832457	0.00000000	0.99155082	0.99492085	0.00000000	0.00000000
0.99923833	0.99073964	0.99871755	0.99308299	0.99611210	0.99602336	0.99621315	0.99898369	0.99906975	0.00000000
0.99516701	0.99669474	0.99773920	0.99029989	0.00000000	0.00000000	0.00000000	0.99373331	0.00000000	0.99844566
0.99617008	0.99831584	0.99853163	0.99163380	0.99146683	0.00000000	0.00000000	0.99816270	0.99519892	0.99401713
0.99027655	0.00000000	0.99401642	0.99514580	0.99449412	0.00000000	0.00000000	0.99136700	0.00000000	0.99386651
0.00000000	0.00000000	0.99648885	0.99361019	0.99601029	0.99096611	0.99244150	0.99406536	0.00000000	0.99125717
0.99111477	0.99769064	0.99386724	0.00000000	0.00000000	0.99889246	0.99779968	0.99036040	0.00000000	0.00000000
0.00000000	0.00000000	0.00000000	0.99380968	0.00000000	0.99426231	0.99033111	0.99674947	0.99919328	0.99388826
0.00000000	0.99440315	0.99256895	0.99002835	0.99829687	0.00000000	0.99334231	0.00000000	0.99328678	0.99218457

Таблица Д.30

0.96310912	0.99627778	0.99257895	0.98511596	0.98966731	0.96286529	0.99247715	0.00000000	0.97480580	0.97788506
0.99927142	0.95068365	0.00000000	0.98295110	0.99078834	0.00000000	0.97637803	0.96938725	0.00000000	0.00000000
0.98752148	0.95970738	0.98305241	0.99873610	0.97107432	0.99740683	0.97389656	0.95798009	0.99653078	0.00000000
0.96539783	0.98706100	0.98557168	0.97287668	0.00000000	0.00000000	0.00000000	0.98130241	0.00000000	0.97571847
0.98268028	0.99385744	0.99250164	0.96045769	0.98641913	0.00000000	0.00000000	0.98761262	0.95627830	0.99740416
0.96856233	0.00000000	0.99212344	0.95050097	0.99520078	0.00000000	0.00000000	0.99657894	0.00000000	0.95567397
0.00000000	0.00000000	0.95592318	0.97641592	0.95917915	0.98226319	0.97151156	0.95904095	0.00000000	0.98320486
0.98640668	0.99611552	0.97081433	0.00000000	0.00000000	0.99576088	0.95223134	0.96180334	0.00000000	0.00000000
0.00000000	0.00000000	0.00000000	0.97327071	0.00000000	0.98163799	0.98817775	0.97796330	0.98913074	0.96307413
0.00000000	0.99626795	0.99049688	0.97230791	0.97391219	0.00000000	0.96993816	0.00000000	0.96452118	0.97087654

Таблиця Д.31

0.00000000	0.00000000	0.00000000	0.99873338	0.99444916	0.00000000	0.99980237	0.00000000	0.00000000	0.99045534
0.00000000	0.00000000	0.99530249	0.99423852	0.00000000	0.99352027	0.99353037	0.99707287	0.00000000	0.99495374
0.99846958	0.99180187	0.99818025	0.00000000	0.99567550	0.99976378	0.00000000	0.00000000	0.00000000	0.99528192
0.00000000	0.00000000	0.00000000	0.99678236	0.99414035	0.99412851	0.99670102	0.00000000	0.99342340	0.99768816
0.99006214	0.99172712	0.99127761	0.00000000	0.99399384	0.00000000	0.99334527	0.00000000	0.99331590	0.99552422
0.99965243	0.99688023	0.99428395	0.99102249	0.00000000	0.99506008	0.99392830	0.99799519	0.99509485	0.00000000
0.99428753	0.99792719	0.00000000	0.00000000	0.99999163	0.00000000	0.99963862	0.99804201	0.99469971	0.00000000
0.00000000	0.00000000	0.99231138	0.99626146	0.99695815	0.99756756	0.00000000	0.00000000	0.99740086	0.00000000
0.99173128	0.00000000	0.99538654	0.99050744	0.99696322	0.99780260	0.00000000	0.99376802	0.99090120	0.99964942
0.99579331	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.99525109	0.99380187	0.00000000	0.99647200

Таблиця Д.32

0.00000000	0.00000000	0.00000000	0.99952454	0.99468397	0.00000000	0.98353490	0.00000000	0.00000000	0.99476233
0.00000000	0.00000000	0.96636597	0.96805761	0.00000000	0.99648998	0.98381114	0.95619163	0.00000000	0.95126657
0.97021961	0.96568850	0.97819923	0.00000000	0.95737210	0.98421362	0.00000000	0.00000000	0.00000000	0.99328703
0.00000000	0.00000000	0.00000000	0.95964860	0.99899132	0.97508245	0.99810672	0.00000000	0.97832538	0.97536960
0.95407073	0.95063560	0.96970418	0.00000000	0.95629537	0.00000000	0.99246915	0.00000000	0.99914379	0.95051172
0.99362057	0.96198224	0.96848792	0.98024889	0.00000000	0.96822319	0.99641406	0.95846442	0.99650867	0.00000000
0.96678704	0.96209843	0.00000000	0.00000000	0.97566590	0.00000000	0.97917970	0.97319986	0.97636474	0.00000000
0.00000000	0.00000000	0.99004088	0.97566771	0.96818037	0.96382759	0.00000000	0.00000000	0.98968362	0.00000000
0.96621852	0.00000000	0.99535520	0.98303087	0.99048702	0.99568882	0.00000000	0.96072252	0.95827123	0.97215036
0.98406609	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.99675917	0.97136155	0.00000000	0.95928565

Таблиця Д.33

0.00000000	0.00000000	0.00000000	0.97978704	0.96442609	0.00000000	0.99375573	0.00000000	0.00000000	0.95673998
0.00000000	0.00000000	0.95816665	0.96117689	0.00000000	0.99739116	0.98890535	0.97561227	0.00000000	0.97459001
0.97706237	0.99365281	0.96400099	0.00000000	0.99511363	0.97381741	0.00000000	0.00000000	0.00000000	0.97966373
0.00000000	0.00000000	0.00000000	0.96707797	0.96629428	0.98162702	0.99717322	0.00000000	0.96536490	0.97382061
0.96968657	0.98529321	0.96926235	0.00000000	0.98482272	0.00000000	0.96181563	0.00000000	0.95167709	0.95723062
0.97254496	0.99989473	0.99596131	0.99113332	0.00000000	0.97802584	0.95919227	0.95467900	0.95114099	0.00000000
0.96506778	0.99340780	0.00000000	0.00000000	0.98427739	0.00000000	0.95221180	0.97407642	0.99178299	0.00000000
0.00000000	0.00000000	0.98740096	0.98284956	0.98626686	0.96774463	0.00000000	0.00000000	0.97177956	0.00000000
0.98407896	0.00000000	0.96466196	0.98462179	0.98375774	0.95702089	0.00000000	0.98181862	0.98090805	0.99275525
0.98225487	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.96047672	0.98564509	0.00000000	0.95996764

Таблиця Д.34

0.00000000	0.00000000	0.00000000	0.98462286	0.99462736	0.00000000	0.98195747	0.00000000	0.00000000	0.98971239
0.00000000	0.00000000	0.98241130	0.98952677	0.00000000	0.99341170	0.99446340	0.98581894	0.00000000	0.98839144
0.98000548	0.99478507	0.98533897	0.00000000	0.99565383	0.98572916	0.00000000	0.00000000	0.00000000	0.99320707
0.00000000	0.00000000	0.00000000	0.99073143	0.99801541	0.98594736	0.98151158	0.00000000	0.99149083	0.99183314
0.99136914	0.98744752	0.99407619	0.00000000	0.98872149	0.00000000	0.98696349	0.00000000	0.98330445	0.99314961
0.99810065	0.99455237	0.99310701	0.99554843	0.00000000	0.99831842	0.99647232	0.99375120	0.99393146	0.00000000
0.98609870	0.98766501	0.00000000	0.00000000	0.98303821	0.00000000	0.99171965	0.99070509	0.99912051	0.00000000
0.00000000	0.00000000	0.99438175	0.98528508	0.99673393	0.98441426	0.00000000	0.00000000	0.98067529	0.00000000
0.99076551	0.00000000	0.99372465	0.98823781	0.98473061	0.99174872	0.00000000	0.99161643	0.98772141	0.98328571
0.98018022	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.99393845	0.99634524	0.00000000	0.99963030

