Хмельницький національний університет
Міністерство освіти і науки України

Хмельницький національний університет
Міністерство освіти і науки України

Кваліфікаційна наукова
праця на правах рукопису

СТЕЦЮК МИКОЛА ВАСИЛЬОВИЧ

УДК 004.75:004.49

ДИСЕРТАЦІЯ

МЕТОДИ ТА ЗАСОБИ ЗАБЕЗПЕЧЕННЯ ВІДМОВОСТІЙКОСТІ ТА ЖИВУЧОСТІ СПЕЦІАЛІЗОВАНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В УМОВАХ ВПЛИВІВ ЗЛОВМИСНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

123 Комп'ютерна інженерія
(шифр і назва спеціальності)

12 Інформаційні технології
(галузь знань)

Подається на здобуття наукового ступеня доктора філософії

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

_____ Стецюк Микола Васильович
підпис

Науковий керівник: Савенко Олег Станіславович,
доктор технічних наук, професор

Хмельницький – 2022

АНОТАЦІЯ

Стецюк М. В. Методи та засоби забезпечення відмовостійкості та живучості спеціалізованих інформаційних технологій в умовах впливів зловмисного програмного забезпечення. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 123 – Комп'ютерна інженерія. – Хмельницький національний університет, Хмельницький, 2022.

Вирішення задачі підтримання постійної доступності та актуальності інформації в умовах впливів зловмисного програмного забезпечення (ЗПЗ), є однією із важливих наукових задач в сфері інформаційних технологій (ІТ), орієнтованих на побудову та подальшу експлуатацію спеціалізованих інформаційних систем (ІС).

У дисертації здійснено аналіз загроз від зловмисного програмного забезпечення та комп'ютерних атак для апаратно-програмних засобів та підтримки функціонування в них інформаційних систем в умовах впливів зловмисного програмного забезпечення. В роботі розроблено методи забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих інформаційних технологій, які покращують їх стійкість щодо впливів зловмисного програмного забезпечення та комп'ютерних атак, а також розроблено відповідні засоби і проведено з ними експериментальні дослідження.

Об'єктом дослідження є процес забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих інформаційних технологій в умовах впливів зловмисного програмного забезпечення та комп'ютерних атак.

Предметом дослідження є методи та алгоритми забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих інформаційних технологій в умовах впливів зловмисного програмного забезпечення та комп'ютерних атак.

Метою дисертаційного дослідження є покращення забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих інформаційних

технологій в умовах впливів зловмисного програмного забезпечення та комп'ютерних атак.

Наукова новизна одержаних результатів полягає в наступному:

1) *вперше розроблено* метод забезпечення відмовостійкості ІТ згідно інтеграції компонентів надмірностей, який на відміну від відомих методів, надає змогу розширити можливості ІТ в частині їх адаптивності та відповідно автоматичної зміни апаратно-програмної конфігурації, що дозволяє створювати відмовостійкі ІТ щодо впливів ЗПЗ та комп'ютерних атак;

2) *вперше розроблено* метод забезпечення живучості спеціалізованих ІТ згідно аналізу маркерів та збереженої інформації для самодослідження, який на відміну від відомих методів, зберігає інформацію про ключові процеси та здійснює їх самоаналіз, що дає можливість покращити забезпечення живучості ІТ в умовах впливів ЗПЗ та комп'ютерних атак;

3) *вперше розроблено* метод забезпечення захисту інформації спеціалізованих ІТ, який на відміну від відомих, полягає в поєднані із організаційними заходами інтеграцію в ІТ методів сегментування мережі, криптографічного захисту, двофакторної автентифікації програмного забезпечення, створення хибних об'єктів атаки, резервного копіювання з територіальним розмежуванням місць зберігання копій, що дозволяє створювати засоби з покращеним захистом інформації в умовах впливів ЗПЗ та комп'ютерних атак;

4) *вперше розроблено* метод забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих ІТ, який на відміну від відомих, полягає в інтеграції в ІТ методів забезпечення відмовостійкості, живучості та захисту інформації згідно їх збігів в станах при реагуванні на впливи ЗПЗ та комп'ютерних атак, що надало змогу створювати спеціалізовані ІС з покращеними характеристиками відмовостійкості, живучості та захисту інформації до цих впливів.

Практичне значення отриманих результатів. За результатами виконаних досліджень здобувачем розроблено методи, алгоритми та засоби забезпечення

відмовостійкості, живучості та захисту інформації спеціалізованих ІТ, в яких здійснено інтеграцію засобів забезпечення відмовостійкості, живучості та захисту інформації згідно їх збігів в станах при реагуванні на впливи ЗПЗ та комп'ютерних атак. Це дало змогу створювати спеціалізовані ІС з покращеними характеристиками відмовостійкості, живучості та захисту інформації до цих впливів. Дослідження методу забезпечення відмовостійкості спеціалізованих ІТ щодо показників надмірності та автоматичної зміни апаратно-програмного конфігурування дало змогу отримати покращення ефективності на 87% порівняно з спеціалізованою ІТ, в яку не було імплементовано цей метод. Крім того, в результаті проведених експериментальних досліджень з засобами, в які імплементовано розроблені методи, отримано покращені характеристики відмовостійкі, живучості та захисту інформації до впливів ЗПЗ та комп'ютерних атак, оціночні значення яких становлять окремо для спеціалізованої ІТ з імплементованим методом забезпечення відмовостійкості 76%, з імплементованим методом забезпечення живучості 72% та при інтеграції в спеціалізовану ІТ методу забезпечення відмовостійкості, живучості та захисту інформації 67%.

Теоретичні та практичні результати дослідження впроваджені при розробці компонентів ІС в бухгалтерії Хмельницького національного університету, при створенні ІТ в ТОВ «ІТТ» та ТОВ «Деймос», а також, в освітньому процесі Хмельницького національного університету на кафедрі комп'ютерної інженерії та інформаційних систем при викладанні дисциплін «Безпека та захист комп'ютерних систем», «Комп'ютерні мережі, адміністрування та кібербезпека», «Безпека та якість інформаційних систем та технологій».

У вступі представлено обгрунтування актуальності наукової задачі із забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих ІТ в умовах впливів ЗПЗ та комп'ютерних атак. Також, представлено зв'язок тематики дослідження з напрямками наукових досліджень відомих дослідників цієї проблеми в світі та відображено основні наукові результати роботи та її практичне значення.

У першому розділі здійснено аналіз предметної області дослідження, відомих методів забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих ІТ, а також здійснено постановку задачі дослідження.

У другому розділі представлено розробку методу забезпечення відмовостійкості ІТ згідно інтегрованого залучення компонентів резервування та надмірностей, який на відміну від відомих методів, надає змогу розширити можливості ІТ в частині її адаптивності та відповідно автоматичної зміни апаратно-програмної конфігурації, що дозволяє створювати відмовостійкі ІТ щодо впливів ЗПЗ та комп'ютерних атак. А також обгрунтовано його ефективність.

У третьому розділі представлено розроблений метод забезпечення живучості спеціалізованих ІТ згідно аналізу маркерів та збереженої інформації для самодослідження, який на відміну від відомих методів, зберігає інформацію про ключові процеси та здійснює їх самоаналіз, що дає можливість покращити забезпечення живучості ІТ в умовах впливів ЗПЗ та комп'ютерних атак та обгрунтовано його ефективність. Також, представлено розроблений метод забезпечення захисту інформації спеціалізованих ІТ, який на відміну від відомих, полягає в поєднані із організаційними заходами інтегроване в ІТ залучення механізмів сегментування мережі, криптографічного захисту, двофакторної автентифікації програмного забезпечення, хибних об'єктів атаки, резервного копіювання з територіальним розмежуванням місць зберігання копій, що дозволяє створювати засоби з покращеним рівнем захищеності інформації в умовах впливів ЗПЗ та комп'ютерних атак.

У четвертому розділі представлено розроблений метод забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих ІТ, який на відміну від відомих, полягає в поєднані та інтегруванні в ІТ механізмів забезпечення відмовостійкості, живучості та захисту інформації згідно їх збігів в станах при реагуванні на впливи ЗПЗ та комп'ютерних атак, що надало змогу створювати

спеціалізовані ІС стійкі до цих впливів та представлено архітектуру засобів, в які він імплементований та його ефективність.

У висновках представлено отримані наукові та практичні результати дослідження.

У Додатках представлено наукові публікації, в яких відображено основні наукові результати роботи, акти впровадження результатів роботи, лістинг програмного забезпечення, таблиці взаємозв'язків та блок-схеми алгоритмів.

Ключові слова: комп'ютерна система, інформаційна технологія, інформаційна система, відмовостійкість, живучість, захист інформації, зловмисне програмне забезпечення, комп'ютерні атаки, апаратно-програмні засоби.

ANNOTATION

Stetsiuk M. V. Methods and means of ensuring fault tolerance and survivability of specialized information technologies under the influence of malicious software. - Qualifying scientific work on the rights of the manuscript.

Dissertation for the degree of Doctor of Philosophy in specialty 123 - Computer Engineering. - Khmelnytsky National University, Khmelnytsky, 2022.

Solving the problem of maintaining constant availability and relevance of information in the face of malicious software (SPR) is one of the important scientific tasks in the field of information technology (IT), focused on the construction and further operation of specialized information systems (IS).

The dissertation analyzes the threats from malicious software and computer attacks for hardware and software and supports the functioning of information systems in them under the influence of malicious software. The paper develops methods to ensure fault tolerance, survivability and information protection of specialized information technologies that improve their resilience to malware and computer attacks, as well as developed appropriate tools and conducted experimental studies with them.

The object of research is the process of ensuring the resilience, survivability and protection of information of specialized information technologies in the face of malicious software and computer attacks.

The subject of research is methods and algorithms to ensure fault tolerance, survivability and protection of information of specialized information technologies in the face of malicious software and computer attacks.

The aim of the dissertation research is to improve the resilience, survivability and protection of information of specialized information technologies in the face of malicious software and computer attacks.

The scientific novelty of the obtained results is as follows:

1) for the first time, a method was developed to ensure IT resiliency according to the integration of redundancy components, which, unlike known methods, allows to expand the capabilities of IT in terms of their adaptability attacks;

2) for the first time, a method was developed to ensure the viability of specialized IT according to the analysis of markers and stored information for self-examination, which, unlike known methods, stores information about key processes and self-analyzes, which allows to improve IT viability;

3) for the first time a method of providing information protection of specialized IT was developed, which, unlike known ones, consists in integration with organizational measures of integration of IT network segmentation methods, cryptographic protection, two-factor software authentication, creation of false attack objects, backup with territorial delimitation of storage locations. copies, which allows you to create tools with improved protection of information in the face of malicious software and computer attacks;

4) for the first time, a method of ensuring the resilience, survivability and protection of information of specialized IT, which, unlike the known ones, consists in integrating into IT methods of ensuring resilience, survivability and protection of information according to their coincidences in response to malicious software and computer attacks. the ability to create

specialized IP with improved fault tolerance, survivability and protection of information to these effects.

The practical significance of the results obtained. Based on the results of the research, the applicant has developed methods, algorithms and means of ensuring resilience, survivability and protection of information of specialized IT, which integrates means of ensuring resilience, survivability and protection of information according to their coincidences in response to malicious software and computer attacks. This has made it possible to create specialized IS with improved fault tolerance, survivability and information protection against these impacts. The study of the method of ensuring the resilience of specialized IT in terms of redundancy and automatic change of hardware and software configuration allowed to obtain an efficiency improvement of 87% compared to specialized IT, which did not implement this method. In addition, experimental studies with tools that have implemented the developed methods have improved the characteristics of fault tolerance, survivability and protection of information against the effects of malicious software and computer attacks, the estimated values of which are separately for specialized IT with implemented method of ensuring resilience 76%, with the implemented method of ensuring the survivability of 72% and the integration into the specialized IT method of ensuring resilience, survivability and protection of information 67%.

Theoretical and practical results of the study were implemented in the development of IS components in the accounting department of Khmelnytsky National University, in the creation of IT in ITT and "Deimos", as well as in the educational process of Khmelnytsky National University at the Department of Computer Engineering and Information Systems "The protection of computer systems", "Computer networks, administration and cybersecurity", "Security and quality of information systems and technologies".

The introduction presents the rationale for the relevance of the scientific problem of ensuring resilience, survivability and protection of information of specialized IT in the face of malicious software and computer attacks. Also, the connection of the research topic with the directions of scientific research of famous researchers of this problem in the world is

presented and the main scientific results of the work and its practical significance are reflected.

In the first section the analysis of the subject area of research, known methods of ensuring fault tolerance, survivability and protection of information of specialized IT, and also the statement of the research task is carried out.

The second section presents the development of a method to ensure IT resiliency according to the integrated involvement of redundancy and redundancy, which, unlike known methods, allows to expand the capabilities of IT in terms of its adaptability and automatic change of hardware and software configuration to create fault-tolerant IT and computer attacks. And also its efficiency is proved.

The third section presents the developed method of ensuring the viability of specialized IT according to the analysis of markers and stored information for self-examination, which, unlike known methods, stores information about key processes and self-analysis, which improves the viability of IT user attacks and substantiated its effectiveness. Also, the developed method of providing information protection of specialized IT, which, in contrast to the known, is combined with organizational measures integrated into IT involvement of network segmentation mechanisms, cryptographic protection, two-factor software authentication, false attack objects, backup with territorial delimitation copy storage sites, which allows you to create tools with an improved level of information security in the face of malicious software and computer attacks.

The fourth section presents the developed method of ensuring resilience, survivability and protection of information of specialized IT, which, unlike the known ones, is to combine and integrate into IT mechanisms to ensure resilience, survivability and protection of information according to their coincidences in response to malicious software and computer This allowed the architect to create resistant IS to these influences and presented to the architect the means in which it is implemented and its effectiveness.

The conclusions present the obtained scientific and practical results of the study.

The Appendices present scientific publications, which reflect the main scientific results of the work, acts of implementation of work results, software listing, tables of relationships and flowcharts of algorithms.

Keywords: computer system, information technology, information system, fault tolerance, survivability, information protection, malware, computer attacks, hardware and software.

Список публікацій здобувача за темою дисертації

Наукові праці, в яких опубліковані основні наукові результати дисертації

1. Стецюк, М.В.; Стецюк, В.М.; Савенко, О.С. Модель архітектури автоматизованих інформаційних систем супроводу фінансово-господарських процесів та підтримки управлінських рішень в закладах вищої освіти. Вимірювальна та обчислювальна техніка в технологічних процесах 2019, №2, с 91 - 98. http://elar.khnu.km.ua/jspui/handle/123456789/9220

2. Стецюк, М.В.; Горошко, А.В.; Савенко, Б.О. Модель забезпечення живучості та відмовостійкості спеціалізованих інформаційних технологій в умовах руйнуючого впливу зловмисного програмного забезпечення. Вимірювальна та обчислювальна техніка в технологічних процесах 2020, №1, с. 97-103. https://doi.org/10.31891/2219-9365-2020-65-1-15

3. Стецюк, М.В.; Каштальян, А.С.; Грибинчук, В.І. Архітектура спеціалізованих інформаційних систем з врахуванням вимог живучості та відмовостійкості в умовах впливів зловмисного програмного забезпечення. Вимірювальна та обчислювальна техніка в технологічних процесах 2020, №2, с 69-77. https://doi.org/10.31891/2219-9365-2020-66-2-12

4. Стецюк, М.В. Метод забезпечення захисту інформації в спеціалізованих інформаційних технологіях при впливах зловмисного програмного забезпечення.

Вимірювальна та обчислювальна техніка в технологічних процесах 2021, №2, с 57-68. https://doi.org/10.31891/2219-9365-2021-68-2-7

5. Стецюк, М.В.; Каштальян, А.С. Абстрактна модель впливів зловмисного програмного забезпечення та метод забезпечення відмовостійкості спеціалізованих інформаційних технологій. Вісник Хмельницького національного університету 2022, №1, с 30 - 42. https://doi.org/10.31891/2307-5732-2022-305-1-31-42

6. Stetsiuk, M.V.; Kashtalian, A.S. The methods of ensuring fault tolerance, survivability and protection of information of specialized information technologies under the influence of malicious software. Computer Systems And Information Technologies (Комп'ютерні системи та інформаційні технології) 2022, №1, pp 36 - 44. http://csitjournal.khmnu.edu.ua/index.php/csit/article/view/126/78

Праці, які засвідчують апробацію матеріалів дисертації

1. Stetsyuk, M.; Bedratyuk, L.; Savenko, B.; Stetsyuk, V.; Savenko O. Providing the Resilience and Survivability of Specialized Information Technology Across Corporate Computer Networks. 1st International Workshop on Intelligent Information Technologies & Systems of Information Security. Khmelnytskyi, Ukraine, June 10-12, 2020; CEUR Workshop Proceedings, 2020; vol 2623, pp 219-238. (*Scopus*)

2. Stetsyuk, M.V.; Stetsyuk, V.M.; Savenko, B.O.; Savenko, O.S.; Dobrowolski M. Implementation of contropl by parameters of client automated workplaces of specialized information systems for neutralization malware. 2st International Workshop on Intelligent Information Technologies & Systems of Information Security. Khmelnytskyi, Ukraine, March 24-26, 2021; CEUR Workshop Proceedings, 2021; vol 2853, pp 340–352. (*Scopus*)

3. Нічепорук, А.О.; Стецюк, М.В.; Сорочинський, О.Ю.; Шаповалов Ф.В. Система для виявлення шкідливого програмного забезпечення на основі дослідження структурних особливостей виконуваних файлів. Матеріали Міжнародної науково-

практичної конференції «Інформаційні технології та взаємодії», Київ, Україна, листопад 20-21, 2018, с 291-292.

4. Стецюк, М.В.; Стецюк, В.М.; Савенко, О.С. Модель архітектури автоматизованих інформаційних систем супроводу фінансово-господарських процесів у корпоративних мережах в умовах впливу зловмисних дій / Актуальні проблеми комп'ютерних наук. Збірник наукових праць за матеріалами XII всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2020», Хмельницький: ХНУ, 2020, с 288-291.

5. Stetsiuk, M.; Nicheporyk, A.; Savenko, B. Ensuring the Fault Tolerance And Survivability of Specialized Information Technologies in Corporate Computer Networks Under the Influence of Malicious Software. Proceedings of VII International conference "Information Technology and Interactions" (IT&I-2020), Taras Shevchenko National University, Kyiv, December 02-04, 2020; Snytyuk, V., Anisimov, A., Krak, I., Nikitchenko, M. Eds.; pp 105-106.

6. Стецюк, М.В.; Савенко, О.С.; Стецюк, В.М. Модель архітектури автоматизованих інформаційних систем супроводу фінансово-господарських процесів та підтримки управлінських рішень в закладах вищої освіти. Тези II Всеукраїнської науково-практичної конференції здобувачів вищої освіти й молодих учених "Комп'ютерна інженерія і кібербезпека: досягнення та інновації", м. Кропивницький, Україна, листопад 25–27, 2020; Кропивницький: ЦНТУ, 2020; с 34 - 35.

Публікації, які додатково відображають наукові результати дисертації

1. Стецюк, М.В.; Стецюк, В.М.; Нічепорук, А.А.; Савенко, Б.О. А. с. 112335, Україна. Комп'ютерна програма «Розподілена інформаційна система з підсистемами забезпечення відмовостійкості, живучості та захисту інформації». Дата реєстрації 14.03.2022.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

АРМ – автоматизоване робоче місце

БД – база даних

ЕОМ – електронно-обчислювальна машина

ЗПЗ – зловмисне програмне забезпечення

ІС – інформаційна система

ІТ – інформаційна технологія

КС - комп’ютерна система

ОП – оперативна пам’ять

ПБЖ – пристрій безперебійного живлення

ПЗ – програмне забезпечення

СКБД – система керування базами даних

ВСТУП

Актуальність роботи. Важливою умовою існування та ведення успішної економічної діяльності для будь-якої організації є забезпечення безпеки та неперервності технологічного процесу, що неможливо без підтримання постійної доступності та актуальності інформації, з якою оперує дана організація, її партнери та клієнти. Зловмисники для досягнення певних вигод прагнуть отримати доступ до такої інформації. Для досягнення цієї мети вони використовують різноманітні засоби, зокрема зловмисне програмне забезпечення (ЗПЗ). Тому, вирішення задачі підтримання постійної доступності та актуальності інформації в умовах впливів зловмисного ПЗ, є однією із самих важливих наукових задач в сфері інформаційних технологій (ІТ), орієнтованих на побудову та подальшу експлуатацію спеціалізованих інформаційних систем (ІС). Незважаючи на великий обсяг виконаних в цьому напрямку наукових досліджень і, відповідно, отриманих наукових результатів та розробок, на сьогодні, надзвичайно актуальною, залишається задача покращення забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих ІТ в умовах впливів ЗПЗ та комп'ютерних атак.

Значний внесок в розробку теорії відмовостійкості та живучості ІТ зробили українські та іноземні вчені: Б.Г. Мудла [116], Д.В. Ланде [100, 114], А. Г. Додонов [100, 101], А. А. Білаш [96], Грищенко І. В [98], Річард Ліндер (Richard C. Linger) [48], Г. С. Теслер [116], Д. В. Флейтман [101], Могенс Бланке (Mohens Blanke) [10], Р. Р. Брукс (Richard R. Brooks) [16].

Забезпеченню працездатності ІС, постійної доступності їх функцій, гарантуванню збереження інформації за будь-яких умов їх функціонування, у тому числі в умовах впливів ЗПЗ, присвячені роботи А.А. Білаш [96], О.Г. Корченко [113], Н.В. Лукова-Чуйко [115], В.Є. Мухін [117].

Дослідженню щодо різного роду вразливостей сучасного системного та прикладного рівнів та особливостей ЗПЗ і комп'ютерних атак в комп'ютерних

системах присвячені роботи Джон Еріксон [25], Хатамі Е. (Hatami E.) [37], Роланд Крофт (Roland Croft) [19], Хан М.І. (Khan M. I.) [44], А.О. Саченко [126].

Незважаючи на велику кількість проведених наукових досліджень в сфері побудови спеціалізованих ІТ з високими параметрами відмовостійкості, живучості та захисту інформації, на сьогодні ці результати не забезпечують належного рівня для користувачів через впливи ЗПЗ та комп'ютерних атак, які призводять до збоїв в апаратно-програмних засобах та втрати інформації.

Тому, актуальною науковою задачею є розробка методів забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих ІТ, які б покращували їх стійкість щодо впливів ЗПЗ та комп'ютерних атак, а також розробка відповідних засобів.

Зв'язок роботи з науковими програмами, планами, темами.

Дисертаційне дослідження виконувалось у рамках науково-дослідних тематик Хмельницького національного університету: держбюджетної науково-дослідної теми «Агентно-орієнтована система підвищення безпеки та якості програмного забезпечення комп'ютерних систем» №1Б-2019 (№ держреєстрації 0119U100662); держбюджетної науково-дослідної теми 1Б-2021 «Самоорганізована розподілена система виявлення зловмисного програмного забезпечення в комп'ютерних мережах» (ДР № 0121U109936), в яких автор дисертації був виконавцем.

Мета і завдання дослідження.

Об'єкт дослідження – процес забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих ІТ в умовах впливів ЗПЗ та комп'ютерних атак.

Предмет дослідження – методи і алгоритми забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих ІТ в умовах впливів ЗПЗ та комп'ютерних атак.

Метою дисертаційного дослідження є покращення забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих ІТ в умовах впливів ЗПЗ та комп'ютерних атак.

Задачі дослідження формулюються в роботі наступним чином:

1. Провести аналіз методів забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих ІТ, типів ЗПЗ і комп'ютерних атак та їх потенційно можливі впливи на апаратно-програмні засоби комп'ютерних систем.

2. Розробити абстрактну модель впливів ЗПЗ та комп'ютерних атак на об'єкти комп'ютерних систем для формалізованого представлення їх в якості процесів, що протікають в комп'ютерних системах і впливають на їх працездатність.

3. Розробити метод забезпечення відмовостійкості спеціалізованих ІТ в умовах впливів ЗПЗ та комп'ютерних атак на об'єкти та процеси, що приймають участь у відновленні працездатності ІС та апаратно-програмних засобів після збоїв, які викликані внутрішніми нерегламентованими діями.

4. Розробити метод забезпечення живучості спеціалізованих ІТ в умовах впливів ЗПЗ та комп'ютерних атак на об'єкти комп'ютерних систем та процеси, які використовують механізми забезпечення живучості для відновлення працездатності ІС та апаратно-програмних засобів після збоїв, які викликані зовнішніми нерегламентованими діями та впливами ЗПЗ і комп'ютерними атаками.

5. Розробити метод забезпечення захисту інформації спеціалізованих ІТ в умовах впливів ЗПЗ та комп'ютерних атак на об'єкти комп'ютерних систем та процеси, які використовують механізми забезпечення збереження інформації в процесі одночасної її обробки та впливів.

6. Розробити метод забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих ІТ в умовах впливів ЗПЗ та комп'ютерних атак на об'єкти комп'ютерних систем та процеси, в якому поєднати впливи та стани забезпечення відмовостійкості, живучості та захисту інформації до впливів.

7. Розробити ІС з підсистемами забезпечення відмовостійкості, живучості та захисту інформації, провести з нею експериментальні дослідження щодо встановлення покращення її характеристик при впливах ЗПЗ і комп'ютерних атак та впровадити її у виробництво.

Методи дослідження. Для розв'язання поставлених задач використовуються основні положення абстрактної алгебри, теорія комп'ютерних мереж, теоретичні основи ІТ, методи захисту інформації в комп'ютерних системах, методи проєктування ІС.

Наукова новизна одержаних результатів полягає в наступному:

1) *вперше розроблено* метод забезпечення відмовостійкості ІТ згідно інтеграції компонентів надмірностей, який на відміну від відомих методів, надає змогу розширити можливості ІТ в частині їх адаптивності та відповідно автоматичної зміни апаратно-програмної конфігурації, що дозволяє створювати відмовостійкі ІТ щодо впливів ЗПЗ та комп'ютерних атак;

2) *вперше розроблено* метод забезпечення живучості спеціалізованих ІТ згідно аналізу маркерів та збереженої інформації для самодослідження, який на відміну від відомих методів, зберігає інформацію про ключові процеси та здійснює їх самоаналіз, що дає можливість покращити забезпечення живучості ІТ в умовах впливів ЗПЗ та комп'ютерних атак;

3) *вперше розроблено* метод забезпечення захисту інформації спеціалізованих ІТ, який на відміну від відомих, полягає в поєднані із організаційними заходами інтеграцію в ІТ методів сегментування мережі, криптографічного захисту, двофакторної автентифікації програмного забезпечення, створення хибних об'єктів атаки, резервного копіювання з територіальним розмежуванням місць зберігання копій, що дозволяє створювати засоби з покращеним захистом інформації в умовах впливів ЗПЗ та комп'ютерних атак;

4) *вперше розроблено* метод забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих ІТ, який на відміну від відомих, полягає в інтеграції в ІТ методів забезпечення відмовостійкості, живучості та захисту інформації згідно їх збігів в станах при реагуванні на впливи ЗПЗ та комп'ютерних атак, що надало змогу створювати спеціалізовані ІС з покращеними характеристиками відмовостійкості, живучості та захисту інформації до цих впливів.

Обґрунтованість і достовірність наукових положень, висновків і рекомендацій. Наукові положення, висновки і рекомендації дисертації обґрунтовані коректним та доцільним використанням математичного апарату, алгоритмами забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих ІТ, успішною програмною реалізацією розробленої ІС, ефективним практичним впровадженням результатів дисертаційного дослідження на підприємствах, що експлуатують комп'ютерні системи, яке продемонструвало відповідність теоретичних досліджень з реальними результатами застосування.

Практичне значення отриманих результатів. За результатами виконаних досліджень здобувачем розроблено методи, алгоритми та засоби забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих ІТ, в яких здійснено інтеграцію засобів забезпечення відмовостійкості, живучості та захисту інформації згідно їх збігів в станах при реагуванні на впливи ЗПЗ та комп'ютерних атак. Це дало змогу створювати спеціалізовані ІС з покращеними характеристиками відмовостійкості, живучості та захисту інформації до цих впливів. Дослідження методу забезпечення відмовостійкості спеціалізованих ІТ щодо показників надмірності та автоматичної зміни апаратно-програмного конфігурування дало змогу отримати покращення ефективності на 87% порівняно з спеціалізованою ІТ, в яку не було імплементовано цей метод. Крім того, в результаті проведених експериментальних досліджень з засобами, в які імплементовано розроблені методи, отримано покращені характеристики відмовостійкі, живучості та захисту інформації до впливів ЗПЗ та комп'ютерних атак, оціночні значення яких становлять окремо для спеціалізованої ІТ з імплементованим методом забезпечення відмовостійкості 76%, з імплементованим методом забезпечення живучості 72% та при інтеграції в спеціалізовану ІТ методу забезпечення відмовостійкості, живучості та захисту інформації 67%.

Теоретичні та практичні результати дослідження впроваджені при розробці компонентів ІС в бухгалтерії Хмельницького національного університету, при створенні ІТ в ТОВ ІТТ (м. Хмельницький) та ТОВ «Деймос», а також, в освітньому процесі Хмельницького національного університету на кафедрі комп'ютерної інженерії та інформаційних систем при викладанні дисциплін «Безпека та захист комп'ютерних систем», «Комп'ютерні мережі, адміністрування та кібербезпека», «Безпека та якість 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: [132] – розроблено метод забезпечення захисту інформації в спеціалізованих інформаційних технологіях при впливах зловмисного програмного забезпечення. У роботах, опублікованих у співавторстві, автору належать основні ідеї, теоретична та практична розробка положень, відображених у характеристиці наукової новизни отриманих результатів, а саме: [119, 127 - 129, 131] – розроблено архітектуру спеціалізованих інформаційних технологій з врахуванням вимог живучості та відмовостійкості в умовах впливів зловмисного програмного забезпечення та методи забезпечення відмовостійкості і живучості; [130, 133] – розроблено метод забезпечення відмовостійкості спеціалізованих інформаційних технологій; [74 - 77] – розроблено методи забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих інформаційних технологій в умовах впливів зловмисного програмного забезпечення та комп'ютерних атак; [134] – розроблено архітектуру інформаційної системи з врахуванням забезпечення відмовостійкості, живучості та захисту інформації в умовах впливів зловмисного програмного забезпечення та комп'ютерних атак.

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 також на: Міжнародній

науково-практичній конференції «Інформаційні технології та взаємодії» (м. Київ, 2018); 1st International Workshop on Intelligent Information Technologies & Systems of Information Security. - Khmelnytskyi, Ukraine, June 10-12, 2020; 2st International Workshop on Intelligent Information Technologies & Systems of Information Security. - Khmelnytskyi, Ukraine, March 24-26, 2021; XII всеукраїнської науково-практичній конференції «Актуальні проблеми комп'ютерних наук АПКН-2020» – Хмельницький: ХНУ, 2020; Proceedings of VII International conference "Information Technology and Interactions" (IT&I-2020) , 02-04 December 2020. – Taras Shevchenko National University, Kyiv; II Всеукраїнській науково-практичній конференції здобувачів вищої освіти й молодих учених "Комп'ютерна інженерія і кібербезпека: досягнення та інновації", м. Кропивницький, 25–27 листопада 2020 р.; 3st International Workshop on Intelligent Information Technologies & Systems of Information Security. - Khmelnytskyi, Ukraine, May 25-27, 2022.

Публікації. За результатами проведених досліджень основні наукові результати опубліковано у 6 наукових статтях у фахових наукових журналах України [76, 127, 129, 130, 132, 133]. Апробація засвідчена публікаціями 6 праць в матеріалах міжнародних та всеукраїнських конференцій [74, 75, 77, 119, 128, 131], з яких дві праці індексовані у наукометричній базі Scopus [74, 77]. Опубліковано 1 свідоцтво про реєстрацію авторського права на твір (програму) [134].

Структура та обсяг дисертації. Дисертаційна робота складається з анотації, змісту, переліку умовних скорочень, вступу, чотирьох розділів, висновку, списку використаних джерел та п'яти додатків. Повний обсяг роботи містить 249 сторінок друкованого тексту, з них анотація – на 11 стор., зміст – на 4 стор., перелік умовних скорочень – на 1 стор., основний текст – на 160 стор., список із 136 використаних джерел – на 19 стор., додатки – на 54 стор. Дисертація містить 55 рисунків та 4 таблиці.

РОЗДІЛ 1.
АНАЛІЗ ВІДОМИХ МЕТОДІВ І ТЕХНОЛОГІЙ ЗАБЕЗПЕЧЕННЯ ВІДМОВОСТІЙКОСТІ, ЖИВУЧОСТІ ТА ЗАХИСТУ ІНФОРМАЦІЇ СПЕЦІАЛІЗОВАНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

1.1. Роль відмовостійкості та живучості в забезпеченні ефективної роботи спеціалізованих ІТ в умовах впливів зловмисного програмного забезпечення

Інформаційні технології (ІТ) є основою для створення інформаційних систем [102 - 107]. До інформаційних систем замовниками висуваються певні вимоги. Тому, розробка самих ІТ повинна враховувати вимоги, які висуваються до таких систем. Ці вимоги повинні поєднуватись із складовими ІТ. Розглядатимемо інформаційну технологію як сукупність методів, виробничих процесів та програмно-технічних засобів, об'єднаних у технологічний ланцюжок, що забезпечує виконання інформаційних процесів з метою підвищення їхньої надійності та оперативності і зниження трудомісткості ходу використання інформаційного ресурсу [98].

Виділимо з усіх ІТ підмножину таких ІТ, які можна вважати спеціалізованими, через вирішення ними вузькоспеціалізованих задач в певних предметних областях. Використання загальних підходів і методів в ІТ, які б дозволили створювати такі спеціалізовані ІТ, є недостатніми, бо наступне їх використання відбуватиметься в умовах впливів зловмисного програмного забезпечення. А до певного класу задач, які забезпечуватимуть ІТ, є зацікавленість в зловмисників. Тому, такі ІТ потрібно розглядати і створювати як спеціалізовані з орієнтуванням на можливі впливи зловмисників через відповідні їх засоби. Таким чином, вимогами до спеціалізованих ІТ є забезпечення їх надійності в роботі через покращення в них відмовостійкості і живучості в умовах впливів зловмисного програмного забезпечення. Цей напрям потребує врахування сукупності методів, виробничих процесів та програмно-технічних засобів, які об'єднані у технологічний ланцюжок, що забезпечує виконання інформаційних процесів з метою покращення їхньої надійності, оперативності і

зниження трудомісткості ходу використання інформаційного ресурсу [122]. Все це в результаті повинно забезпечити покращення подання, отримання, зберігання, передавання, опрацювання та захист інформації в обчислювальних процесах, які регламентуватимуться ними, зокрема із врахуванням архітектури та організації функціонування відповідних програмно-технічних засобів з урахуванням державних стандартів [102 - 107].

1.1.1. Відмовостійкість та живучість як параметри забезпечення ефективної роботи спеціалізованих ІТ

Питання відмовостійкості ІТ стало нагальним з появою саме автоматизованих ІС, тому має досить тривалу історію. Вона базується на теорії маскування та практичних технологіях виявлення помилок, діагностиці несправностей і відмов та їх парируванні.

В науковий обіг його ввів ще у 1967 році д-р А. Авіженіс (департамент Computer science Каліфорнійського університету Лос-Анджелеса – UCLA) в процесі виконаних досліджень, метою яких було узагальнення різноманітних існуючих теорій та методів в уніфіковану концепцію, що об’єднала б різноманітні підходи у єдине подання усіх системних ознак виживання у структурованій системі понять і термінів. В результаті досить плідної роботи вдалось все розмаїття понять вкласти у фундаментальне поняття – концепцію відмовостійких систем (the concept of fault-tolerant systems) [5].

У цій науковій роботі, що об’єднала два фундаментальних підходи – безвідмовність (надійність) комп’ютерних систем (КС) та їх стійкість до відмов – у єдине поняття відмовостійкість (fault tolerance; resilience) і визначила його як «властивість архітектури цифрових систем, що дозволяє логічній машині продовжувати роботу і тоді, коли в реальній системі, що є її носієм, виникають різного роду несправності, збої та відмови компонентів».

На сьогодні поняття відмовостійкості не втратило своєї актуальності у забезпеченні безвідмовного функціонування ІС. Воно включено в "Словник термінів"

інформаційних технологій Національного стандарту України як «здатність функціонального блока (системи) виконувати необхідну функцію за даних умов для заданого інтервалу часу» [107, 109, 111]. Особливої актуальності воно набуло в теперішній час з появою та розвитком ефективних зловмисних засобів, які можуть використовуватись цілеспрямовано на ресурси інформаційних та комп'ютерних систем [108]. Тому, врахування цієї особливості ставить нові вимоги до забезпечення змісту та реалізації відмовостійкості в ІТ [7].

Відмовостійкість – це властивість системи зберігати повну або часткову працездатність у випадках відмов окремих елементів, що непов'язані із зовнішніми нерегламентованими діями [111].

Під живучістю інформаційної системи розуміють її властивість залишатися працездатною з допустимим зменшенням продуктивності в умовах негативних зовнішніх впливів (нерегламентованих дій) [99, 107].