Таблиця Д.35

0.00000000	0.00000000	0.00000000	0.99440581	0.99742976	0.00000000	0.99918968	0.00000000	0.00000000	0.99363184
0.00000000	0.00000000	0.99099366	0.99721181	0.00000000	0.99167058	0.99423685	0.99849685	0.00000000	0.99979966
0.99975181	0.99091534	0.99535244	0.00000000	0.99929703	0.99254544	0.00000000	0.00000000	0.00000000	0.99228318
0.00000000	0.00000000	0.00000000	0.99629165	0.99177118	0.99595310	0.99063799	0.00000000	0.99993713	0.99785007
0.99319100	0.99753452	0.99836510	0.00000000	0.99654340	0.00000000	0.99250533	0.00000000	0.99399268	0.99767401
0.99856330	0.99041621	0.99993635	0.99752255	0.00000000	0.99216805	0.99097029	0.99794544	0.99718701	0.00000000
0.99521423	0.99723304	0.00000000	0.00000000	0.99304380	0.00000000	0.99612621	0.99400701	0.99414492	0.00000000
0.00000000	0.00000000	0.99027816	0.99293278	0.99084006	0.99122968	0.00000000	0.00000000	0.99957959	0.00000000
0.99627988	0.00000000	0.99931870	0.99759429	0.99197637	0.99045048	0.00000000	0.99006450	0.99113588	0.99204685
0.99846765	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.99838673	0.99567486	0.00000000	0.99807102

Таблиця Д.36

0.00000000	0.00000000	0.00000000	0.98729642	0.98416881	0.00000000	0.99526360	0.00000000	0.00000000	0.98488838
0.00000000	0.00000000	0.99801170	0.99098444	0.00000000	0.99565689	0.99497829	0.99848801	0.00000000	0.98524439
0.98449054	0.98514324	0.98301248	0.00000000	0.98625307	0.98927698	0.00000000	0.00000000	0.00000000	0.98178501
0.00000000	0.00000000	0.00000000	0.98264294	0.99897300	0.99829913	0.98807587	0.00000000	0.99950907	0.99590683
0.98920765	0.98488892	0.98084604	0.00000000	0.99603079	0.00000000	0.99241470	0.00000000	0.98070544	0.98007300
0.99870476	0.99804925	0.99333526	0.98497062	0.00000000	0.98588269	0.98490059	0.98570720	0.99796154	0.00000000
0.99344552	0.99668208	0.00000000	0.00000000	0.98766195	0.00000000	0.99837961	0.99685653	0.98017430	0.00000000
0.00000000	0.00000000	0.99622579	0.98499503	0.98207256	0.99529376	0.00000000	0.00000000	0.99460298	0.00000000
0.99389147	0.00000000	0.99790777	0.98300854	0.99972801	0.99143210	0.00000000	0.98381260	0.99294337	0.98523691
0.99345772	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.99377867	0.98937085	0.00000000	0.99182154

Таблиця Д.37

0.00000000	0.00000000	0.00000000	0.99013291	0.99285299	0.00000000	0.99589456	0.00000000	0.00000000	0.99601110
0.00000000	0.00000000	0.99302386	0.99457150	0.00000000	0.99525310	0.99904094	0.99760384	0.00000000	0.99150398
0.99137961	0.99497557	0.99503736	0.00000000	0.99961042	0.99641918	0.00000000	0.00000000	0.00000000	0.99279568
0.00000000	0.00000000	0.00000000	0.99043553	0.99465627	0.99668894	0.99418051	0.00000000	0.99000799	0.99353340
0.99136662	0.99867777	0.99963452	0.00000000	0.99386339	0.00000000	0.99723431	0.00000000	0.99144295	0.99135930
0.99530484	0.99538904	0.99626576	0.99462027	0.00000000	0.99287055	0.99628273	0.99796815	0.99016550	0.00000000
0.99027983	0.99495029	0.00000000	0.00000000	0.99040009	0.00000000	0.99730831	0.99998944	0.99733901	0.00000000
0.00000000	0.00000000	0.99695574	0.99713701	0.99882760	0.99174270	0.00000000	0.00000000	0.99176942	0.00000000
0.99378794	0.00000000	0.99738831	0.99587391	0.99631784	0.99267374	0.00000000	0.99566455	0.99755332	0.99776832
0.99444066	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.99350434	0.99134646	0.00000000	0.99252182

Таблиця Д.38

0.00000000	0.00000000	0.00000000	0.99221542	0.99933920	0.00000000	0.99331904	0.00000000	0.00000000	0.99914935
0.00000000	0.00000000	0.99398573	0.99627539	0.00000000	0.99175749	0.99131805	0.99453600	0.00000000	0.99799874
0.99199000	0.99249695	0.99832807	0.00000000	0.99395179	0.99013202	0.00000000	0.00000000	0.00000000	0.99387220
0.00000000	0.00000000	0.00000000	0.99566032	0.99439409	0.99953659	0.99778559	0.00000000	0.99433354	0.99617331
0.99762701	0.99248200	0.99915124	0.00000000	0.99971718	0.00000000	0.99642610	0.00000000	0.99385485	0.99790915
0.99653926	0.99713900	0.99633254	0.99982858	0.00000000	0.99494708	0.99662489	0.99023707	0.99649925	0.00000000
0.99252121	0.99243294	0.00000000	0.00000000	0.99354444	0.00000000	0.99731865	0.99030439	0.99148000	0.00000000
0.00000000	0.00000000	0.99151062	0.99949467	0.99031669	0.99249115	0.00000000	0.00000000	0.99094487	0.00000000
0.99597189	0.00000000	0.99417500	0.99875045	0.99160919	0.99697653	0.00000000	0.99383016	0.99009810	0.99133914
0.99252939	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.99382087	0.99168905	0.00000000	0.99260446

Таблиця Д.39

0.00000000	0.00000000	0.00000000	0.99783994	0.99643364	0.00000000	0.99344599	0.00000000	0.00000000	0.99781684
0.00000000	0.00000000	0.99362292	0.99361254	0.00000000	0.99988999	0.99701984	0.99227605	0.00000000	0.99483417
0.99591953	0.99212742	0.99805986	0.00000000	0.99443182	0.99967349	0.00000000	0.00000000	0.00000000	0.99649549
0.00000000	0.00000000	0.00000000	0.99663900	0.99160375	0.99700239	0.99071995	0.00000000	0.99847103	0.99945588
0.99417345	0.99131779	0.99823923	0.00000000	0.99621039	0.00000000	0.99169974	0.00000000	0.99859989	0.99247434
0.99290089	0.99605443	0.99791734	0.99251592	0.00000000	0.99416089	0.99289747	0.99046586	0.99691584	0.00000000
0.99600042	0.99737972	0.00000000	0.00000000	0.99872764	0.00000000	0.99263112	0.99907064	0.99121693	0.00000000
0.00000000	0.00000000	0.99577371	0.99654118	0.99317413	0.99557056	0.00000000	0.00000000	0.99538331	0.00000000
0.99945846	0.00000000	0.99312819	0.99898220	0.99381355	0.99861880	0.00000000	0.99268903	0.99940524	0.99850143
0.99389734	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.99696683	0.99150715	0.00000000	0.99725669

Таблиця Д.40

0.00000000	0.00000000	0.00000000	0.98693496	0.96844337	0.00000000	0.97811415	0.00000000	0.00000000	0.95643650
0.00000000	0.00000000	0.97643199	0.99488685	0.00000000	0.99773929	0.99782592	0.99442966	0.00000000	0.96472664
0.96472379	0.99744157	0.99404856	0.00000000	0.95905362	0.97424233	0.00000000	0.00000000	0.00000000	0.97002606
0.00000000	0.00000000	0.00000000	0.96063587	0.96372632	0.95262283	0.96695946	0.00000000	0.96038800	0.96923403
0.99952641	0.97600265	0.95025690	0.00000000	0.99577175	0.00000000	0.99488142	0.00000000	0.98601853	0.95776382
0.97690596	0.96289042	0.97351442	0.97554976	0.00000000	0.97524175	0.95989335	0.97439476	0.97023964	0.00000000
0.97257341	0.98933886	0.00000000	0.00000000	0.95834252	0.00000000	0.95878929	0.96879446	0.95593899	0.00000000
0.00000000	0.00000000	0.97997122	0.96371607	0.99177579	0.96252302	0.00000000	0.00000000	0.96960044	0.00000000
0.96516194	0.00000000	0.95649053	0.99065157	0.98896882	0.97337901	0.00000000	0.99959167	0.95937194	0.96873922
0.98380017	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.97430786	0.96767780	0.00000000	0.95643246

Таблиця Д.41

0.00000000	0.00000000	0.00000000	0.99288895	0.99879735	0.00000000	0.99948470	0.00000000	0.00000000	0.99633574
0.00000000	0.00000000	0.99028938	0.99989500	0.00000000	0.99875712	0.99077099	0.99494374	0.00000000	0.99287661
0.99074183	0.99006315	0.99301262	0.00000000	0.99716584	0.99670959	0.00000000	0.00000000	0.00000000	0.99387878
0.00000000	0.00000000	0.00000000	0.99443522	0.99973883	0.99208491	0.99654738	0.00000000	0.99589697	0.99944774
0.99772644	0.99510921	0.99128928	0.00000000	0.99801508	0.00000000	0.99044482	0.00000000	0.99619315	0.99168484
0.99473409	0.99646580	0.99028184	0.99893870	0.00000000	0.99071247	0.99940306	0.99714723	0.99194799	0.00000000
0.99339097	0.99795778	0.00000000	0.00000000	0.99361666	0.00000000	0.99292609	0.99432192	0.99945729	0.00000000
0.00000000	0.00000000	0.99612224	0.99730554	0.99462102	0.99434368	0.00000000	0.00000000	0.99575514	0.00000000
0.99021524	0.00000000	0.99503234	0.99008238	0.99559065	0.99300180	0.00000000	0.99772579	0.99854106	0.99802883
0.99559112	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.99559298	0.99886180	0.00000000	0.99744126

Таблиця Д.42

0.00000000	0.00000000	0.00000000	0.95153098	0.96126324	0.00000000	0.96202172	0.00000000	0.00000000	0.98175572
0.00000000	0.00000000	0.99059633	0.98981604	0.00000000	0.99582780	0.97718532	0.96665901	0.00000000	0.97397964
0.95025652	0.96797115	0.98764704	0.00000000	0.96262284	0.96754991	0.00000000	0.00000000	0.00000000	0.99188306
0.00000000	0.00000000	0.00000000	0.95319550	0.95718896	0.96946406	0.99799015	0.00000000	0.95360298	0.97101055
0.95878509	0.99276551	0.96294520	0.00000000	0.95572468	0.00000000	0.95574171	0.00000000	0.97030361	0.96092096
0.96121610	0.97983529	0.99262298	0.99016440	0.00000000	0.99999489	0.99788929	0.97040319	0.95693581	0.00000000
0.96665687	0.96145485	0.00000000	0.00000000	0.97487772	0.00000000	0.96750514	0.98882038	0.95775513	0.00000000
0.00000000	0.00000000	0.96216661	0.98674871	0.99857442	0.95752120	0.00000000	0.00000000	0.97141207	0.00000000
0.95133280	0.00000000	0.96219428	0.97735105	0.95060368	0.99102898	0.00000000	0.99670407	0.95260289	0.95756776
0.98361114	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.99660270	0.95493288	0.00000000	0.99212782

Таблиця Д.43

0.00000000	0.00000000	0.00000000	0.98638160	0.96506226	0.00000000	0.95359946	0.00000000	0.00000000	0.99464796
0.00000000	0.00000000	0.95995734	0.97847959	0.00000000	0.97006312	0.97949271	0.97096618	0.00000000	0.99368426
0.96230287	0.95274251	0.97208472	0.00000000	0.99239967	0.96061426	0.00000000	0.00000000	0.00000000	0.98682371
0.00000000	0.00000000	0.00000000	0.96565465	0.95980539	0.95913878	0.99937996	0.00000000	0.95268947	0.98223910
0.99300392	0.99644426	0.96681239	0.00000000	0.98719659	0.00000000	0.95152909	0.00000000	0.97096670	0.95748875
0.95891227	0.96692790	0.99548805	0.99612161	0.00000000	0.98395432	0.96394422	0.99116264	0.98607666	0.00000000
0.98716331	0.96717076	0.00000000	0.00000000	0.98062442	0.00000000	0.99694913	0.95198397	0.95763536	0.00000000
0.00000000	0.00000000	0.98700361	0.95523286	0.98652886	0.98644118	0.00000000	0.00000000	0.97723833	0.00000000
0.98354922	0.00000000	0.97358302	0.98426367	0.97162242	0.97838396	0.00000000	0.99175866	0.99904617	0.99074647
0.95168667	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.97403091	0.96129513	0.00000000	0.95018338

Таблиця Д.44

0.00000000	0.00000000	0.00000000	0.99793956	0.99663798	0.00000000	0.99330968	0.00000000	0.00000000	0.99225653
0.00000000	0.00000000	0.99118089	0.99123784	0.00000000	0.99718274	0.99124136	0.99691381	0.00000000	0.99948086
0.99780647	0.99485601	0.99909399	0.00000000	0.99863681	0.99973706	0.00000000	0.00000000	0.00000000	0.99328693
0.00000000	0.00000000	0.00000000	0.99594405	0.99417926	0.99535720	0.99665021	0.00000000	0.99570625	0.99674062
0.99614787	0.99532333	0.99386492	0.00000000	0.99440381	0.00000000	0.99286121	0.00000000	0.99848832	0.99815506
0.99040097	0.99961173	0.99943636	0.99769880	0.00000000	0.99788689	0.99490456	0.99199114	0.99005770	0.00000000
0.99556796	0.99455402	0.00000000	0.00000000	0.99800111	0.00000000	0.99316461	0.99446609	0.99946167	0.00000000
0.00000000	0.00000000	0.99143024	0.99192466	0.99834513	0.99312694	0.00000000	0.00000000	0.99312557	0.00000000
0.99090791	0.00000000	0.99652956	0.99072523	0.99131325	0.99094809	0.00000000	0.99979933	0.99005390	0.99599589
0.99955715	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.99206611	0.99563569	0.00000000	0.99054924

Таблиця Д.45

0.00000000	0.00000000	0.00000000	0.96121110	0.97701454	0.00000000	0.99343517	0.00000000	0.00000000	0.97913690
0.00000000	0.00000000	0.99136833	0.95260582	0.00000000	0.98629572	0.97609763	0.95576805	0.00000000	0.97837295
0.96500752	0.96203794	0.99408872	0.00000000	0.97159614	0.95736312	0.00000000	0.00000000	0.00000000	0.96969226
0.00000000	0.00000000	0.00000000	0.96686962	0.95348405	0.95256802	0.98893051	0.00000000	0.95079205	0.95829317
0.99126854	0.97199086	0.99854779	0.00000000	0.95571957	0.00000000	0.98645452	0.00000000	0.96355167	0.97826828
0.96054517	0.97497975	0.98676683	0.97628242	0.00000000	0.95954995	0.99415270	0.99069080	0.95964912	0.00000000
0.99352239	0.98694747	0.00000000	0.00000000	0.97562333	0.00000000	0.95767534	0.96560553	0.95822662	0.00000000
0.00000000	0.00000000	0.95348010	0.97781570	0.98225474	0.95557935	0.00000000	0.00000000	0.99390898	0.00000000
0.99039195	0.00000000	0.99397300	0.95270972	0.96385517	0.96840765	0.00000000	0.97514395	0.96307502	0.97890114
0.95021650	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.98235752	0.99182275	0.00000000	0.97312323

Таблиця Д.46

0.99450542	0.99083821	0.99228977	0.99913337	0.99152378	0.99825817	0.99538342	0.99996135	0.99078176	0.99442678
0.99106653	0.99961898	0.99004634	0.99774910	0.99817303	0.99868695	0.99084436	0.99399783	0.99259870	0.99800068
0.99431414	0.99910648	0.99181847	0.99263803	0.99145539	0.99136069	0.99869292	0.99579705	0.99549860	0.99144955
0.99853031	0.99622055	0.99350952	0.99513250	0.99401808	0.99075967	0.99239916	0.99123319	0.99183908	0.99239953
0.99417267	0.99049654	0.99902716	0.99944787	0.99490864	0.99489253	0.99337719	0.99900054	0.99369247	0.99111203
0.99780252	0.99389739	0.99241691	0.99403912	0.99096455	0.99131973	0.99942051	0.99956135	0.99575209	0.99059780
0.99234780	0.99353159	0.99821194	0.99015403	0.99043024	0.99168990	0.99649115	0.99731722	0.99647746	0.99450924
0.99547009	0.99296321	0.99744693	0.99188955	0.99686775	0.99183511	0.99368485	0.99625619	0.99780227	0.99081126
0.99929386	0.99775713	0.99486792	0.99435859	0.99446784	0.99306349	0.99508509	0.99510772	0.99817628	0.99794831
0.99644318	0.99378609	0.99811580	0.99532826	0.99350727	0.99939002	0.99875943	0.99550156	0.99622475	0.99587045