Тому, вони визначають одну мету - забезпечення доступності функцій ІС, що досягається різними шляхами [100]. Від забезпечення цих параметрів в прямій залежності знаходиться ефективність функціонування всієї ІТ. Одним із її параметрів є час недоступності, тобто час, коли система не здатна виконувати свої функції в рамках певних вимог до неї [96, 101]. Для різних систем цей час різний і може знаходитись в діапазоні від нуля до певної, ще прийнятної величини. Так для автоматизованих систем управління складними технологічними процесами час недоступності дорівнює нулю. Для таких, критично-важливих систем сама вірогідність виникнення недоступності повинна прямувати до нуля.

Загалом, на сьогодні існує два основних підходи в побудові відмостійкої ІС [99, 136]. Перший базується на використані відмовостійких компонентів. Така ІС буде забезпечувати свої функції навіть при виході з ладу підкомпонентів деяких компонентів. Це самий простий метод, але разом з тим і самий дорогий, через застосування самих дорогих складових – відмовостійких компонентів ІС. Крім того, цей підхід має недолік, бо набір складових для покращення відмовостійкості відомий,

також, потенційним зловмисникам і відповідно може бути ними використаний. Тому, саме для спеціалізованих ІТ необхідним є використання таких методів і засобів забезпечення відмовостійкості, відомості про які не відомі зловмисникам.

Другий спосіб полягає в побудові відмовостійкої ІС з використанням компонентів, які не є відмостійкими. Відмостійкість в таких системах досягається за рахунок введення в них надмірності через резервування критичних ланок апаратного забезпечення, програмного забезпечення, між компонентних зв'язків та спеціальних алгоритмів функціонування ІС [4, 11, 67, 85], що передбачають її реконфігурацію при відмові деяких компонентів. Такий напрям, особливо в умовах наявності іструментарію в зловмисників для здійснення можливого доступу до компонент ІС та КС, є перспективним, бо потребує використання саме методів та засобів забезпечення відмовостійкості орієнтованих на можливі зловмисні прояви.

Методи підвищення живучості і відмовостійкості складних систем, до яких без сумніву відносяться ІС, можуть бути активними чи пасивними відносно зовнішніх негативних впливів [39, 72]. При активному методі факт виникнення відмови виявляється засобами контролю, локалізується діагностуванням і усувається шляхом автоматичної реконфігурації системи, намагаючись в умовах відмови, привести виконувану функцію до успішного завершення.

Пасивні методи засновані на функціональному резервуванні, при якому ті ж самі елементи можуть виконувати різні функції у системі, а також резервування одних елементів іншими [3, 114].

Як відомо, коефіцієнт доступності *Кд* технічної системи визначається формулою:

$$Кд=Тр/(Тр+Твз+Тв), \qquad (1.1)$$

де *Тр* – час між сусідніми збоями; *Твз* – час, необхідний для виявлення збою та пошуку шляху його обходу; *Тв* – час, необхідний для відновлення ІС після збою [9].

Її аналіз показує, що для ІС із системою забезпечення відмовостійкості на базі активних методів, коефіцієнт доступності системи буде наближатись до одиниці, а відмовостійкість, відповідно, до своєї верхньої межі, через час реакції $Твз+Тв$, який для таких систем прямує до нуля.

Для побудови таких систем немає теоретичних перешкод, але в практиці при їх реалізації, потрібно враховувати ряд значимих факторів: фінансові витрати реалізації автоматичної системи із забезпеченням її живучості та відмовостійкості; складність системи, яка призведе до зменшення надійності системи в цілому [120].

Для більшості спеціалізованих багатокористувацьких ІС, нереального часу, буде доцільним відмовитись від автоматичної системи керування відмовостійкістю (на базі активних методів) на користь автоматизованої [17]. При такому підході частина дорогих функцій управління надмірностями, присутніми в ІС, буде покладена на людину, якщо це не загрожує можливими значними втратами. В цьому випадку, час реакції на відмову буде нижчою, ніж в першому випадку. Але вирішенням задачі побудови ІС (так як і в інших задачах проєктування), є не забезпечення максимально можливої відмовостійкості системи, а знаходження прийнятного балансу параметрів системи, в рамках певного технологічного базису. В тому числі і згідно вимог критерію «відмовостійкість \ вартість».

Таким чином, забезпечення відмовостійкості та живучості спеціалізованих ІТ повинно здійснюватись із врахування можливих потреб їх функціонування в умовах впливів зловмисного програмного забезпечення [74, 75, 77, 119, 127].

1.1.2. Аналіз методів забезпечення в ІТ відмовостійкості, живучості та захищеності інформації в умовах впливів зловмисного програмного забезпечення

Відмостійкість та живучість в ІТ досягається за рахунок введення до їх складу певних надмірностей. Розрізняють наступні види надмірностей, які знайшли широке

практичне застосування [116]: структурна; часова; інформаційна; функціональна; алгоритмічна; програмна; апаратна; багаторівнева.

Методи забезпечення відмовостійкості та живучості в ІТ, що базуються на структурній надмірності передбачають наявність у об'єкта (системи) надлишкових елементів, вузлів, які замінюють основні вузли або пристрої, що відмовили, запобігаючи виходу з ладу всього об’єкта (системи). Надлишкові вузли можуть працювати паралельно з основними, або знаходитись в режимі очікування. Застосування структурної надмірності для забезпечення відмостійкості та живучості в ІТ ускладнює та обставина, що її використання знаходиться в прямій залежності від якісного складу засобів діагностики елементів, які відмовили у системі, що теж є досить складним завданням. Цей тип надмірностей може бути реалізований через мультиагентний метод підвищення надійності ІТ за допомогою зменшення числа критеріїв оптимізації [10, 36, 50, 122].

Часова надмірність [30] полягає у використанні певної частини продуктивності комп'ютера для контролю за виконанням програм та відновлення (рестарту) обчислювального процесу (запас часу для повторного виконання операції). Наприклад, з подвійним або потрійним перерахунком [2]. Розрізняють природну та штучну часові надмірності. Штучна часова надмірність передбачає введення резерву часу для повторного виконання передбачених завдань з наступним порівнянням отриманих результатів [21].

Інформаційна надмірність реалізовується через деяке повторення інформації в тій чи іншій формі, що дозволяє відновлювати вихідні дані в разі будь-яких порушень в роботі системи. Це процес дублювання частини даних інформаційної системи для забезпечення надійності і контролю даних. Досить часто застосовується в методах виявлення помилок та їх корекції. На ній заснований гібридний крипто-кодовий метод контролю і відновлення цілісності даних. Його особливістю є можливість врахування структури багатовимірного представлення даних, заснований на спільному застосуванні хеш-функції і надлишкових кодів [6, 61].

При реалізації функційної надмірності враховується можливість виконання однієї і тієї ж задачі різними засобами, що входять до складу системи. Наявність дублюючих пристроїв, дозволяє забезпечувати максимально ефективне завантаження технічних засобів при вирішенні конкретного класу задач, отримувати максимальну продуктивність при розв'язанні заданого класу задач. Разом з тим, у разі відмови пристрою, його задачі візьме на себе дублюючий, але при цьому ефективність роботи системи в цілому може зменшитися. Прикладом використання такої надмірності є кластерний метод побудови ІС. Він передбачає створення географічно розподілених, кластерних структур [37, 57].

Алгоритмічна надмірність полягає в застосуванні алгоритмів роботи системи, які включають додаткові правила та приписи понад мінімально необхідне, які забезпечують задовільні результати в разі наявності або виникнення помилок в процесі обробки інформації. Алгоритмічна надмірність передбачає наявність тимчасової надмірності і є засобом її реалізації. Цей тип надмірності використовується в методі локальних контрольних точок [58, 79]. Його особливістю є необхідність безпосередньої участі розробника прикладної програми в реалізації методу забезпечення відмовостійкості, зокрема у формуванні контрольних точок і процедур відновлення. Він включає схему збереження в пам'яті обчислювальних вузлів даних прикладної програми, які формують узгоджену глобальну контрольну точку. В її рамках здійснюється дублювання локальних контрольних точок, що дозволяє відновити обчислювальний процес, якщо число відмов не перевищує допустимого для даної схеми рівня. Вадою методу «контрольна точка / перезапуск» є збільшення часу виконання програм, що збільшує вартість виконання задачі в цілому [33].

На використані алгоритмічної надмірності базується ще один метод – метод багатоатрибутного прийняття рішень. Його суть в формуванні програмного забезпечення таким чином, щоб він складався із надлишкового набору програмних модулів, що характеризуються певним набором атрибутів, оцінка яких в ситуації

відмови, дозволяє знайти максимальне наближення до ідеального вирішення задачі [58, 125].

Програмна надмірність використовується для контролю і забезпечення достовірності найбільш важливих рішень з управління та обробки інформації. Оскільки комп'ютерна програма є результатом кодування алгоритму засобами деякої мови програмування, то фактично, програмна надмірність - це наслідок алгоритмічної надмірності.

Врахування в ІТ апаратної конфігурації та її можливих комбінацій чи модифікацій є важливим при створенні хибних об'єктів атаки для комп'ютерних атак чи зловмисного програмного забезпечення. Це дає змогу сформувати надмірність за рахунок не тільки функційної надмірності, а саме апаратної архітектури і її модифікації, яка наявна.

Багаторівнева надмірність полягає у комплексному застосуванні декількох видів надмірності. Також при її реалізації можливі поєднання алгоритмічної, програмної, функціональної і часової надмірності.

Всі розглянуті надмірності знайшли широке застосування при вирішенні задач, пов'язаних з розробкою технологій реалізації ІТ з прийнятними параметрами відмовостійкості та живучості. В архітектурі ІТ чітко вирізняються дві функціональні частини – серверна та клієнтська, кожна з яких, в свою чергу, поділяються на апаратну платформу та програмні комплекси. Тому, задача забезпечення відмовостійкості та живучості вирішується для кожної із цих складових своїми методами, які є актуальними науковими задачами. Загалом, якщо абстрагуватись від деталізації процесів та алгоритмічного забезпечення, задача забезпечення відмовостійкості та живучості ІТ, вирішується з використанням методів, які є спільними як для серверної, так і для клієнтської частин ІС, які базуються на використані різних типів надмірностей. Але при реалізації кожної складової ІТ, через їх високу складність і насиченість різноманітними, нетривіальними об'єктами, набір методів та їх застосування в кожній окремій ситуації сильно різняться. Незважаючи на те, що

обидві частини будучи складовими єдиної, логічно нерозривної ІТ, виконують в рамках неї свої, специфічні функції. Тому, забезпечення відмовостійкості для кожної складової ІТ досягається все ж різними шляхами. Але при цьому ІТ повинна бути так спроєктована, щоб відповідати не тільки наступним основним вимогам з [122], але і враховувати можливість функціонування в умовах впливів зловмисного програмного забезпечення. ІТ повинна проєктуватись так, щоб у ній був відсутній компонент, відмова якого приведе до повної відмови всієї системи. Для систем реального часу [24, 86], додатково, накладаються часові обмеження досягнення результату.

Таким чином, забезпечення відмовостійкості та живучості в спеціалізованих ІТ в умовах впливів зловмисного програмного забезпечення є актуальним напрямом досліджень. Наявні види надмірностей, такі як структурна, часова, інформаційна, функціональна, алгоритмічна, програмна, апаратна, багаторівнева, використовуються як можливі варіанти забезпечення відмовостійкості та живучості в спеціалізованих ІТ і це застосування збалансовано в поєднанні з можливими впливами зловмисного програмного забезпечення. Разом з тим, побудовані ІС згідно спроєктованих ІТ без врахування можливих впливів зловмисного програмного забезпечення можуть призводити до неправильного функціонування, незавершення розпочатих операцій та втрати інформації [74, 75, 77, 127-130].

1.2. Аналіз ЗПЗ та методів його виявлення

1.2.1. Аналіз відомих вразливостей програмного забезпечення

Розрізняють вразливості конфігурації програмного забезпечення (ПЗ) та вразливості самого ПЗ. Вразливості конфігурації пов’язані із помилками налаштування програмних систем. Цю роботу виконують адміністратори систем. Оскільки це – адміністратори/оператори, то вони мають різний рівень кваліфікації,

можуть помилятись з різних причин. Тому, причиною вразливостей конфігурації може бути людський фактор.

Стосовно вразливостей самого ПЗ, то до них відносяться недоліки в програмному забезпечені, які можуть використовуватись ЗПЗ або зловмисниками для нелегального проникнення в комп'ютерні системи [8, 26, 55, 64]. Причина їх появи в більшості випадків полягає у зростаючій складності програмних систем та обмежених можливостях людини. Аналіз цих вразливостей ПЗ показує, що їх можна виділити в такі групи: переповнення буфера; підвішений вказівник; вбудовування SQL-коду; міжсайтовий скриптинг тощо [1, 45, 56, 62].

Переповнення буфера є одним із самих часто застосовуваних способів перехоплення управління комп'ютерною системою [25]. Це пов'язано з тим, що більшість мов програмування високого рівня використовують технологію, яка передбачає розміщення в стекові процесора даних, що є сумішшю даних програми з даними керування, а саме адреси початку стека та адреси повернення з виконуваної процедури. Це проявляється тоді, коли обсяг даних, які приймає програма для розміщення в стекові, не контролюється. Наприклад, вразливість стека протоколів TCP/IP [28, 38, 63, 94] – жорстка логіка функціонування, відсутність перевірки автентичності суб'єктів взаємодії в базових протоколах тощо. Як наслідок, це можливість підміни DHCP сервера протоколу динамічного налаштування вузла через фальсифікацію першої відповіді

Метод згідно використання підвішеного вказівника (Dangling pointer,wild pointer dangling reference) реалізовується так, що вказівник вказує на дані, яких уже не існує. На сьогодні відомі три випадки появи таких покажчиків [29].

Перший випадок полягає у звільненні пам'яті. Виникає у випадку звільнення сторінок пам'яті програмою без обнулення вказівника, що на неї вказував. Це особливі випадки порушення безпеки пам'яті. В багатьох випадках вони призводять до таких явищ, як неіснуючі Інтернет-посилання [59;65]. Другий випадок полягає у виклику неіснуючої функції. Таке може статись після знищення об'єкта, наприклад функції [41].

Таке може статись при використані програмістом динамічного завантаження процедур в деяку область пам’яті. При видалені такої процедури з оперативної пам’яті (ОП), без відповідної корекції вказівника на неї, гарантовано отримується підвішений покажчик. Третій випадок може відбутись через вихід змінної за межі визначення, наприклад адресної зміної. Для випадку, коли адреса виклику функції, або адреса переходу формується динамічно, відповідно до деякого правила. У випадку якогось, не врахованого програмістом виключення, можлива ситуація, коли отримана адреса буде некоректною, а по суті підвішеним покажчиком. Підвішений покажчик, для всіх розглянутих випадків, може вказувати на пам’ять, яка містить дані, програмний код, або код операційної системи. У випадку здійснення переходу за цим покажчиком, це може призвести до різних за наслідками, проблем. Наявність такої вразливості в ПЗ, у випадку її виявлення зловмисниками, може стати місцем атаки ЗПЗ на КС [41]. Відповідальність за появу в ПЗ підвішених покажчиків можна розділити між програмістом та інструментом, який він використовує. У багатьох мовах програмування (наприклад, C) явне видалення об'єкта з пам'яті або знищення стекового кадру при поверненні не змінює значення пов'язаного покажчика. Покажчик все одно вказує на те саме місце в пам'яті, навіть якщо посилання було видалено, воно може використовуватися для іншої мети [80, 84].

Метод згідно вбудовування SQL-коду полягає в підміні тексту SQL-оператора у випадках, коли відсутній контроль за даними, що вводяться операторами ІС. В результаті цього можуть бути модифіковані дані в базі даних (БД), що приведе її в неактуальний стан або буде отримано нелегальний доступ до даних. Це одна із самих небезпечних вразливостей для спеціалізованих ІТ і побудованих на їх основі ІС. Суть таких атак полягає у впровадженні в дані (передані через GET, POST запити або значення Cookie) довільного SQL коду. Якщо атакований сайт вразливий і виконує спотворені SQL-оператори, то у зловмисника появляється повний контроль над БД [118]. Найбільш відомим прикладом такої вразливості є вставлені введені

користувачем дані в динамічний оператор SQL прямо без перевірки,. Таке можливо для коду написаного на мові Java [110].

Метод здійснення міжсайтового скриптингу (Cross-Site Scripting), або XSS-атака реалізовується зловмисником шляхом впровадження на стороні клієнта скрипта, який буде в подальшому виконувати потрібні для зловмисника дії. XSS-атака відбувається, коли зловмисник отримує можливість впровадити скрипт (найчастіше JavaScript) в сторінку, що видається веб-застосунком, і виконати його в браузері клієнта. Переважно це здійснюється за допомогою перемикання контексту даних HTML в скриптовому контексті. Найчастіше тоді, коли впроваджується новий код HTML, Javascript або CSS-розмітка [59].

Наведені групи вразливостей не є вичерпними і можуть бути інші класифікації вразливостей ПЗ. Наприклад, Американська організація MITRE опублікувала список з 25 найбільш небезпечних вразливостей ПЗ, із зазначенням ідентифікаторів CWE (Common Weakness Enumeration), які використовуються як інструмент ранжирування вразливостей за рівнем небезпеки [18].

Тому, із зростанням складності ПЗ кількість його вразливостей та їх видів може зростати, що потребуватиме врахування цієї особливості при проєктуванні спеціалізованих ІТ.

1.2.2. Вразливості мов програмування

Вразливості ПЗ, які можуть бути використані ЗПЗ, не завжди є результатом помилок розробників. Часто вони пов’язані із вразливостями присутніми безпосередньо в засобах мов програмування і, в такому випадку, представляють загрозу безпеці створюваних з їх допомогою програмних засобів [88]. В зв’язку з цією причиною постає питання вибору безпечної програмної платформи для реалізації свого проєкту.

Компанія Veracode опублікувала результати дослідження [83] залежності числа вразливостей в коді від використовуваної мови програмування. В рамках дослідження було виконано статичний аналіз понад 200 тисяч застосунків, який показав, що найбільша кількість пов'язаних з безпекою помилок присутня в коді проєктів на ASP, ColdFusion і PHP. Рейтинг платформ за кількістю критичних вразливостей, що привноситься в код розроблюваного програмного забезпечення представлено у [81].

1. Classic ASP - 1686 проблем на мегабайт коду (з них критичних 1112).
2. ColdFusion - 262 проблеми на мегабайт коду (з них критичних 227).
3. PHP - 184 проблеми на мегабайт коду (з них критичних 47).
4. Java - 51 проблема на мегабайт коду (з них критичних 5.2).
5. .NET - 32 проблеми на мегабайт коду (з них критичних 9.7).
6. C ++ - 26 проблем на мегабайт коду (з них критичних 8.8).
7. iOS - 23 проблеми на мегабайт коду (з них критичних 0.9).
8. Android - 11 проблем на мегабайт коду (з них критичних 0.4).
9. JavaScript - 8 проблем на мегабайт коду (з них критичних 0.09).

При цьому була помічена тенденція, яка говорить, що отримані рейтинги мов програмування, з точки зору продукування ними вразливостей в створене з їх використанням ПЗ, не завжди відображають реальну картину. Це пов’язано з тим, що для більш часто використовуваних мов програмування, відповідно, виявляється більше вразливостей і, як наслідок, вони втрачають в рейтингу.

Сам розвиток мов програмування йде по шляху збільшення продуктивності роботи програміста, зменшення вимог до його кваліфікації, що в свою чергу, веде до значного розширення їх функціональних можливостей, збільшення гнучкості програмування, впровадження все охоплючих алгоритмів контролю, а це означає, що їх складність буде тільки зростати, а відповідно, кількість присутніх в ньому вразливостей, як мінімум не буде зменшуватись. Загалом же, із проведеного дослідження компанії Veracode [83], можна зробити висновок, що розроблене ПЗ,

незважаючи на заходи протидії, завжди буде містити вразливості, хоча б по тій причинні, що їх містять інструменти, які використовувались для його створення.

Тому, врахування вразливостей мов програмування при забезпеченні відмовостійкості та живучості спеціалізованих ІТ повинно відбуватись вже не на рівні систем програмування, а саме на рівні їх проєктування.

1.2.3. Методи виявлення зловмисного програмного забезпечення

Все, існуюче на сьогодні, програмне забезпечення, покликане боротись із ЗПЗ [40, 55], класифікується за групами, відповідно до реалізованого в ній конкретного методу пошуку ЗПЗ. Але в цьому контексті важливим є не реалізація одного чи групи методів в спеціалізованій ІТ, а розробка методів забезпечення відмовостійкості та живучості спеціалізованих ІТ в умовах впливу ЗПЗ без використання в них саме механізмів виявлення ЗПЗ. Спрямованість щодо об’єкту виявлення в антивірусних методах потребує дослідження для врахування при забезпеченні відмовостійкості та живучості спеціалізованих ІТ.

Результати статистики кількості ЗПЗ зображені на рис. 1.1 демонструють стрімку динаміку збільшення кількості щорічно [15]. Це суттєво актуалізує проблему створення спеціалізованих ІТ з врахуванням їх можливих функціонувань в умовах впливу ЗПЗ.

Тому, проаналізуємо відомі методи виявлення ЗПЗ [13, 55, 29, 56, 51, 69, 119, 132]. Метод сканування базується на послідовному перегляді пам'яті комп'ютера, завантажувальних секторів і файлів, що перевіряються в пошуку сигнатур відомих вірусів [126]. За сигнатури вірусу береться унікальна послідовність байтів, що належить вірусу і не зустрічається в інших програмах, тобто вона є ідентифікатором або підписом вірусу. Вона дозволяє однозначно ідентифікувати наявність вірусу в файлі, навіть якщо файл цілком є вірусом. Всі сигнатури відомих вірусів складають антивірусну базу [84]. Надійність методу пошуку за сигнатурою обмеженої довжини

не надто висока. Віруси модифікують і це збільшує їх живучість. Зберігання сигнатур усіх відомих вірусів вимагає невиправдано багато пам'яті. Тому, досить зберігати

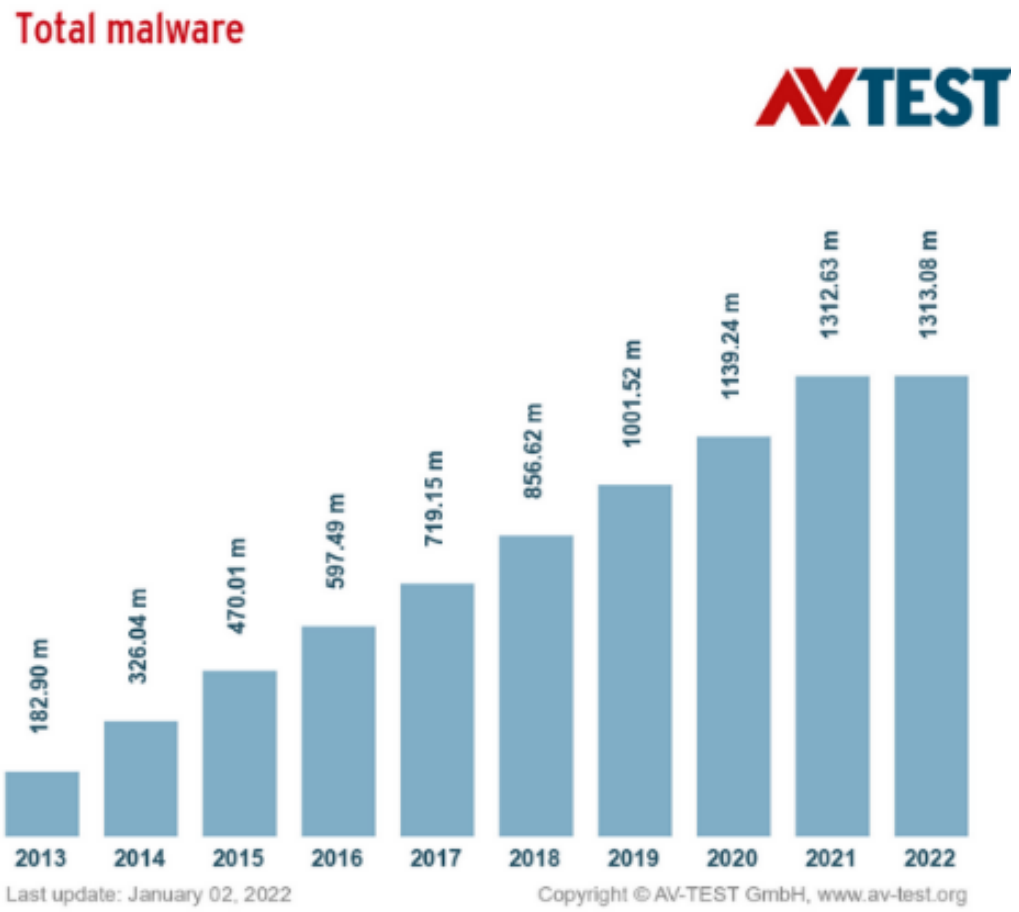


Рис. 1.1 – Результати статистики кількості ЗПЗ з ресурсу https://www.av-test.org/en/statistics/malware/

тільки контрольну суму сигнатур вірусів. При підозрі на вірус необхідно привести підозрюваний код до канонічного вигляду, підрахувати контрольну суму і порівняти з еталоном. За сигнатури можна брати характерний для цього вірусу фрагмент коду, наприклад, фрагмент обробника переривання. Не всі віруси мають сигнатури у вигляді рядків байт, для деяких вдається, як сигнатури використовувати регулярні вирази, а деякі віруси можуть взагалі не мати сигнатур, наприклад, поліморфні. Незважаючи на свої переваги в точності виявлення ЗПЗ, сигнатурний аналіз, як метод,

має недоліки, які часто стають причиною вірусних атак [60]: зростання баз сигнатур та залежність від них; неможливість виявлення нових вірусів, відомості про які не містяться в базах сигнатур. Застосуванням даного методу за змістом можна віднести до комп'ютерних атак. Також, використовується цей метод і до поведінкових сигнатур. Але до забезпечення відмовостійкості та живучості ІТ він не може бути застосовний чи використаний як частина чи повністю, бо не передбачається наповнення кожної ІТ базами сигнатур вірусів. Крім того, технології з використанням баз сигнатур не пов'язані із завданнями проєктованих спеціалізованих ІТ.

Метод контролю цілісності заснований на використані підрахунку контрольної суми програмних модулів. На відміну від методу сканування сигнатур, метод контролю цілісності дозволяє виявляти сліди діяльності будь-яких, в тому числі невідомих вірусів, для яких в базі даних ще не внесені сигнатури. Під контрольною сумою розуміють деяке значення, розраховане за набором даних із застосуванням певного алгоритму. Алгоритм підрахунку контрольних сум задається вибором хеш-функції згортки [84]. Значення контрольних сум програмного модуля може зберігатись в самому файлі, наприклад дописуватись в його кінець, або в спеціальний службовий файл. Кожен із способів має свої недоліки та переваги. При зберіганні контрольних сум в контрольованому файлі ЗПЗ може, після інфікування файлу програми, перерахувати і змінити його контрольну суму. Таке може відбутись при компрометації хеш-функції. В другому випадку ЗПЗ не зможе змінити контрольну суму, але виявити факт порушення цілісності зможе лише та програма, яка раніше розрахувала і зберегла контрольну суму контрольованого файлу. В подальшому, отримана контрольна сума може бути використана для перевірки цілісності цих же даних при їх передачі або зберіганні. Ця обставина дозволяє використовувати наперед підраховані контрольні суми для детектування присутності ЗПЗ в файлух програм [112]. Проблеми методу контролю цілісності полягають в тому, що для підрахунку контрольних сум використовуються хеш-функції, які генерують бітові згортки певної довжини. Переважну частку застосувань хеш-функцій «беруть на себе» алгоритми

MD5, SHA-1, SHA-256, STREEBOG. Незважаючи на те, що використовувані для контролю цілісності інформації хеш-функції повинні гарантувати стійкість до колізій, розвиток засобів дешифрування, використання наперед згенерованих Rainbow tables - «Райдужних» таблиць та інших методів дозволяє знайти колізії за прийнятний для зловмисника час і, таким чином, уникнути перебору всіх можливих варіантів [49, 61]. Перші успішні спроби злому даної хеш-функції MD5 датуються 1993 роком: дослідники Берт ден Боєр і Антон Боссіларіс показали, що в алгоритмі можливі псевдоколізії [104, 107]. Індійські дослідники Сомітра Кумар Санада і Палаш Саркар опублікували знайдені ними колізії для 22 ітерацій SHA-256 і SHA-512. У вересні того ж року вони представили метод конструювання колізій для врізаних варіантів SHA-2 (21 ітерація). З приведених прикладів видно, що з розвитком розрахункових засобів зростають і можливості по компрометації хеш-функцій підрахунку контрольних сум, а відповідно ці вразливості можуть використовуватись розробниками ЗПЗ [6, 89]. Оскільки криптографічні хеш-функції використовуються для підтвердження незмінності вхідної інформації, можливість швидкого знаходження колізій для них рівноцінна їх дискредитації. В загрозі компрометування використовуваних хеш-функцій для контролю цілісності програмних файлів проявляється недолік методу контролю цілісності.

Крім проаналізованих проблем, метод контролю цілісності має ще один недолік, який звужує сферу його застосування. Тому, цей метод не є всеосяжним. Існує досить чисельна група програмних файлів, щодо яких метод контролю цілісності на основі підрахунку контрольної суми, на практиці, не може бути застосованим. Її особливістю є складна внутрішня структура файлів, яка змінюється в процесі функціонування програми. До таких файлів, наприклад відносяться файли СУБД, розроблені в середовищі MS Access. Отримуються вони шляхом компіляції розроблених в MS Access БД суміщених з СУБД, написаною на мові VBA в цьому середовищі. В результаті компіляції отримуються файли з розширенням MDE, ACCDE. Ці програмні файли мають складну внутрішню стуктуру. Окрім, суто програмних модулів вони

включають в себе такі складні об'єкти, як БД. В процесі свого функціонування такі файли постійно змінюють вміст цих внутрішніх об’єктів і тим самим змінюють свою контрольну суму. Тому, цей клас програм не має сталої контрольної суми, що робить неможливим використання методу контролю цілісності для виявлення ЗПЗ в них. Наведений приклад є не єдиним. До цієї групи файлів відносяться всі програмні файли, які можуть вносити зміни у свій вихідний файл. Але це може бути використано при забезпеченні відмовостійкості і живучості проєктованих спеціалізованих ІТ.

1.3. Аналіз негативних впливів ЗПЗ на фунціонування спеціалізованих ІТ

1.3.1. Негативні впливи, що діють на клієнтську частину ІС

Проаналізуємо фактори, що негативно впливають на відмовостійкість ІС на стороні клієнта. Це потрібно для того, щоб виробити адекватні заходи протидії. За результатами проведеного аналізу, побудовано модель впливу [120] негативних факторів на відмостійкість клієнтської частини ІС.

Серед зовнішніх факторів найбільшу загрозу представляють собою збої в роботі енергосистем живлення та природні явища (наприклад, гроза), які можуть призвести до відмов компонентів комп’ютерів та комп’ютерних мереж [46, 22, 32].

Не менш деструктивним по відношенню до забезпечення роботи ІС є вплив зловмисного ПЗ. Причому він проявляється тим сильніше, чим більше вразливостей містить вся сукупність програмного забезпечення ІС [68, 23]. Це пов’язано з тим, що велика частина зловмисного ПЗ, цілеспрямовано використовує вразливості ПЗ для проникнення в нього з подальшою реалізацією своїх деструкцій.

Наступний фактор - помилки в коді системного програмного забезпечення. Раніше цей фактор мав непереборну силу. На сьогодні ситуація змінилась і ліцензійні ОС можуть налаштовуватись із включеною функцією автоматичного оновлення програмного забезпечення ОС [51, 53]. Це виключає людський фактор та зменшує

навантаження на персонал, що займається обслуговуванням ІС, хоча в повній мірі це не вирішує проблему, а лише зменшує вірогідність деструктивного прояву. Причина в тому, що системне ПЗ представляє собою складну систему і кожне виправлення, раніше знайденої помилки, не гарантує відсутності привнесення нової.

Програмне забезпечення клієнтських АРМ на протязі досить тривалого життєвого циклу ІС, з різних причин, в тому числі і через виявленні в ньому помилки, може змінюватись, проходячи свої власні цикли оновлення.

1.3.2. Негативні впливи, що діють на серверну частину ІС

Зовнішні фактори, які зменшують відмовостійкість серверної частини нейтралізуються у той же спосіб, що і в клієнтській частині ІС. Через те, що сервер ІС є місцем знаходження БД, де зосереджена вся інформація, що обробляється в системі, то для нього такий фактор, як збої в системі живлення є особливо небезпечним. Це пов’язано з тим, що БД, будучи сама складно організованою системою, є чутливою до порушення технології поводження з нею [44, 54]. Раптове зникнення живлення, або вихід з ладу апаратного забезпечення сервера в силу флуктуацій напруги може призвести, з високою ймовірністю, до пошкодження БД з негативними наслідками для інформації [47, 48].

З метою недопущення такого розвитку подій, в контур системи живлення вводиться ПБЖ з подвійним перетворенням напруги та достатнім часом забезпечення автономної роботи сервера. Крім того, ПБЖ повинен мати контролер стану, що включає в себе функцію передачі серверу сигналу розвантаження у випадку, коли через тривале зникнення зовнішнього живлення, буде вичерпано внутрішній запас його електроенергії, що не допустить руйнування БД.

Другим негативним фактором є ЗПЗ. Тут дуже дієвими для захисту серверної частини ІС є використання файрвола, з виписаними для нього правилами таким чином, щоб тільки з визначених IP-адрес була можливість створення з’єднань. Крім

цього обов'язково береться під контроль підозріла активність в мережі, мінімізується ПЗ і служби, що працюють на сервері, залишаються відкритими лише необхідні порти. Вся робота сервера підпорядковується основній задачі - забезпечення інформаційних потреб клієнтів ІС через безперешкодне функціонування БД.

Не менш загрозливими для серверної частини, які зменшують її відмовостійкість, є внутрішні фактори. Серед них найбільш важким по наслідках є вихід з ладу апаратних засобів, а саме накопичувачів.

Накопичувачі сервера ІС є самою тонкою ланкою системи, відмова якої може спричинити не тільки тривалу недоступність до інформаційних ресурсів, але і безповоротну втрату даних. Оскільки втрата БД є неприйнятною, то необхідно прийняти заходи для нейтралізації загрози раптової втрати накопичувача, що вирішується резервуванням накопичувачів. Замість окремого накопичувача застосовується RAID масив накопичувачів типу 1 [87, 66, 124].

Окрім того, проводиться періодичне діагностування накопичувачів, для завчасного виявлення проблем накопичувача. Для цього слугує ряд утиліт, що входить в офіційні репозитарії більшості дистрибутивів ОС. Звіт про кожне діагностування вноситься в log-файл.

Ще одним заходом недопущення втрати БД є організація резервного копіювання. Незважаючи на описані вище прийняті заходи забезпечення відмовостійкості серверної частини ІС, вони все ж не можуть претендувати на абсолютність. Для недопущення неконтрольованих змін даних у БД, які можуть порушити цілісність даних ІС, всі зміни виконуються під управлінням транзакцій. Такий підхід, гарантовано забезпечує перехід БД з одного погодженого стану в інший, при маніпулюванні даними.

Отже, процес забезпечення відмовостійкості є неперервним на протязі всього життєвого циклу ІС [76]. Він розпочинається з планування заходів забезпечення відмовостійкості ІС, що проєктується і триває до часу завершення її функціонування взагалі.

Загалом, задача забезпечення відмовостійкості серверної частини ІС вирішувалась, як і для клієнтської частини, як комплекс заходів з протидії негативним факторам. Він включає [95]:

- включення до контуру підсистеми живлення інтелектуального ПБЖ, що взаємодіє із ОС сервера, забезпечуючи автоматичне коректне закриття всіх серверних застосунків, не допускаючи аварії БД, раптового зникнення живлення та виключення флуктацій напруги живлення;

- використання RAID масиву накопичувачів типу 1, що з високою ймовірністю, виключає втрату БД ІС через відмову накопичувача;

- автоматичне діагностування стану накопичувачів за встановленим графіком, що дозволяє оперативно виявляти причини майбутніх відмов;

- організацію роботи підсистеми резервування БД в автоматичному режимі, згідно графіка, із територіальним рознесенням основної БД та її копій, по власному мережевому каналу;

- використанням підсистеми транзакцій ПЗ АРМ, який гарантує, що будь-які маніпуляції з даними в БД, виконуються з узгодженими даними в будь який момент роботи ІС.

Таким чином, негативні впливи ЗПЗ та комп'ютерних атак можуть відбуватись щодо фунціонування спеціалізованих ІТ, як в серверній так і в клієнтській їх частинах.Тому, такі впливи мають бути враховані при проєктуванні спеціалізованих ІТ.

1.4. Аналіз відомих методів забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих ІТ

Важливим методологічним підходом при побудові ІС є вибір її архітектури, від чого, в свою чергу, залежать можливості забезпечення відмовостійкості, живучості та захищеності в рамках застосованої ІТ [127-129, 131]. На сьогодні найбільше

застосування знайшли архітектури, характерною особливістю для яких є інваріантна розподіленість їх компонентів, що уже само по собі забезпечує підвищену живучість ІС в цілому. При цьому дублювання операцій з даними в різних їх компонентах, характерне для розподілених архітектур, дозволяє досягти найвищих показників відмовостійкості та живучості з одночасним забезпеченням захисту інформації з прийнятними рівнем ефективності.