Таблиця Д.47

0.96038711	0.96506232	0.97354617	0.96152441	0.99221544	0.95973821	0.96129609	0.95853540	0.96138321	0.97178493
0.96555511	0.99616898	0.97151037	0.95924082	0.99524405	0.99898742	0.97194350	0.95555596	0.96290323	0.97043599
0.97974480	0.96311059	0.98014215	0.98556079	0.96108734	0.95587088	0.96483379	0.96593892	0.97120834	0.97539291
0.95427579	0.96312411	0.99005073	0.95146101	1.00497105	0.98651654	0.97443045	0.97892625	0.96186418	0.97294244
0.99815443	0.97734029	0.97605679	0.96157972	0.97444489	0.98120300	0.98395678	0.96977576	0.96837183	0.99939910
0.95188694	0.99425840	0.99566434	0.98980919	0.95493561	0.96309356	0.96676784	0.98398640	0.95682766	0.98606137
0.95533809	0.98268787	0.97470870	0.98895259	0.98575185	0.99518603	0.99454613	0.96670815	0.98493729	0.95989049
0.95152705	0.98720371	0.97500112	0.97399611	0.99523611	0.98049333	0.98088332	0.99297212	0.99027447	0.97883608
0.95914612	0.96199660	0.99432560	0.95143371	0.97449507	0.95839636	0.99893403	0.98563472	0.97502358	0.97355442
0.95298094	0.98409860	0.95212156	0.95357227	0.97608249	0.95483650	0.99090743	0.99087735	0.98612198	0.95749327

Таблиця Д.48

0.98298026	0.97592975	0.99864873	0.98244957	0.99001653	0.97268989	0.97161958	0.99126569	0.95417349	0.95665855
0.95866943	0.96954689	0.99156899	0.99016822	0.95302356	0.96996289	0.97634379	0.97083997	0.98284299	0.98139867
0.96459920	0.97158256	0.95077436	0.99920319	0.95835842	0.95531082	0.96862049	1.16743413	0.97448438	0.96697467
0.99758152	0.99601660	0.95263385	0.98689290	0.96345597	0.97114178	0.97739355	0.99713685	0.97088721	0.99915262
0.96507275	0.98505494	0.98331694	0.97695632	0.98490528	0.98332640	0.95890662	0.95640072	0.99995402	0.95855605
0.95163004	0.97805999	1.13428736	0.98345877	0.95952166	0.96844583	0.97303630	0.99908190	0.95782025	0.99277614
0.98223823	0.96881361	0.95954618	0.97141265	0.97410110	0.95603058	0.97947537	0.96130938	0.96923096	0.97914932
0.96259031	0.96452203	0.98085454	0.96326405	0.99121881	0.99913317	0.98651244	0.96719385	0.97920347	0.95538845
0.99531541	0.99398269	0.99088803	0.96303640	0.97971781	0.95112563	0.97126297	0.96563594	0.95807424	0.95893831
0.97114428	0.95471147	0.97992618	0.97354621	0.98479747	0.98499439	0.98192654	0.95168019	0.95344030	0.96597999

Таблиця Д.49

0.99061729	0.99308891	0.98815238	0.99639962	0.99436718	0.99937299	0.99062668	0.98650291	0.98211258	0.99221917
0.99557604	0.98846906	0.98181647	0.98532943	0.98307313	0.98562011	0.98880170	0.99054285	0.98914849	0.99750743
0.99036104	0.99887245	0.99275418	0.99915388	0.98481414	0.99352245	0.98578129	0.99343616	0.99390281	0.98135986
0.98509580	0.98448080	0.99335665	0.99688784	0.98688925	0.99561039	0.99350664	0.98013431	0.99204341	0.98773542
0.99831982	0.98002302	0.98924898	0.98848698	0.98921833	0.99540319	0.98644944	0.99569479	0.98942714	0.98071525
0.98351749	0.99443516	1.19429993	0.98305442	0.98682249	0.99214778	0.98383491	0.99476854	0.98485699	0.99834849
0.98538123	0.99531000	0.98377324	0.98574996	0.98182227	0.99152419	0.99366726	0.99093186	0.98851458	0.99288886
0.99295235	0.99358034	0.99271573	0.99890348	0.98417870	0.99418563	0.98472461	0.98238792	0.99214608	0.98900275
0.98917451	0.99323890	0.99540571	0.98700436	0.99324019	0.98832317	0.99683858	0.99665834	0.98512882	0.99226921
0.99164498	0.99081479	0.99739882	0.98529558	0.98636148	0.98238429	0.99879659	0.99291104	0.98958926	0.99278634

Таблиця Д.50

0.99544716	0.99647311	0.99543886	0.99721047	0.99522495	0.99993705	0.99218677	0.99105798	0.99109697	0.99063591
0.99404580	0.99448373	0.99365816	0.99763505	0.99627896	0.99771980	0.99932854	0.99972741	0.99192028	0.99138874
0.99696266	0.99093820	0.99525404	0.99530344	0.99861140	0.99484853	0.99393456	0.99671431	0.99741258	0.99520052
0.99347713	0.99149997	0.99586092	0.99262145	0.99044454	0.99754933	0.99242785	0.99442402	0.99687796	0.99359228
0.99736340	0.99394707	0.99683416	0.99704047	0.99442305	0.99019578	0.99330858	0.99424309	0.99270270	0.99197054
0.99821721	0.99429921	0.99887771	0.99391183	0.99769114	0.99396792	0.99808514	0.99755077	0.99377396	0.99216019
0.99790407	0.99949304	0.99327565	0.99671264	0.99438645	0.99833501	0.99768854	0.99167254	0.99861980	0.99989872
0.99514423	0.99884281	0.99588026	0.99154752	0.99199863	0.99406955	0.99748706	0.99825584	0.99789963	0.99318524
0.99534064	0.99089951	0.99111706	0.99136293	0.99678652	0.99495177	0.99189710	0.99495006	0.99147608	0.99054974
0.99850713	0.99560560	0.99929609	0.99696667	0.99582791	0.99815397	0.99879014	0.99988912	0.99000522	0.99865439

Таблиця Д.51

0.99225133	0.99979900	0.99055360	0.98959047	0.99602695	0.98455686	0.98996189	0.99801705	0.99149322	0.99690356
0.99477281	0.99171974	0.98493469	0.99332832	0.98166966	0.99251920	0.99321889	0.99459504	0.99781504	0.99964606
0.99538058	0.99162893	0.99856626	0.99160181	0.98033966	0.98241719	0.99725421	0.98968593	0.99689711	0.98418810
0.99104583	0.99259767	0.98063982	0.99229427	0.98724823	0.98099065	0.98979140	0.98385021	0.98246167	0.98410988
0.98293030	0.98378144	0.98085305	0.99270396	0.98563734	0.99077193	0.99390326	0.98998232	0.99071602	0.98890366
0.98247865	0.98980715	0.99705996	0.99747855	0.98540589	0.98416923	0.99129959	0.99280624	0.98834058	0.98411951
0.99895866	0.98164142	0.98211419	0.98284082	0.98332921	0.99241917	0.99147420	0.98104156	0.99862403	0.99457323
0.99475683	0.98126809	0.99720881	0.99868810	0.99968797	0.99717878	0.99571118	0.99026755	0.98355205	0.98797179
0.98267863	0.98061779	0.99878283	0.98602612	0.98591068	0.98665873	0.98934136	0.99296397	0.98050456	0.99684413
0.99118065	0.99708200	0.98695758	0.98892053	0.98108479	0.98354215	0.99325616	0.98661658	0.99796972	0.98236310

Таблиця Д.52

0.99988418	0.99539982	0.99706917	0.99999492	0.99287849	0.99414523	0.99464840	0.99763957	0.99818204	0.99100222
0.99178117	0.99359635	0.99056705	0.99521886	0.99335849	0.99175669	0.99208947	0.99905154	0.99675391	0.99468468
0.99912132	0.99104012	0.99745546	0.99736267	0.99561861	0.99184194	0.99597211	0.99299937	0.99134123	0.99212602
0.99894942	0.99071453	0.99242487	0.99053754	0.99441722	0.99013283	0.99897191	0.99196658	0.99093371	0.99307367
0.99456058	0.99101669	0.99995390	0.99332093	0.99297347	0.99062045	0.99298244	0.99046351	0.99505428	0.99761426
0.99631070	0.99089892	0.99080862	0.99777241	0.99905135	0.99533772	0.99109154	0.99825809	0.99338098	0.99293973
0.99746313	0.99010337	0.99048447	0.99667916	0.99603468	0.99526102	0.99729709	0.99707253	0.99781377	0.99287977
0.99692532	0.99556670	0.99396521	0.99061591	0.99780176	0.99337584	0.99607866	0.99741254	0.99104813	0.99127888
0.99549540	0.99485229	0.99890476	0.99798960	0.99734341	0.99051332	0.99072885	0.99088527	0.99798351	0.99943008
0.99683716	0.99132083	0.99722725	0.99110353	0.99117493	0.99640718	0.99328814	0.99653812	0.99749131	0.99583186

Таблиця Д.53

0.99740032	0.99234827	0.99734958	0.99970599	0.99866930	0.99086235	0.99366437	0.99369199	0.99685028	0.99597942
0.99789364	0.99367653	0.99206028	0.99086667	0.99771934	0.99205675	0.99388272	0.99551779	0.99228953	0.99641941
0.99484480	0.99151846	0.99781932	0.99100606	0.99294066	0.99237373	0.99530872	0.99091499	0.99405315	0.99104846
0.99112284	0.99784428	0.99291570	0.99603533	0.99964423	0.99432485	0.99694752	0.99758099	0.99432642	0.99655498
0.99109755	0.99933760	0.99187461	0.99266179	0.99797830	0.99487604	0.99768958	0.99396007	0.99272939	0.99037235
0.99673295	0.99429564	0.99451739	0.99609857	0.99059403	0.99315811	0.99772722	0.99696433	0.99125332	0.99130151
0.99092352	0.99007820	0.99423109	0.99655573	0.99722923	0.99531209	0.99108818	0.99631766	0.99126500	0.99134303
0.99098594	0.99142027	0.99168251	0.99196249	0.99317480	0.99316429	0.99217563	0.99251042	0.99892922	0.99703223
0.99555738	0.99184434	0.99212031	0.99077347	0.99913800	0.99706715	0.99557789	0.99313429	0.99166204	0.99622497
0.99987935	0.99170432	0.99257792	0.99396799	0.99073995	0.99684096	0.99402388	0.99982835	0.99402184	0.99620672

Таблиця Д.54

0.99154370	0.99381345	0.99161134	0.99758112	0.99871111	0.99350777	0.99685536	0.99294149	0.99530629	0.99832423
0.99597490	0.99335311	0.99299225	0.99452593	0.99422646	0.99359606	0.99558319	0.99742545	0.99424335	0.99429356
0.99124873	0.99024434	0.99290185	0.99317521	0.99653690	0.99956936	0.99935731	0.99457886	0.99240478	0.99763898
0.99759327	0.99740648	0.99743688	0.99105920	0.99681560	0.99463261	0.99212163	0.99098519	0.99823574	0.99175010
0.99163570	0.99665987	0.99894389	0.99516558	0.99702702	0.99153590	0.99953457	0.99540884	0.99679734	0.99036563
0.99809204	0.99748619	0.99120187	0.99525045	0.99325834	0.99546449	0.99398881	0.99415093	0.99180738	0.99255387
0.99020536	0.99923676	0.99653700	0.99932614	0.99163512	0.99921097	0.99794658	0.99577394	0.99440036	0.99257614
0.99751946	0.99228669	0.99064187	0.99767330	0.99671202	0.99715213	0.99642061	0.99419048	0.99390762	0.99816140
0.99317428	0.99814540	0.99789074	0.99852264	0.99505637	0.99635661	0.99950894	0.99443964	0.99060019	0.99866750
0.99631189	0.99355074	0.99997003	0.99224171	0.99652451	0.99604991	0.99387245	0.99142187	0.99025135	0.99421112

Таблица Д.55

0.95920501	0.98628876	0.96851813	0.99207800	0.98671148	0.97855129	0.95884275	0.99786920	0.96326610	0.99622904
0.96118852	0.96867819	0.95437502	0.98200583	0.95903084	0.95225256	0.98615867	0.96737188	0.98303084	0.96919343
0.98136733	0.95108249	0.99552850	0.99002793	0.98729237	0.99065564	0.96916532	0.98086396	0.97877474	0.97650259
0.96375349	0.96243145	0.97258194	0.96138564	0.99022248	0.99930521	0.95149960	0.97678321	0.95435386	0.99010457
0.99945725	0.95334731	0.99696992	0.95090888	0.98419193	0.98918682	0.97670688	0.99426797	0.99495024	0.98129688
0.95689345	0.96089008	0.95910705	0.95209099	0.95534708	0.98082217	0.99698305	0.96772279	0.97053145	0.99921747
0.99727896	0.98383223	0.99941511	0.98834157	0.96683496	0.98311909	0.96220826	0.96477536	0.98400892	0.97639234
0.97057968	0.98013191	0.98752600	0.97917666	0.97758963	0.97917853	0.97559100	0.95412964	0.98597851	0.99980781
0.96772672	0.99856294	0.96732244	0.99432719	0.97273474	0.97067136	0.96088660	0.95628273	0.96544573	0.98630522
0.98914360	0.98468938	0.95049011	0.99216067	0.99611660	0.98854771	0.95213299	0.96890931	0.98521698	0.98647565

Таблица Д.56

0.99224277	0.99269055	0.99673031	0.99477492	0.99623716	0.99236445	0.99177124	0.99829643	0.99766922	0.99934478
0.99107889	0.99182228	0.99099095	0.99489764	0.99193245	0.99895892	0.99099090	0.99044166	0.99557295	0.99772495
0.99311940	0.99178982	0.99338956	0.99210146	0.99510153	0.99906364	0.99628924	0.99101534	0.99390855	0.99054617
0.99501283	0.99431721	0.99997560	0.99811603	1.01138658	0.99894448	0.99137547	0.99390005	0.99927356	0.99917494
0.99713574	0.99618337	0.99343288	0.99936027	0.99124774	0.99730585	0.99646477	0.99833152	0.99398282	0.99749822
0.99835221	0.99322460	0.99552262	0.99979129	0.99549309	0.99330424	0.99619472	0.99360637	0.99756510	0.99413901
0.99492345	0.99694743	0.99972734	0.99327755	0.99837803	0.99739072	0.99954174	0.99031923	0.99356869	0.99662654
0.99281502	0.99230383	0.99711129	0.99624573	0.99590609	0.99660438	0.99047555	0.99348785	0.99451341	0.99240905
0.99715045	0.99856182	0.99281508	0.99731051	0.99137763	0.99836723	0.99138602	0.99588209	0.99366157	0.99806760
0.99503781	0.99489594	0.99877049	0.99353142	0.99449444	0.99963530	0.99042298	0.99972958	0.99189207	0.99667120

Таблица Д.57

0.97932198	0.98375562	0.96805110	0.98101392	0.99055754	0.95096287	0.95419368	0.99874008	0.98256748	0.96156189
0.97017456	0.95610103	0.96342194	0.96289231	0.96658326	0.95761170	0.96740038	0.95608292	0.99420765	0.95471392
0.99650203	0.96995100	0.95237007	0.96711868	0.98679831	0.98973411	0.97724529	0.98431117	0.99468163	0.95273959
0.96518307	0.95230958	0.95977384	0.98600829	0.98608766	0.99388995	1.09006476	0.95353422	0.99613723	0.99001860
0.96429734	0.97718316	0.99923881	0.98578390	0.99194848	0.97166303	0.97353124	0.97803567	0.96345458	0.98745092
0.97519439	0.98234048	0.96538728	0.95693623	0.97377865	0.96812296	0.98940567	0.98901479	0.98342561	0.95667519
0.95107779	0.97799204	0.96504095	0.99697049	0.99904518	0.96433102	0.99004101	0.99480557	0.97987633	0.99420084
0.99718658	0.97745790	0.98641934	0.97883791	0.95129287	0.97232655	0.98231510	0.97606015	0.96861563	0.99685673
0.99147664	0.99245427	0.96862671	0.97965923	0.99362763	0.99667508	0.98342321	0.96033882	0.98269253	0.95360258
0.97033635	0.98334658	0.99668628	0.99054750	0.97422741	0.98783746	0.97085237	0.99858930	0.99939874	0.99320738

Таблица Д.58

0.96944419	0.97273709	0.96233436	0.98922115	0.99414188	0.99568558	0.97791425	0.97994341	0.95744384	0.99498567
0.97251968	0.96028362	0.99498255	0.98812928	0.99412432	0.96424751	0.98366130	0.98321400	0.95614075	0.97036592
0.96376435	0.98583349	1.116494060	0.99480994	0.99132894	0.96950133	0.97489515	0.98474026	0.99171845	0.98048148
0.97873686	0.96630211	0.97282123	0.98568978	0.99422025	0.98604278	0.95093064	0.98373882	0.97192544	0.97189101
0.95585184	0.99073408	0.96624277	0.96231141	0.96713566	0.96878461	0.97732769	0.97809601	0.96979111	0.96990654
0.97576836	0.98287653	0.99754576	0.98611743	0.97000399	0.99159357	0.95671692	0.95302334	0.95421235	0.95819492
0.96621100	0.96508634	0.95058405	0.97699525	0.95476863	1.07476953	0.98155706	0.99296602	0.99871108	0.97854192
0.99984251	0.97767708	0.97577292	0.96653410	0.97150009	0.97459031	0.95355185	0.99438696	0.95323168	0.97180925
0.99133148	0.96972673	0.98067374	0.99093204	0.99431175	0.99655558	0.95953923	0.96292911	0.99489328	0.97966809
0.97519200	0.98064048	0.99097111	0.97659446	0.96010375	0.97269467	0.97139555	0.99830264	0.98100275	0.98476950