В [3] автори відмічають, що організаційна інтеграція ІС супроводжується підвищеними ризиками вторгнення та втрати даних. Пропонують шляхи пом'якшення цих проявів, включивши заходи живучості до складу ІС. При цьому живучість розглядається як здатність системи своєчасно виконувати свою місію за наявності атак, збоїв або аварій. Автори акцентують на питаннях живучості як на безперервності процесу, розуміючи, що ніякі одноразові заходи безпеки не зможуть гарантувати ІС від проникнення та компрометації. Наголошується, що забезпечення живучості, відмовостійкості має бути зосереджено на мережевих ІС, де традиційно заходи безпеки є недостатніми.

В [35, 43] розглянуто проблему забезпечення та підвищення надійності багатофункційної інформаційної системи, внаслідок виникнення загроз втрати або спотворення інформації, що обробляється у системі. Проведено аналіз існуючих підходів та методів забезпечення та підвищення надійності, складовою частиною якої є показники відмовостійкості та живучості комплексів програмних та технічних засобів. Обґрунтована необхідність використання методів підвищення надійності, основним з яких, автор вважає метод застосування структурної надмірності, а всі інші методи підвищення надійності можуть використовуватись як додаткові до основного.

В [114] відмічається, що методи підвищення живучості можуть бути активними та пасивними відносно зовнішніх шкідливих впливів, що прикладаються до системи. При активному методі загрози виявляються засобами контролю, локалізуються діагностуванням і усуваються автоматичною реконфігурацією системи. Пасивні методи засновані на функціональному резервуванні, при якому одні і ті ж елементи

можуть брати на себе функції тих, що вибули, можливо із погіршенням продуктивності.

Пропонуються регулярна перебудова метаданих та індексів; дублювання даних; організація резервного копіювання різними способами; децентралізація зберігання даних; хешірування даних, для захисту від спотворення; розподіл прав доступу; шифрування трафіка.

В [101] запропонована базова структура системи забезпечення живучості корпоративної ІС. Живучість тлумачиться як властивість, що характеризує можливість системи ефективно функціонувати при отримані пошкоджень або відновлювати її на протязі заданого відрізку часу.

Робиться висновок, що через складність задачі забезпечення живучості ІС вирішити її разовими заходами неможливо. Необхідна неперервна система заходів, контроль стану ІС у режимі реального часу.

В роботах [58, 74, 77] розглядаються методи високої відмовостійкості програмних компонентів та методи її досягнення, таких як відмовостійка кластеризація; переміщення журналів (log shipping) – технологія, яка полягає в автоматизації резервного копіювання БД та її відновлення на іншому сервері.

В роботах [31, 93, 20] розглянута нетривіальна наукова задача самовідновлення та самоорганізації функцій кібернетичних систем, яка є фундаментальною в забезпеченні відмовостійкості та живучості ІС. Приділена увага теорії маскування та надлишковості компонент яка узагальнена W.H. Pierce у концепцію стійкості до відмов (the concept of failure tolerance) або відмовостійкості. Пропонується застосування нейромереж в діагностиці несправностей.

В роботах [90, 76] аналізуються два підходи до забезпечення відмовостійкості – архітектурний та алгоритмічний. Архітектурні, будучи алгоритмічно незалежними, можуть базуватись на ручних методах реконфігурації масиву елементів системи, відновлюючи у такий спосіб працездатність системи. Інший підхід полягає у маскуванні відмов шляхом негайного відновлення працездатності системи, при якому

виникнення відмови не помічається. Цей спосіб ефективний, але потребує триразової апаратурної надмірності. Алгоритмічні підходи використовують властивості алгебри для цифрової обробки даних. Прикладом є забезпечення відмовостійкості згідно надлишкового кодування даних, що дозволяє відновити правильний результат при несправному елементі системи.

В роботах [40, 34, 38] запропонована методика для порівняльної оцінки інформаційних мереж, щодо їх стійкості до відмов. Для досягнення технічного результату враховуються динаміка впливів на вузли мережі випадкових і навмисних перешкод, і, навіть можливості відновлення зв'язку між транзитними вузлами. Для цього обчислюють значення показників доступності вузлів інформаційних мереж, час досягнення критичного співвідношення ”небезпечних” та ”безпечних” вузлів для кожного варіанта підключення абонентів.

В [91, 97] пропонується спосіб згідно декомпозиційного підходу отримання моделі проєктування технічних систем, які дозволяють об'єднати окремі показники надійності і безпеки у функцію живучості. Запропонована інформаційна технологія конструювання моделей дозволяє раціонально упорядковувати проєктні варіанти технічних систем. Розроблені імітаційні моделі функцій живучості, що дозволяють будувати ці функції як за статистичними даними, так і за даними імітаційного моделювання та використовувати методи теорії ймовірності і нечіткості в проєктному аналізі. Показано, що існуючу множину показників живучості можна подати як інтервали на шкалі вартості наслідків відмов.

В роботах [14, 27] відмічається, що для швидкого вирішення проблеми підвищення відмовостійкості ІС необхідно враховувати досвід, накопичений природою в процесі еволюційного розвитку об'єктів живої природи. Відмічається, що надійність та відмовостійкість біологічних систем набагато вище, ніж будь-якої технічної системи. Робиться висновок що, в забезпеченні відмовостійкості комп'ютерних систем важливе місце займають контроль правильності їх функціонування (виконується в точках біфуркації протікаючих в системі процесів) та

достатній рівень надлишковості і мережевої взаємодії на всіх рівнях складових системи. При цьому відмічається, що найбільш ефективним методом досягнення високого рівня відмовостійкості комп'ютерних систем є використання всього спектру видів надлишковості закладених на елементно-технологічному, організаційному, інформаційному та алгоритмічному базисах.

В роботах [12, 75, 78] розглядаються методи забезпечення надійності інформаційно-автоматизованих систем на основі експертних оцінок. Розробка нової інформаційно-автоматизованої системи, зазвичай, супроводжується труднощами в ризиках. Процес виявлення та пом'якшення ризиків є одним із важливих напрямків розвитку програмної системи. Методи в розглянутих роботах ґрунтуються на оцінці та пом'якшенні ризиків при їх застосуванні для підвищення надійності та відмовостійкості інформаційно-автоматизованих систем.

В роботах [21, 16, 42, 92, 70] приводиться вичерпний огляд методів відмовостійкості для високопродуктивних обчислень. Акцент робиться на аналітичних моделях ефективності. Приводиться огляд методів загального призначення, включаючи кілька протоколів відновлення контрольних точок і відкату. Розглядається застосування алгоритму Брукса-Айенгара при аналізі джерел помилок і несправностей у великих системах; розглядається набір методів, які можна застосувати для розробки відмовостійкого програмного забезпечення при побудові FTCS систем, а саме: метод прогнозування, який передбачає наявність механізму, який попереджає користувача про майбутні несправності в системі; метод реплікації, який полягає в дублюванні всіх обчислень. Загалом, автори пропонують для забезпечення відмовостійкості ПЗ введення інформаційної надлишковості у дані та її підтримку під час обчислень.

В роботах [82, 71, 73] представлено огляд питань безпеки та конфіденційності в мережах Інтернету, розглядаються методи пов’язані з безпекою і захистом даних при цифровому переході до сучасних систем із виявленням несанкціонованого доступу до них, приводяться схеми шифрування від примітивних до використання гомоморфізму

для забезпечення конфіденційності даних, що відкриває шлях широкого використання хмарних технологій.

В результаті проведеного аналізу відомих методів забезпечення відмовостійкості, живучості та захищеності інформації в ІТ встановлено, що на теперішній час відомі методи для забезпечення належної роботи ІС в умовах впливів зловмисного ПЗ та комп'ютерних атак [119, 127-130] мають недоліки та не забезпечують належними чином стійкість до впливів.

1.5. Постановка задачі дослідження

Для вирішення задачі синтезу спеціалізованих ІТ стійких щодо впливів ЗПЗ та комп'ютерних атак на об'єкти комп'ютерних систем необхідно здійснити розробку методів забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих ІТ і розв'язати такі наукові задачі:

1. Провести аналіз методів забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих ІТ, типів ЗПЗ та комп'ютерних атак і їх потенційно можливих впливів на апаратно-програмні засоби комп'ютерних систем.

2. Розробити абстрактну модель впливів ЗПЗ та комп'ютерних атак на об'єкти комп'ютерних систем для формалізованого представлення їх в якості процесів, що протікають в комп'ютерних системах і впливають на їх працездатність.

3. Розробити метод забезпечення відмовостійкості спеціалізованих ІТ в умовах впливів ЗПЗ та комп'ютерних атак на об'єкти та процеси, що приймають участь у відновленні працездатності ІС та апаратно-програмних засобів після збоїв, які викликані внутрішніми нерегламентованими діями.

4. Розробити метод забезпечення живучості спеціалізованих ІТ в умовах впливів ЗПЗ та комп'ютерних атак на об'єкти комп'ютерних систем та процеси, які використовують механізми забезпечення живучості для відновлення працездатності

ІС та апаратно-програмних засобів після збоїв, які викликані зовнішніми нерегламентованими діями та впливами ЗПЗ і комп'ютерними атаками.

5. Розробити метод забезпечення захисту інформації спеціалізованих ІТ в умовах впливів ЗПЗ та комп'ютерних атак на об'єкти комп'ютерних систем та процеси, які використовують механізми забезпечення збереження інформації в процесі одночасної її обробки та впливів.

6. Розробити метод забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих ІТ в умовах впливів ЗПЗ та комп'ютерних атак на об'єкти комп'ютерних систем та процеси, в якому поєднати впливи та стани забезпечення стійкості до впливів.

7. Розробити ІС з підсистемами забезпечення відмовостійкості, живучості та захисту інформації, провести з нею експериментальні дослідження щодо встановлення покращення її характеристик при впливах ЗПЗ і комп'ютерних атак та впровадити її у виробництво.

1.6. Висновки до першого розділу

Дослідження методів забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих ІТ, показало наступні результати:

1. Впливи зловмисного ПЗ та комп'ютерних атак на функціонування спеціалізованих ІС є суттєвими, навіть в умовах застосування засобів протидії та їх виявлення.

2. Аналіз зловмисного ПЗ та комп'ютерних атак підтверджує їх різноманітність і як наслідок, складність в забезпеченні їм протидії.

3. Методи виявлення зловмисного ПЗ та комп'ютерних атак та протидії їм не забезпечують високу надійність захисту від них ІС, які мають спеціалізоване призначення для вирішення задач користувача і не містять спеціальних методів протидії та виявлення зловмисного ПЗ та комп'ютерних атак.

4. Аналіз методів забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих ІТ показав недостатню їх орієнтацію на врахування руйнуючих впливів різних типів ЗПЗ та комп'ютерних атак, їх потенційно можливі впливи на апаратно-програмні засоби комп'ютерних систем зокрема.

5. Перспективним напрямом згідно проведеного дослідження предметної області є проєктування спеціалізованих ІТ із імплементованими в них методами забезпечення відмовостійкості, живучості та захисту інформації в умовах руйнуючих впливів різних типів ЗПЗ та комп'ютерних атак. Такі методи додатково до методів виявлення та протидії ЗПЗ та комп'ютерних атак, які імплементовані в антивірусні засоби різних типів, можуть покращити функціонування ІС.

Основні результати розділу опубліковані у працях [74-77, 119, 127-133].

РОЗДІЛ 2.
АБСТРАКТНА МОДЕЛЬ ВПЛИВІВ ЗЛОВМИСНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА МЕТОД ЗАБЕЗПЕЧЕННЯ ВІДМОВОСТІЙКОСТІ СПЕЦІАЛІЗОВАНИХ ІТ

2.1. Абстрактна модель впливів зловмисного програмного забезпечення на об’єкти комп’ютерних систем

Проведений аналіз розвитку і поширення зловмисного програмного забезпечення та різноманітності в проведенні комп’ютерних атак підтверджує, що проблема протидії зловмисному програмному забезпеченню та комп’ютерним атакам буде залишатись актуальною і, її актуальність, буде тільки зростати по мірі становлення інформаційного суспільства, базованого на використані комп’ютерних інформаційних систем. Вирішення цієї проблеми є безперервним процесом, успішність якого полягає в уникненні загроз, які створює зловмисне ПЗ та комп’ютерні атаки, можливе за умови використання наукових підходів, в основі яких використовуються моделі загроз та методи, розроблені на їх основі. Вони, як правило, є локальними і, не охоплюють весь спектр загроз, які створюють ЗПЗ та комп’ютерні атаки функціонуванню комп’ютерної системи. Крім великого спектру напрямків для зловмисників в комп’ютерних системах, важливим напрямом для них є інформаційні системи. Як правило, вони на сьогодні є переважно розподіленими. Тому, їх проєктування має враховувати особливості функціонування при виконанні поставлених на них задач, які можуть виконуватись при впливах ЗПЗ та комп’ютерних атаках. В зв’язку з цим необхідним науковим завданням є розробка спеціалізованих ІТ, в яких будуть закладені можливості протидії зловмисним проявам, що дасть змогу розробляти на їх основі стійкі до таких впливів інформаційні системи.

Для забезпечення стійкості ІС до впливів ЗПЗ та комп’ютерних атак в процесі їх функціонування, синтезуємо в ІТ сумісно з спеціалізованим функціоналом її призначання, а також складові елементи, призначення яких полягатиме у підтримці

працездатності ІС з виконання спеціалізованого функціоналу для виконання основного завдання в умовах впливів ЗПЗ та комп'ютерних атак. Задамо складові елементи спеціалізованої ІТ M_{IT} так:

$$M_{IT} = \{F_0, F_1, F_2, \dots, F_{N_{IT}}, A_{IT}\}, \quad (2.1)$$

де F_0 – функціонал основного завдання ІТ і обов'язково присутній в M_{IT}; F_i – i – тий складовий елемент в ІТ, що забезпечує додатковий функціонал; $i = 1,2, \dots, N_{IT}$; N_{IT} – кількість додаткових складових в ІТ; A_{IT} – складовий елемент в ІТ, що активізує елементи $F_1, F_2, \dots, F_{N_{IT}}$ в ІТ за настання певних подій чи запитів від елементу F_0 і він не містить додаткового функціоналу для виконання інших дій.

Оскільки сучасні ІС можуть мати різні архітектури, що впливатиме і на проєктування ІТ, а також вони переважно є розподіленими, то представлені в формулі (2.1) її складові елементи вважатимемо такими, що об'єднують відповідно в своїх елементах всі складові, які розміщені в різних комп'ютерних станціях, але мають складову, що відноситься до складової певної типу. Зокрема, при такому представленні матимемо такі співвідношення:

$$M_{IT} = \begin{Bmatrix} F_0 | \, F_0 = \bigcup_{i=1}^{N_{IT}} F_{0,i} \\ F_1 | \, F_1 = \bigcup_{i=1}^{N_{IT}} F_{1,i} \\ \dots \\ F_{N_{IT}} | \, F_{N_{IT}} = \bigcup_{i=1}^{N_{IT}} F_{N_{IT},i} \\ A_{IT} | \, A_{IT} = \bigcup_{i=1}^{N_{IT}} A_{IT,i} \end{Bmatrix}, \quad (2.2)$$

де $F_{0,i}$ – функціонал основного завдання ІТ в i – ій комп'ютерній станції і обов'язково присутній в M_{IT}; $i = 1,2, \dots, N_{ks}$; N_{ks} – кількість комп'ютерних станцій, в яких встановлено компоненти ІС; $F_{j,i}$ - j – тий складовий елемент в ІТ в i – ій комп'ютерній

станції, що забезпечує додатковий функціонал; $j = 1,2,\dots,N_{IT}$; N_{IT}, i – кількість додаткових складових в ІТ; $A_{IT,i}$ – складовий елемент в ІТ в i – ій комп'ютерній станції, що активізує елементи $F_{1,i}, F_{2,i}, \dots, F_{N_{IT},i}$ в ІТ за настання певних подій чи запитів від елементу $F_{0,i}$ і він не містить додаткового функціоналу для виконання інших дій.

Не в усіх компонентах ІС, які розміщені в комп'ютерних станціях, можуть бути розміщені всі складові елементи $F_{j,i}$, де $i = 1,2,\dots,N_{ks}$; N_{ks} – кількість комп'ютерних станцій, в яких встановлено компоненти ІС; $j = 1,2,\dots,N_{IT}$. Також, в різних комп'ютерних станціях можуть бути різні складові елементи, які відносяться до одного і того ж типу. Зокрема, ці складові можуть відрізнятись для сервера від компонентів в комп'ютерній станції. Але більшість складових елементів одного типу в різних компонентах комп'ютерних станцій може бути однаковою. Це, крім спрощення розробки спеціалізованої ІТ, дає змогу синхронізувати, також, засоби підтримки відмовостійкості ІС при впливах ЗПЗ чи комп'ютерних атак за рахунок координації і взаємодії між ними напряму чи із залученням серверної частини ІС.

Це демонструє можливість до масштабування компонентів спеціалізованих ІТ між різними комп'ютерними станціями в мережах та можливість до виконання завдання в межах однієї комп'ютерної станції.

Таким чином, спеціалізована ІТ M_{IT} представлена множиною (2.2) дає можливість врахувати різні архітектури при проєктуванні ІТ та різні наповнення функціоналом окремих її складових елементів, що узагальнює спеціалізовану ІТ через її таку архітектуру для використання в подальших дослідженнях щодо впливу на неї чи її компоненти в комп'ютерних станціях зі сторони ЗПЗ та комп'ютерних атак.

До складових елементів в ІТ синтезуватимемо такі характеристики, які забезпечуватимуть її відмовостійкість, живучість, захист інформації. Ці складові елементи реалізовуватимемо як окремі завершені модулі, але з можливістю активації в умовах сигналізації про впливи та потреби, які вимагатиме функціонал основного

завдання. Тобто при $N_{IT} = 3$, тоді узагальнена структура спеціалізованої ІТ матиме представлення зображене на рис. 2.1.

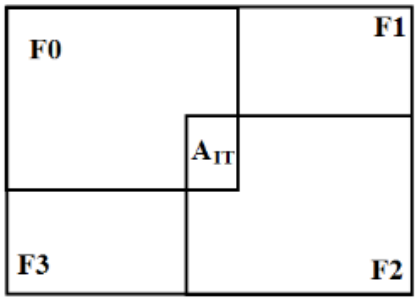


Рис. 2.1 – Узагальнена структура спеціалізованої ІТ з елементами

Розглянемо представлення можливих впливів ЗПЗ та комп'ютерних атак на ІС в комп'ютерних мережах. Дослідження таких впливів важливо на всіх етапах функціонування ІС та в усіх комп'ютерних станціях в цілому та окремо. Впливи ЗПЗ та комп'ютерних атак на комп'ютерні станції в мережі можуть бути спрямовані на їх різні об'єкти, як апаратні, апаратно-програмні так і на програмні. Причому, ці впливи можуть бути багатоетапні та одноетапні, віддалені, безпосередньо спрямовані на об'єкт та опосередковані. Вони можуть реалізовуватись різноманітними засобами, які можуть бути звичайними засобами для роботи в мережі та комп'ютерних системах, а також спеціально створеними зловмисниками засобами. Все це урізноманітнення засобів для проведення зловмисних впливів не тільки ускладнює процес виявлення ЗПЗ та комп'ютерних атак спеціальними антивірусними засобами, але й ускладнює класифікацію саме зловмисних дій. На сьогодні для здійснення точної класифікації зловмисних впливів є невелике ознакове поле характеристик, тому спеціальні антивірусні засоби не забезпечують повного виявлення. Залишається множина ЗПЗ та комп'ютерних атак, які проникають через ці спеціальні антивірусні засоби. Тому, незважаючи на різноманітність ЗПЗ та комп'ютерних атак, досліджуючи саме їх особливості та відображаючи їх у базах зловмисних програм та атак, що здійснюється в реалізаціях спеціальних антивірусних засобів, доцільним є дослідження можливих

їх варіантів зловмисних впливів на конкретні об'єкти комп'ютерних систем. Тобто, побудова можливих впливів саме через формування множини таких впливів щодо конкретних об'єктів в комп'ютерних системах дасть змогу сформувати обмежену множину зловмисних впливів, кожен з елементів якої буде пов'язаний з певним об'єктом комп'ютерної системи, а також, наприклад, з певними елементами ІС, етапами її функціонування, включаючи початок роботи та завершення, зокрема і її різних компонент. В зв'язку з таким співвіднесенням впливів до об'єктів комп'ютерних систем з врахуванням їх часу функціонування, отримуємо зв'язки конкретних об'єктів в часі з можливими впливами на них. Крім того, впливи ЗПЗ та комп'ютерних атак можуть бути руйнуючими і неруйнуючими. Неруйнуючі впливи можуть поділятись на такі, що не досягли своєї мети і, тому процеси, які створені ними функціонують сумісно чи паралельно з процесами створеними заданими користувачем чи комп'ютерною системою і такі, що націлені на інші об'єкти в комп'ютерній системі, тобто на об'єкти від заданої ІС і ресурсів, необхідних для її функціонування. Частина неруйнуючих впливів в певні моменти в майбутньому може перейти до категорії руйнуючих. Руйнуючі впливи можуть бути спрямовані на ІС, в яку будуть імплементовані механізми протидії, або на інші об'єкти комп'ютерної системи, які не пов'язані з ІС та ресурсами для її функціонування. Крім того, такі впливи можуть досягати як часткової мети в певний момент часу, так і в подальшому результуючої мети з виведення з ладу комп'ютерної станції, вузла мережі, призупинки або знищення процесів, знищення інформації на жорсткому диску тощо.

Задамо впливи ЗПЗ та комп'ютерних атак множиною M_{VP} так:

$$M_{VP} = M_{VP,r} \cup M_{VP,nr}, \tag{2.3}$$

де $M_{VP,r}$ – множина руйнуючих впливів; $M_{VP,nr}$ – множина неруйнуючих впливів.

Віднесення впливів до підмножин множини M_{VP} залежить від поточного моменту часу і може змінюватись. Задамо підмножини впливів переліком їх елементів так:

$$M_{VP,r} = \{m_{VP,r,1}, \dots, m_{VP,r,n_{VP,r}}\}, M_{VP,nr} = \{m_{VP,nr,1}, \dots, m_{VP,nr,n_{VP,nr}}\}, \quad (2.4)$$

де $m_{VP,r,i}$ – елемент множини $M_{VP,r}$, який означає i -тий руйнуючий вплив в певний момент часу; $i = 1,2, \dots, n_{VP,r}$; $n_{VP,r}$ – загальна кількість руйнуючих впливів; $m_{VP,r,j}$ – елемент множини $M_{VP,nr}$, який означає j -тий неруйнуючий вплив в певний момент часу; $j = 1,2, \dots, n_{VP,nr}$; $n_{VP,nr}$ – загальна кількість неруйнуючих впливів.

Частина неруйнуючих впливів множини $M_{VP,nr}$ в процесі свого здійснення може не зашкодити об’єктам комп’ютерної станції. Це може відбутись через задану в них змістовність функціоналів так і через недосконаліть функціоналів в певному середовищі комп’ютерної станції. Інша частина неруйнуючих впливів в певний момент часу може перейти до руйнуючих. Такий розгляд впливів в динаміці є необхідним для побудови моделі впливів у співвіднесенні з об’єктами комп’ютерної станції, які динамічно змінюються.

Спрямування впливів ЗПЗ та комп’ютерних атак можу бути здійснене на ІС, для якої проєктуються механізми забезпечення стійкості при впливах, яка задана множиною M_{IT}, та ресурси, які забезпечують її функціонування. Також, спрямування впливів може бути здійснене на об’єкти комп’ютерної станції, які не пов’язані з ІС і вплив на них не впливатиме на функціонування ІС. Тому, розглядатимемо, як можливі варіанти, два типи таких впливів. Оскільки впливи динамічно можуть змінюватись з неруйнуючих в руйнуючі, то задамо множину M_{VP} переліком її елементів так:

$$M_{VP} = \{m_{VP,1}, \dots, m_{VP,n_{VP}}\}, \quad (2.5)$$

де $m_{VP,i}$ – елемент множини M_{VP}, який означає i -тий вплив в певний момент часу; $i = 1,2,\dots,n_{VP}$; n_{VP} – загальна кількість впливів.

Результатом впливів ЗПЗ та комп'ютерних атак на об'єкти комп'ютерних систем будуть наслідки, множину яких задамо так:

$$M_r = \{m_{r,1},\dots,m_{r,n_r}\}, \tag{2.6}$$

де $m_{r,i}$ – елемент множини M_r, який означає i -тий наслідок впливу в певний момент часу; $i = 1,2,\dots,n_r$; n_r – загальна кількість наслідків впливів.

Якщо впливи ЗПЗ та комп'ютерні атаки пов'язати з об'єктами комп'ютерних систем, на які вони спрямовані, і результатом таких взаємодій будуть наслідки, то ці наслідки задамо так:

$$M_r = \begin{pmatrix} m_{r,1,1} & \cdots & m_{r,1,N_{VP}} \\ \vdots & \ddots & \vdots \\ m_{r,N_{IT},1} & \cdots & m_{r,N_{IT},N_{VP}} \end{pmatrix}, \tag{2.7}$$

де $m_{r,i,j}$ – елемент множини наслідків впливів на об'єкти комп'ютерних систем; $i = 1,2,\dots,N_{VP}$; $j = 1,2,\dots,N_{IT}$.

Введемо для множини об'єктів комп'ютерної системи та впливів ЗПЗ і комп'ютерних атак алгебраїчну структуру так:

$$\Omega = \langle\, \Omega_{ks},\ \ \Omega_{VP},\ \ \Omega_{RVP}\rangle, \tag{2.8}$$

де Ω_{ks} – множина об'єктів комп'ютерної системи, на які можуть бути здійснені впливи ЗПЗ та комп'ютерних атак; Ω_{VP} – множина функцій, які реалізують впливи

ЗПЗ та комп'ютерних атак; Ω_{RVP} – множина предикатів заданих на множині Ω_{ks}, які відображають успішність/неуспішність при реалізації функцій з множини Ω_{VP}; $\alpha = 1,\ \beta = 1$ – арності операцій, тому тип системи $\tau = (1,\ 1)$.

За елементи множини Ω_{ks} об'єктів комп'ютерної системи розглядатимемо всі об'єкти файлової системи, завантажувального сектору диску, оперативної пам'яті, мережні пакети, які можуть бути об'єктами впливів ЗПЗ та комп'ютерних атак. Елементами множини Ω_{VP} є одиничні елементи, які містять єдиний функціонал, реалізація якого надає змогу здійснювати зловмисний вплив ЗПЗ та комп'ютерних атак на конкретний єдиний об'єкт комп'ютерної системи та їх комбінації. Для досягнення результату щодо впливу одиничний елемент з множини Ω_{VP} може залучати деякі з інших об'єктів комп'ютерної системи, тобто здійснювати опосередкований вплив, але вплив спрямований винятково на один об'єкт. Тоді, комбінація таких елементів формуватиме решту елементів цієї множини Ω_{VP}. Такі елементи множини Ω_{VP} є породжуючими для решти різних елементів цієї множини. Функції з множини Ω_{VP} успішно реалізовуватимуть впливи не завжди, тому задамо впливи ЗПЗ та комп'ютерних атак множиною предикатів Ω_{RVP}. Вона відображатиме результат успішного / неуспішного впливу ЗПЗ та комп'ютерних атак на об'єкти комп'ютерних систем. Предикати, які належать множині Ω_{RVP} визначимо так, що вони будуть істинними, якщо результат здійснення зловмисних впливів ЗПЗ та комп'ютерних атак на об'єкт чи об'єкти комп'ютерної системи буде успішним, тобто функція з множини Ω_{VP} виконається. Інакше, результат предикату буде хибним.

Тоді, перейдемо з формули (2.8) до абстрактно моделі, яку задамо так [133]:

$$\Re = \langle \Omega_{ks},\ \Omega_{RVP} \rangle, \tag{2.9}$$

де Ω_{ks} – множина об'єктів комп'ютерної системи, на які можуть бути здійснені впливи ЗПЗ та комп'ютерних атак; Ω_{RVP} – множина предикатів заданих на множині

Ω_{ks}, які відображають успішність / неуспішність при реалізації функцій з множини Ω_{VP}; $\alpha = 1,\ \beta = 1$ – арності операцій, тому тип системи $\tau = (1,\ 1)$.

Якщо впливи ЗПЗ та комп'ютерних атак будуть успішними, тоді вони матимуть наслідки, тобто відноситимуться до множини M_r, яку задано за формулою (2.6). В результаті функція відображення елементів множини впливів M_{VP} в множину наслідків M_r:

$$\Omega_{RVP}\colon M_{VP} \xrightarrow{\Omega_{VP}} M_{RVP}. \tag{2.10}$$

Результатом такого представлення є абстрактна модель і множина функцій, які надають можливість представити процеси, які здійснюються в комп'ютерних системах при функціонування ІС та можливих впливів ЗПЗ і комп'ютерних атак на об'єкти комп'ютерних систем. Вона поєднує такі складові, як об'єкти комп'ютерних систем, зокрема і компоненти та елементи ІС, впливи на об'єкти та наслідки впливів.

Таким чином, отримана абстрактна модель [133] надає змогу деталізувати об'єкти для впливів і можливі наслідки, стає основою для розробки методів, які забезпечуватимуть відмовостійкість, живучість ІС та захист інформації в ІС від таких впливів. Абстрактна модель є основою для створення спеціалізованої ІТ, стійке фукнціонування якої можливе в умовах впливів ЗПЗ та комп'ютерних атак. Також, ця модель може включати особливість, яка полягатиме в розподіленні об'єктів комп'ютерних систем в комп'ютерній мережі та компонентів спеціалізованої ІТ.

2.2. Модель впливів та метод забезпечення відмовостійкості спеціалізованих ІТ

2.2.1. Модель впливів ЗПЗ та комп’ютерних атак на відмовостійкість спеціалізованих ІТ

Розглянемо забезпечення відмовостійкості та її реалізацію в спеціалізованих ІТ. Позначимо відмовостійкість як F_1 (рис. 2.1) в структурі спеціалізованої ІТ. Відмовостійкість спеціалізованої ІТ згідно [76] полягатиме у забезпеченні властивості системи зберігати повну або часткову працездатність у випадках відмов окремих її елементів, що не пов'язані із зовнішніми нерегламентованими діями. Основним об’єктом в такому випадку виступає компонент, а також елемент (наприклад, зокрема і запущений процес), ІС, але це стосується не зовнішніх нерегламентованих дій. Бо тоді, така постановка задачі щодо моделі впливів і методу забезпечення відмовостійкості не відноситься саме до відмовостійкості ІТ. В процесі функціонування компоненти та елементи ІС, які проєктуються в ІТ, можуть мати різні стадії виконання, і з них потребуватимуть переходу з одних станів в інші, і саме можливий вплив ЗПЗ та комп’ютерних атак на них можуть активувати механізми, що забезпечуватимуть відмовостійкість. Якщо в множині засобів забезпечення відмовостійкості ІТ відсутні такі, що враховують можливі впливи ЗПЗ та комп’ютерних атак, то в результаті виконання регламентних заходів із забезпечення відмовостійкості може бути не виконано, що порушить функціонування відповідного компоненту ІС. Таким чином, потреба у залученні механізмів забезпечення відмовостійкості спеціалізованих ІТ викликана відмовами окремих її елементів, що не пов'язані із зовнішніми нерегламентованими діями, зокрема і з впливами ЗПЗ та комп’ютерних атак, але при виконанні функціоналу підсистеми, що реалізує забезпечення відмовостійкості, можуть бути впливи ЗПЗ та комп’ютерних атак на об’єкти комп’ютерних систем, що унеможливлюватиме ефективне і результативне виконання відновлення компоненти ІС з використанням засобів, що реалізують

відмовостійкість. Тому, при реалізації відмовостійкості в спеціалізованій ІТ повинні бути механізми, що забезпечують її виконання в умовах впливів ЗПЗ та комп'ютерних атак.

Розглянемо конструктивно компонент спеціалізованої ІТ, наприклад, як автоматизоване робоче місце (АРМ). Враховуючи, що спеціалізована ІТ буде розглядатись як система з розподіленими компонентами, то АРМ тоді виступає як компонент такої розподіленої ІС. Також, це надає можливість масштабувати результати та компоненти ІТ в локальній комп'ютерній мережі. Тоді, в спеціалізованій ІТ, якщо вона враховуватиме проєктування з можливістю функціонування в розподіленому середовищі, виділимо серверну частину і клієнтську частину. Клієнтську частину розглядатимемо як сукупність компонентів АРМ, що встановлені в комп'ютерних станціях. В зв'язку з таким представленням компонент АРМ в спеціалізованій ІТ розглянемо їх можливе наповнення щодо вирішення завдань із забезпечення відмовостійкості, живучості та захисту інформації.

АРМ представимо, як блок, в якому є такі елементи: функціонал для виконання завдань ІС; функціонал для відмовостійкості; функціонал для живучості; функціонал для захисту інформації.

Розглянемо події, які викликають відмови окремих елементів спеціалізованої ІТ, що не пов'язані із зовнішніми нерегламентованими діями, тобто такі, які відноситимуться до відмовостійкості.

Для серверної частини спеціалізованої ІТ такі події можуть виникнути через повну або часткову відмову апаратних компонентів, збоїв у живленні та вичерпуванні вмісту пам'яті. Розглянемо ці події детальніше з метою дослідження можливості впливів на них ЗПЗ та комп'ютерних атак в процесі самовідновлення механізмами відмовостійкості, які закладені в спеціалізовану ІТ.

Відмова апаратних компонентів (пам'яті, дискових накопичувачів, тощо) може спричинити складні проблеми: від сповільнення виконання запитів користувачів (зменшення продуктивності роботи ІС) до повного припинення роботи. При цьому

можливими наслідками можуть бути: спотворення інформації; втрата інформації; недоступність інформації.

Деградація апаратних компонентів теж впливає на серверну частину ІС. Компонент ІС не одномоментно втрачає роботоздатність (відмова), а поступово деградує. Це найбільш характерно для оперативної та постійної пам'яті, поверхонь дискових накопичувачів. При деградації оперативної пам'яті можливе спотворення інформації в базі даних ІС. Часткові несправності постійної пам'яті можуть призвести до програмних збоїв на командному рівні, і наслідком стане зависання процесора сервера. Деградація поверхонь дискових накопичувачів з часом призведе до втрати інформації, що обробляється в ІС.

Збої живлення апаратної частини сервера може викликати руйнування баз даних на сервері ІС, що в свою чергу, призведе до втрати інформації.

Вичерпання пам’яті сервера (витоки пам'яті), внаслідок помилок, що присутні в програмному забезпечені сервера, може призвести до загального зменшення продуктивності роботи всієї ІС в цілому.

Для клієнтської частини спеціалізованої ІТ такими подіями, що викликані збоями в роботі, але не пов’язаними з зовнішніми нерегламентованими діями та впливами, можуть бути такі: відмова апаратних компонентів (пам'яті, дискових накопичувачів, тощо); деградація апаратних компонентів; збої живлення апаратної частини сервера чи комп’ютерної станції; самопошкодження програмного файлу в процесі його виконання.

Відмова апаратних компонентів (пам'яті, дискових накопичувачів, тощо) може проявитись сповільненням роботи АРМ, або повною неможливістю його функціонування, або втратою результатів останньої виконуваної операції.

Деградація апаратних компонентів призводить як правило до частих збоїв в роботі комп’ютерних станцій. Деградація поверхонь дискових накопичувачів з часом призведе до сповільнення роботи комп’ютерної станції, та можливої втрати локальної інформації, що не критично для ІС в цілому.

Збої живлення апаратної частини сервера може викликати збої в роботі комп'ютерної станції, що в свою чергу, призведе до втрати результатів останньої виконуваної операції.

Самопошкодження програмного файлу в процесі його виконання спричинить збій в роботі ПЗ АРМ ІС. В більшості випадків наслідками будуть втрачені результати останньої операції з наступною недоступністю функції ІС.

Виникнення таких подій пов'язуватимемо з одночасними проявами в комп'ютерній мережі чи станціях впливів ЗПЗ та комп'ютерних атак і розглянемо можливі механізми забезпечення відмовостійкості з урахуванням такого поточного стану.

До переліку впливів ЗПЗ та комп'ютерних атак віднесемо такі.

1. Контроль над КС. Наприклад, КС стала вузлом бот-мережі.
2. Атака та її різні етапи.
3. Виконання деструктивних дій ЗПЗ.
4. Виконання розмноження ЗПЗ.
5. Комбінації 1-4, що посилюють вплив на комп'ютерні станції.

Розділимо виконання завдань ІС на етапи: старт, виконання завдань, запуск підсистем, запуск засобів відмовостійкості, запуск засобів живучості, запуск засобів захисту інформації, коректне завершення роботи, некоректне завершення роботи. Це надає можливість представити засоби, які виконують кожен з цих етапів, як об'єкт на який можуть бути спрямовані впливи ЗПЗ та комп'ютерних атак. При цьому засоби забезпечення відмовостійкості можуть повторювати всі розглядувані етапи.