Таблица Д.59

0.99720165	0.99346895	0.99516990	0.99556695	0.99156495	0.99562056	0.99694803	0.99426456	0.99836270	0.99731387
0.99360031	0.99454212	0.99386390	0.99775555	0.99734271	0.99430278	0.99693753	0.99945213	0.99784233	0.99705572
0.99109334	0.99389931	0.99590905	0.99459380	0.99050340	0.99228688	0.99834189	0.99015645	0.99863711	0.99078069
0.99669043	0.99500211	0.99217994	0.99571616	0.99122189	0.99671166	0.99599586	0.99055976	0.99056343	0.99152501
0.99019621	0.99435176	0.99832221	0.99617390	0.99520129	0.99863868	0.99097698	0.99908052	0.99108017	0.99516997
0.99143156	0.99559371	0.99004580	0.99766682	0.99848709	0.99916821	0.99986968	0.99505133	0.99271422	0.99100751
0.99507849	0.99585609	0.99762887	0.99082963	0.99661596	0.99516979	0.99171048	0.99938558	0.99590483	0.99440635
0.99941919	0.99655914	0.99451946	0.99839697	0.99532624	0.99553887	0.99680066	0.99367190	0.99239291	0.99578923
0.99866887	0.99406777	0.99112615	0.99443846	0.99300184	0.99401387	0.99833364	0.99403629	0.99390176	0.99360449
0.99140255	0.99260130	0.99086815	0.99429397	0.99257283	0.99297555	0.99424858	0.99119207	0.99495067	0.99706407

Таблица Д.60

0.96217867	0.98925350	0.95370448	0.96969417	0.95016971	0.96103384	0.95006503	0.95945898	0.95712420	0.96340380
0.95874460	0.95693245	0.97994428	0.99505290	0.99696899	0.96105922	0.97413357	0.96880056	0.97618900	0.96324363
0.95341786	0.97181635	0.95869265	0.95130536	0.99773391	0.97152983	0.99807793	0.98812072	0.95036743	0.98400193
0.98529754	0.98225644	0.97761549	0.96090544	0.98861831	0.96140142	0.96854324	0.99454644	0.99281885	0.97012168
0.96590096	0.98043177	0.99550976	0.99545491	0.97957972	0.96662857	0.99265318	0.97211989	0.99521777	0.95165897
0.97662132	0.98582487	0.95896509	0.96682665	0.95938565	0.96609636	0.97019284	0.97742831	0.95243693	0.97763661
0.96374057	0.96207509	0.96215726	0.95770797	0.99782082	0.99678307	0.99093572	0.98641309	0.95879059	0.96801855
0.95943950	0.95005992	0.96582098	0.98498085	0.98126276	0.97715311	0.97195186	0.96437136	0.97508296	0.98807731
0.98812040	0.97880280	0.98738314	0.98227673	0.95616098	0.97521989	0.96736307	0.95460738	0.95739247	0.95990849
0.98361351	0.97157556	0.98472020	0.96283923	0.95048793	0.97661415	0.96396960	0.99731151	0.99532216	0.96963423

ДОДАТОК Е.

РЕЗУЛЬТАТИ ТРЕТЬОГО ЕКСПЕРИМЕНТУ

Таблиця Е.1

Результати серії 1 експерименту («0» - негативний результат виконання поставленого завдання; «1» - позитивний результат виконання поставленого завдання)

Network host	Detection result									
1-10	1	1	1	1	1	1	1	1	1	1
11-20	1	1	1	1	1	1	1	1	1	1
21-30	1	1	1	1	1	1	1	1	1	1
31-40	1	1	1	1	1	1	1	1	1	1
41-50	1	1	1	1	1	1	1	1	1	1
51-60	1	1	1	1	1	0	1	1	1	1
61-70	1	1	1	1	1	1	1	1	1	1
71-80	1	1	1	1	1	1	1	1	1	1
81-90	1	1	1	1	1	1	1	1	1	1
91-100	1	1	1	1	1	1	1	1	0	1

Таблиця Е.2

Значення рівнів безпеки в серії 1 експерименту

Network host	Statistical probability of malware detection									
1-10	0,9415	0,9981	0,9799	0,9889	0,9766	0,9705	0,9992	0,9856	0,9932	0,9785
11-20	0,9999	0,9836	0,9840	0,2935	0,9860	0,9959	0,3972	0,9858	0,9323	0,9987
21-30	0,9722	0,9996	0,9988	0,9904	0,9803	0,9824	0,3512	0,9868	0,9905	0,9974
31-40	0,9840	0,9952	0,9743	0,9746	0,9849	0,9769	0,9894	0,9790	0,9963	0,9985
41-50	0,9712	0,5609	0,9680	0,9728	0,6343	0,9832	0,4970	0,9845	0,9202	0,0450
51-60	0,9897	0,9964	0,0450	0,9222	0,9745	0,9145	0,9918	0,9943	0,9683	0,9760
61-70	0,9911	0,9881	0,9956	0,9720	0,9787	0,9875	0,9831	0,9469	0,9802	0,9958
71-80	0,9734	0,9713	0,9882	0,9931	0,9721	0,9998	0,9152	0,9947	0,9862	0,9917
81-90	0,9875	0,9822	0,9768	0,4245	0,9869	0,9727	0,9737	0,9492	0,3167	0,9711
91-100	0,9976	0,9946	0,9990	0,9973	0,9863	0,9878	0,9793	0,9768	0,2378	0,9822

Таблиця Е.3

Результати серії 2 експерименту («0» - негативний результат виконання поставленого завдання; «1» - позитивний результат виконання поставленого завдання)

Network host	Detection result									
1-10	1	1	1	1	1	1	1	1	1	1
11-20	1	1	1	1	1	0	1	1	1	1
21-30	1	1	1	1	1	1	1	1	1	1
31-40	1	1	1	1	1	1	1	1	1	1
41-50	1	1	1	1	1	1	1	1	1	1
51-60	1	0	1	1	1	1	1	1	1	1
61-70	1	1	1	1	1	1	1	1	1	1
71-80	1	1	1	1	1	1	1	1	1	0
81-90	1	1	1	1	1	1	1	1	1	1
91-100	1	1	1	1	1	1	1	1	1	1

Таблиця Е.4

Значення рівнів безпеки в серії 2 експерименту

Network host	Statistical probability of malware detection									
1-10	0,9771	0,9857	0,9782	0,9835	0,9705	0,9864	0,9893	0,9912	0,9839	0,9969
11-20	0,9904	0,9859	0,9872	0,9960	0,9887	0,2223	0,9719	0,9840	0,9999	0,9902
21-30	0,9829	0,9904	0,9737	0,9925	0,9758	0,9974	0,9719	0,9736	0,9784	0,9882
31-40	0,9750	0,9905	0,9855	0,9834	0,9852	0,9888	0,9881	0,9925	0,9807	0,9715
41-50	0,9863	0,9757	0,9724	0,9882	0,9844	0,9855	0,9822	0,9990	0,9922	0,9852
51-60	0,9877	0,6217	0,9811	0,9910	0,9913	0,9861	0,9832	0,9960	0,9864	0,9790
61-70	0,9943	0,9885	0,9902	0,9770	0,9907	0,9748	0,9724	0,9964	0,9871	0,9772
71-80	0,9842	0,9978	0,9708	0,9836	0,9918	0,9741	0,9822	0,9752	0,9722	0,0106
81-90	0,9976	0,9750	0,9896	0,9714	0,9832	0,9879	0,9734	0,9786	0,9725	0,9968
91-100	0,9959	0,9959	0,9968	0,9883	0,9799	0,9882	0,9999	0,9955	0,9759	0,9716

Таблиця Е.5

Результати серії 3 експерименту («0» - негативний результат виконання поставленого завдання; «1» - позитивний результат виконання поставленого завдання)

Network host	Detection result									
1-10	1	1	0	1	1	1	1	1	1	1
11-20	1	1	1	1	1	1	1	0	1	1
21-30	0	1	1	1	1	1	1	1	1	1
31-40	1	1	1	1	1	1	1	1	1	1
41-50	1	1	1	1	1	1	1	1	1	1
51-60	1	1	1	1	1	1	0	1	1	1
61-70	1	1	1	1	1	1	1	1	1	1
71-80	1	1	1	1	1	1	1	1	1	1
81-90	1	1	1	1	1	1	1	1	1	1
91-100	1	1	1	1	1	1	1	1	1	1