Накладаємо на ці стани функціонування ІС впливи ЗПЗ та комп'ютерних атак і аналізуємо, які з них можливі. Тобто, беремо множину впливів ЗПЗ та комп'ютерних атак M_{VP} і множину наслідків M_{RVP} згідно формули (2.10) і встановлюємо між їх елементами зв'язки, тобто формуємо множину предикатів Ω_{RVP} з врахуванням особливостей множини функцій Ω_{VP}, які реалізують впливи. В результаті такого поєднання формуємо матрицю спряження M_r за формулою (2.7).

Розглянемо множину впливів ЗПЗ та комп'ютерних атак M_{VP}, як таку що містить, наприклад, п'ять елементів:

1) $m_{VP,1}$ – над комп'ютерною станцією встановлено зовнішній сторонній контроль;
2) $m_{VP,2}$ – здійснення різних етапів атаки;
3) $m_{VP,3}$ – виконання деструктивних дій ЗПЗ;
4) $m_{VP,4}$ – здійснення розмноження ЗПЗ;
5) $m_{VP,5}$ – комбінації елементів з підмножини $\{m_{VP,1}, \dots, m_{VP,4}\}$.

Наслідками можливих впливів на об'єкти комп'ютерних систем і, зокрема, компоненти та елементи ІС в них можуть бути такі:

1) зниження продуктивності;
2) взаємоблокування при змаганні процесів за ресурси (пам'ять, реакція на події, тощо);
3) блокування доступу до ресурсів (принтери, мережі, тощо);
4) блокування запуску;
5) інфікування програмних файлів АРМ;
6) зменшення часу реакції на події;
7) майбутня загроза, яка призводить до подій представлених в 1 та 3;
8) захоплення чужих прав доступу.

Тому, множину наслідків M_r впливів ЗПЗ та комп'ютерних атак визначимо, як таку, що містить, наприклад, вісім елементів:

1) $m_{r,1}$ – зниження продуктивності операцій в комп'ютерній системі;
2) $m_{r,2}$ – здійснення стану взаємоблокування при змаганні процесів за ресурси;
3) $m_{r,3}$ – здійснення блокування доступу до ресурсів;
4) $m_{r,4}$ – здійснення блокування запуску;
5) $m_{r,5}$ – інфікування програмних файлів АРМ в комп'ютерній станції;
6) $m_{r,6}$ – зменшення часу реакції на події;

7) $m_{r,7}$ – майбутня загроза, яка призводить до подій представлених елементами $m_{r,1}$ і $m_{r,3}$;

8) $m_{r,8}$ – здійснення захоплення чужих прав доступу.

Оскільки важливим в роботі ПЗ клієнтського АРМ є вдале або невдале виконання завдань, реалізованих як множина функцій F1-Fn під керуванням транзакцій n – n+i, як зображено на рис. 2.2, то з такого представлення роботи АРМ та опису елементів множин впливів і наслідків результати їх взаємозв'язків представлено в табл. Г.1 (Додатку Г) саме для комп'ютерних станцій, в яких знаходяться компоненти ІС, відмінні від серверної частини.

Такі елементи множин впливів і наслідків, враховуючи розподілення компонентів ІС, масштабуються на залучені комп'ютерні станції, в яких встановлено ІС. Компоненти ІС, які встановлені в них можуть відрізнятись, але при цьому вони можуть отримувати впливи, викликані одним чи декількома джерел.

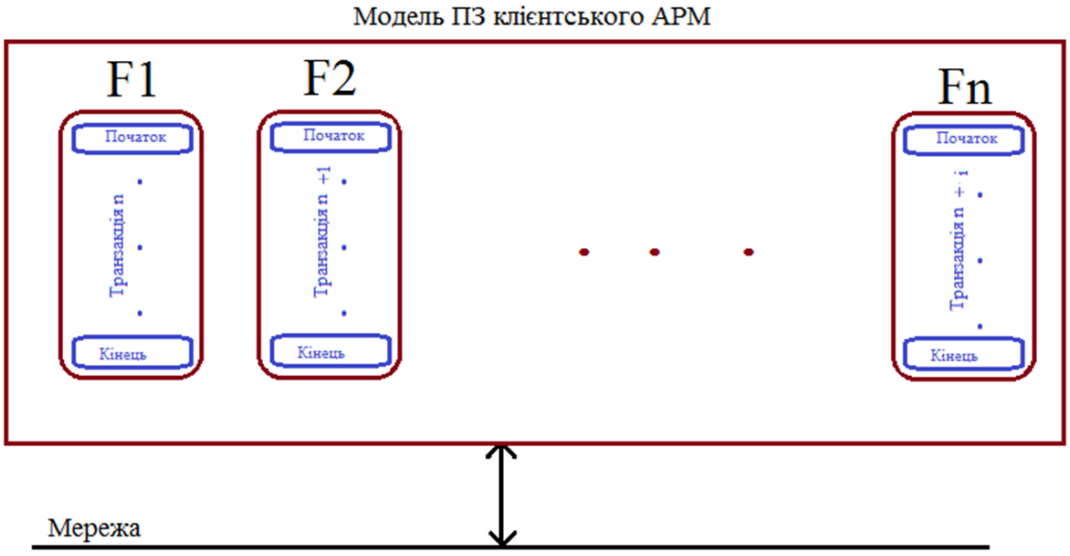


Рис. 2.2 – Відображення транзакцій в структурі ІС

Таким чином, модель впливів ЗПЗ та комп'ютерних атак на відмовостійкість [130, 133] спеціалізованих ІТ міститиме об'єкти комп'ютерних систем, на які спрямовані зловмисні дії, впливи ЗПЗ та комп'ютерних атак, наслідки впливів. Ці

результати деталізовані для застосування в спеціалізованих ІТ і базуються на абстрактній моделі впливів ЗПЗ та комп'ютерних атак.

2.2.2. Моделі проникнення, виживання та деструктивних впливів зловмисного програмного забезпечення і комп'ютерних атак в комп'ютерних системах

Джерелами впливів є ЗПЗ та комп'ютерні атаки. Методи, технології та засоби для здійснення проникнення, виживання та деструктивних впливів зловмисного програмного забезпечення і комп'ютерних атак в комп'ютерних системах постійно урізноманітнюються зловмисниками. Тому, які б вони не були, для розробки спеціалізованої ІТ стійкої до таких впливів, необхідна комплексна оцінка впливів всього інваріантного сімейства ЗПЗ на об'єкти комп'ютерної системи, способів проникнення в них та його подальшого функціонування в інфікованій системі. Систематизація даних щодо здійснення проникнення, виживання та деструктивних впливів зловмисного програмного забезпечення і комп'ютерних атак в комп'ютерних системах цілісно без врахування засобів, які виступають джерелами таких впливів, зображена на рис. 2.3 схемою моделей впливів на об'єкти комп'ютерної системи.

Зображена на рис. 2.3 комп'ютерна система може масштабуватись від окремого комп'ютера до глобальної ком'ютерної мережі.

Така узагальнена модель враховує, що КС керується людиною і ця обставина призводить до прояву людського фактору, який в свою чергу може стати причиною проникнення зловмисного ПЗ в КС. Також, запропонована модель враховує наявність вразливостей як в програмному забезпеченні КС так і в її апаратній платформі, що також може стати причиною враження зловмисним ПЗ КС. За способами проникнення ЗПЗ в КС, воно представлено вірусами, які проникають в КС, використовуючи її вразливості. При цьому шляхом проникнення слугують різні накопичувачі. Черв'яками, які схожі з вірусами, але шляхом їх проникнення в КС є комп'ютерна мережа. Троянське ПЗ може проникати систему різними шляхами. Його особливість

полягає в маскуванні під корисну програму. Ще один клас ЗПЗ – це роботи та мережі на їх основі, для яких шляхом проникнення слугує комп'ютерна мережа.

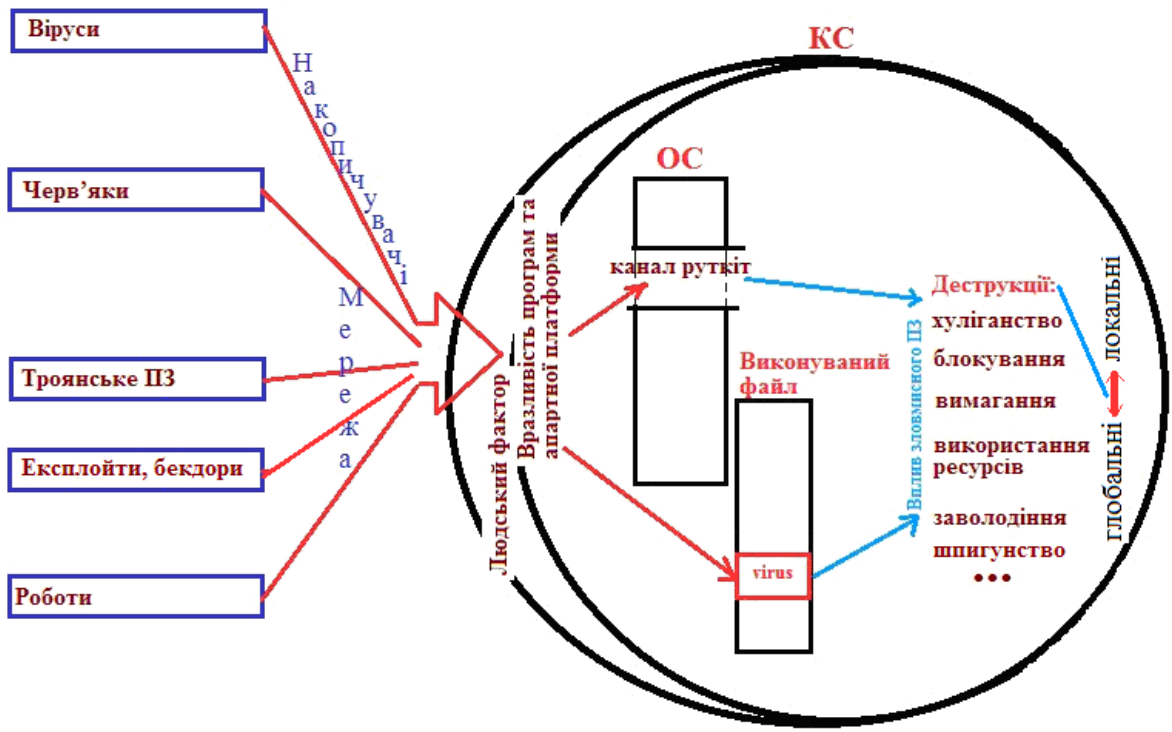


Рис. 2.3 – Узагальнена схема моделей проникнення, виживання та деструктивних впливів зловмисного програмного забезпечення в комп'ютерній системі

Модель враховує способи маскування ЗПЗ в КС. Це і використання руткітів, які створюють неконтрольований «чорний вхід» в КС з метою його подальшого використання зловмисним ПЗ, вживлення вірусу, або його частини в виконуваний файл з метою перехоплення управління КС.

Після проникнення зловмисного ПЗ в КС, воно готове до своїх деструктивних проявів, приведених в моделі, масштаб яких може коливатись від локального прояву в окремій КС до глобального в мережах КС.

Дана модель дає відповідь на питання, які саме загрози зі сторони зловмисного ПЗ є найбільш небезпечними при побудові відмовостійких та живучих спеціалізованих ІТ, що мають бути основою інформаційної системи.

З аналізу впливів зловмисного ПЗ встановлено, що їх перелік для серверної та клієнтської частин ІС, практично однаковий. Бо обидві частини ІС містяться в комп'ютерних системах і, відповідно, узагальнені моделі загроз (рис. 2.3), для кожної із них є однаковими. Проте, якщо аналізувати вірогідність прояву того, чи іншого деструктивного впливу, то вони різні для кожної частини ІТ. Причину, також, можна побачити в представленій моделі (рис. 2.3). Керування комп'ютерною системою, якого б масштабу вона не була, здійснює користувач чи адміністратор. Тобто в обох випадках присутній людський фактор. Але його вага, а відповідно, вірогідність прояву ЗПЗ буде різною. Ця різниця полягає в рівні кваліфікації персоналу, що працює з клієнтською та серверною частинами ІС. Якщо з серверною частиною всі операції з її обслуговування виконують фахівці ІТ-сфери, та ще і, як правило, високої кваліфікації, то з клієнтською частиною, більшу частину часу працює персонал, який не відноситься до ІТ-сфери, а тому забезпечує низький поріг протидії ЗПЗ.

Оскільки з контуру керування комп'ютерною системою, користувача, на сьогодні, прибрати неможливо, то необхідно ввести додаткові засоби, які б підвищили поріг протидії ЗПЗ. Особливо це стосується клієнтських АРМ ІТ, блокування роботи якого зі сторони зловмисного ПЗ веде до недоступності функцій ІС, що є неприпустимим для більшості спеціалізованих ІС.

Як видно з моделі (рис. 2.3), об'єктом атаки ЗПЗ на комп'ютерну систему може бути її ОС та виконувані файли програм, що використовуються в ній. Всі ці об'єкти присутні і в комп'ютерній системі, яка виконує функцію клієнтського АРМ ІС. Характерною особливістю сучасного клієнтського АРМ є необхідність його розміщення не тільки в комп'ютерній мережі, а і підключення до мережі Internet. Ця обставина суттєво збільшує ризик інфікування комп'ютерної системи. До складу її ПЗ входять компоненти клієнтського АРМ ІС, які реалізують функції системи управління

даними. Їх інфікування може привести до втрати узгодженості даних бази даних, та їх достовірності, що є неприпустимо для ІС. Тому, гарантувати повну цілісність компонентів ПЗ клієнтських АРМ ІС неможливо.

Одна із причин – це використання програмних комплексів постійно зростаючої складності. В цьому є як позитивні моменти, так і негативні - чим складніша система, тим більше вразливостей вона містить в собі, її складніше надійно захищати.

Друга причина неможливості гарантування захисту від ЗПЗ полягає тому, що воно, також знаходиться в постійному розвитку, відшуковуючи все нові шляхи проникнення в комп'ютерну систему.

Пріоритетною стратегією протидії ЗПЗ є забезпечення зменшення величини загрози. Саме цьому слугує, представлена дворівнева модель забезпечення відмовостійкості в спеціалізованих ІТ.

Таким чином, розроблені моделі [74, 75, 77] проникнення, виживання та деструктивних впливів зловмисного програмного забезпечення і комп'ютерних атак в комп'ютерних системах відображають особливості ЗПЗ та комп'ютерних атак і потребують врахування при проєктуванні спеціалізованих ІТ, в яких повинно бути забезпечення відмовостійкості, живучості і захист інформації в умовах впливів ЗПЗ та комп'ютерних атак.

2.2.3. Дворівнева модель протидії впливам ЗПЗ

В зображеній на рис. 2.3 узагальненій моделі відображено, що більш високою вразливістю від впливів ЗПЗ страждають клієнтські АРМ ІС. Тому, в пропонованій концепції щодо забезпечення відмовостійкості ІТ розробимо спосіб послаблення залежності від людського фактора. Шляхом вирішення поставленої задачі є використання двох рівнів протидії впливам ЗПЗ, перший із яких, загальносистемний, що вибудовується з використанням загальноприйнятих механізмів протидії, а другий,

локальний, що реалізується в рамках самої спеціалізованої ІТ, з використанням закладених в неї особливостей функціонування та архітектури (рис. 2.4).

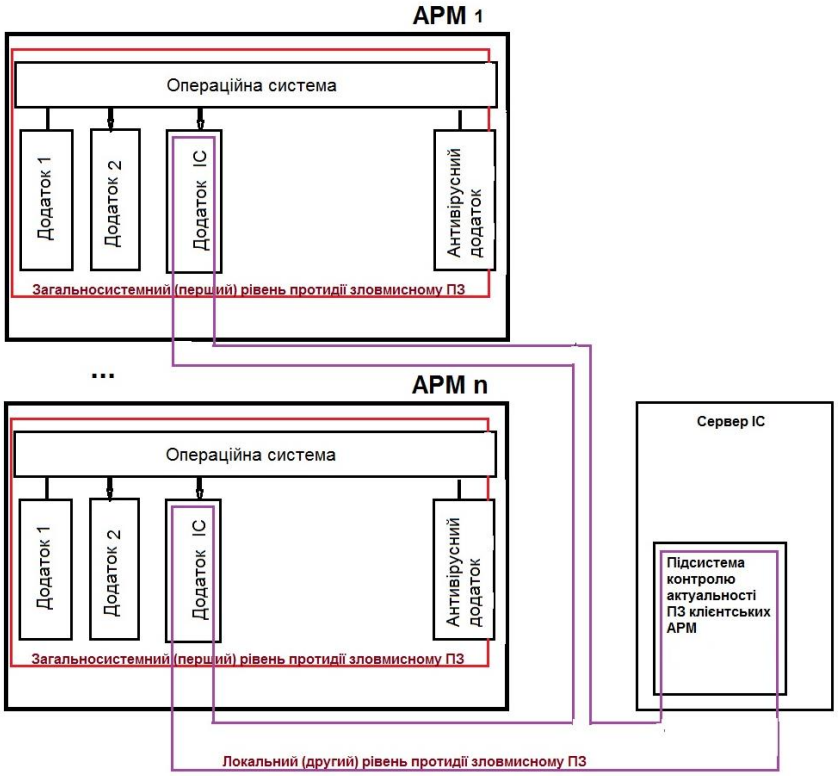


Рис. 2.4 – Дворівнева схема протидії ЗПЗ для клієнтських АРМ спеціалізованої ІС

Такий концептуальний підхід дозволяє збільшити вірогідність нейтралізації цільової атаки ЗПЗ на об’єкти спеціалізованої ІС в комп’ютерних системах, максимально утруднити роботу ЗПЗ з метою підвищення доступності ІС в будь-який момент часу.

При цьому пропонується використати в якості механізму протидії атакам другого рівня, уже існуючу в більшості розвинутих спеціалізованих ІС, службу підтримки актуальності клієнтського ПЗ, наділивши її новими функціональними можливостями.

Задача підтримки актуальності ПЗ клієнтських робочих місць спеціалізованої ІС є досить об’ємною, тому, як правило, природній хід розвитку ІС призводить до переходу на автоматизовану підсистему оновлення актуальності версій ПЗ. В

подальшому будемо її називати службою підтримки актуальності ПЗ клієнтських АРМ. Її задача полягає в виявлені в автоматичному режимі оновлень версій ПЗ клієнтських АРМ ІС з наступним виконанням тиражування змін на всі робочі місця ІС, де вони стались, забезпечуючи роботу клієнтських АРМ з самими новими версіями ПЗ.

Основною складовою цієї служби є банк еталонного ПЗ. В процесі удосконалення ІС, в ПЗ вносяться зміни, що в свою чергу, призводить до заміни еталона ПЗ в банку. Задача служби підтримки полягає в виявлені факту зміни еталона ПЗ і, як реакцію на цю подію, запуск процедури оновлення ПЗ всіх клієнтських робочих місць, де воно використовується.

Розглянемо запропоновану дворівневу модель протидії ЗПЗ (рис. 2.4) і сформулюємо об'єм вирішуваних задач, які будуть покладені саме на її другий рівень протидії. Очевидно, що цей рівень не повинен дублювати задачі виконувані на загальносистемному першому рівні.

Тому, сформулюємо його задачу як локальну, вирішувану тільки в інтересах клієнтського АРМ ІС. І характер вирішуваної задачі буде полягати не в знешкодженні ЗПЗ, а в спробі відновлення працездатності клієнтського ПЗ без встановлення причини її втрати, маючи на меті отримання доступу до функцій ІС, тобто основного її ресурсу.

Процес відновлення працездатності ПЗ клієнтського АРМ, враженого ЗПЗ, або само пошкодженого, що часто буває з складними програмними системами, похожий на процес оновлення версій ПЗ.

Тому, методи забезпечення відмовостійкості та живучості ІС в умовах впливів зловмисного ПЗ схожі за своєю метою і алгоритмами роботи із роботою служби підтримки актуальності ПЗ ІС. Різниця лише в тому, що алгоритми служби підтримки актуальності ПЗ реагують на зміну еталона ПЗ, а алгоритми методів забезпечення відмовостійкості та живучості ІС на втрату відповідності еталону екземпляра ПЗ n-го

клієнтського робочого місця. Реакція ж обох випадках буде однаковою – відновлення відповідності ПЗ клієнтського робочого місця його еталону в банку.

Тому, в основу роботи запропонованої технології забезпечення відмовостійкості та живучості спеціалізованої ІС в умовах впливів зловмисного ПЗ є зміст покласти ідею використання функціональних можливостей уже працюючої служби підтримки актуальності ПЗ клієнтських АРМ. Для цього її алгоритми роботи було удосконалено, шляхом включення до її складу функцій, які забезпечують відмовостійкість та живучість клієнтської частини ІС в умовах дії зловмисного ПЗ.

Процес відновлення пошкодженого, або знищеного клієнтського ПЗ є автоматичним. Задача ця покладена на фоновий процес, який виконується на захищеному комп'ютері (сервер БД, або ПК адміністратора ІС). Він в циклічному режимі із заданою дискретністю виконує перевірку програмних модулів всіх зареєстрованих у ІС АРМ на відповідність еталонним параметрам. Запуск клієнтського ПЗ виконується завантажувачем, якому в якості параметра передається ім'я потрібного модуля. Запуск завантажувача виконується з клієнтського ПК, але сам він знаходиться на тому ж ПК, що і програма фонового процесу. В необхідні моменти часу він взаємодіє із фоновим процесом, вимагаючи позачергового відновлення ПЗ.

2.2.4. Модель забезпечення відмовостійкості та живучості спеціалізованої ІТ в умовах впливів ЗПЗ та комп'ютерних атак з використанням параметричного контролю актуальності модулів ПЗ клієнтських АРМ та їх маскуванням

Для вирішення проблеми відновлення працездатності ІС з метою унеможливлення впливів ЗПЗ, підвищення ступеня гарантоздатності ІС, пропонується забезпечення відмовостійкості та живучості згідно схеми моделі зображеної схемою на рис. 2.5.

В ній відображено реалізацію другого локального рівня забезпечення доступності функцій ІС і, таким чином підвищується відмовостійкість та живучість ПЗ клієнтських робочих місць ІС.

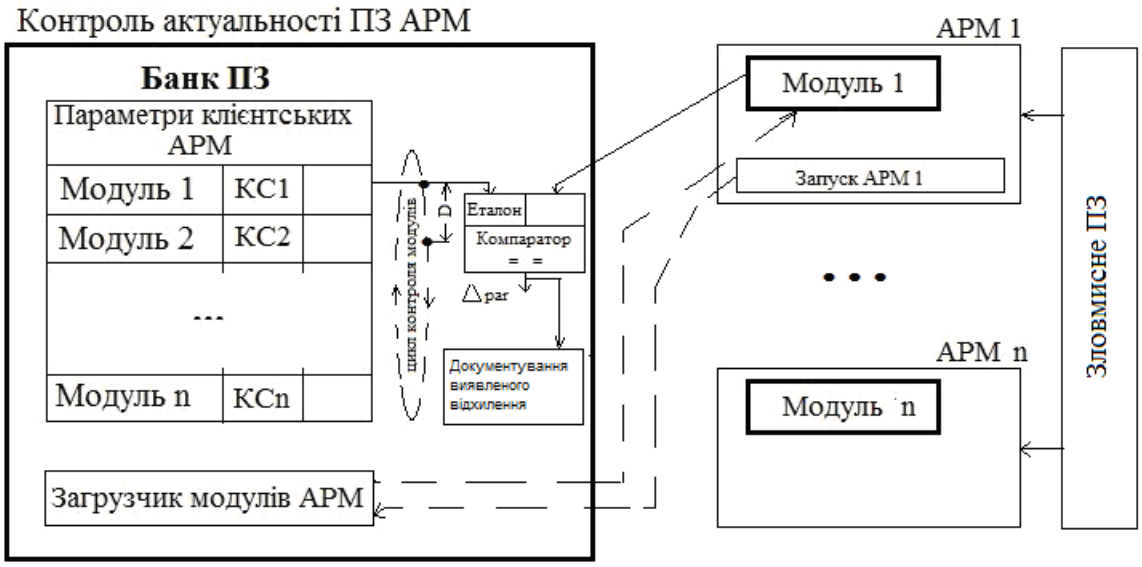


Рис. 2.5 – Схема моделі забезпечення відмовостійкості та живучості ІТ в умовах впливів ЗПЗ з використанням параметричного контролю актуальності модулів ПЗ клієнтських АРМ та їх маскуванням

Суть забезпечення відмовостійкості та живучості ІТ полягає в здійснені постійного циклічного контролю параметрів модулів клієнтських АРМ із заданою дискретністю D. Дискретність є параметром, значення якого вибирається виходячи із рівня загальносистемної продуктивності роботи комп'ютерної мережі та клієнтських ПК, на яких базуються АРМ. Це дозволяє адаптувати дану технологію під апаратні платформи ІС різної продуктивності.

Для забезпечення працездатності такого концептуального підходу його модель включає банк ПЗ, що містить в собі програмні модулі всіх клієнтських АРМ ІС та їх еталоні параметри. В якості таких параметрів слугують контрольні суми КС1 - КСn кодових сторінок, пораховані за заданим правилом, значення маркерів границь

програмних модулів, кількість програмних модулів контрольованого файлу. Набір параметрів контролю може змінюватись, відповідно до структури програмних модулів, що контролюються.

Модель забезпечення відмовостійкості та живучості ІТ заснована на використані маркерів, якими позначаються границі програмних модулів, що включені до таких файлів. Це дозволяє виокремити програмну частину із загальної структури файлу і, таким чином, виконувати розрахунок контрольної суми тільки для сталої частини файлу.

Програмний файл з несталою контрольною сумою необхідно певним чином опрацювати. Для цього в процесі розробки, початок та кінець програмних модулів, що будуть включені до його складу, відмічаються спеціальним маркером, об'єктний код яких відомий антивірусній програмі, яка контролює його цілісність. Одним із способів реалізації таких маркерів в програмному модулі можуть слугувати оператори програмного коду, які присвоюють унікальне, наперед задане значення, програмній змінній. В процесі контролю актуальності стану програмного модуля клієнтського АРМ перевіряється його наявність по заданому шляху, обчислюються його параметри та порівнюються із еталонними. Задачу параметричного контролю актуальності модулів в рамках цієї концептуальної моделі покладається на програмно реалізований компаратор.

У випадку відсутності контрольованого модуля в заданому місці, або отримання розходження між фактичними та еталонними параметрами Δpar на виході компаратора, виконується відновлення ПЗ клієнтського АРМ з використанням еталонного ПЗ, що зберігається в банку. Сам факт виявлення розбіжностей Δpar автоматично документується із збереженням необхідних для подальшого аналізу параметрів в базі даних. У випадку, якщо розходжень між еталоном та модулем, що пройшов контроль актуальності не виявлено, то він, додатково, може маскуватись шляхом перейменування. Це дозволить зменшити вірогідність атаки модуля з боку ЗПЗ, яке, як відомо, в першу чергу вражає виконувані файли.

Такий підхід дозволяє здійснювати контроль актуальності модулів ПЗ клієнтських АРМ в автоматичному режимі, що в свою чергу, дозволяє забезпечувати відмовостійкість та живучість ІТ в умовах впливів зловмисного ПЗ. При цьому характер атаки ЗПЗ на клієнтське ПЗ особливого впливу не матиме. Оскільки на клієнтських ПК дані ІС не зберігаються, то ЗПЗ може лише пошкодити файли програм. Цей факт виявляється в процесі контролю актуальності модулів ПЗ АРМ і вони замінюються еталонними. Таким чином відновлюється доступ до функцій ІС.

2.2.5. Дослідження впливу формату виконуваних файлів на частоту атак ЗПЗ

Операційна система MS Windows є програмною платформою, яка широко використовується для організації роботи на її базі клієнтських робочих місць, практично всіх ІС. З метою пошуків шляхів підвищення ефективності роботи із забезпечення відмовостійкості та живучості спеціалізованих ІС було проведено аналіз частоти атак зловмисного ПЗ в залежності від типу виконуваних файлів в рамках ОС MS Windows. Ця операційна система допускає роботу з достатньо широким спектром виконуваних файлів. Це файли програм COM, EXE-формату та системні драйвери, що мають розширення SYS або BIN. Також, до виконуваних файлів відносяться командні файли (BAT-файли) та файли оверлеїв і динамічно завантажуваних бібліотек, які використовуються програмами по мірі необхідності.

Проведений аналіз (табл. 2.1) показав, що ЗПЗ найчастіше використовує в якості об’єктів своєї атаки файли в COM та EXE - форматах. За ними слідують CMD та BAT - файли та файли драйверів ОС SYS та BIN. А використання інших типів файлів (INF, INS, MSC, MSI, PIF, REG, VBS, MDB, MDE) в деструктивних цілях ЗПЗ є поодиноким.

Зі всіх наведених типів файлів розглядатимемо файл з форматом MDE пакета MS Access. Будучи виконуваним з функціональними можливостями, що дозволяють реалізувати програмну систему будь-якої складності, він має найнижчий рівень

ризику стати об’єктом атаки зловмисного ПЗ. Це означає, що він ігнорується розробниками зловмисного коду.

Таблиця 2.1

Залежність частоти атак ЗПЗ від формату виконуваного файлу

Тип файлу	**Оцінка ризику стати об’єктом атаки зловмисного ПЗ**
EXE	Найвищий
COM	Високий
CMD та BAT	Середній
SYS та BIN	Низький
MDB, MDE	Найнижчий
Інші типи	Не аналізувались

На сьогодні виявлено лише спроби знищення вмісту MDE-файлу деструктивними діями вірусу Blackmal, шляхом вписування рядка "DATA Error". Але знищення вмісту файлу не є інфікуванням файлу і, відповідно, такі деструкції з боку злоякісного ПЗ не загрожують наслідками для даних, а лише потребують заміни спотвореного файлу новим.

Тому, ця обставина (наявність власного формату виконуваного MDE-файлу, мало схильного до інфікування зловмисним ПЗ) поряд з іншими, суттєво вплинула на рекомендацію вибору MS Access в якості інструмента розробки програмного забезпечення клієнтських АРМ спеціалізованих ІС.

MDE-файл є спеціальним форматом бази даних MS Access, і свою чергу є похідним від БД MDB-типу MS Access. Його особливістю є те, що частина компонентів БД, які можуть включати до свого складу виконувані модулі - форми, звіти, модулі, макроси – зберігається в середині MDE-файлу в скомпільованому вигляді, який не допускає внесення будь яких змін в їх вихідний текст, а також їх

перегляд, але при цьому залишається можливість внесення змін у таблиці та запити. Він позиціонується як файл СУБД з розширеними можливостями маніпулювання даними. При цьому дані БД можуть знаходитись у цьому ж файлі, або в іншому MDE файлі, або MDB-файлі. Також можливою є робота з даними, що містяться в будь якій, не MS Access БД, що підтримує технологію ODBC доступу до даних.

MDE-файл є виконуваним файлом в середовищі ОС MS Windows та MAC. Запуск його може бути виконаний програмами MS Microsoft Access або RUN Time Access.

Таким чином, вибір типу файлу [119] є суттєвим при забезпеченні захисту інформації в спеціалізованих ІТ при впливах ЗПЗ.

2.3. Метод забезпечення відмовостійкості спеціалізованих ІТ

При забезпеченні відмовостійкості спеціалізованої ІТ механізмами, які унеможливлюватимуть вплив ЗПЗ та комп'ютерних атак розглядатимемо компоненти ІТ як такі, що поділяються на серверні та клієнтські. Якщо ж серверні частини ІТ відсутні, то результати будуть використані і для клієнтських, які розглядатимуться як такі, що можуть мати частину можливостей серверних частин.

Компоненти спеціалізованої ІТ містять програмну частину та вимагають певних апаратно-програмних засобів для свого функціонування, тому розгляд впливів ЗПЗ та комп'ютерних атак потрібно враховувати до цих двох складових.

Згідно даних з таблиць Д.1 та Д.2 (Додатків) і матриці спряження (формула (2.7)) впливів ЗПЗ та комп'ютерних атак з об'єктами комп'ютерних систем, на які вони спрямовані, і результатом таких взаємодій будуть наслідки. Тоді, необхідно розробити метод забезпечення відмовостійкості ІТ, який би унеможливив успішне виконання відображення згідно формули (2.10), тобто наявність елементів в матриці спряження (2.7), або зменшила б їх кількість чи вірогідність появи. Таким чином, була б забезпечена відмовостійкість ІТ в умовах впливів ЗПЗ та комп'ютерних атак.

З врахуванням необхідності інтеграції механізмів протидії ЗПЗ та комп'ютерним атакам, які можуть бути застосовні однаково до змістовних елементів з матриці спряження (2.7), представимо метод забезпечення відмовостійкості ІТ основними кроками, які відноситимуться як до клієнтської частини, так і до серверної частин.

Розглянемо перший крок методу забезпечення відмовостійкості ІТ, суть якого полягатиме у використанні блокових міток. клієнтської частини ІС при реалізації.

Стосовно прикладного програмного забезпечення, до якого відносяться клієнтські АРМ, то критичні помилки, які можуть проявитись в ході експлуатації робочих місць ІС, фіксуються разом зі своїми параметрами в реєстрі системи в автоматичний спосіб і, в подальшому використовується для аналізу з метою усунення причин, що їх викликали. Це стало можливим завдяки стратегії, яка базована на привнесенні деякої надмірності в програмне забезпечення АРМ ІС, по аналогії із методами забезпечення відмовостійкості апаратної частини ІС.

З цією метою всі розрахункові процедури, які гіпотетично, можуть містити критичні для функціонування АРМ помилки, розробляються із дотриманням певного однотипного шаблону побудови алгоритмів їх виконання. Суть цього першого кроку методу, в подальшому кроку згідно блокових міток, відображена на рис. 2.7.

В структурі етапів першого кроку методу алгоритм виконання будь-якої нетривіальної процедури розділяється на два взаємодіючих блоки. В першому блоці реалізується функція процедури ІС, а в другому обробник помилок. В процесі виконання деякої процедури, яка реалізує одну із функцій АРМ ІС, обидва блоки взаємодіють між собою, передаючи управління обчислювальним процесом один одному, поки виконувана функція не завершиться.

Суть першого кроку методу згідно блокових міток полягає в тому, що алгоритм, який реалізує функцію ІС, розділяється маркерами (мітка 1, ..., мітка n на рис. 2.6) на фрагменти за принципом функціональної завершеності.

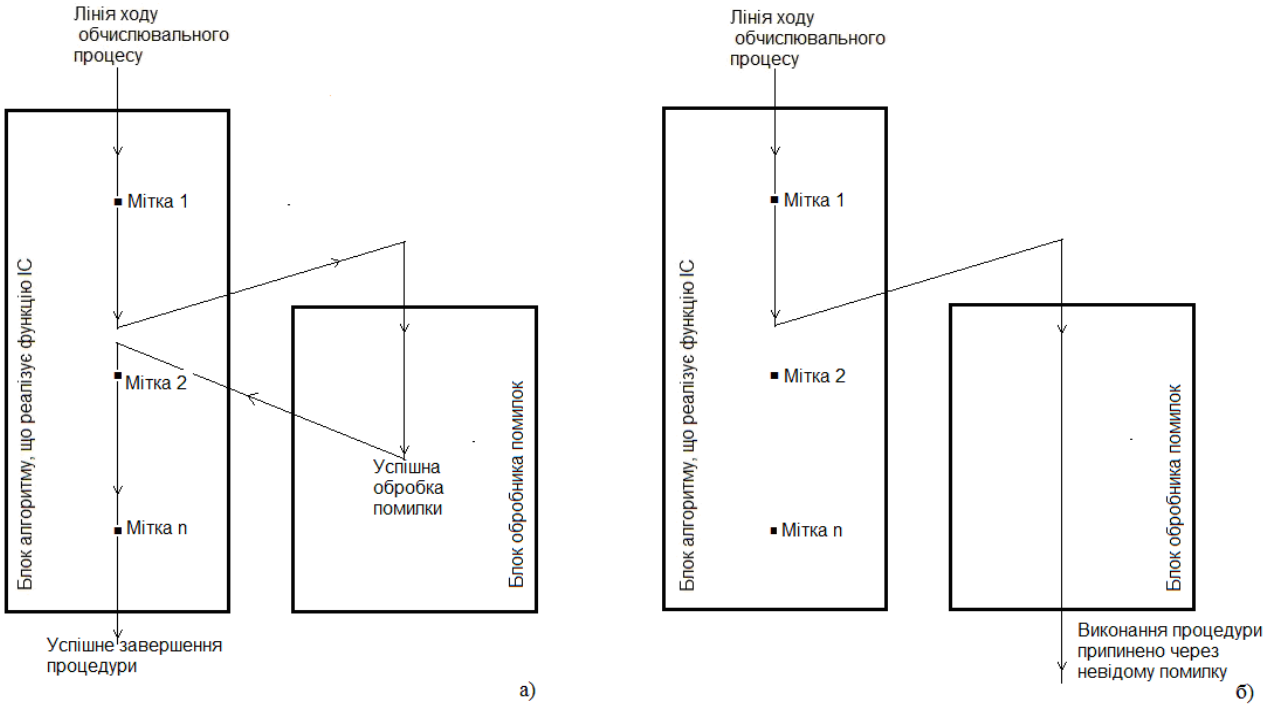


Рис. 2.6 – Етапи першого кроку методу реалізації відмовостійкої процедури: а) для випадку успішного завершення процедури після виникнення помилки; б) для випадку, коли помилка невідома для обробника помилок

Перед початком виконання поточного фрагменту алгоритму в реєстр фатальних помилок заноситься інформація про гіпотетично можливу помилку (код екземпляра АРМ, код функції, № мітки, час і т. і.). В подальшому можливі наступні варіанти розвитку подій:

1. Фрагмент алгоритму функції успішно виконався. В цьому випадку інформація в реєстрі про помилку, що не сталась, знищується, а обчислювальний процес переходить до виконання наступного фрагмента.

2. В процесі виконання фрагмента сталась помилка, але вона успішно локалізована обробником помилок (рис. 2.6 а). В цьому випадку інформація про помилку також може бути видалена з реєстру.

3. В процесі виконання фрагмента сталась помилка, яка не була локалізована обробником помилок (рис. 2.6 б). В цьому випадку інформація про можливу помилку залишиться в реєстрі.

Розглянемо варіант можливої деталізації першого кроку методу. Перший крок методу згідно міток можна представити на прикладі реалізації з більш детальнішим алгоритмом виконання нетривіальної процедури із застосуванням вище згаданого шаблону, що зображено на рис. Д.1 (Додатку Д). Такий підхід до побудови алгоритму виконання функції ІС дозволяє фіксувати в реєстрі помилок, також, і інформацію про помилки, викликані збоями або критичними відмовами в апаратному забезпечені клієнтського комп'ютера, його системного програмного забезпечення.

Зібрана в такий спосіб інформація про фатальні помилки, що стались в ІС, дозволяє їх класифікувати та в процесі подальшого аналізу виявити слабкі ланки в ІС з метою їх усунення, шляхом удосконалення програмного забезпечення АРМ.

Тепер розглянемо внутрішні фактори, що впливають на відмовостійкість клієнтської частини ІС (рис. Д.1 Додатку Д) та методи, які були застосовані з метою його зменшення. Перший з них, по частоті виникнення, це помилки оператора АРМ. Ця проблема вирішена шляхом використання типового редактора даних, в якому всі процедури внесення змін в БД реалізовані з використанням шаблона (рис. Д.2 Додатку Д), структура якого включає надмірності у вигляді блоків алгоритму, які передбачають перевірку дій оператора АРМ. Тут і далі під типовим редактором вважатимемо базовий елемент, який береться за основу при розробці всієї множини редакторів, що застосовуються в ІС.

Як видно з блок-схеми шаблона типового редактора (рис. Д.2 Додатку Д), оператор АРМ не має змоги прямого редагування даних. Всі його дії по маніпулюванню даними знаходяться під контролем процедур попередньої перевірки, які включають в себе набір правил (допустима повнота внесення даних, знаходження значень в заданих діапазонах, відсутність протиріч з раніше внесеними даними і т.д.),

контекстно зв’язаних із виконуваною функцією. Таким чином, фактор помилок оператора АРМ знімається, шляхом ускладнення алгоритму роботи редактора даних.

Таким чином, запропонований перший крок методу згідно блокових міток дозволяє типовим чином вирішувати задачу забезпечення відмовостійкості для всієї множини функцій клієнтської частини ІС.

Другий крок методу забезпечення відмовостійкості ІТ полягає у використанні функціонального резервування. Наступним із значимих внутрішніх факторів, що негативно впливають на відмостійкість є перевантаження апаратної платформи клієнтського ПК задачами, що може різко погіршити часові параметри виконуваних АРМ завдань, або навіть зробити неможливою його роботу, через вичерпання технічних ресурсів. Щоб нейтралізувати дію цього фактора на ІС, при розробці програмного забезпечення, а саме тієї його частини, яка відповідальна за реалізацію "бізнес-логіки" використано функціональне резервування (рис. Д.3 Додатку Д).

Наявність функціонального резерву "важких" розрахункових функцій дозволяє здійснювати маневр обчислювальними потужностями апаратної платформи ІС, в разі перевантаження окремих її ланок, підвищуючи таким чином відмовостійкість ІС.

Оскільки процедура, яка функціонально резервується (наприклад Funk1 на рис. Д.3 Додатку Д) розробляється в двох варіантах за одним і тим же алгоритмом, але в різних програмних середовищах, то для виконання на різних технічних засобах цей факт можна використати для нейтралізації такого негативного фактора, як наявність помилки в прикладному програмному забезпеченні АРМ, у випадку, коли в одному із варіантів процедури проявиться помилка. В цьому проявляється позитивна мультиплікативність ефекту функціонального резервування, що підвищує загальну відмовостійкість ІТ.

Третій крок методу забезпечення відмовостійкості ІТ полягає у перехресному резервуванні. Вирішення задачі, як завжди в таких випадках, полягає в створені деякого резерву. Аналіз роботи АРМ ІС показав, що деякі із них мають резерв часу та надлишковість продуктивності роботи. Тому, природнім було рішення

використовувати цей резерв в критичні моменти в роботі клієнтської частини ІС. В якості резерву тут слугує будь який інший, клієнтський ПК (рис. 2.7), який згідно плану подолання критичної ситуації, може взяти на себе забезпечення роботи АРМ, чий ПК вийшов з ладу. Такий підхід дозволяє не тримати в якості резерву окремий ПК, а також мати запаси комплектуючих, що зменшує експлуатаційні витрати, без втрати показників відмовостійкості системи в цілому.

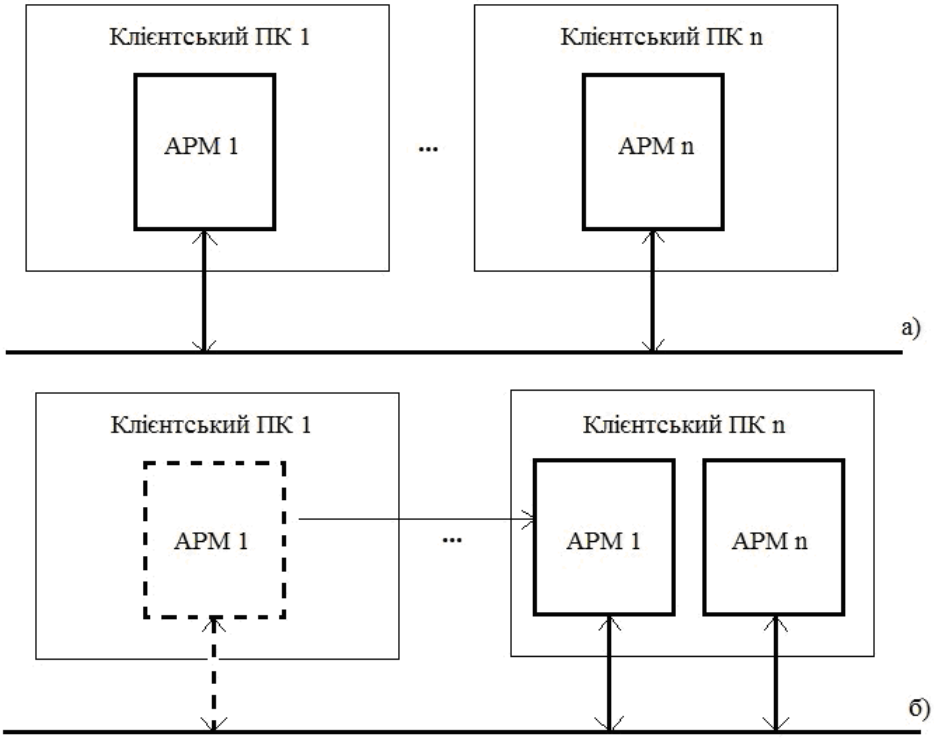


Рис. 2.7 – Приклад структурного взаєморезервування клієнтської частини ІС: а) ІС до виходу з ладу клієнтського ПК1; б) ІС після реконфігурації системи в результаті виходу із ладу ПК1.

Як правило, модулі програмного забезпечення, в налаштованому вигляді, зберігаються в репозитарії програмного забезпечення ІС та на тих клієнтських ПК, де вони плануються бути використаними в критичні моменти згідно плану резервування. У випадку виходу з ладу критичного обладнання комп'ютера, яке зробить неможливим виконання АРМ своїх функцій, воно переноситься на підходящий інший

комп’ютер. Витрати часу на реконфігурацію клієнтської частини обчислюються хвилинами, що є прийнятною величиною для забезпечення живучості ІС, які виконують інформаційне забезпечення, наприклад, в такій предметній прикладній області, як фінансово-господарська діяльність ЗВО.

Така реконфігурація клієнтської частини стала можливою завдяки тому, що на клієнтських комп’ютерах, на яких виконується програмне забезпечення АРМ не зберігаються абсолютно жодні дані. При цьому, сам програмний модуль АРМ, для зручності, скомпонований в один файл і не потребує процедури інсталяції. Її достатньо скопіювати на інший комп’ютер. Після чого вона буде готовою до роботи. Такий підхід, дозволяє навіть після виходу із ладу кількох ПК, що само по собі має низьку вірогідність, зберегти повну функціональність ІС.

Є лише одне обмеження – кожен екземпляр програмного забезпечення АРМ попередньо повинен бути зареєстрований в ІС. Інакше, спроба запуску такої програми буде розглядатись як спроба несанкціонованого доступу до системи, навіть при правильних реєстраційних даних користувача. Контроль ІС за всіма екземплярами своїх АРМ дозволяє блокувати спроби зловмисників, яким вдалось оволоділи даними аккаунта користувача, отримати доступ до системи.

При цьому програма якою оволодів зловмисник, не отримує доступу до даних ІС, а сам факт спроби такої програми підключитись до системи фіксується в реєстрі фатальних помилок з відповідними даними, що дозволяє з їх використанням вжити організаційних заходів проти зловмисника.

Четвертий крок методу забезпечення відмовостійкості ІТ орієнтований на застосування в серверній частині і, тому, в клієнтській частині переважно не буде застосовний, крім випадків поєднання задач і особливостей обох частин ІС.

Неможливо реалізувати спеціалізовану ІТ, що представляє собою ІС з БД з достатньо високими параметрами відмовостійкості, якщо вона не буде опиратись на реалізацію своєї серверної частини з достатнім рівнем резервування. З рис. 2.8 видно, як пропонується вирішення задачі підвищення відмовостійкості ІС, шляхом

структурного резервування основних її компонентів, а саме її серверної частини. У випадку виходу з ладу основного сервера ІС, його функції може взяти на себе резервний, який має абсолютно однакові налаштування з основним.

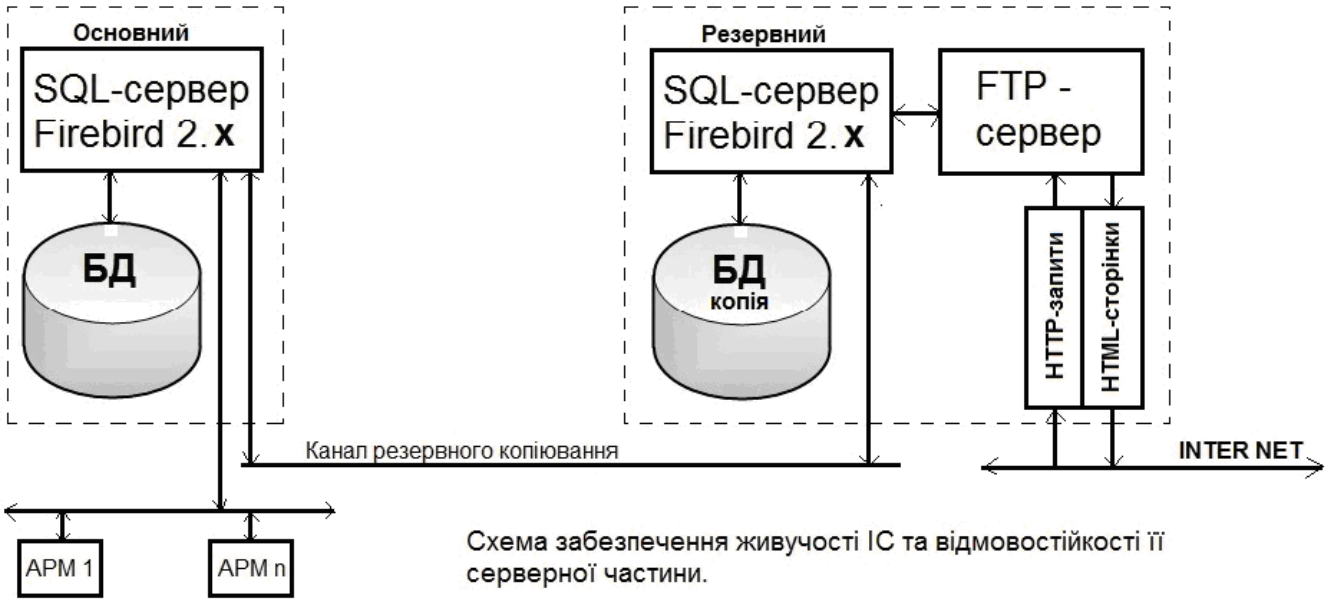


Рис. 2.8 – Схема структурного резервування серверної частини ІС

При цьому основний та резервний сервери мають бути рознесені територіально і повинні живитись з різних ліній. Оскільки вихід з ладу зразу двох серверів є подія маловірогідна, то тим самим забезпечується висока відмовостійкість серверної частини АІС. Реконфігурація реальної системи, незважаючи на ручний режим перемикання, виконується за прийнятний відрізок часу для ІС, яка працює в ірреальному часі. Оскільки дзеркальна копія БД підтримується в актуальному стані службою реплікацій, то перемикання основної бази даних на БД - копію виконується за звичай без втрат інформації. Але незначна втрата інформації при такій схемі все ж можлива. Це може трапитись при відмові деяких чутливих компонентів апаратної платформи сервера. Як правило, це останні запущені транзакції, виконання яких буде припинене через відмову обладнання. І якщо це транзакції на зміну інформації в базі даних, то в цьому випадку інформація буде втрачена. Але оскільки така подія в

життєвому циклі ІС сама по собі рідкісна, то такою можливою кількістю втрати інформації можна знехтувати. Після відновлення роботи серверної частини, операторам АРМ, чиї транзакції були втрачені, потрібно повторно виконати останні операції, для відновлення втраченої інформації.

Значно зменшити вірогідність втрати інформації можна, якщо робота серверної частини ІС буде знаходитись під постійним контролем. Для цього організується регулярне діагностування критичного обладнання сервера. Такий підхід дозволить виявляти назріваючу відмову і вчасно замінювати відповідний компонент, ще до виходу його із ладу. Наприклад, це може стосуватись дискових накопичувачів, якість дискових поверхонь котрих є надзвичайно критичними для функціонування всієї ІС. Така організація роботи дозволяє зменшити вірогідність виходу з ладу серверної частини ІС і тим самим призвести до збереження інформації.

Згідно з рис. 2.8 резервний сервер, окрім виконання функції резервування основного сервера, слугує джерелом даних для WEB-сервера, через який ІС видає інформацію для своїх віддалених користувачів. Такий крок методу згідно резервування серверної частини гарантує достатню високий рівень відмовостійкості в цілому.

Розроблений метод передбачає можливість самостійної перебудови ІС в процесі функціонування із залученням при цьому апаратно-програмних засобів. В процесі перебудови ІС виконання заданих функцій продовжується. Таким чином, метод забезпечення відмовостійкості ІТ в умовах впливів ЗПЗ та комп'ютерних атак надає змогу розширити можливості ІС в частині її адаптивності і відповідно автоматичної зміни апаратно-програмної конфігурації. Крім того, в кроках розробленого методу інтегровано два способи забезпечення відмовостійкості ІТ: залучення резервування; залучення надмірностей. Ця інтеграція поєднана з адаптивністю ІС.

Таким чином, розроблено [74 - 76] метод забезпечення відмовостійкості ІТ згідно інтеграції резервування та надмірностей, який надає змогу розширити можливості ІТ в частині її адаптивності та відповідно автоматичної зміни апаратно-програмної

конфігурації, що дозволятиме створювати відмовостійкі ІТ щодо впливів ЗПЗ та комп'ютерних атак.

2.4. Експериментальні дослідження та оцінювання ефективності методу забезпечення відмовостійкості спеціалізованих ІТ

Встановлення можливості застосування методу забезпечення відмовостійкості спеціалізованої ІТ в умовах впливів ЗПЗ та комп'ютерних атак здійснимо проведенням відповідних експериментальних досліджень та оцінюванням його ефективності.

Оцінювання ефективності методу забезпечення відмовостійкості спеціалізованої ІТ в умовах впливів ЗПЗ та комп'ютерних атак здійснимо за критеріями, що відповідатимуть залученим показникам і відповідно функційним можливостям. Зокрема, такими досліджуваними показниками є наступні: надмірності; автоматична зміна апаратно-програмного конфігурування ІС.

Здійснимо оцінювання впливу різних надмірностей на забезпечення відмовостійкості ІТ розробленим методом. Задамо множину надмірностей так:

$$M_{nd} = \{m_{nd,1}, \dots, m_{nd,p}\}, \tag{2.11}$$

де $m_{nd,i}$ – i - та надмірність у спеціалізованій ІТ; p – кількість розглядуваних надмірностей, які можуть бути реалізовані в ІТ.

Вважатимемо, що в структурі спеціалізованої ІТ, враховуючи особливості її застосування в умовах впливів ЗПЗ та комп'ютерних атак, будуть такі надмірності: $m_{nd,1}$ – структурна; $m_{nd,2}$ – часова; $m_{nd,3}$ – інформаційна; $m_{nd,4}$ – функціональна; $m_{nd,5}$ – алгоритмічна; $m_{nd,6}$ – програмна; $m_{nd,7}$ – апаратна; $m_{nd,8}$ – багаторівнева. Задамо їх внесок в спеціалізовану ІТ в залежності від вагових коефіцієнтів:

$$O_{nd} = \alpha_i \cdot m_{nd,i}, \tag{2.12}$$

де α_i – коефіцієнт ваги внеску надмірності в забезпечення відмовостійкості спеціалізованої ІТ; $m_{nd,i}$ – i-та надмірність; $i = 1, \dots, p$; p – кількість надмірностей.

Тоді, унормуємо величину внеску надмірностей O_{nd} в забезпечення відмовостійкості спеціалізованої ІТ для встановлення її взаємозв'язку з впливами ЗПЗ і комп'ютерних атак на об'єкти комп'ютерних систем та наслідками так:

$$Q_{nd} = \frac{\sum_{i=1}^{p} \alpha_i \cdot m_{nd,i}}{\sum_{i=1}^{p} m_{nd,i}}, \tag{2.13}$$

де $\sum_{i=1}^{p} m_{nd,i} = p$, $\sum_{i=1}^{p} \alpha_i = 1$.

З формули (2.13) випливає, що для певних впливів ЗПЗ та комп'ютерних атак можуть застосовуватись не всі надмірності. І це буде відображуватись відповідними величинами коефіцієнтів. Кількість успішно виконаних функцій з множини впливів M_{VP} буде зменшуватись при застосуванні надмірностей і залежатиме від кількості задіяних надмірностей, що виражатиметься величиною їх оцінки застосування Q_{nd}. Тому, множина предикатів Ω_{RVP}, заданих на множині Ω_{VP}, які будуть істинними зменшиться. В зв'язку з цим потрібно оцінити елемент матриці спряження (2.7) в контексті застосування методу, в якому використано надмірності. Для цього кожен елемент матриці спряження розглядатимемо окремо і так що до нього застосовано метод. А, також, випадок коли метод застосовний до декількох елементів матриці спряження одночасно. В цьому випадку необхідно встановити можливість втрати його ефективності. Для випадку застосування кроку методу з використанням надмірностей до одного елементу матриці спряження введемо функцію ефективності і задамо її в залежності від чинників впливу і протидії так:

$$Q_{m_{r,i}} = \frac{1}{Q_{nd}} \sum_{j=1}^{N_{VP}} Q_{m_{VP,j}}, \quad (2.14)$$

де $m_{r,i}$ – елемент множини M_r, який означає i -тий наслідок впливів в певний момент часу; $i = 1,2,\dots,n_r$; n_r – загальна кількість наслідків впливів; $m_{VP,j}$. – елемент множини M_{VP}, який означає i -тий вплив в певний момент часу; $i = 1,2,\dots,N_{VP}$; N_{VP} – загальна кількість впливів; Q_{nd} - унормована величина внеску надмірностей для протидії впливам; $Q_{m_{r,i}}$ – величина, яка відображає наслідок впливів після протидії впливам надмірностей; $Q_{m_{VP,j}}$ – унормована величина впливів, що реалізовані функціями і виражена відповідними їх оцінками порівняно між всіма функціями впливів.

Впливів може бути декілька або один, або всі наявні. Тому, протидія їм засобами відмовостійкості може знижуватись при одночасному здійсненні широкого спектру різних впливів. Це відображено в формулі (2.14). Але всі ці впливи чи один вплив зосереджені на один об’єкт комп’ютерної системи в формулі (2.14). Результатом цієї формули (2.14) буде наслідок впливів відмінний від наслідку, який отримувався б без залучення надмірностей з першого кроку методу забезпечення відмовостійкості.

Якщо об’єктів комп’ютерної системи декілька і на них будуть зосереджені впливи, тоді це теж понижуватиме результат стійкості до впливів, бо засоби забезпечення відмовостійкості будуть додатково витрачати ресурси комп’ютерної системи. Тоді, результат щодо впливів оцінимо так:

$$\sum_{i=1}^{N_r} Q_{m_{r,i}} = \frac{N_r}{Q_{nd}} \sum_{j=1}^{N_{VP}} Q_{m_{VP,j}}. \quad (2.15)$$

Права частина рівності відображає, що загальна оцінка відмовостійкості в цьому випадку відображає зниження можливості зміни наслідків впливів. Таке оцінювання

масштабуємо в межах розподіленої системи і отримуємо результат для сервера та комп'ютерних станцій, в які встановлено компоненти ІС.

Таким чином, отримані формули (2.14), (2.15) дають змогу оцінити вплив надмірностей щодо наслідків впливів для забезпечення відмовостійкості ІТ.

Резервування в спеціалізованій ІТ, яке впливає на забезпечення її відмовостійкості, є частиною заходів з динамічної перебудови системи і може бути оцінене виходячи із часового використання серверних компонент, часу їх використання.

Експериментальні дослідження щодо перевірки ефективності розробленого методу забезпечення відмовостійкості ІТ проводимо в два етапи. Спочатку досліджуємо ІС без імплементованого в неї методу. Після цього на другому етапі досліджуємо ІС з імплементованим в неї розробленим методом. При постановці такого експерименту суттєвим аспектом виступають джерела впливів. Можуть бути варіанти, коли ІС буде працювати тривалий час, щоб за тривалий час з певною ймовірністю можливо було отримати впливи, які призведуть до активізації засобів забезпечення відмовостійкості або якщо їх не імплементовано в ІС, тоді фіксації таких впливів. Але тоді вплив на ІС для експериментів для двох таких випадків не буде однаковим, бо він буде реальним і випадковим. Для проведення потрібна тривалість експерименту протягом дуже тривалого часу, наприклад року. Це пов'язано з тим, що аналіз повідомлень про комп'ютерні атаки в межах України дає статистику масованих атак приблизно три на два роки за останні 6-8 років. Для двох експериментів, можна вирішити питання проведення їх послідовно, тоді потрібно два роки, або паралельно експлуатувати дві однакових ІС, в одній з яких наявні засоби забезпечення відмовостійкості, а в іншій відсутні. Крім того, тривалість експериментів можна зменшити, створивши в закритому середовищі кіберполігон і встановити в ньому штучні джерела впливів.

Результати експериментів ІС записує в свій внутрішній формат, який за потреби після проведених експериментів може бути досліджений. На рис.2.9 зображено

фрагменти з результатів роботи двох ІС. В одній ІС не було імплементовано засобів забезпечення відмовостійкості і, тому, результатом стала статистика виведення з ладу компонентів ІС в процесі функціонування в умовах впливів ЗПЗ та комп'ютерних атак. В другій ІС, в яку було імплементовано засоби забезпечення відмовостійкості, результатом стали час впливів, час залучення засобів забезпечення відмовостійкості і тривалість та результативність. В обох випадках дослідження впливів фіксувались саме щодо необхідності забезпечення подій, які викликані внутрішніми нерегламентованими діями.

Logfile001.txt [vk.com]

NPP	ARM	PVR	KVR	Error	NAMERROR	IP_BD	IP_ARM
200732	108	02.01.2021 16:10:07	02.01.2021 19:41:47			192.168.168.1	192.168.168.15
200733	40	04.01.2021 8:33:11	04.01.2021 9:02:26			192.168.168.1	192.168.168.10
200734	105	04.01.2021 8:35:43	04.01.2021 11:21:23			192.168.168.1	192.168.168.10
200735	51	04.01.2021 8:37:18	04.01.2021 11:21:26			192.168.168.1	192.168.168.10
200736	208	04.01.2021 8:42:08	04.01.2021 16:57:05			192.168.168.1	192.168.168.9
200737	83	04.01.2021 8:44:09	04.01.2021 8:44:46			192.168.168.1	192.168.168.9
200738	89	04.01.2021 8:48:36	04.01.2021 11:21:21			192.168.168.1	192.168.168.10
200739	69	04.01.2021 8:49:59	04.01.2021 13:45:48			192.168.168.1	192.168.168.9
200740	8	04.01.2021 8:55:38	04.01.2021 13:34:37			192.168.168.1	192.168.168.6

Рис.2.9 – Log-файл подій в ІС

Розглянемо постановку проведеного експерименту з перемиканням БД на резервний сервер в ручному режимі. Зображена на рис.2.10 архітектурна схема ІС включає в себе два сервери керування БД – основний та резервний та n її клієнтських АРМ, об'єднаних в єдину систему апаратними засобами комп'ютерної мережі. Робота основного та резервного сервера БД базується на використанні однакового переліку програмного забезпечення з однаковими налаштуваннями. Різниця лише в мережевих адресах, режимах роботи та виконуваних функціях. В звичайному режимі запити з клієнтських АРМ адресуються до основного сервера. Резервний сервер виконує

функцію гарячого резерву. В деяких випадках може бути доцільним його використання як джерело даних для вирішення другорядних задач ІС.

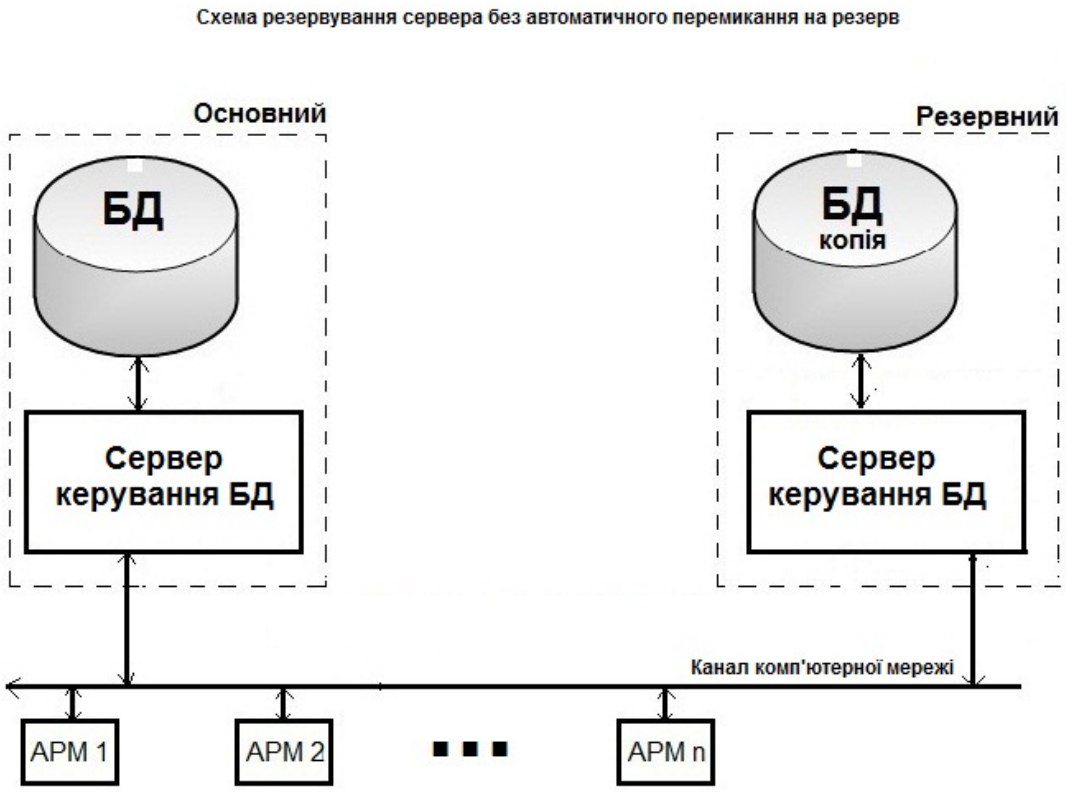


Рис.2.10 – Структурна схема ІС з резервним сервером БД

Розглянемо постановку експерименту з перемиканням БД на резервний сервер в автоматичному режимі. Схема ІС представлена на рис. 2.11.

Метою проведення експериментів є отримання на практиці прямих результатів про час відновлення доступу до функцій ІС в ситуації, коли з певної причини став недоступним основний сервер. Перший експеримент мав надати інформацію про час перемикання на резервний сервер в ручному режимі, а другий – в автоматичному.

Розглянемо експеримент з перемиканням клієнтських АРМ на резервний сервер в ручному режимі. Зображена на рис.2.10 архітектурна схема ІС включає в себе два сервери керування БД – основний та резервний та n її клієнтських АРМ, об’єднаних в єдину систему апаратними засобами комп’ютерної мережі.

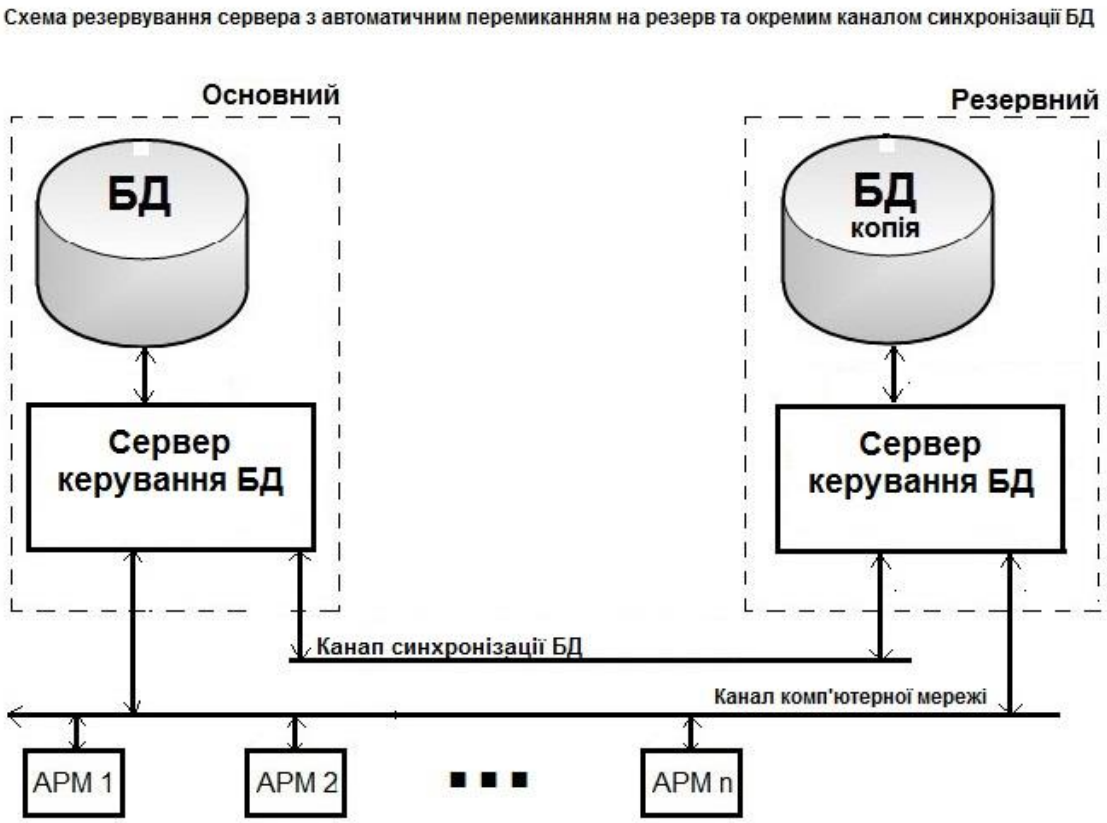


Рис.2.11 – Структурна схема ІС з резервним сервером БД

Робота основного та резервного сервера БД базується на використанні однакового переліку програмного забезпечення з однаковими налаштуваннями. Різниця лише в мережевих адресах, режимах роботи та виконуваних функціях. В звичайному режимі запити з клієнтських АРМ адресуються до основного сервера. Резервний сервер виконує функцію гарячого резерву. В деяких випадках може бути доцільним його використання як джерело даних для вирішення другорядних задач ІС.

Для проведення експерименту залучимо ІС з основним та резервним серверами та клієнтські АРМ №3 та №76. Змоделюємо ситуацію виходу з ладу основного сервера через впливи ЗПЗ або внутрішні причини, наприклад, шляхом відключення його від комп’ютерної мережі. Те, як будуть розвиватись події в ІС, будемо відслідковувати на прикладі АРМ №3.

Результати експеримента зображені фрагментом log-файлу подій в ІС (рис.2.12).

File001.log [vk,com]

210174	3	13.09.2021 8:42:25	13.09.2021 8:49:32			192.168.168.1	192.168.168.5
210176	3	13.09.2021 8:49:43				192.168.168.1	192.168.168.5
210177	76	13.09.2021 8:49:55				192.168.168.1	192.168.168.14
210178	3	13.09.2021 8:53:34		3050	The server is not responding	192.168.168.1	192.168.168.5
210179	76	13.09.2021 8:55:06		3050	The server is not responding	192.168.168.1	192.168.168.14
210181	3	13.09.2021 8:59:39	13.09.2021 8:59:55	3055	unknown server error ... STOP	192.168.168.1	192.168.168.5
210182	76	13.09.2021 9:01:30	13.09.2021 9:01:38	3055	unknown server error ... STOP	192.168.168.1	192.168.168.14
210183	3	13.09.2021 9:33:11			redirect to backup server	192.168.168.2	192.168.168.5
210186	3	13.09.2021 9:41:20				192.168.168.2	192.168.168.5
210187	3	13.09.2021 9:44:28				192.168.168.2	192.168.168.5
210188	76	13.09.2021 9:49:26			redirect to backup server	192.168.168.2	192.168.168.14
210189	76	13.09.2021 9:50:32	13.09.2021 10:02:33			192.168.168.2	192.168.168.14
210190	3	13.09.2021 9:51:15	13.09.2021 9:51:21			192.168.168.2	192.168.168.5
210194	3	13.09.2021 12:05:08				192.168.168.1	192.168.168.5
210195	3	13.09.2021 12:06:25				192.168.168.1	192.168.168.5
210196	76	13.09.2021 12:29:54				192.168.168.1	192.168.168.14
210197	76	13.09.2021 12:31:55	13.09.2021 12:48:30			192.168.168.1	192.168.168.14
210200	53	13.09.2021 12:35:01	13.09.2021 12:37:02			192.168.168.1	192.168.168.13
210201	82	13.09.2021 12:37:42	13.09.2021 12:38:58			192.168.168.1	192.168.168.13

Рис. 2.12 – Log-файл подій в ІС при ручному відновлені її роботи

Ситуації, пов’язані з цим АРМ, які склались в ІС в ході експеримента, будемо відслідковувати починаючи з події 210176. Аналіз подій показаний в табл. 2.2.

Аналізуючи данні табл. 2.2 видно, що між подіями 210178 та 210183 пройшло трохи менше 40 хвилин, за які вдалось відновити параметри відмовостійкості. Але при цьому слід розглянути, якими діями був наповнений цей час. Він в ході експерименту був витрачений на виконання наступних дій:

– блокування стартів клієнтських АРМ;

– перевірка актуальності встановленої копії БД на резервному сервері;

– відновлення останньої актуальної копії БД;

– установка параметра в локальній БД АРМ «Адміністратора ІС», який вказує клієнтським АРМ адресу резервного серверу;

– зняття блокування старту клієнтських АРМ.

Таблиця 2.2

Аналіз ситуацій з АРМ №3 в ході першого експерименту

№ події в ІС	Розшифровка змісту події
210176	Старт АРМ №3 в 8:49:43 з підключенням до сервера БД з IP адресою 192.168.168.1.
210178	В 8:53:34 АРМ №3 запустив деяку процедуру. В ході її виконання виявилась помилка №3050 (Сервер не відповідає). Після трьох невдалих спроб виконати цю процедуру, в локальну БД АРМ Адмістратора було відправлено інформацію про помилку.
210181	В 8:59:39 АРМ №3 завершило свою роботу. При цьому відправлено інформацію про помилку №3053 (Невідома помилка сервера. Робота АРМ зупинена.)
210183	В 9:33:11 АРМ №3 успішно запускається із під'єднанням до БД резервного сервера із IP адресою 192.168.168.2. При цьому оператору АРМ повідомляється, що система працює з підключенням до резервного сервера. Робиться це для того, щоб він міг оцінити актуальність даних.
210190	В 9:51:15 АРМ №3, після успішного виконання деякої процедури, завершило свою роботу.
210194	В 12:05:08 АРМ №3 успішно стартувало. При цьому із запису в log-файлі видно, що на цей момент робота основного сервера відновлена і ІС повернулась до штатного режиму роботи.