Таблиця Е.6

Значення рівнів безпеки в серії 3 експерименту

Network host	Statistical probability of malware detection									
1-10	0,97433	0,99847	0,83774	0,97243	0,99027	0,98756	0,99817	0,97569	0,99047	0,97551
11-20	0,97702	0,99819	0,99104	0,99354	0,97332	0,98264	0,99490	0,45687	0,99136	0,97964
21-30	0,29488	0,99201	0,99481	0,97540	0,98302	0,97106	0,98365	0,99632	0,98440	0,97485
31-40	0,99613	0,98051	0,98402	0,97978	0,99504	0,98978	0,98217	0,97568	0,97852	0,99226
41-50	0,98662	0,98016	0,97543	0,97886	0,97290	0,99336	0,99966	0,97019	0,97355	0,98700
51-60	0,99693	0,99367	0,99921	0,97310	0,99353	0,97529	0,97837	0,97649	0,98607	0,99234
61-70	0,99845	0,99383	0,97758	0,99827	0,98207	0,99760	0,97900	0,98629	0,98641	0,97863
71-80	0,97898	0,98055	0,98398	0,97620	0,98937	0,97979	0,98302	0,98587	0,99468	0,97282
81-90	0,98498	0,98003	0,97106	0,98293	0,98832	0,97847	0,98319	0,99262	0,99076	0,98746
91-100	0,98181	0,98216	0,97401	0,97151	0,97043	0,99106	0,98438	0,98934	0,99124	0,97256

Таблиця Е.7

Результати серії 4 експерименту («0» - негативний результат виконання поставленого завдання; «1» - позитивний результат виконання поставленого завдання)

Network host	Detection result									
1-10	1	1	1	0	1	1	0	1	1	1
11-20	1	1	1	1	1	1	1	1	1	1
21-30	1	1	1	1	1	1	1	1	1	1
31-40	1	1	1	1	1	1	1	1	1	1
41-50	1	1	0	1	1	1	1	1	1	1
51-60	1	1	1	1	1	1	1	1	1	1
61-70	1	1	1	1	1	1	1	1	1	1
71-80	1	1	1	1	1	1	1	1	1	1
81-90	1	1	1	1	1	1	0	1	1	1
91-100	1	1	1	1	1	1	1	1	1	1

Таблиця Е.8

Значення рівнів безпеки в серії 4 експерименту

Network host	Statistical probability of malware detection									
1-10	0,9998	0,9891	0,9819	0,9759	0,9825	0,9802	0,2876	0,9779	0,9981	0,9811
11-20	0,9832	0,9782	0,9825	0,9824	0,9788	0,9708	0,9767	0,9810	0,9989	0,9952
21-30	0,9761	0,9802	0,9792	0,9878	0,9702	0,9769	0,9765	0,9751	0,9704	0,9873
31-40	0,9922	0,9940	0,9844	0,9816	0,9805	0,9893	0,9777	0,9920	0,9711	0,9736
41-50	0,9993	0,9771	0,9791	0,9883	0,9850	0,9863	0,9967	0,9794	0,9986	0,9915
51-60	0,9851	0,9757	0,9791	0,9705	0,9725	0,9784	0,9921	0,9897	0,9812	0,9726
61-70	0,9987	0,9962	0,9949	0,9855	0,9753	0,9809	0,9853	0,9899	0,9875	0,9780
71-80	0,9977	0,9703	0,9849	0,9862	0,9893	0,9879	0,9839	0,9858	0,9873	0,9730
81-90	0,9854	0,9944	0,9885	0,9981	0,9979	0,9950	0,1871	0,9871	0,9824	0,9819
91-100	0,9806	0,9839	0,9902	0,9982	0,9951	0,9915	0,9827	0,9933	0,9859	0,9895

Таблиця Е.9

Результати серії 5 експерименту («0» - негативний результат виконання поставленого завдання; «1» - позитивний результат виконання поставленого завдання)

Network host	Detection result									
1-10	1	1	1	1	1	1	1	1	1	1
11-20	1	1	1	1	1	1	1	1	1	1
21-30	1	1	1	1	1	1	1	1	1	1
31-40	1	1	1	1	1	1	1	1	1	1
41-50	1	1	1	1	1	1	1	1	1	1
51-60	1	1	1	1	1	1	1	1	1	1
61-70	1	1	1	1	1	1	1	1	1	1
71-80	1	1	1	1	1	1	1	1	1	1
81-90	1	1	1	1	1	0	1	0	1	1
91-100	1	1	1	1	1	1	1	1	1	1

Таблиця Е.10

Значення рівнів безпеки в серії 5 експерименту

Network host	Statistical probability of malware detection									
1-10	0,9985	0,9707	0,9857	0,9850	0,9979	0,9886	0,9705	0,9958	0,9997	0,9876
11-20	0,9873	0,9883	0,9844	0,9974	0,9830	0,9906	0,9887	0,9954	0,9814	0,9854
21-30	0,9998	0,9981	0,9836	0,9748	0,9883	0,9750	0,9902	0,9947	0,9998	0,9891
31-40	0,9757	0,9916	0,9802	0,9788	0,9906	0,9781	0,9820	0,9876	0,9779	0,9974
41-50	0,9985	0,9714	0,9818	0,9951	0,9938	0,9851	0,9966	0,9995	0,9895	0,9807
51-60	0,9894	0,9959	0,9817	0,9936	0,9960	0,9762	0,9845	0,9775	0,9915	0,9828
61-70	0,9826	0,9715	0,9878	0,9977	0,9834	0,9836	0,9844	0,9925	0,9915	0,9939
71-80	0,9735	0,9944	0,9728	0,9764	0,9916	0,9941	0,9973	0,9793	0,9921	0,9789
81-90	0,9782	0,9792	0,9818	0,9807	0,9933	0,0536	0,9794	0,9706	0,9733	0,9879
91-100	0,9717	0,9963	0,9969	0,9908	0,9841	0,9779	0,9784	0,9782	0,9741	0,9950

Таблиця Е.11

Результати серії 6 експерименту («0» - негативний результат виконання поставленого завдання; «1» - позитивний результат виконання поставленого завдання)

Network host	Detection result									
1-10	1	1	1	1	1	1	1	1	1	1
11-20	1	1	0	1	1	1	1	1	1	1
21-30	1	1	1	1	1	1	1	1	1	1
31-40	1	1	1	1	1	1	1	1	1	1
41-50	1	1	1	0	1	1	1	1	1	1
51-60	1	1	1	1	1	1	1	1	1	1
61-70	1	1	1	1	1	1	1	1	1	1
71-80	1	1	1	1	1	1	1	1	1	1
81-90	1	1	1	1	1	1	1	1	1	1
91-100	1	1	1	1	1	1	1	1	1	1

Таблиця Д.12

Значення рівнів безпеки в серії 6 експерименту

Network host	Statistical probability of malware detection									
1-10	0,9824	0,9844	0,9831	0,9831	0,9999	0,9956	0,9714	0,9767	0,9747	0,9725
11-20	0,9970	0,9816	0,9716	0,9970	1,0000	0,9957	0,9849	0,9963	0,9959	0,9814
21-30	0,9818	0,9812	0,9957	0,9975	0,9789	0,9927	0,9716	0,9956	0,9759	0,9857
31-40	0,9876	0,9987	0,9947	0,9968	0,9853	0,9769	0,9961	0,9796	0,9988	0,9991
41-50	0,9798	0,9749	0,9844	0,6296	0,9953	0,9857	0,9800	0,9815	0,9809	0,9832
51-60	0,9783	0,9843	0,9896	0,9888	0,9873	0,9855	0,9827	0,9771	0,9701	0,9944
61-70	0,9808	0,9990	0,9777	0,9733	0,9854	0,9887	0,9701	0,9981	0,9957	0,9969
71-80	0,9889	0,9978	0,9903	0,9704	0,9708	0,9851	0,9823	0,9750	0,9954	0,9949
81-90	0,9833	0,9704	0,9780	0,9848	0,9704	0,9794	0,9908	0,9888	0,9862	0,9847
91-100	0,9899	0,9706	0,9958	0,9822	0,9902	0,9709	0,9828	0,9742	0,9843	0,9930

Таблиця Е.13

Результати серії 7 експерименту («0» - негативний результат виконання поставленого завдання; «1» - позитивний результат виконання поставленого завдання)