Це максимальний час, ручного перемикання ІС на резервний сервер. Його можна значно зменшити, якщо постійно синхронізувати БД резервного сервера із БД

основного сервера, але при цьому слід відмітити, що підтримання такого режиму готовності резервного сервера є досить витратним.

Розглянемо експеримент з перемиканням клієнтських АРМ на резервний сервер в автоматичному режимі.

Для проведення даного експеримента була взята ІС з архітектурою показаною на рис. 2.11. Вона відрізняється від приведеної на рис.2.11 тим, що обидва сервери – основний та резервний з'єднані додатковим каналом передачі даних, утворюючи свою локальну мережу, яка використовується для синхронізації основної БД та її копії, яка керується резервним сервером. Крім того, в даній архітектурі підкреслюється роль АРМ «Адміністратора ІС». Дане АРМ є центральним в організації процесу автоматичного перемикання клієнтських робочих місць з основного сервера на резервний.

Як і в попередньому випадку, основний та резервний сервери мають однакову номенклатуру ПЗ та однакові параметри їх налаштування. Відміна тільки в режимі роботи та виконуваних функціях. Особливістю є більш жорсткі вимоги до синхронізації основної БД та її копії на резервному сервері. Якщо буде втрачено актуальність БД резервного сервера відносно БД основного, то перемикання клієнтських АРМ на нього в автоматичному режимі стане недоцільним через загрозу втрати значної кількості даних.

При проведені експеримента, для даного типу ІС, можна вважати прийнятною несинхронність БД тривалістю не більше 10 хвилин.

Контроль за роботою основного сервера та організацію перемикання клієнтських АРМ на резервний сервер виконує АРМ «Адміністратор ІС», який в своїй роботі опирається на службову БД, доступну в режимі читання всім клієнтським АРМ. Воно запускає фоновий процес, узагальнений алгоритм роботи якого зображено на рис. Д.4 (Додатку Д).

Фоновий процес, зображений на рис. Д.4 (Додатку Д), запускається з дискретністю один раз за хвилину. Після запуску він звертається до основного сервера

(рис. Д.4 Додатку Д, оператор 2). Якщо з певних причин основний сервер не відгукнеться, то процедура звернення до нього повториться ще двічі. Якщо після третього запиту відгуку від сервера не буде, то фоновий процес переходить до перевірки актуальності копії БД резервного сервера (рис. Д.4 Додатку Д, оператор 5). Потрійний запит основного сервера робиться для того, щоб уникнути випадкового переходу ІС на роботу з резервним сервером в силу якихось короткотермінових нестабільностей в роботі сервера або комп'ютерної мережі.

Якщо проблем з актуальністю копії БД (рівнем синхронізації) немає, фоновий процес переходить до процедури перемикання клієнтських АРМ на роботу з резервним сервером (рис. Д.4 Додатку Д, оператор 6). Дана процедура включає виконання наступних кроків:

- блокування доступу клієнтських АРМ до БД резервного сервера;

- зміна значення адресної змінної, яка знаходиться в локальній БД цього АРМ, таким чином, щоб вона вказувала на резервний сервер; її значення зчитується клієнтськими АРМ при їх кожному звернені до ресурсів ІС;

- зняття блокування доступу до копії БД.

Після виконання цієї процедури клієнтські АРМ будуть переадресовані в автоматичному режимі на резервний сервер.

Для проведення експеримента використовується ІС з основним та резервним серверами, АРМ «Адміністратора ІС» та клієнтські АРМ №3 та №76. Змоделюємо ситуацію, коли через впливи ЗПЗ або з якихось інших причин став недоступний основний сервер шляхом його відключення від комп'ютерної мережі. Те як будуть розвиватись події в ІС будемо відслідковувати на прикладі АРМ №3.

Результати експеримента зображені фрагментом log-файлу подій в ІС (рис. 2.13). На рис. 2.13 жовтим кольором відмічені задукоментовані події в ІС пов'язані з виконанням фонового процесу АРМ «Адміністратор ІС» по переключенню клієнтських АРМ на резервний сервер. Червоним – події пов'язані з клієнтським АРМ №3, на якому демонструється процес перемикання на резерв.

file017.log [vk.com]

210301	76	16.09.2021 8:43:26	16.09.2021 8:45:07			192.168.168.1	192.168.168.14
210302	3	16.09.2021 8:45:15	16.09.2021 13:01:39			192.168.168.1	192.168.168.5
210303	3	16.09.2021 8:48:14				192.168.168.1	192.168.168.5
210304	3	16.09.2021 8:52:47		3050	The server is not responding	192.168.168.1	192.168.168.5
210305	50	16.09.2021 8:52:54		3050	server IP 192.168.168.1 is not available	192.168.168.1	192.168.168.10
210306	50	16.09.2021 8:53:56		3050	server IP 192.168.168.1 is not available	192.168.168.1	192.168.168.10
210307	50	16.09.2021 8:54:12		3050	server IP 192.168.168.1 is not available	192.168.168.1	192.168.168.10
210308	50	16.09.2021 8:54:38			blocking connection to SQL server reserve IP 192.168.168.2	192.168.168.2	192.168.168.10
210309	3	16.09.2021 8:55:13		3050	The server is not responding	192.168.168.1	192.168.168.5
210310	50	16.09.2021 8:55:17			change address bar for clients to IP 192.168.168.2	192.168.168.2	192.168.168.10
210311	50	16.09.2021 8:55:25			Unblocking the connection to the SQL server of the reserve IP	192.168.168.2	192.168.168.10
210312	76	16.09.2021 8:55:57	16.09.2021 10:15:52		redirect to backup server	192.168.168.2	192.168.168.14
210313	3	16.09.2021 8:58:06	16.09.2021 10:08:27		redirect to backup server	192.168.168.2	192.168.168.5
210314	76	16.09.2021 10:12:22	16.09.2021 10:25:31			192.168.168.2	192.168.168.14
210315	3	16.09.2021 10:21:32				192.168.168.2	192.168.168.5
210316	76	16.09.2021	16.09.2021			192.168.168.2	192.168.168.14

Рис. 2.13 – Log-файл подій в ІС (автоматичний перехід на резервний сервер)

Процес переключення на резервний сервер клієнтського АРМ №3 та дії АРМ «Адміністратора ІС», що забезпечував його, приведено в табл. 2.3 починаючи з події 210303.

Як видно з наведеного аналізу результатів експерименту, час перемикання ІС на резервний сервер склав близько двох з половиною хвилин, що суттєво скоротило час недоступності функцій ІС порівняно з результатом першого експеримента, де недоступність функцій склала близько сорока хвилин.

Оціночні значення відмовостійкості при імплементації в ІС розробленого методу забезпечення відмовостійкості ІТ щодо наслідків впливів становить $\sum_{i=1}^{N_r} Q_{m_{r,i}} = 0{,}87$, де $m_{r,i}$ – елемент множини M_r, який означає i -тий наслідок впливів в певний момент часу; $i = 1,2,\dots,n_r$; n_r – загальна кількість наслідків впливів; $i = 1,2,\dots,N_{VP}$; N_{VP} – загальна кількість впливів; $Q_{m_{r,i}}$ – величина, яка відображає наслідок впливів після

протидії впливам надмірностей. Результати відображають належний рівень відмовостійкості щодо впливів ЗПЗ та комп'ютерних атак в процесі активізації підсистеми забезпечення відмовостійкості в ІС.

Таблиця 2.3

Аналіз ситуацій з АРМ №3 для експеримента
з автоматичним переключенням на резервний сервер

№ події в ІС	Розшифровка змісту події
210303	Старт АРМ №3 в 8:48:14 з підключенням до сервера БД з IP адресою 192.168.168.1.
210304	В процесі виконання деякої функції АРМ №3, зв'язаної з отриманням інформаційних ресурсів, в 8:52:47 основний сервер не відповів, що і було відмічено в log-файлі ІС.
210305	В 8:52:54 ту ж ситуацію виявив фоновий процес АРМ «Адміністратора ІС» - основний сервер з IP адресою 192.168.168.1 був недоступний при звернені до нього.
210306	В 8:53:56 фоновий процес АРМ «Адміністратора ІС» робить другу спробу зв'язатись з основним сервером і знову невдалу.
210307	В 8:54:12 зафіксована третя невдала спроба фонового процесу АРМ «Адміністратора ІС» зв'язатись з основним сервером.
210308	Після трьох невдалих спроб зв'язатись з основним сервером фоновий процес перейшов до виконання процедури перемикання клієнтських АРМ ІС на резервний сервер. А саме виконано блокування під'єднання до БД резервного сервера на час виконання внутрішніх функцій. Поява в log-файлі цього запису означає, що процедура перевірки актуальності копії БД завершилась без проблем.

Кінець таблиці 2.3

210309	В 8:55:13 АРМ №3 робить другу невдалу спробу зв’язатись з основним сервером.
210310	В 8:55:17 фоновий процес АРМ «Адміністратора ІС» змінює адресну зміну, перенаправляючи запити клієнтських АРМ на резервний сервер з IP 192.168.168.2
210311	В 8:55:25 фоновий процес АРМ «Адміністратора ІС» знімає заборону під’єднання до БД резервного сервера. З цього моменту трафік клієнтських АРМ перенаправлено на резервний сервер.
210313	В 8:58:06 АРМ №3 успішно запускається із під’єднанням до БД резервного сервера із IP адресою 192.168.168.2. При цьому оператору АРМ повідомляється, що система працює з підключенням до резервного сервера. Робиться це для того, щоб він міг оцінити актуальність даних копії БД.
210315	Повторний успішний старт АРМ №3 в 10:21:32 після припинення роботи в 10:08:27 з під’єднанням до резервного сервера.

Результати експериментальних досліджень [74, 75, 133] підтвердили ефективність розробленого методу забезпечення відмовостійкості ІТ ЗПЗ та комп’ютерних атак. Розрахунки оціночних значень для надмірностей та резервування за даними з експерименту над розробленою ІС вказують приблизно на 87 відсотків більше порівняно з ІС, в яку не імплементовано розроблений метод.

2.5. Висновки до другого розділу

Розроблена абстрактна модель впливів ЗПЗ та комп’ютерних атак на об’єкти комп’ютерних систем є основою для розроблюваних методів забезпечення стійкості ІТ. Розроблені моделі проникнення, виживання та деструктивних впливів

зловмисного програмного забезпечення і комп'ютерних атак в комп'ютерних системах відображають особливості ЗПЗ та комп'ютерних атак і потребують врахування при проєктуванні спеціалізованих ІТ, в яких повинно бути забезпечення відмовостійкості, живучості і захист інформації в умовах впливів ЗПЗ та комп'ютерних атак.

Метод забезпечення відмовостійкості ІТ згідно інтеграції резервування та надмірностей надає змогу розширити можливості ІТ в частині її адаптивності та відповідно автоматичної зміни апаратно-програмної конфігурації, що дозволяє створювати відмовостійкі ІТ щодо впливів ЗПЗ та комп'ютерних атак. Особливістю основних кроків розробленого методу є врахування параметричного контролю цілісності програмних файлів та можливість його застосування до групи програмних файлів, які не мають сталої контрольної суми і, цим самим він розширює можливості відомого методу виявлення ЗПЗ, а саме методу контролю цілісності програм,

Застосування розробленого методу здійснюється в системі, яка має засоби для перебудови та використовує надмірності. Для дослідження розробленого методу розроблено методику оцінювання його ефективності в частині надмірностей та резервування. Проведені експериментальні дослідження та оціночні розрахунки підтверджують ефективність розробленого методу забезпечення відмовостійкості ІТ в умовах впливів ЗПЗ та комп'ютерних атак.

Основні результати розділу опубліковані у працях [74 - 76, 119, 129, 130, 133].

РОЗДІЛ 3.
МЕТОДИ ЗАБЕЗПЕЧЕННЯ ЖИВУЧОСТІ ТА ЗАХИСТУ ІНФОРМАЦІЇ В СПЕЦІАЛІЗОВАНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В УМОВАХ ВПЛИВІВ ЗЛОВМИСНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1. Абстрактна модель впливів ЗПЗ та комп'ютерних атак на об'єкти комп'ютерних систем в контексті забезпечення живучості спеціалізованих ІТ

На відміну від відмовостійкості живучість ІТ полягає у здатності залишатися працездатною системою з допустимим зменшенням її продуктивності в умовах негативних зовнішніх впливів (нерегламентованих дій) [74 - 76, 129, 130].

Оскільки впливи ЗПЗ та комп'ютерних атак спричиняються із зовні і, крім того, з зовні можуть бути дії, які не пов'язані із впливами ЗПЗ та комп'ютерних атак. Це можуть бути інші зовнішні впливи. Але ці дві групи впливів будемо розглядати як такі, що обов'язково матимуть вплив ЗПЗ та комп'ютерних атак. Тобто, множину зовнішніх впливів розподілимо на дві підмножини: підмножина впливів безпосередньо ЗПЗ та комп'ютерних атак, яка призводить до необхідності залучення в спеціалізованій ІТ механізмів живучості; підмножина зовнішніх впливів, які не пов'язані першочергово з впливами попередньої підмножини, але наслідки таких впливів в подальшому можуть мати вплив, також, і від впливів ЗПЗ та комп'ютерних атак, і при залученні механізмів забезпечення живучості в спеціалізованій ІТ можуть теж здійснювати такі впливи. Таким чином, якщо розглядати модель впливів в часовому вимірі, то типових класів зовнішніх впливів в контексті забезпечення живучості спеціалізованих ІТ може бути декілька:

1) впливи безпосередньо від ЗПЗ та комп'ютерних атак;

2) впливи початково не пов'язані із ЗПЗ та комп'ютерними атаками, але в подальшому після активізації підсистеми забезпечення живучості поява ще впливів ЗПЗ та комп'ютерних атак;

3) впливи початково не пов’язані із ЗПЗ та комп’ютерними атаками, але в подальшому в процесі активізації підсистеми забезпечення живучості поява ще впливів ЗПЗ та комп’ютерних атак;

4) впливи початково не пов’язані із ЗПЗ та комп’ютерних атак, але в процесі реагування на них для активізації підсистеми забезпечення живучості долучення ще впливів ЗПЗ та комп’ютерних атак;

5) впливи початково пов’язані із ЗПЗ та комп’ютерними атаками та іншими причинами, тобто одночасно різними джерелами.

Описані дії, які відбуватимуться в різні часові проміжки, будуть мати різні результати, тому це повинно бути враховано в абстрактній моделі впливів ЗПЗ та комп’ютерних атак на об’єкти комп’ютерних систем в контексті забезпечення живучості спеціалізованих ІТ. Наприклад, впливи ЗПЗ можуть бути до активізації підсистеми забезпечення живучості ІТ і можуть бути під час її активізації або після активізації. Таким чином, впливи матимуть різні наслідки.

Розглянемо формування матриці спряження (формула (2.7)) у випадку забезпечення живучості ІТ. Множина впливів в цьому випадку буде та ж, що і у випадку із забезпеченням відмовостійкості ІТ, але буде враховано часову модель з п’яти розглянутих періодів для впливів. В якості об’єктів комп’ютерних систем, на які можуть бути здійснені впливи, розглядатимемо такі: центральний процесор; оперативна пам’ять; постійна пам’ять; дискові накопичувачі; засоби відмовостійкості; засоби живучості; засоби захисту інформації; мережні пристрої; периферійні пристрої. Наслідки впливів відрізнятимуться в контексті відмовостійкості і живучості. Крім того, для живучості буде застосована часова модель, що означатиме групу наслідків з врахуванням різних часових періодів та етапів від впливів ЗПЗ та комп’ютерних атак.

Сформуємо множину наслідків впливів ЗПЗ та комп’ютерних атак в контексті живучості ІТ аналогічно як для відмовостійкості. Частина елементів множин наслідків впливів ЗПЗ та комп’ютерних атак для відмовостійкості і живучості будуть збігатись.

Тому, розглядатимемо спочатку їх як підмножини, а після формування об'єднаємо їх в одну множину.

Елементами підмножини наслідків впливів ЗПЗ та комп'ютерних атак на об'єкти комп'ютерних систем в контексті забезпечення живучості ІТ будуть такі: $m_{g,r,1}$ – зниження продуктивності з виконання операцій в комп'ютерній системі; $m_{g,r,2}$ – взаємоблокування при змаганні процесів за ресурси (пам'ять, реакція на події тощо); $m_{g,r,3}$ – блокування доступу до ресурсів (пам'ять, периферія, мережа, ...); $m_{g,r,4}$ – блокування запуску, роботи; $m_{g,r,5}$ – інфікування програмних файлів АРМ; $m_{g,r,6}$ – зменшення часу реакції на події; $m_{g,r,7}$ – неефективне використання ресурсів системи; $m_{g,r,8}$ – прискорення деградації; $m_{g,r,9}$ – тимчасове виведення з ладу. Тоді, підмножина наслідків впливів ЗПЗ та комп'ютерних атак в контексті забезпечення живучості ІТ визначатиметься так:

$$M_{g,r} = \{m_{g,r,1}, m_{g,r,2}, m_{g,r,3}, m_{g,r,4}, m_{g,r,5}, m_{g,r,6}, m_{g,r,7}, m_{g,r,8}, m_{g,r,9}\}. \quad (3.1)$$

Елементів множини $M_{g,r}$ буде більше, ніж задано формулою (3.1). До цієї множини входитимуть, також, наслідки отримані за результатами розгляду моделі впливів в часовому вимірі.

Множину наслідків впливів ЗПЗ та комп'ютерних атак в контексті забезпечення відмовостійкості та живучості ІТ задамо так:

$$M_{vg} = M_r \cup M_{g,r}. \quad (3.2)$$

Задані елементи в множині $M_{g,r}$ згідно формули (3.1) відносяться до клієнтської частини ІС. Для серверної частини ІС характерні такі наслідки впливів, які представимо наступними елементами множини $M_{g,r}$: $m_{g,r,10}$ – підсистема

переривання; $m_{g,r,11}$ – BIOS система; $m_{g,r,12}$ – система живлення зі зворотним зв'язком. Елементи множини впливів можуть бути поділені в залежності від типів та етапів атак, зокрема: мережева розвідка, сканування портів; віддалена відмова (DOS-DDOS-атаки); віддалене проникнення (брутфорс-атака); виконання деструктивних дій ЗПЗ.

Проведений аналіз інформації [96, 98, 105, 125] про ЗПЗ та його впливи на об'єкти комп'ютерних систем, надав змогу систематизувати її до вигляду наведенного в табл. Г.2 (Додатку Г). Згідно систематизованих даних таблиці Д.3 була побудована абстрактна модель впливів ЗПЗ та комп'ютерних атак на об'єкти комп'ютерних систем, показана на рис. 2.3. Ці впливи в частині відношення до об'єктів комп'ютерних систем при забезпеченні живучості спеціалізованих ІТ будуть збіжними з впливами в контексті забезпечення відмовостійкості. Тому, і можливі наслідки згідно матриці спряження (формула (2.7)) для клієнтської частини відображено в табл. Г.3 (Додатку Г), а для серверної частини відображено в табл. Г.4 (Додатку Г). Ці данні в таблицях підтверджують існування наслідків від впливів на об'єкти комп'ютерних систем. Тому, таке представлення впливів може бути використано в абстрактній моделі впливів ЗПЗ та комп'ютерних атак на забезпечення живучості спеціалізованих ІТ.

В моделі, представленій на рис. 2.3 схемою, також, потрібно врахувати, що КС керується людиною і ця обставина призводить до прояву людського фактору, який в свою чергу може стати причиною проникнення ЗПЗ в КС. Також, запропонована модель повинна враховувати наявність вразливостей як в програмному забезпеченні КС, так і в її апаратній платформі, що також може стати причиною інфікування ЗПЗ КС. Крім того, потрібно враховувати способи маскування ЗПЗ в КС. Це і використання руткітів, які створюють неконтрольований «чорний вхід» в КС з метою його подальшого використання ЗПЗ, втілення вірусу, або його частини у виконуваний файл з метою перехоплення управління КС. Після проникнення ЗПЗ в КС, воно готове до своїх деструктивних проявів, наведених в моделі, масштаб яких може коливатись від

локального прояву в окремій КС до глобального в мережах КС. Дана модель дає відповідь на питання, які саме загрози зі сторони ЗПЗ є найбільш небезпечними при побудові живучих спеціалізованих ІТ, що мають бути основою інформаційної системи.

Впливи ЗПЗ, згідно рис. 2.3, за переліком для серверної та клієнтської частин ІС, суттєво збіжні. Причина в тому, що обидві частини ІС є комп'ютерними системами і, відповідно, узагальнені моделі загроз (рис. 2.1.), для кожної із них є однаковими. Проте, якщо аналізувати вірогідність прояву того, чи іншого деструктивного впливу, то вони різні для кожної частини ІТ. Причину також можна побачити в представленій моделі (рис. 2.1.). Як видно з неї, керування комп'ютерною системою, якого б масштабу вона не була, здійснює людина. Тобто в обох випадках присутній людський фактор. Але його вага, а відповідно, вірогідність прояву ЗПЗ буде різною. Ця різниця полягає в рівні кваліфікації персоналу, що працює з клієнтською та серверною частинами ІС. Якщо з серверною частиною всі операції з її обслуговування виконують фахівці ІТ-сфери, та ще і, як правило, високої кваліфікації, то з клієнтською частиною, більшу частину часу працює персонал, який не відноситься до ІТ-сфери, а тому забезпечує низький поріг протидії ЗПЗ. Оскільки з контуру керування комп'ютерною системою, людину, на сьогодні, прибрати неможливо, то необхідно ввести додаткові засоби, які б підвищили поріг протидії ЗПЗ. Особливо це стосується клієнтських АРМ ІС, блокування роботи якого зі сторони ЗПЗ призведе до недоступності функцій ІС, що є неприпустимим для більшості спеціалізованих ІС.

Як видно з моделі (рис. 2.1.), об'єктом атаки зловмисного ПЗ на комп'ютерну систему може бути її ОС та виконувані файли програм, що використовуються в ній. Всі ці об'єкти присутні і в комп'ютерній системі, яка виконує функцію клієнтського АРМ ІС. Характерною особливістю сучасного клієнтського АРМ є необхідність його підключення не тільки до комп'ютерної мережі, а і до мережі Internet. Ця обставина різко збільшує ризик інфікування комп'ютерної системи. До складу її ПЗ входять компоненти клієнтського АРМ ІС, які реалізують функції системи управління даними.

Їх інфікування може привести до втрати узгодженості даних бази даних, та їх достовірності, що є неприпустимо для ІС.

Забезпечити повну цілісність компонентів ПЗ клієнтських АРМ ІС складно. Це пов'язано із використанням програмних комплексів постійно зростаючої складності. В цьому є як позитивні моменти, так і негативні - чим складніша система, тим більше вразливостей вона містить в собі, її складніше надійно захищати. Інша причина полягає в неможливості гарантування захисту від ЗПЗ через те, що воно знаходиться в постійному розвитку, відшуковуючи все нові шляхи проникнення в комп'ютерну систему. Тому, стратегією для протидії ЗПЗ є зменшення величини загрози. Для досягнення цієї мети при забезпеченні відмовостійкості спеціалізованої ІТ була використана дворівнева модель забезпечення відмовостійкості ІТ. Аналогічно і для забезпечення живучості ІТ використаємо дворівневу модель, представлену схемою на рис. 2.2.

Задамо абстрактну модель впливів ЗПЗ та комп'ютерних атак для відображення впливів на об'єкти комп'ютерних систем та процеси в кожній комп'ютерній станції та на сервері так:

$$\Re_i = \langle \Omega_{ks,i},\ \Omega_{RVP,i} \rangle, \tag{3.3}$$

де $\Omega_{ks,i}$ – множина об'єктів комп'ютерної системи, на які можуть бути здійснені впливи ЗПЗ та комп'ютерних атак; $\Omega_{RVP,i}$ – множина предикатів заданих на множині $\Omega_{ks,i}$, які відображають успішність / неуспішність при реалізації функцій з множини Ω_{VP}; $\alpha = 1,\ \beta = 1$ – арності операцій, тому тип системи $\tau = (1,\ 1)$; i – кількість розподілених компонент ІТ.

Якщо впливи ЗПЗ та комп'ютерних атак будуть успішними, тоді вони матимуть наслідки, тобто відноситимуться до множини M_r, яку задано за формулою (2.6). В

результаті функцію відображення елементів множини впливів M_{VP} в множину наслідків M_r з врахуванням часу функціонування системи задамо так:

$$\Omega_{RVP,i}: M_{VP,i} \xrightarrow{\Omega_{VP,i};\, T} M_{RVP,i}, \tag{3.4}$$

де T – час функціонування системи; i – кількість розподілених компонент ІТ.

Результатом такого представлення є абстрактна модель і множина функцій, які надають можливість представити процеси в комп'ютерних станціях в мережі, які здійснюються при функціонування ІС та можливих впливів ЗПЗ і комп'ютерних атак на об'єкти комп'ютерних систем.

Таким чином, отримана абстрактна модель [133] надає змогу деталізувати об'єкти для впливів і можливі наслідки в кожному елементі розподіленої спеціалізованої ІТ, що стає основою для розробки методів [76], які забезпечуватимуть відмовостійкість, живучість та захист інформації від таких впливів. Ця модель включає особливість, яка полягатиме в розподіленні об'єктів комп'ютерних систем в комп'ютерній мережі та компонентів спеціалізованої ІТ.

3.2. Метод забезпечення живучості спеціалізованих ІТ в умовах впливів ЗПЗ та комп'ютерних атак з використанням параметричного контролю актуальності модулів ПЗ клієнтських АРМ та їх маскуванням

Для вирішення проблеми відновлення роботоздатності ІС з метою унеможливлення впливів ЗПЗ, підвищення ступеня гарантоздатності ІС, пропонується забезпечення живучості згідно моделі зображеної схемою на рис. 3.1.

В ній відображено реалізацію другого локального рівня забезпечення доступності функцій ІС і, таким чином підвищується живучість ПЗ клієнтських робочих місць ІС.

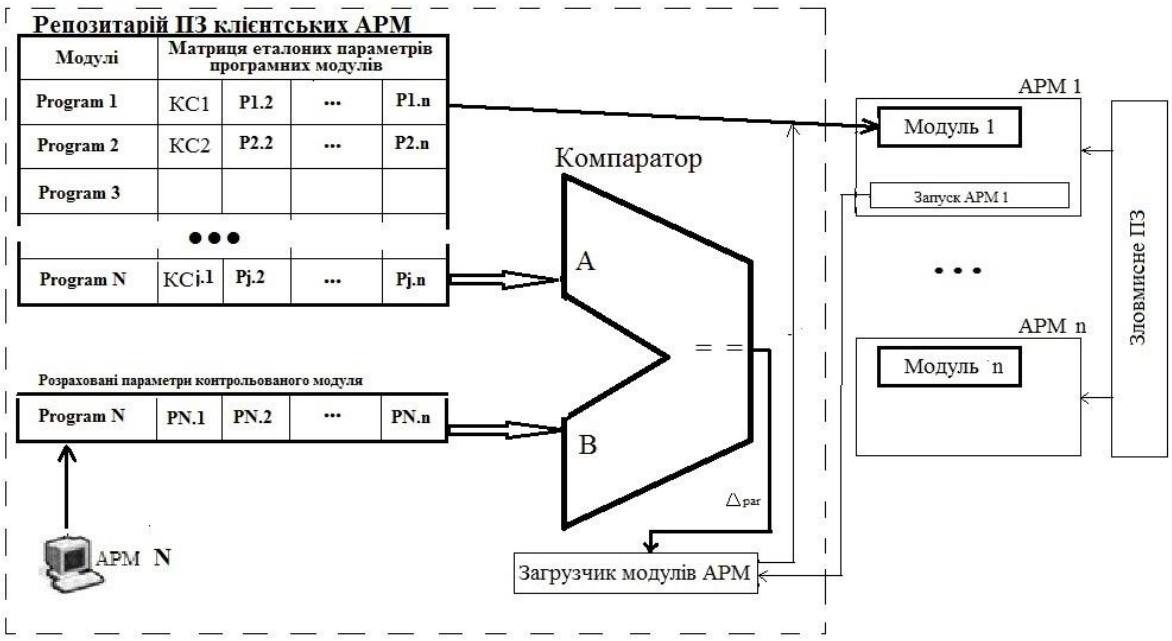


Рис. 3.1 – Схема моделі забезпечення живучості ІТ в умовах впливів ЗПЗ з використанням параметричного контролю актуальності модулів ПЗ клієнтських АРМ та їх маскуванням

Для забезпечення працездатності методу, його модель включає банк ПЗ, що містить в собі програмні модулі всіх клієнтських АРМ ІС та їх еталоні параметри. В якості таких параметрів слугують контрольні суми КС1 - КСn кодових сторінок, полічені за заданим правилом, значення маркерів границь програмних модулів, кількість програмних модулів контрольованого файлу. Набір параметрів контролю може змінюватись, відповідно до структури програмних модулів, які контролюються.

Метод живучості спеціалізованих ІТ базується на використані маркерів, якими позначаються границі програмних модулів, що включені до таких файлів. Це дозволяє виокремити програмну частину із загальної структури файлу і таким чином виконувати розрахунок контрольної суми тільки для сталої частини файлу.

Програмний файл з несталою контрольною сумою необхідно певним чином опрацювати. Для цього в процесі розробки, початок та кінець програмних модулів, що будуть включені до його складу, відмічаються спеціальним маркером, об'єктний код

яких відомий антивірусній програмі, яка контролює його цілісність. Одним із способів реалізації таких маркерів в програмному модулі можуть слугувати оператори програмного коду, які присвоюють унікальне, наперед задане значення, програмній змінній. В процесі контролю актуальності стану програмного модуля клієнтського АРМ перевіряється його наявність по заданому шляху, обчислюються його параметри та порівнюються із еталонними. Задачу параметричного контролю актуальності модулів в рамках цієї концептуальної моделі покладається на програмно реалізований компаратор.

У випадку відсутності контрольованого модуля в заданому місці, або отримання розходження між фактичними та еталонними параметрами Δpar на виході компаратора, виконується відновлення ПЗ клієнтського АРМ з використанням еталонного ПЗ, що зберігається в банку. Сам факт виявлення розбіжностей Δpar автоматично документується із збереженням необхідних для подальшого аналізу параметрів в базі даних У випадку, якщо розходжень між еталоном та модулем, що пройшов контроль актуальності не виявлено, то він, додатково, може маскуватись шляхом перейменування. Це дозволить зменшити вірогідність атаки модуля з боку зловмисного ПЗ, яке, як відомо, в першу чергу вражає виконувані файли.

Така стратегія дозволяє здійснювати контроль актуальності модулів ПЗ клієнтських АРМ в автоматичному режимі, що в свою чергу, дозволяє забезпечувати живучість спеціалізованих ІТ в умовах впливів ЗПЗ та комп'ютерних атак. При цьому характер атаки ЗПЗ на клієнтське ПЗ особливого впливу не матиме. Оскільки на клієнтських ПК дані ІС не зберігаються, то ЗПЗ може лише пошкодити файли програм. Цей факт виявляється в процесі контролю актуальності модулів ПЗ АРМ і вони замінюються еталонними. Результатом стане відновлення доступу до функцій ІС.

Ще однією важливою особливістю розробленого методу згідно параметричного контролю цілісності програмних файлів є можливість його застосування до групи програмних файлів, які не мають сталої контрольної суми і, цим самим він розширює

можливості відомого методу виявлення ЗПЗ, а саме методу контролю цілісності програм, базованого на підрахунку контрольних сум. Суть методу полягає в здійснені постійного циклічного контролю параметрів модулів клієнтських АРМ, як це зображено на рис. 3.2.

Ця задача покладена на фоновий процес, який виконується на захищеному комп'ютері (сервер БД, або комп'ютерній станції адміністратора ІС). Він в циклічному режимі із заданою дискретністю D виконує перевірку програмних модулів всіх зареєстрованих у ІС АРМ на відповідність еталонним параметрам.

Запуск клієнтського ПЗ виконується спеціальною програмою-завантажувачем, якому в якості параметра передається код, який ідентифікує потрібний програмний модуль. Запуск завантажувача виконується з клієнтського ПК, але сам він знаходиться на тому ж фізичному комп'ютері, що і програма фонового процесу. В необхідні моменти часу він взаємодіє із фоновим процесом, подаючи заявку на запуск необхідного АРМ, або вимагаючи позачергового відновлення ПЗ цього АРМ.

Такий алгоритм запуску ПЗ клієнтського АРМ унеможливлює запуск старих версій ПЗ, пошкодженого вірусами ПЗ та його нелегальних копій.

Дискретність D є параметром, значення якого вибирається виходячи із рівня загальносистемної продуктивності роботи комп'ютерної мережі та клієнтських ПК, на яких базуються АРМ. Це дозволяє адаптувати дану технологію під апаратні платформи ІС різної продуктивності.

Цей метод дозволяє здійснювати контроль актуальності модулів ПЗ клієнтських АРМ в автоматичному режимі, що в свою чергу забезпечує живучість ІТ в умовах впливів ЗПЗ. При цьому характер атаки ЗПЗ на клієнтське ПЗ особливої ролі не грає. Оскільки на клієнтських ПК жодні дані ІС не зберігаються, то ЗПЗ може лише пошкодити файли програм. Цей факт виявляється в процесі контролю актуальності модулів ПЗ АРМ і вони заміняються еталонними і, таким чином відновлюється доступ до функцій ІС.

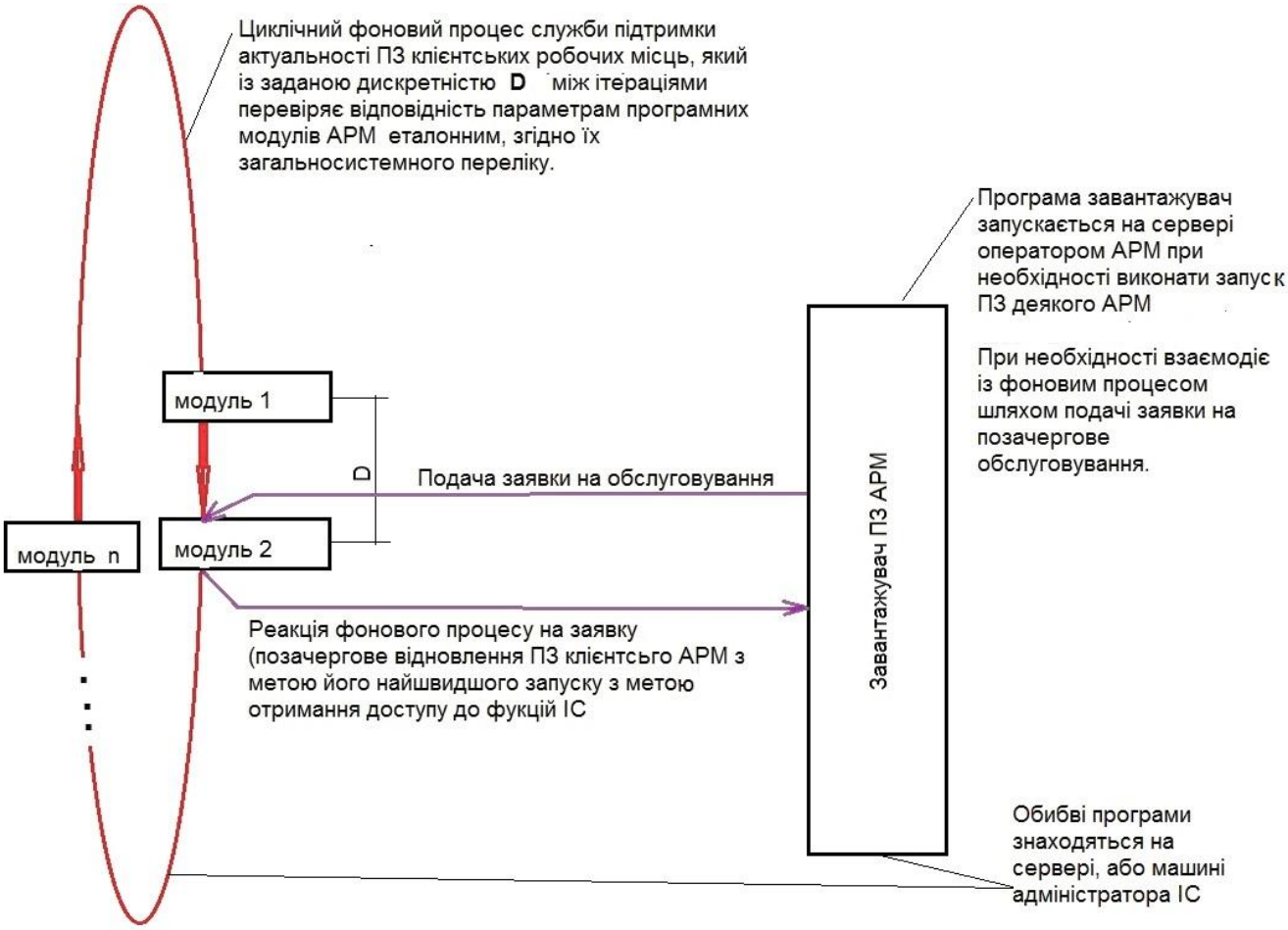


Рис. 3.2 – Взаємодія фонового процесу із завантажувачем модулів ПЗ клієнтського АРМ

Відомо, що всі ІТ характеризуються тривалим життєвим циклом, під час якого їх ПЗ, під дією багатьох зовнішніх та внутрішніх чинників підлягає змінам. І ці зміни тим значимі, чим більшу предметну область охоплює ІТ.

Метод забезпечення живучості спеціалізованої ІТ для протидії атакам ЗПЗ, модель якого зображена на рис. 2.4, реалізований так, що включає в себе кілька взаємодіючих на рівні програмного забезпечення процесів. Ця технологія є доповненням до уже відомих способів захисту ІС методом протидії атакам ЗПЗ.

Вона включає в себе фоновий процес, алгоритм якого в спрощеному вигляді зображений на рис. Д.5 (Додатку Д), під час якого, власне, і перевіряється актуальність

модулів ПЗ клієнтських АРМ та спеціальну процедуру запуску модулів ПЗ АРМ на виконання, зображення якої на рис.Д.6 (Додатку Д).

Фоновий процес служби актуальності ПЗ, виконуючись із заданою періодичністю, моніторить ПЗ кожного зареєстрованого в ІС клієнтського АРМ. В кожній ітерації процес виконує задану послідовність операцій, яка реалізує алгоритм моніторингу актуальності ПЗ клієнтських АРМ:

1. Фоновий процес перевіряє наявність заявки на обслуговування від завантажувача модулів клієнтських АРМ. Це необхідно для того, щоб реакція на факт невдалого запуску деякого програмного модуля АРМ була найшвидшою (рис. Д.5 Додатку Д, оператор 2).

Така ситуація в ІС може виникнути тоді, коли оператор деякого АРМ робить спробу запустити свою клієнтську програму на виконання, а вона із якихось причин недоступна. Завантажувальник програми, в момент спроби її запуску, цей факт виявляє і подає заявку фоновому процесу на першочергове відновлення ПЗ вказаного в заявці АРМ. При отримані заявки, фоновий процес зчитує з неї номер програмного модуля, який потребує першочергового обслуговування і переходить до пункту 6.

2. Якщо заявка на позачергове обслуговування відсутня, то фоновий процес переходить до перевірки наступного в черзі модуля (рис. Д.5 Додатку Д, оператор 3).

3. На наступному кроці перевіряється, чи знаходиться в визначеному місці файлового каталогу клієнтського комп’ютера модуль клієнтського АРМ, що аналізується,. У випадку його відсутності з будь-якої причини виконується перехід до пункту 6 (рис. Д.5 Додатку Д, оператор 4).

4. Якщо модуль в наявності і в заданому місці, то виконується перевірка його активності (рис. Д.5 Додатку Д, оператор 5). На цьому етапі виясняється, чи він завантажений у пам’ять ПК і виконує покладену на нього в рамках ІС функцію. Якщо в момент перевірки модуль активний, то виконується перехід до пункту 7.

5. Контроль параметрів чергового програмного модуля n-го АРМ на відповідність еталонним, що зберігаються в банку служби актуальності ПЗ (рис. Д.5

Додатку Д, оператор 6). Якщо відхилень від еталонних параметрів не виявлено, то файл маскується шляхом перейменування (рис. Д.5 Додатку Д, оператори 9,10). Це дозволяє вивести його з під ймовірної атаки зі сторони ЗПЗ, знаючи, що воно атакує виконувані файли, орієнтуючись на їх розширення. Потім виконується перехід до пункту 7, інакше до наступного.

6. Виконується заміна пошкодженого, чи відновлення відсутнього програмного модуля еталонним з банка ПЗ (рис. Д.5 Додатку Д, оператор 8).

7. Перевіряється подача команди зупинки фонового процесу (рис. Д.5 Додатку Д, оператор 11).. Якщо ні, то виконується завершення поточної ітерації з наступним переходом до пункту 1.

8. Завершення фонового процесу.

Оскільки дана технологія призначається для спеціалізованих ІТ, які самі по собі можуть стати об’єктом цілеспрямованої атаки, то в алгоритм (рис. Д.5 Додатку Д), може бути включено ще одну функцію, завдання якої полягає в формуванні за місцями знаходження виконуваних модулів клієнтських АРМ спеціальних файлів-пасток, які мають слугувати хибними об’єктами атаки для ЗПЗ, в той час, як реальні модулі замасковані. Файли-пастки не відрізняються від програмних модулів АРМ ІС за винятком того, що вони ніколи не зможуть бути завантаженні на виконання. Перед запуском реального модуля вони знищуються і потім створюються заново, після того як модуль завершить свою роботу. Разом із функцією маскування програмних модулів АРМ, функція формування хибних об’єктів атаки дозволяє направити деструктивні дії ЗПЗ в напрямку, який нічим не загрожує функціонуванню ІС.

Процес запуску модулів ПЗ клієнтських АРМ на виконання, також, має свою особливість - він виконується в два етапи. Спочатку, із клієнтського ПК, запускається коротка програма завантажувач, яка постійно зберігається в захищеному каталозі служби актуальності ПЗ спеціалізованої ІТ. Завантажувач стартує, знаходить в визначеному місці файл відповідного модуля, відновлює його ім'я і передає йому управління.

Якщо файл з певної причини не буде знайдено, то завантажувач модулів видасть фоновому процесу заявку на позачергове оновлення ПЗ вказаного АРМ і перейде в режим очікування (рис. Д.6 Додатку Д, оператори 4, 5). Після того, як фоновий процес виконає заявку завантажувальника, то той в свою чергу, повторить процедуру запуску відповідного модуля (рис. Д.6 Додатку Д, оператори 6, 7).

Можливості даної технології відновлювати роботу пошкодженого та знищеного ПЗ клієнтських АРМ дозволяє ліквідувати наслідки атак ЗПЗ як того, що проникає в модулі АРМ ІС, так і того, чиї деструктивні дії проявляється в шифруванні даних.

Таким чином, основні кроки методу забезпечення живучості спеціалізованих ІТ.

1. Включення маркерів до складу ПЗ на підготовчому етапі забезпечення живучості спеціалізованої ІТ.

Наприклад, здійснимо підготовку ПЗ клієнтського АРМ для застосування методу. Для того, щоб стало можливим застосування маркерів ПЗ клієнтського АРМ необхідно включити маркери до складу програмного модуля, як це показано на рис. 3.3. На рисунку лінія червоного кольору підкреслює маркери початку і кінця модуля. Перший маркер "M00002" зі зміщенням 0018A906 знаходиться на початку програмного модуля, а маркер "M00003" зі зміщенням 0018AFD8 є його закінченням.

Все, що знаходиться між маркерами і самі маркери є кодом програмного модуля. Фрагменти об’єктного коду зображені на рис. 3.3. Їх особливість в сталості значення контрольної суми.

2. Розрахунок контрольних сум маркованого відповідно до п.1 виконуваного модуля ПЗ клієнтського АРМ. Їх число може складати множину від 1 до n.

3. Виконувані модулі ПЗ клієнтського АРМ розміщаються в базі еталонного ПЗ служби контролю актуальності.

4. Збереження списку значень маркерів та контрольних сум маркованих ними модулів.

5. Якщо фоновий процес (рис. 3.2) запущений, то в певній його ітерації буде виконано перевірку ПЗ деякого АРМ і, якщо його список маркерів і значення

контрольних сум, що їм відповідають не збігаються, або потрібних маркерів взагалі не буде знайдено, то буде виконана процедура оновлення ПЗ даного АРМ.

6. Факт оновлення ПЗ деякого АРМ заноситься в log-файл для подальшого аналізу причин, що викликали його заміну.

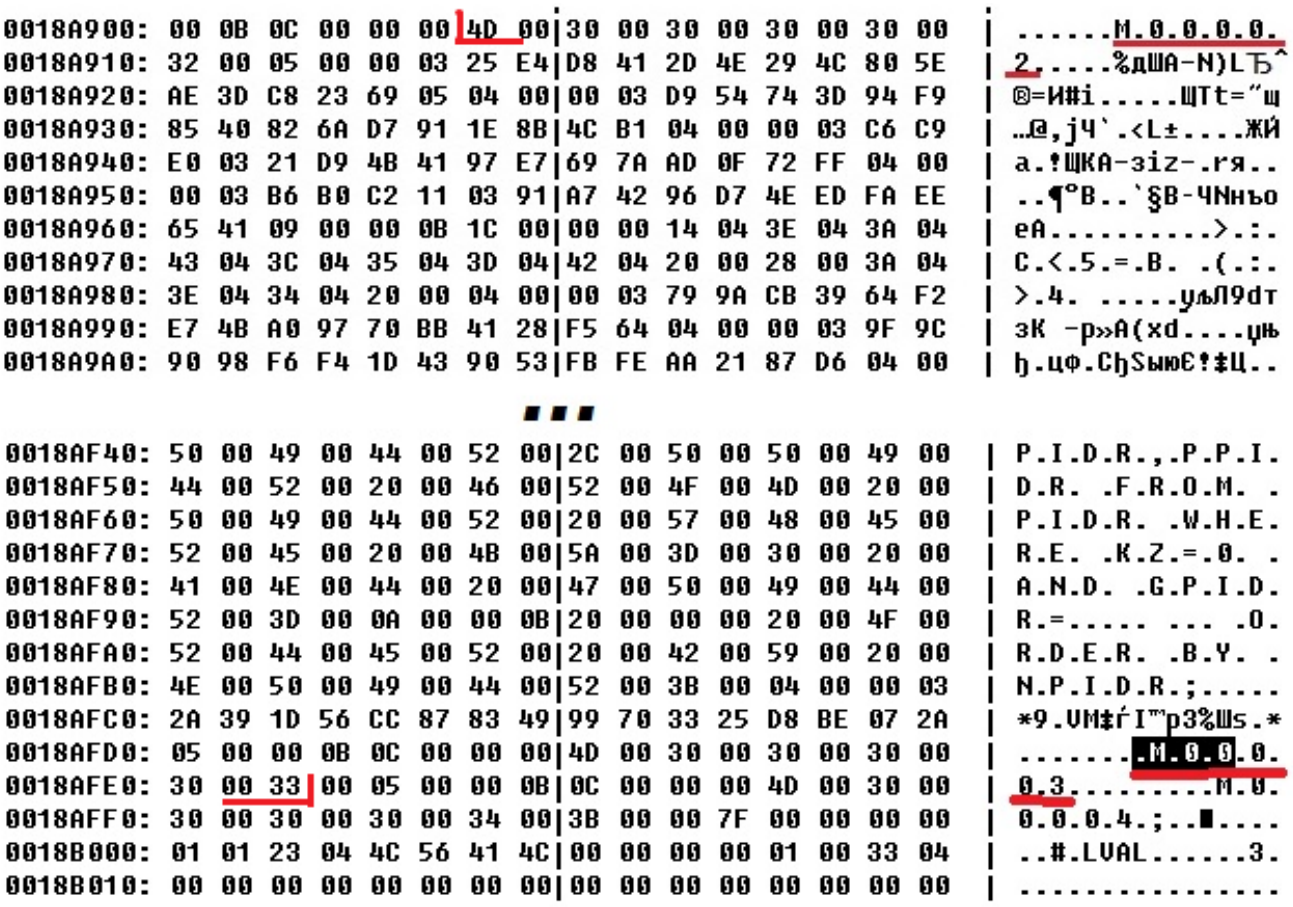


Рис. 3.3 – Фрагмент об’єктного коду ПЗ клієнтського АРМ з маркерами початку і кінця програмного модуля

Розроблений метод забезпечення живучості [74 - 76, 129, 130] спеціалізованих ІТ базується на аналізі маркерів та збереженні ключової інформації, яка потрібна для дослідження в процесі функціонування ІС. Це дає можливість покращити забезпечення живучості ІТ в умовах впливів ЗПЗ та комп’ютерних атак.

3.3. Оцінка ефективності методу забезпечення живучості спеціалізованих ІТ

Для кожної компоненти M_i, $(i = 1, 2, \dots, n,\ n$ – кількість компонентів спеціалізованої ІТ) застосуємо функцію, яка включатиме всі критерії ефективності в корпоративних комп'ютерних мережах, застосування яких при розробці спеціалізованої ІТ необхідно для подальшого користування нею. Зокрема, серед таких критеріїв буде, також, критерій забезпечення живучості. Задамо критерій ефективності живучості спеціалізованих ІТ вектором, компонентами якого будуть функції ефективності для живучості ІТ, що відповідатимуть конкретним підкритеріям:

$$K_e = (f_1, f_2, \dots, f_m), \tag{3.5}$$

де f_j – j-та функція, яка задає один з підкритеріїв ефективності, $j = 1, 2, \dots, m$, m - кількість функцій.

Враховуючи те, що в цілому задача досягнення максимальної ефективності залежить від конкретних підкритеріїв, які можуть бути пов'язані між собою і відповідно впливати один на одного, при цьому покращення ефективності одного може призводити погіршення іншого. Крім того, оскільки спеціалізовані ІТ складаються з компонентів, до яких застосовуються ті ж критерії із заданого вектору, то задача ускладнюється тим, що частина компонентів ІТ є різною і, відповідно, досягнення ефективності за тими ж наборами критеріїв буде різною. Тому, вибір оптимальних розв'язків є складною багатокритеріальною задачею. Загальну постановку задачі пошуку найкращої ефективності для забезпечення живучості спеціалізованих ІТ в корпоративних комп'ютерних мережах сформулюємо так:

$$\begin{cases} K_e(M_{IT}) \to max; \\ f_j(M_i) \to max, i = 1, 2, \dots, n, j = 1, 2, \dots, m \end{cases}. \tag{3.6}$$

Крім того, частина компонентів спеціалізованої ІТ може повторюватись за функціоналом в залежності від задач та розміщення в корпоративних комп'ютерних мережах. Це вплине на загальну ефективність забезпечення живучості спеціалізованої ІТ. Але досягнення ефективності за певними підкритеріями в однакових компонентах спеціалізованої ІТ не обов'язково повинно бути однаковим, бо в цих компонентах вирішуватимуться різні завдання чи ті ж завдання, але в поточний момент часу вони проходитимуть різні етапи. Врахування цих особливостей є важливим, тому деталізуємо постановку задачі пошуку найкращої ефективності для спеціалізованих ІТ в корпоративних комп'ютерних мережах так:

$$\begin{cases} K_e(M_{IT}) \to max; \\ f_{j,q}\left(M_{i,p}\right) \to max, i = 1,2,\dots,n, j = 1,2,\dots,m, \\ q = 0,1,\dots,n_q, j = 0,1,\dots,n_p \end{cases} \quad (3.7)$$

де q – номер компоненти спеціалізованої ІТ у певному вузлі корпоративної комп'ютерної мережі; j – індекс для критерію ефективності компоненти спеціалізованої ІТ у певному вузлі корпоративної комп'ютерної мережі; $q = 0,1,\dots,n_q, j = 0,1,\dots,n_p$; n_q – кількість однакових компонент спеціалізованої ІТ у корпоративній комп'ютерній мережі; n_p – номер критерію для однакових компонент спеціалізованої ІТ у корпоративній комп'ютерній мережі.

Введемо функцію, яка буде визначати максимальне значення критерію ефективності:

$$F\colon K_e(M_{IT}) \to max; \quad (3.8)$$

Значення критерію ефективності задамо виразом з врахуванням вагових коефіцієнтів:

$$K_e(M_{IT}) = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{p=0}^{n_p}\sum_{q=0}^{n_q}\left(\alpha_{i,j,p,q}\cdot f_{j,q}\left(M_{i,p}\right)\right), \quad (3.9)$$

де $\alpha_{i,j,p,q}$ - вагові коефіцієнти.

Розглянемо досягнення максимізації критеріїв за показниками живучості в конфігураціях інформаційних технологій, які побудовані на основі архітектури «клієнт – сервер» з їх забезпеченням за усіма ланками системи: від користувачів (клієнтська частина) до критично важливої серверної частини. Вибір для розгляду саме архітектури «клієнт – сервер» залежить від її особливостей, які проявляються в наступному: базові функції клієнтського застосунку розподіляються між клієнтом та сервером; програмне забезпечення автоматизованого робочого місця клієнтського комп'ютера працює з даними через запити до серверного програмного забезпечення; здійснюється повна підтримка багатокористувацької роботи; гарантується цілісність даних. Це відрізняє її від інших архітектур і дозволяє здійснити забезпечення живучості до кожної з ланок системи окремо.

Основними напрямами для підвищення живучості спеціалізованих ІТ є внесення надмірності в конфігурацію апаратних і програмних засобів, підтримуючої інфраструктури, резервування інформаційних ресурсів (програм та даних). При цьому ІТ повинна відповідати наступним основним вимогам: система повинна будуватись так, щоб в ній був відсутній компонент (ресурс), відмова якого призведе до повної відмови всієї системи. Для систем реального часу, додатково, накладаються часові обмеження досягнення результату.

Розглядатимемо спеціалізовану ІТ, яка відноситься до систем ірреального часу. Тому, часові обмеження для неї набагато менш жорсткі, порівняно з системами реального часу. В зв'язку з вибраною для розгляду архітектурою реалізації ІТ, яка складається із серверної та клієнтської частин, то відповідно розглядатимемо питання відмовостійкості та живучості в співвіднесенні до функцій покладених на них. Незважаючи на те, що ці дві частини будучи складовими єдиної, логічно нерозривної

ІТ, виконують в рамках неї свої, специфічні функції, забезпечення живучості для кожної складової ІС досягається різними шляхами.

Головною властивістю живучості є прозорість відмов її окремих компонентів для кінцевого користувача. Це означає, що живучість системи автоматично змінює свою конфігурацію у випадку відмови. Її програмне забезпечення в процесі виконання шукає обхідні шляхи, намагаючись в умовах відмови, привести виконувану функцію до успішного завершення. Задамо функцію $f_1(S_i), i = 1, 2, \dots, n$ визначення живучості в комп'ютерних системах в кількісному вигляді так:

$$f_1(M_i) = \frac{T_{f_1(M_i),1}}{T_{f_1(M_i),1} - (T_{f_1(M_i),2} + T_{f_1(M_i),3})}, \quad (3.10)$$

де i – кількість компонентів спеціалізованої ІТ, $i = 1, 2, \dots, n$, $T_{f_1(M_i),1}$ – час між сусідніми збоями; $T_{f_1(M_i),2}$ – час, необхідний для виявлення збою та пошуку шляху його обходу; $T_{f_1(M_i),3}$ – час, необхідний для відновлення ІТ після збою.

Згідно з формули (3.10), для ІТ з автоматичною системою забезпечення живучості вона буде наближатись до максимуму, через швидкість реакції. Для побудови таких систем немає теоретичних перешкод, але в практиці при їх реалізації, потрібно враховувати ряд значимих факторів: фінансові витрати реалізації автоматичної системи забезпечення живучості; складність системи. Для ІТ, призначених для інформаційного забезпечення у вузькій спеціалізованій предметній області, наприклад фінансово-господарська діяльність закладу вищої освіти, буде доцільним відмовитись від автоматичної системи керування живучістю на користь автоматизованої. При такому підході частина дорогих функцій управління надмірностями, присутніми в ІТ, буде покладена на людину, якщо це не загрожує можливими значними втратами. Тоді, згідно формули (3.10), живучість $f_1(M_i)$ буде нижчою, ніж в першому випадку. Але вирішенням задачі побудови ІТ (аналогічно, як і в інших задачах проєктування), є не забезпечення максимально можливої живучості

системи, а знаходження прийнятного балансу параметрів системи, в рамках певного технологічного базису. А, також, в тому числі враховуючи вимоги критерію «живучість \ вартість». Дослідимо вирішення питань забезпечення живучості ІТ при використанні такої стратегії. Проаналізуємо фактори, що негативно впливають на живучість ІТ зі сторони клієнта. Це потрібно для того, щоб оцінити і виробити адекватні заходи протидії. Схема впливу негативних факторів на живучість клієнтської частини спеціалізованої ІТ зображено на рис. 2.3.

Показники живучості в складній системі: багатофункціональність окремих компонент; наявність єдиної (головної) мети функціонування всієї системи; можливість не тільки інформаційного обміну між окремими компонентами, але й інформаційної взаємодії з користувачами; наявність засобів захисту, контролю, діагностики й самоорганізації. Задача аналізу структурної живучості потребує визначення: системної архітектури, необхідної для виконання цілі функціонування ІТ у деякий момент або проміжок часу, коли виникають небажані впливи на систему; вимог щодо окремих видів ресурсів системи та їхнього взаємозв'язку; вимог щодо функціональних можливостей ресурсів системи; особливостей характеру небажаних впливів чи їхніх наслідків. Задамо функцію $f_2(M_i)$, в якій $i = 1, 2, \dots, n$, визначення живучості в кількісних одиницях в комп'ютерних мережах виразимо так:

$$f_2(M_i) = \frac{T_{f_2(M_i),1} + T_{f_2(M_i),2}}{T_{f_2(M_i),1}}, \tag{3.11}$$

де $T_{f_2(M_i),1}$ – час функціонування процесі ІТ в стандартному режимі роботи, $T_{f_2(M_i),2}$ – час витрачений на процеси забезпечення живучості, $i = 1, 2, \dots, n$.

Таке визначення функції живучості надає можливість відобразити стандартний режим роботи значенням одиниці, а при виникненні потреби у забезпеченні живучості

і у випадку набагато тривалішого часу, ніж час стандартного режиму роботи, значення функції відображатиме кількісну порядкову величину.

Згідно формул (3.9) – (3.11) отримаємо значення ефективності для ІТ з врахуванням показників двох підкритеріїв живучості, отриманих за різними показниками та оцінюванням їх значущих впливів на результат:

$$K_e(M_{IT}) = \sum_{j=1}^{m} \sum_{p=0}^{n_p} \sum_{q=0}^{n_q} \left(\alpha_{1,j,p,q} \cdot \frac{T_{f_1(M_i),1}}{T_{f_1(M_i),1} - (T_{f_1(M_i),2} + T_{f_1(M_i),3})} + \alpha_{2,j,p,q} \cdot \frac{T_{f_2(M_i),1} + T_{f_2(M_i),2}}{T_{f_2(M_i),1}} \right), \quad (3.12)$$

де $\alpha_{1,j,p,q}$ – коефіцієнт для значення, яке визначає відмовостійкість в кількісних одиницях; $\alpha_{2,j,p,q}$ – коефіцієнт для значення, яке визначає живучість в кількісних одиницях; $\alpha_{1,j,p,q} + \alpha_{2,j,p,q} = 1$.

Аналогічно, доданками в формулі (3.9) та її конкретизації формулою (3.12) для двох величин можуть бути інші показники, які характеризують ефективність забезпечення живучості спеціалізованої ІТ.

В результаті використання перелічених заходів було отримано ІТ вузькоспеціалізованого використання для різних сфер застосування, де супроводжуванні процеси відносяться до ірреального або нереального часу із досить високими параметрами живучості і, в той же час, прийнятним рівним фінансових витрат на її експлуатацію.

Для визначення на скільки ефективними є запропоновані рішення із забезпечення живучості проведемо порівняння критерія ефективності для ІТ без забезпечення живучості і з включенням цих характеристик згідно формули (3.11).

Значення величини критерія ефективності забезпечення живучості спеціалізованої ІТ, в якій не забезпечуються вимоги живучості отримуємо з формули

(3.12) так: 1) вирішення проблем, пов'язаних із відсутністю забезпечення в ІТ реалізованій підсистемі живучості, покладено на оператора чи адміністратора, який постійно моніторить функціонування ІТ; вирішення проблемних ситуацій здійснюється тільки при їх виявленні. В першому випадку розрахунок за формулою (3.12) може бути аналогічним і значення отриманої величини на порядки перевищуватиме значення критерію для ІТ, де забезпечується живучість. Якщо ж розглядати другий варіант, тоді $K_e(M_{IT}) = 1$. В цьому випадку, відношення між значеннями визначається за формулою (3.13) і дозволяє встановити ефективність запропонованих рішень із забезпечення живучості, а також покращити досягнення ефективності за рахунок коригування коефіцієнтів:

$$\mu = \frac{1}{\sum_{j=1}^{m}\sum_{p=0}^{np}\sum_{q=0}^{nq}\left(\alpha_{1,j,p,q}\cdot\frac{T_{f_1(M_i),1}}{T_{f_1(M_i),1}-(T_{f_1(M_i),2}+T_{f_1(M_i),3})}+\alpha_{2,j,p,q}\cdot\frac{T_{f_2(M_i),1}+T_{f_2(M_i),2}}{T_{f_2(M_i),1}}\right)}, \quad (3.13)$$

де відсутність збоїв в роботі спеціалізованої ІТ або зовнішніх впливів означатиме, що час витрачений на їх обробку дорівнюватиме нулеві і відповідно відношення стане дорівнювати одиниці.

Якщо ж відбудеться збій або зовнішнє втручання, тоді значення μ буде більше одиниці. Ефективним значенням є значення мінімально відхилене від одиниці.

Результати забезпечення живучості спеціалізованої ІТ зображені в реалізованій ІТ на рис. 3.4.

Для зручності всі рядки фрагмента лог-файлу були пронумеровані, а критичні позиції виділені.

В позиції 19 виявлено фатальну помилку в роботі мережевого адаптера «eth0» в момент звернення користувацького комп'ютера з IP 192.168.168.2.

В позиції 35,36 закривається поточна сесія користувача SWM.

В позиції 37 система сповіщає, що потрібна реконфігурація мережевих пристроїв.

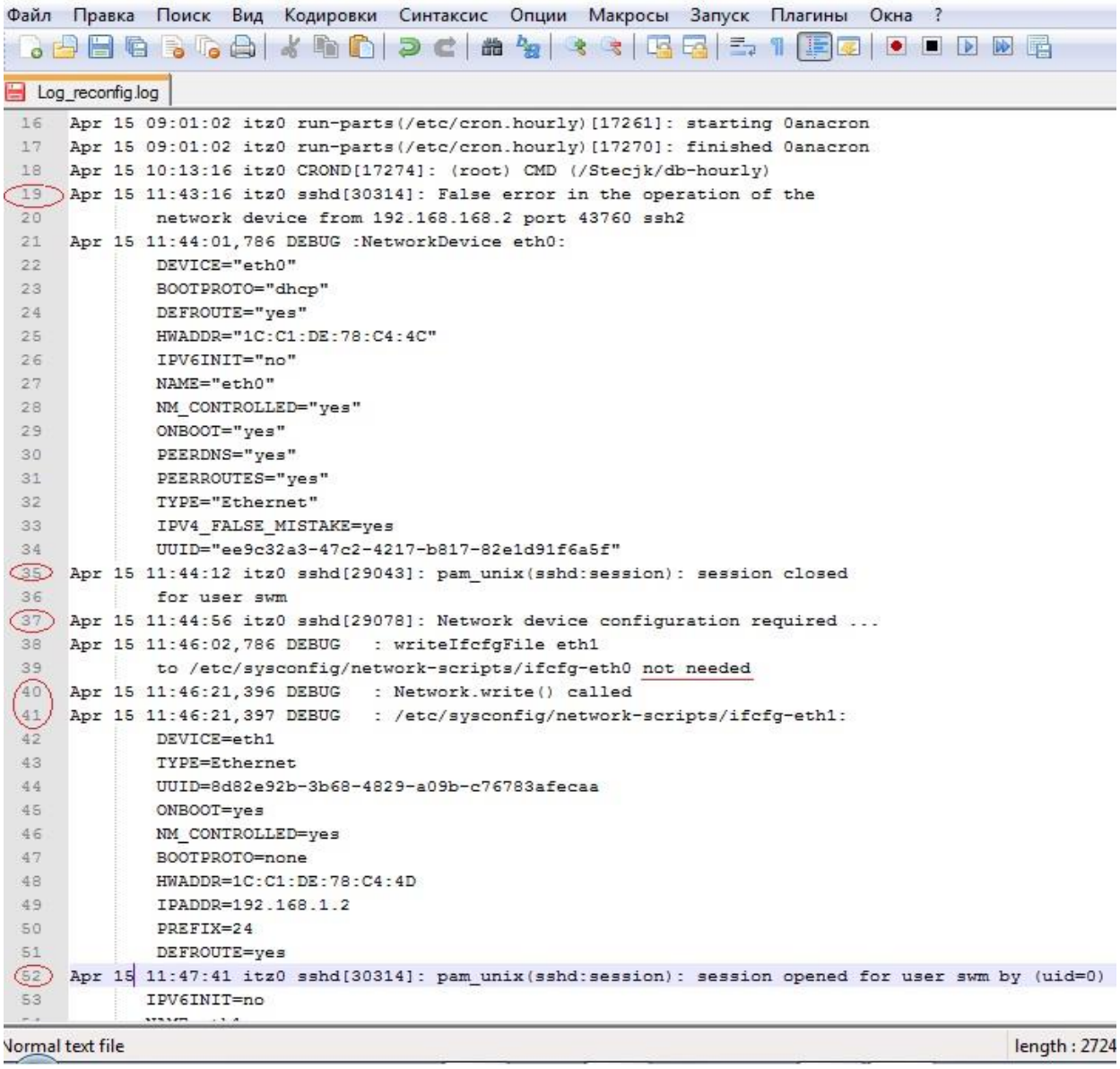


Рис. 3.4 – Вміст файлів-відомостей про збої та зовнішні впливи

В позиції 38 сповіщається, що пристрій «eth0» відключається.

В позиціях 40,41 повідомляється, що активується резервний мережевий адаптер «eth1».

В позиції 52 сповіщається, що відкривається сесія користувача SWM, яка була припинена через вихід із ладу мережевого адаптера «eth0».

Фрагмент лог-файлу на рис. 3.5 відображає роботу підсистеми транзакцій бази даних під час її резервного копіювання.

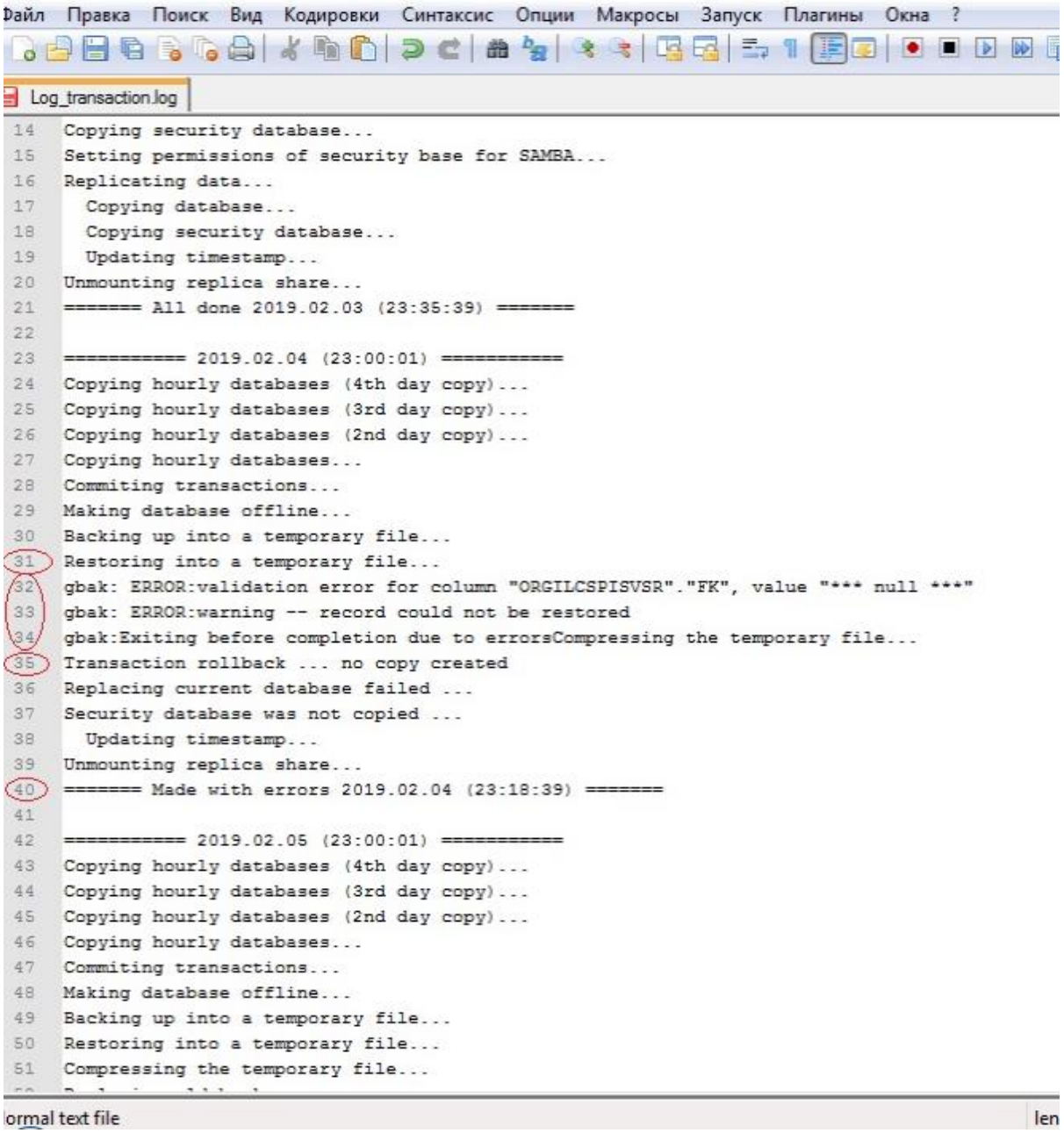


Рис. 3.5 – Фрагмент лог-файлу підсистеми резервного копіювання бази даних

При виконані процедури створення резервної копії бази даних утилітою GBAK 4 лютого 2019 року сталась помилка в даних, що призвела до відкату транзакції. Критичні позиції виділені.

Позиція 31. Створення тимчасового файлу бази даних.

Позиція 32-34. Повідомлення утиліти GBAK про помилку при спробі запису в

поле [FK] таблиці ORGILCSPISVSR значення за замовчуванням визначника NULL.

Позиція 35. Відкат поточної транзакції через помилку.

Позиція 40. Сповіщення, що процедура створення резервної копії завершилась із помилками.

Наведено графіки (рис. 3.6), отримані за розрахунками за формулою (3.12) для результатів живучості (формула (3.13)).

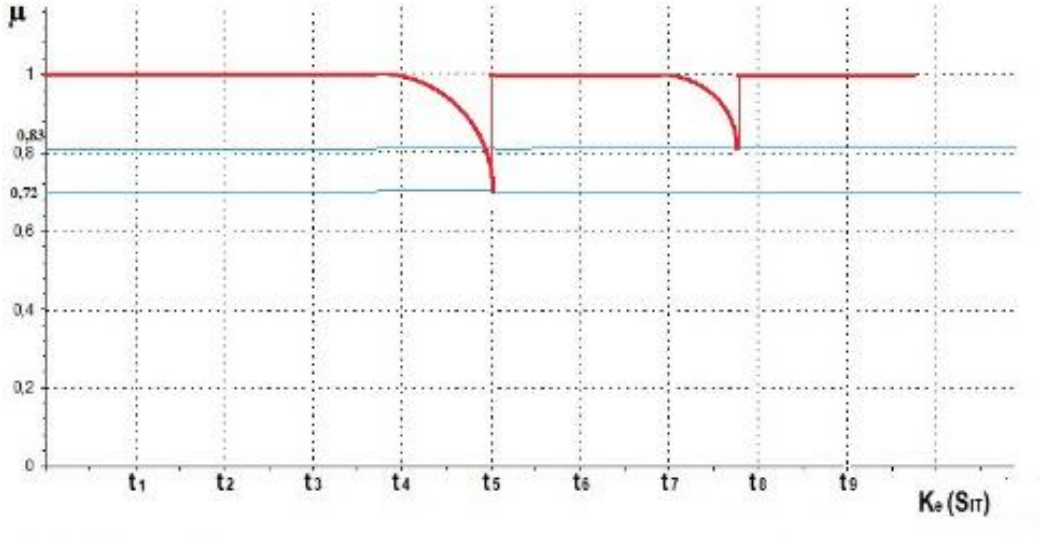


Рис. 3.6 – Графік проявів живучості

Результати дослідження підтверджують високий рівень живучості в корпоративних комп’ютерних мережах, який становить понад 72%.

Оціночні значення відмовостійкості при імплементації в ІС розробленого методу забезпечення відмовостійкості ІТ зображено на рис. 3.7, які розраховано за формулою (3.12), і відображають належний рівень відмовостійкості щодо впливів ЗПЗ та комп’ютерних атак в процесі активізації підсистеми забезпечення відмовостійкості в ІС, який становить не менше 76%.