Network host	Detection result									
1-10	1	1	1	1	1	1	1	1	1	1
11-20	1	1	1	1	1	1	1	1	1	0
21-30	1	1	1	1	1	1	1	1	0	1
31-40	1	1	1	1	1	1	1	1	1	1
41-50	1	1	1	1	1	1	1	1	1	1
51-60	1	1	1	1	1	1	1	1	1	1
61-70	1	1	1	1	1	0	1	1	1	1
71-80	1	1	1	1	1	1	1	1	1	1
81-90	1	1	1	1	1	1	0	1	1	1
91-100	1	1	1	1	1	1	1	1	1	1

Таблиця Е.14

Значення рівнів безпеки в серії 7 експерименту

Network host	Statistical probability of malware detection									
1-10	0,9723	0,9955	0,9780	0,9995	1,0000	0,9841	0,9829	0,9783	0,9918	0,9734
11-20	0,9901	0,9766	0,9924	0,9722	0,9987	0,9919	0,9726	0,9893	0,9877	0,9727
21-30	0,9888	0,9945	0,9883	0,9818	0,9945	0,9947	0,9930	0,9856	0,9720	0,9757
31-40	0,9728	0,9871	0,9800	0,9922	0,9990	0,9938	0,9852	0,9915	0,9771	0,9976
41-50	0,9901	0,9876	0,9805	0,9908	0,9802	0,9806	0,9979	0,9847	0,9742	0,9933
51-60	0,9733	0,9900	0,9843	0,9704	0,9965	0,9744	0,9866	0,9768	0,9941	0,9955
61-70	0,9890	0,9750	0,9809	0,9863	0,9812	0,9746	0,9767	0,9824	0,9941	0,9999
71-80	0,9752	0,9837	0,9919	0,9820	0,9755	0,9808	0,9906	0,9784	0,9901	0,9925
81-90	0,9793	0,9808	0,9847	0,9793	0,9776	0,9848	0,7682	0,9802	0,9825	0,9716
91-100	0,9802	0,9735	0,9717	0,9846	0,9874	0,9849	0,9959	0,9908	0,9808	0,9706

Таблиця Е.15

Результати серії 8 експерименту («0» - негативний результат виконання поставленого завдання; «1» - позитивний результат виконання поставленого завдання)

Network host	Detection result									
1-10	1	1	1	1	1	1	1	0	1	1
11-20	1	1	1	1	1	1	0	1	1	1
21-30	1	1	1	1	1	1	1	1	1	1
31-40	1	1	1	1	1	0	1	1	1	1
41-50	1	1	1	1	1	1	1	1	1	1
51-60	1	1	1	1	1	1	1	1	1	0
61-70	1	1	1	1	1	1	1	1	1	1
71-80	1	1	1	1	1	1	1	1	1	1
81-90	1	1	1	1	1	1	1	1	1	1
91-100	1	1	1	1	1	1	1	1	1	1

Таблиця Е.16

Значення рівнів безпеки в серії 8 експерименту

Network host	Statistical probability of malware detection									
1-10	0,9832	0,9845	0,9941	0,9748	0,9857	0,9753	0,9900	0,5618	0,9804	0,9871
11-20	0,9733	0,9949	0,9738	0,9722	0,9992	0,9967	0,6746	0,9863	0,9715	0,9773
21-30	0,9859	0,9792	0,9814	0,9843	0,9860	0,9911	0,9846	0,9755	0,9769	0,9747
31-40	0,9742	0,9970	0,9762	0,9809	0,9983	0,9726	0,9786	0,9989	0,9763	0,9957
41-50	0,9895	0,9861	0,9755	0,9896	0,9931	0,9898	0,9900	0,9957	0,9761	0,9782
51-60	0,9987	0,9720	0,9920	0,9934	0,9997	0,9973	0,9854	0,9763	0,9911	0,2009
61-70	0,9899	0,9752	0,9928	0,9887	0,9863	0,9920	0,9888	0,9855	0,9755	0,9892
71-80	0,9939	0,9720	0,9993	0,9713	0,9908	0,9777	0,9843	0,9715	0,9799	0,9960
81-90	0,9827	0,9947	0,9905	0,9866	0,9842	0,9749	0,9800	0,9824	0,9776	0,9992
91-100	0,9744	0,9784	0,9947	0,9947	0,9776	0,9776	0,9968	0,9801	0,9856	0,9871

Таблиця Е.17

Результати серії 9 експерименту («0» - негативний результат виконання поставленого завдання; «1» - позитивний результат виконання поставленого завдання)

Network host	Detection result									
1-10	1	1	1	1	1	1	1	1	1	1
11-20	1	1	1	1	1	0	1	1	1	0
21-30	1	1	1	1	1	1	1	1	1	1
31-40	1	1	1	1	1	1	1	1	1	1
41-50	1	1	1	1	1	1	1	1	1	1
51-60	1	1	1	1	1	1	1	1	1	1
61-70	1	1	1	1	1	1	1	1	1	1
71-80	1	1	1	1	1	1	1	1	1	1
81-90	1	1	1	1	1	1	1	1	1	1
91-100	1	1	1	1	1	1	1	1	1	1

Таблиця Е.18

Значення рівнів безпеки в серії 9 експерименту

Network host	Statistical probability of malware detection									
1-10	0,9834	0,9823	0,9999	0,9882	0,9903	0,9817	0,9895	0,9861	0,9732	0,9883
11-20	0,9995	0,9945	0,9914	0,9980	0,9714	0,7208	0,9842	0,9932	0,9728	0,9719
21-30	0,9913	0,9762	0,9988	0,9867	0,9909	0,9848	0,9850	0,9821	0,9915	0,9918
31-40	0,9998	0,9740	0,9936	0,9803	0,9706	0,9915	0,9885	0,9708	0,9768	0,9938
41-50	0,9740	0,9951	0,9775	0,9831	0,9943	0,9831	0,9921	0,9958	0,9938	0,9778
51-60	0,9705	0,9904	0,9702	0,9922	0,9873	0,9973	0,9842	0,9962	0,9795	0,9824
61-70	0,9922	0,9773	0,9863	0,9770	0,9965	0,9836	0,9937	0,9979	0,9949	0,9982
71-80	0,9757	0,9893	0,9999	0,9753	0,9747	0,9829	0,9857	0,9881	0,9961	0,9807
81-90	0,9807	0,9719	0,9886	0,9747	0,9863	0,9849	0,9998	0,9760	0,9895	0,9719
91-100	0,9785	0,9914	0,9780	0,9918	0,9816	0,9842	0,9896	0,9779	0,9968	0,9932

Таблиця Е.19

Результати серії 10 експерименту («0» - негативний результат виконання поставленого завдання; «1» - позитивний результат виконання поставленого завдання)

Network host	Detection result									
1-10	1	1	1	1	1	1	1	1	1	1
11-20	1	1	1	1	1	1	1	1	1	1
21-30	1	1	1	1	1	1	1	1	1	1
31-40	1	1	1	1	1	1	1	1	0	1
41-50	1	1	1	1	0	1	1	1	1	1
51-60	1	1	1	1	1	1	1	1	1	1
61-70	1	1	1	1	1	1	1	1	1	1
71-80	1	1	1	1	1	1	1	1	1	1
81-90	1	1	1	1	1	1	1	1	1	1
91-100	1	1	1	1	1	1	1	1	1	1

Таблиця Е.20

Значення рівнів безпеки в серії 10 експерименту

Network host	Statistical probability of malware detection									
1-10	0,9936	0,9915	0,9785	0,9703	0,9961	0,9898	0,9720	0,9986	0,9983	0,9865
11-20	0,9900	0,9939	0,9919	0,9828	0,9769	0,9835	0,9735	0,4228	0,9798	0,9891
21-30	0,9996	0,9774	0,9710	0,9770	0,9829	0,9759	0,9713	0,9877	0,9999	0,9248
31-40	0,9907	0,9776	0,8622	0,9834	0,9996	0,9762	0,9593	0,9924	0,9423	0,9471
41-50	0,9865	0,9992	0,7708	0,5601	0,3523	0,0822	0,4768	0,9796	0,9927	0,9810
51-60	0,9055	0,9991	0,7481	0,9877	0,9915	0,9911	0,9975	0,9919	0,9747	0,6081
61-70	0,9991	0,9735	0,9899	0,9911	0,9709	0,9730	0,9812	0,9889	0,9944	0,9907
71-80	0,9738	0,9992	0,9871	0,1132	0,9747	0,9872	0,9887	0,9970	0,9993	0,9871
81-90	0,5243	0,9945	0,9961	0,9437	0,9897	0,9951	0,6822	0,9825	0,9820	0,8616
91-100	0,9809	0,9883	0,9789	0,9892	0,9882	0,9717	0,9914	0,9191	0,9679	0,9792