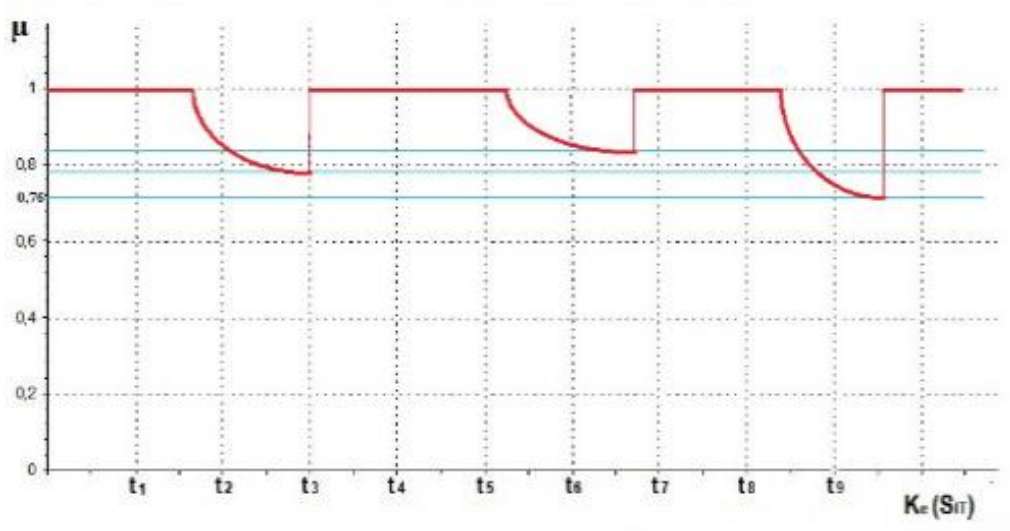


Рис. 3.7 – Графік оцінки значень відмовостійкості

Таким чином, розроблений підхід [74, 77] до визначення ефективності ІТ на основі врахування кількісних величин, які характеризують живучість, та може бути розширений для врахування інших характеристичних величин. Зокрема, запропонована методика визначення ефективності ІТ може бути застосована і для оцінки методу забезпечення відмовостійкості ІТ. Для забезпечення живучості ІТ розроблено систему заходів в результаті виконання яких отримано ІТ вузькоспеціалізованого використання для різних сфер застосування, де супроводжуванні процеси відносяться до ірреального або нереального часу із досить високими параметрами живучості і, в той же час, прийнятним рівним фінансових витрат на її експлуатацію.

3.4. Метод забезпечення захисту інформації спеціалізованих інформаційних технологій в умовах впливів ЗПЗ

Питання забезпечення захисту інформації є фундаментальною складовою, поряд із відмовостійкістю та живучістю, при розробці інформаційної технології для

побудови на її основі спеціалізованих інформаційних систем, що працюють в умовах впливів зловмисного ПЗ.

Проведений аналіз показав, що на сьогодні основна загроза для інформації, що зберігається в комп'ютерній системі, надходить із глобальної комп'ютерної мережі. Тому, структура комп'ютерної мережі, на якій буде базуватись робота ІС повинна передбачати її розділення на локальні сегменти з обмеженням доступу до них. В таких захищених сегментах із контрольованим доступом, розміщуються серверна частина ІС та її клієнтські місця, що забезпечують основну функціональність системи.

Такий підхід (рис. 3.7) до побудови комп'ютерної мережі дозволив зменшити ризик для ІС з боку зловмисного ПЗ та спроб несанкціонованого доступу до інформаційної системи з використанням каналів INTERNET.

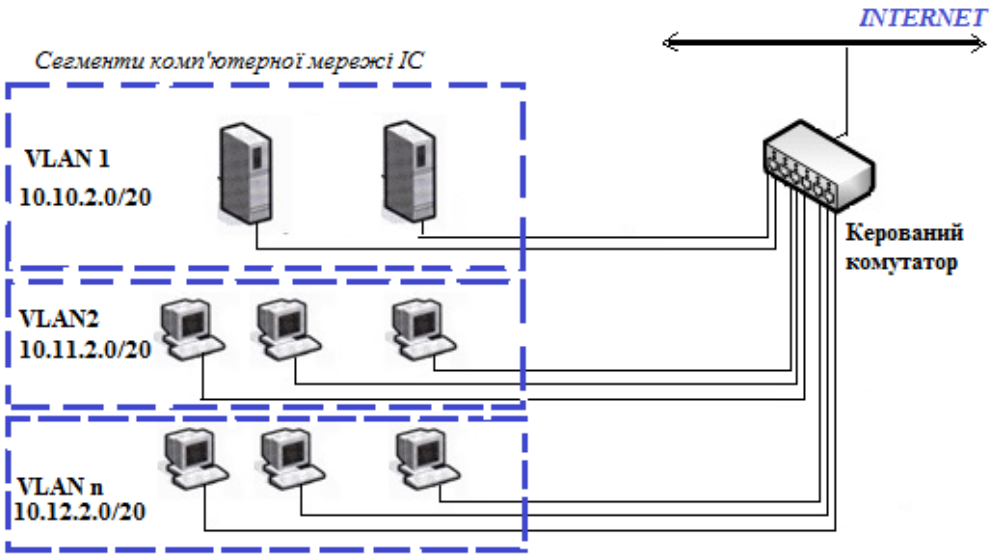


Рис. 3.7 – Спрощена топологія сегментованої комп'ютерної мережі спеціалізованої ІС

Застосування керованого комутатора з функцією створення віртуальних комп'ютерних мереж (VLAN) дозволило отримати можливість.

1. Захистити мережу від стороннього втручання. Порт керованого мережевого комутатора зможе ігнорувати та відсікати пакети, які надходять з інших підмереж, причому незалежно від початкової IP-адреси.

2. Гнучко управляти розділенням комп'ютерів по віртуальним підмережах, забезпечуючи ізольованість одна від одної, при цьому їх топологія не залежить від того, де фізично знаходяться мережі компоненти.

3. Забезпечення зменшення широкомовного трафіку в мережі. Кожна створена віртуальна підмережа є окремим широкомовним доменом, широкомовний трафік якого не транслюватиметься між різними підмережами, зменшуючи навантаження на мережеве обладнання.

4. Розбиття мережі на віртуальні підмережі, дозволило застосовувати свої правила безпеки для кожної із них (рис.3.8), що загалом підвищує безпеку та керованість мережі в цілому.

Це дозволяє гнучко налаштувати фільтрацію пакетів для кожної підмережі, яка виконується з оцінкою даних на основі IP-інформації, що міститься в заголовку пакета, а саме адреси відправника і одержувача пакета. В процесі фільтрації пакетів інформація, отримана з IP-заголовку зіставляється зі списком правил фільтрації для дозволу або заборони передачі пакета. В розробленій ієрархії правил фільтрації враховуються наявні поля IP-адрес, типи протоколів, номери портів відправника і одержувача. Перш, ніж дозволити пакету продовження передбачуваного для нього маршруту, правила фільтрів пакетів перевіряють вказані в них дані на відповідність зумовленим значенням. Це дозволяє, зберігаючи високий рівень захисту інформації в мережі, мати гнучко керований доступ до її ресурсів та різко скоротити об’єм інформації про мережу, яку отримає зловмисник при спробі її розвідки, особливо враховуючи ту обставину, що вміст пакетів має криптографічний захист.

Окрім базування ІС на сегментовану комп’ютерну мережу, передбачається протидія зловмисному ПЗ на рівні окремої комп’ютерної системи, на якій буде базуватись клієнтське робоче місце (рис. 3.9).

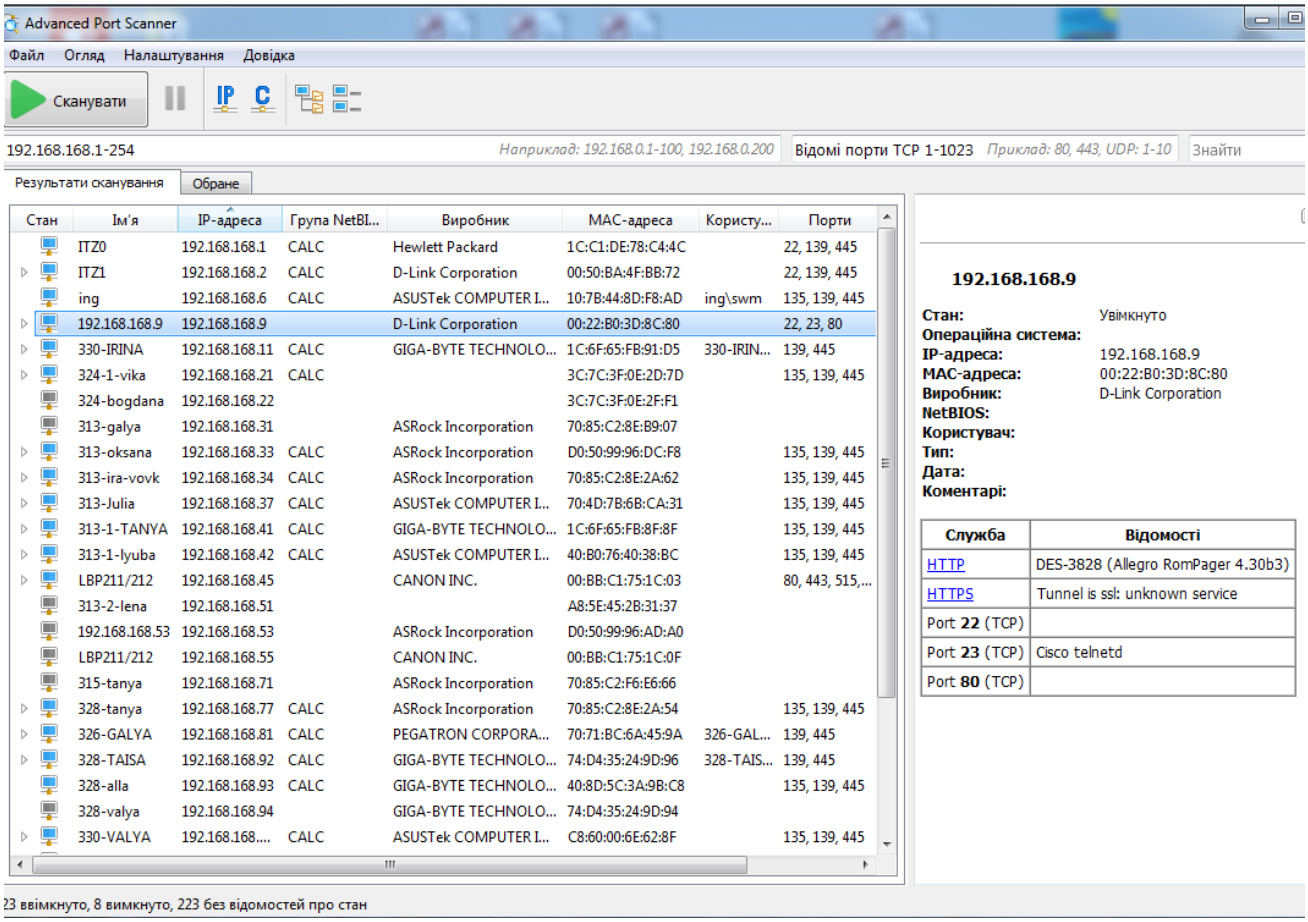


Рис. 3.8 – Налаштування політик безпеки компонентів мережі ІС

Ця протидія здійснюється на двох рівнях - як на загальносистемному із використанням можливостей операційної системи, так і локальному, із використанням ресурсів самої ІС (рис. 3.9). Загальносистемний рівень є основним в протидії ЗПЗ. На ньому, як видно з рис. 3.9, використовуються найпотужніші засоби протидії. Але все одно це не надає повної гарантії безпеки. Тому, локальна протидія ЗПЗ засобами самої ІС, особливо з можливістю самовідновлення пошкодженого ПЗ, дозволяє повернути клієнтському АРМ працездатність навіть в умовах, коли комп'ютер знаходиться під контролем ЗПЗ.

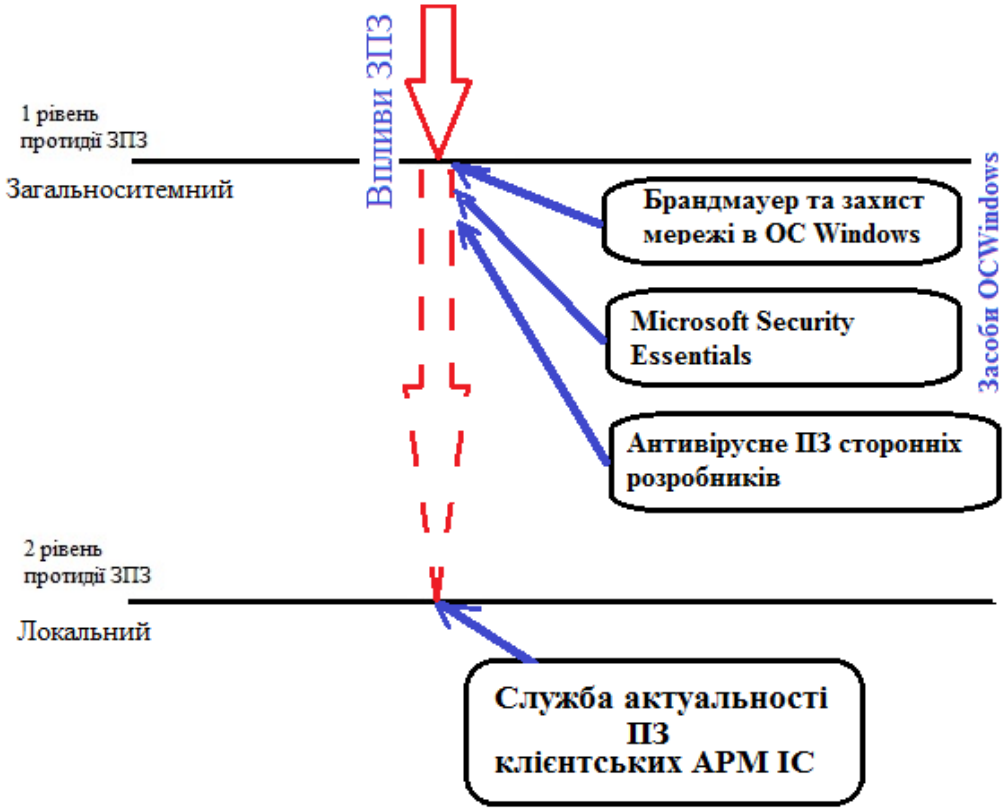


Рис. 3.9 – Модель організації дворівневої схеми протидії ЗПЗ для клієнтських АРМ ІС

Для цього в рамках ІС залучається ПЗ служби контролю актуальності програмного забезпечення клієнтських АРМ ІС. Її основне призначення – автоматичне оновлення версій програмного забезпечення АРМ. Але аналіз її алгоритму роботи показав, що служба контролю актуальності ПЗ АРМ може бути використана і для протидії ЗПЗ.

Вона не зможе виявляти і знешкоджувати зловмисні програми, але відповідно до свого алгоритму роботи, вона може локально ліквідовувати їх прояви, такі як знищення програмних модулів АРМ, їх пошкодження, шифрування і таким чином посилювати захист інформації ІС, відновлювати доступ до її функцій.

Як видно з рис. 3.10, в основі служби контролю актуальності ПЗ лежить процедура програмного компаратора, яка виконується як фоновий процес. Її алгоритм та схема роботи зображені на рис. 3.11.

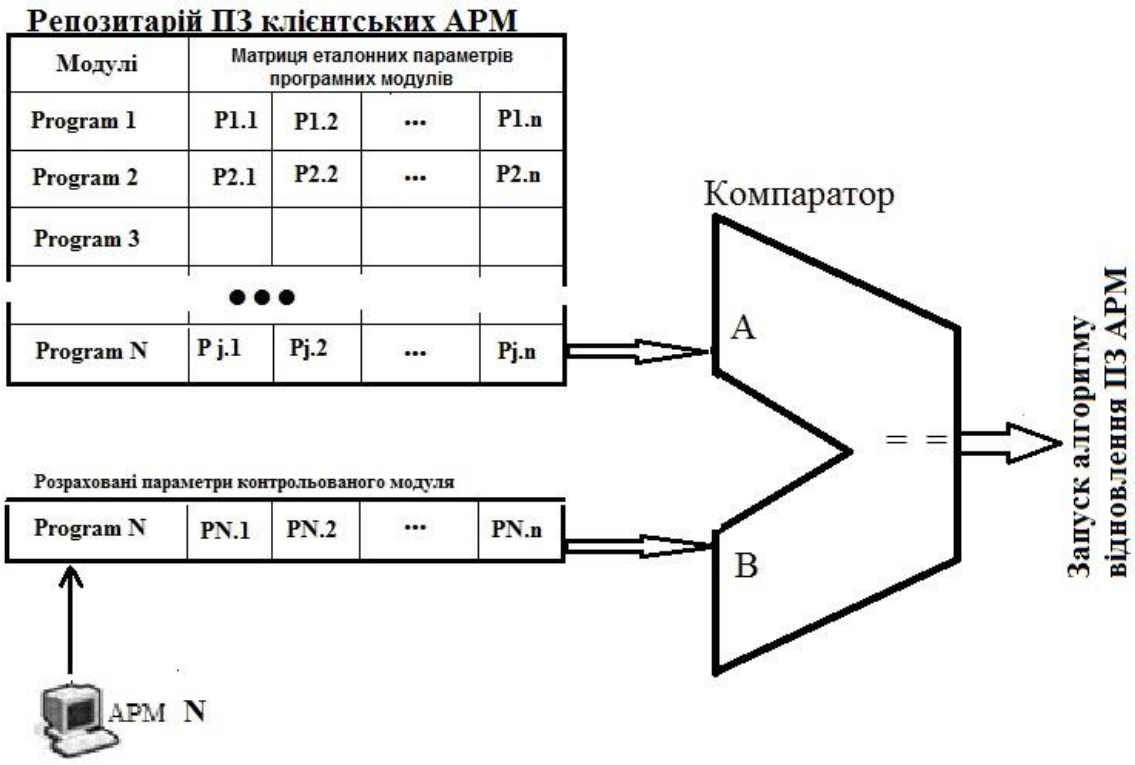


Рис. 3.10 – Архітектура служби контролю актуальності ПЗ клієнтських АРМ

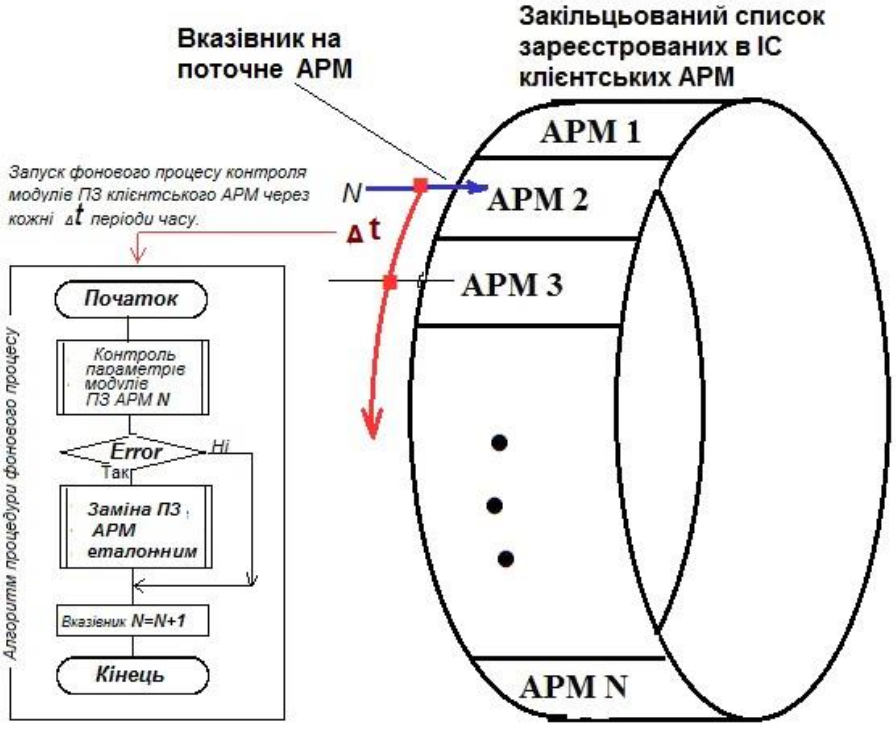


Рис. 3.11 – Схема та алгоритм контролю актуальності ПЗ клієнтських АРМ

Його задача – здійснення із заданим періодом Δt перевірку параметрів модулів ПЗ клієнтських АРМ з еталонними, що зберігаються в репозитарії служби. Якщо в ході перевірки модуль не буде знайдено в місці свого призначення, або його параметри будуть відрізнятись від еталонних, то він буде визнаний як неактуальний. Це призведе до запуску процедури, яка виконає заміщення програмного модуля еталонним із репозитарію модулів. При цьому причина пошкодження або знищення модуля неважливі – основна мета відновлення функціональності АРМ і, таким чином, недопущення блокування доступу до інформації, її втрати, спотворення або пошкодження, при функціонуванні неактуального або пошкодженого ПЗ АРМ з можливим відхиленням від заданих алгоритмів.

Несанкціонований доступ до даних інформаційної системи можна отримати шляхом фільтрації та наступного аналізу її мережевого трафіка, у випадку наявності фізичного доступу до інформаційних каналів системи. Для унеможливлення або утруднення доступу до даних таким шляхом, застосовано криптографічний захист інформації.

Крім того, інформаційні канали окремих сегментів комп'ютерної мережі, наприклад, її ядра, до якого входять серверна група та основні клієнтські місця, виконані таким чином, що унеможливлюють фізичний доступ до них зловмисників (рис. 3.9). В таких сегментах можна вести обмін даними інформаційної системи між її складовими без втрати часу на криптографічний захист, що дозволяє отримати максимальну продуктивність роботи системи.

Наступним шляхом отримання несанкціонованого доступу до даних інформаційної системи може бути використання несанкціонованого підключення до інформаційної системи. Для цього зловмисник може спробувати скористатись як штатним програмним забезпеченням так і власноруч розробленим. Щоб перекрити такий шлях доступу до даних інформаційної системи, ведеться реєстрація всіх екземплярів клієнтського програмного забезпечення та їх параметрів запуску (IP-адреса комп'ютерної системи, дискові шляхи, імена файлів програм і т. і.). Крім цього

використовується спеціальний алгоритм їх запуску, який передбачає двофакторну перевірку легітимності екземпляра програми. Схему двофакторної перевірки легітимності ПЗ АРМ зображено на рис. 3.12. Його особливість в тому, що оператор

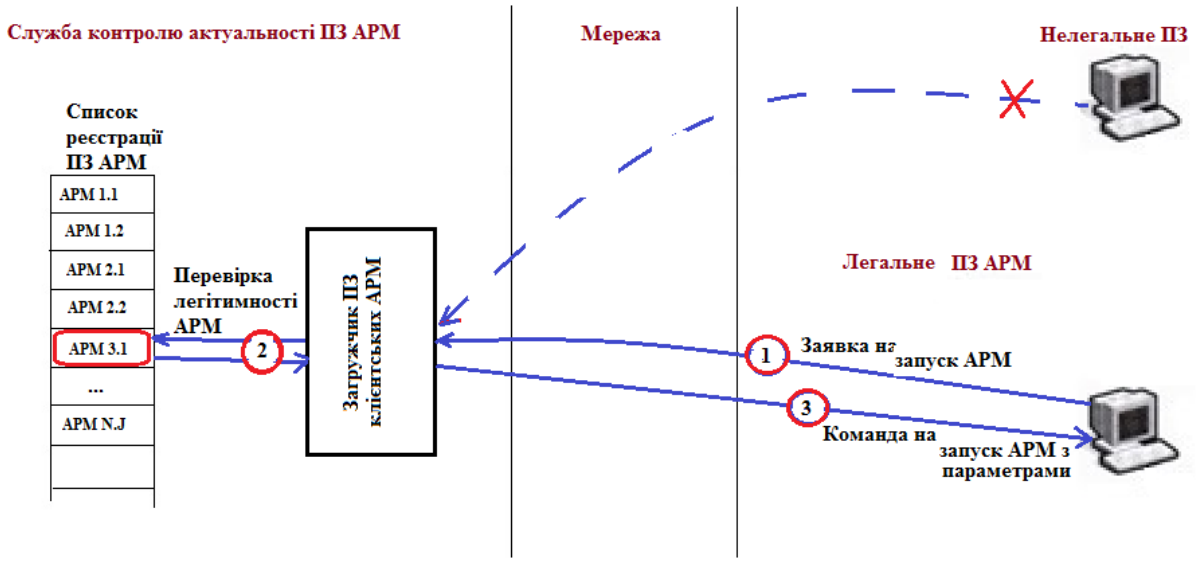


Рис. 3.12 – Схема двофакторної перевірки легітимності ПЗ АРМ

АРМ сам не може запустити програмну систему свого АРМ, через те, що йому невідомі параметри запуску та з'єднання, окрім тих, що ідентифікують його самого. Він лише подає заявку на запуск програмі загружчику, яка входить до складу служби контролю актуальності ПЗ АРМ (позиція 1 рис. 3.12). Завантажувальник отримавши заявку від деякого АРМ перевіряє наявність його реєстрації в ІС (позиція 2 рис. 3.12). Якщо реєстрація підтверджується, то виконує віддалений запуск ПЗ клієнтського АРМ на комп'ютері, де згідно реєстрації має функціонувати АРМ, що подало заявку на запуск і тільки з тими параметрами, що зберігаються в репозитарії ПЗ служби контролю актуальності клієнтського ПЗ (позиція 3 рис. 3.12). В процесі запуску оператор, що подав заявку на запуск, вводить свої реєстраційні дані, ідентифікує себе в ІС, проходячи автентифікацію. Таким чином, унеможливлюється підключення незареєстрованих екземплярів програм та гарантується доступ до інформації тільки

легальним користувачам. Ця робота покладена на службу контролю актуальності програмного забезпечення клієнтських робочих місць.

Питання контрольованого доступу до ПЗ клієнтських АРМ є важливим в переліку заходів забезпечення інформаційної безпеки - це самий простий шлях для зловмисника подолати систему захисту ІС. Тому, саме для цієї ланки системи захисту інформації є важливим дотримання персоналом АРМ організаційно-правових заходів безпеки.

Для отримання доступу до ПЗ є два шляхи доступу до клієнтського АРМ - зовнішній та внутрішній. Зовнішній шлях полягає в отримані віддаленого контролю над комп'ютером, на якому встановлене ПЗ клієнтського АРМ ІС. Оскільки ПЗ АРМ включає в себе алгоритми ідентифікації легітимності користувача, то спроба його віддаленого запуску не буде досить простою. Особливо, коли кількість спроб підключення обмежена і отримати пароль шляхом перебору є практично неможливим.

Зловмисник може діяти не тільки зовні, але і зсередини системи, якщо йому вдасться подолати організаційно-правові заходи безпеки. З метою не допуску такого розвитку ситуації, коли зловмисник отримає доступ до легального ПЗ клієнтського робочого місця, в нього включена функція контролю за активністю його оператора.

Вона реалізована як фоновий процес, який циклічно контролює активність оператора відповідно до алгоритму, наведеному на рис. Д.7 (Додатку Д). Як видно з нього, фоновий процес взаємодіє із будь якою функцією ПЗ, запущеною оператором АРМ.

У випадку відсутності активності оператора АРМ на встановленому проміжку часу, фоновий процес запустить процедуру завершення роботи ПЗ клієнтського робочого місця.

Кожен клієнт ІС при реєстрації його, як користувача ІС, не отримує ніяких прав на інформаційні ресурси системи. Всі необхідні права, для управління даними в такій системі надаються ролі (прообразу посади). Клієнту ж надається право виконувати певну роль в інформаційній системі. Цей підхід дозволяє уникнути проблеми

залишкових прав, коли функції клієнта в системі з часом змінюються, а деякі права доступу до ресурсів інформаційної системи залишаються. Запропонований підхід передбачає зміну ролі користувача в ІС, гарантуючи при цьому, що права по старій ролі будуть ним втрачені, а по новій набуті. Таким чином реалізується перший рівень управління доступом до ресурсів ІС в запропонованій технології реалізації спеціалізованих ІС. Як правило, він реалізується з використанням штатних засобів системи адміністрування ІС.

З метою покращення гнучкості управління правами на інформаційні ресурси, та з метою зменшення часу реакції на різні деструктивні прояви в ІС, дана технологія побудови ІС передбачає введення ще одного рівня управління правами доступу до її інформаційних ресурсів.

На цьому рівні розмежування прав виконується на рівні клієнтського робочого місця. Це дозволяє оперативно реагувати на події в ІС шляхом заборони роботи деяких робочих місць, або переключення їх в режим, що не передбачає внесення змін в дані. При цьому робота інших клієнтських робочих місць ніяк не порушується.

Реалізація другого рівня управління правами доступу до ресурсів ІС стала можливою, завдяки наявності реєстрації всіх екземплярів клієнтського програмного забезпечення в її БД, як це описано вище. Його особливістю є те, що він становить базову фундаментальну частину ПЗ клієнтського робочого місця, яка є типовою для всіх робочих місць в запропонованій технології. Алгоритм управління правами на рівні робочого місця допускає зміну прав зі сторони адміністратора системи у будь який час, а сама ця зміна буде врахована при спробі виконати наступну операцію над даними.

Дворівнева система управління доступом до даних спеціалізованої ІС дозволяє значно зменшити ризик випадкового спотворення даних зі сторони операторів клієнтських робочих місць, через випадкове набуття прав. Але залишається можливість їх спотворення в процесі виконання легальних операцій з даними згідно своєї ролі у системі.

ІС в плані інформаційної безпеки повинна відповідати трьом безпековим принципам: конфіденційності, цілісності та доступності інформації. При вирішенні задачі захисту інформації від несанкціонованого копіювання необхідно забезпечити виконання одночасно двом із них – конфіденційності та доступності.

Аналіз місць вразливості ІС щодо можливості несанкціонованого копіювання показав, що це в принципі неможливо з клієнтського робочого місця, оскільки самі АРМи побудовані з урахуванням правил моделі безпеки Кларка-Вільсона, згідно яких оператор (суб'єкт) не має прямого доступу до даних і відповідно, навіть у випадку коли зловмисник знаходиться всередині системи, він гарантовано позбавлений можливості несанкціонованого копіювання інформації.

І тільки в випадку отримання прямого доступу зловмисником до БД або її копій це стане можливим. Оскільки БД та її копії розміщуються на комп'ютерах серверної групи, відносно яких діють самі жорсткі заходи захисту інформації від організаційно-правових до фізичних, то така ситуація є маловірогідною, незважаючи на територіальне рознесення комп'ютерів цієї групи.

Щоб максимально зменшити ймовірність спотворення інформації в результаті некваліфікованих дій клієнтів інформаційної системи, започаткована наступна організація їх роботи.

Щоб максимально унеможливити її спотворення в процесі функціональної діяльності, технологія розробки ПЗ клієнтських робочих місць, де виконуються операції редагування даних передбачає включення до його складу типових редакторів даних, які включають в себе модулі контролю, які перешкоджають внесенню в БД неузгоджених даних. До його складу включені процедури програмного контролю правильності введених даних, їх несуперечливості раніше введеним. Такий підхід підвищує ступінь актуальності даних, поміщених в базу даних, відхиляє технічні помилки. Схема організації роботи такої структури показана на рис. Д.8 (Додатку Д). Її особливістю є те, що маніпулювання критичними даними з точки зору їх цілісності, зі сторони оператора клієнтського АРМ, знаходиться під програмним контролем.

Як видно з рис. Д.8 (Додатку Д), обрані для редагування оператором елементи даних поміщаються в спеціальний буфер, в якому і здійснюється їх редагування. Після закінчення цієї операції оператор робить спробу зберегти зміни. Але перед їх відправкою на сервер БД нові значення елементів даних перевіряються на непротиріччя та цілісність. Якщо перевірка дасть позитивний результат, то дані відправляються на сервер БД, інакше в їх збережені буде відмовлено. Такий підхід дозволяє в великій мірі уникнути випадкового, ненавмисного спотворення інформації ІС зі сторони операторів клієнтських робочих місць.

Робота підсистеми транзакцій побудована з урахуванням правил абстрактної моделі захисту інформації Кларка-Вільсона (Clark-Wilson) яка біла оприлюднена 1987 році. Вважається однією з найдосконаліших у відношенні підтримки цілісності інформаційних систем. Дана модель заснована на повсюдному використанні транзакцій і ретельному оформленні прав доступу суб'єктів до об'єктів.

Особливістю цієї моделі є розповсюдження системи захисту на третю сторону – на програму (транзакцію). Модель побудована на тристоронніх відносинах суб'єкт-програма-об'єкт (де програма взаємозамінна з транзакцією). В рамках цих відносин суб'єкт немає прямого доступу до об'єкта. Доступ до об'єкта можливий лише через програму (транзакцію).

Крім того, в моделі Кларка-Вільсона транзакції вперше були побудовані за методом верифікації, тобто ідентифікація суб'єкта проводилася не тільки перед виконанням команди від нього, але і повторно після виконання. Це дозволило зняти проблему підміни учасника в момент між його ідентифікацією і власне командою.

Схема роботи підсистеми транзакцій показана на рис. 3.13. Будь-які маніпуляції над інформацією в ІС охоплюються транзакцією. Її властивості гарантують, що в ході виконання транзакції вона або гарантовано приведе БД до нового несуперечливого стану у випадку її підтвердження, або поверне її до попереднього, з якого вона почалась у випадку її відкату.

Ця схема є складовою частиною типового елемента маніпулювання даними ІС показаного на Д.8 (Додатку Д), який працює відповідно до правила моделі Кларка-Вільсона, яке, в свою чергу, проголошує принцип, що тільки процедура (програма) може змінювати інформацію в БД.

Така схема роботи клієнтських АРМ гарантує забезпечення цілісності та несуперечливості інформації та її захист в цілому.

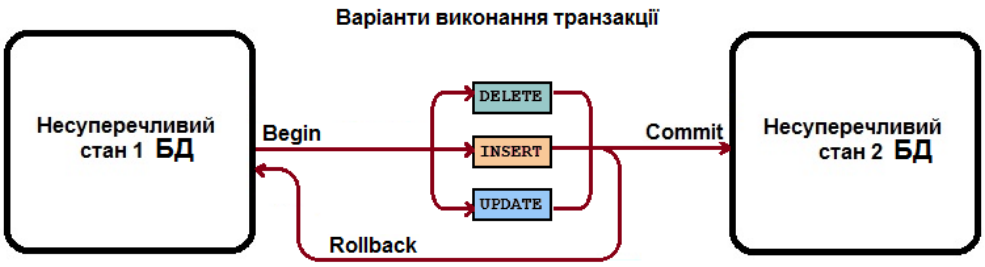


Рис. 3.13 – Схема роботи підсистеми транзакцій як частини системи захисту інформації

Одним із важливих вирішуваних завдань в системі забезпечення захисту інформації під час експлуатації ІС є організація роботи підсистеми резервного копіювання, адже навіть у найнадійнішій системі існує ризик втрати інформації, а особливо при її роботі в умовах впливів ЗПЗ. Для систем, що працюють в таких жорстких умовах наявність механізму швидкого відновлення втрачених даних є необхідністю.

Ключовими параметрами для планування побудови підсистеми резервного копіювання є:

– час, на протязі якого система має бути відновлена;
– часовий період, втрата даних на протязі якого є прийнятною.

Їх значення вибираються залежно від специфіки роботи ІС та критичності інформаційної системи і є компромісом між витратами на її функціонування та допустимими втратами інформації.

Загальний устрій системи резервного копіювання показаний на рис. 3.14. Як видно із схеми, сервером резервних копій слугує фізично інший комп'ютер, який територіально розмежований із основним сервером керування БД ІС.

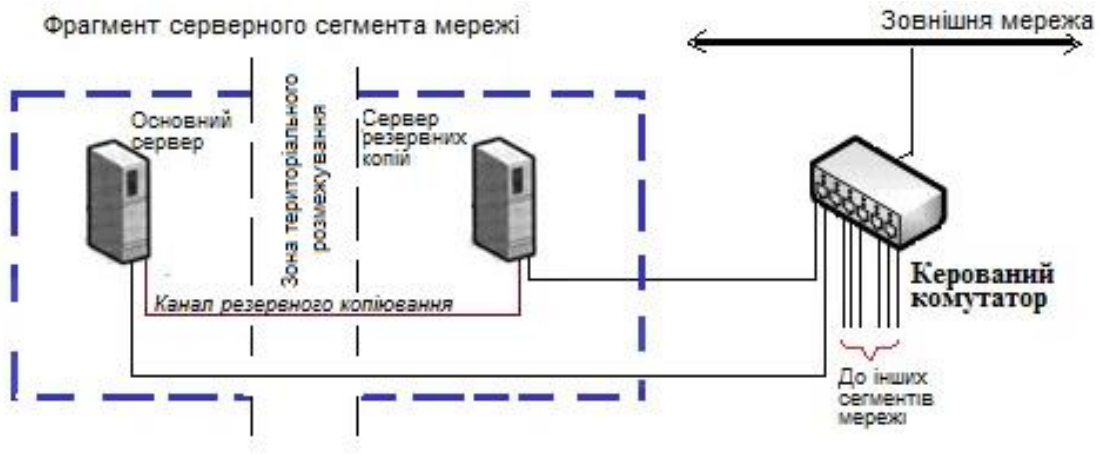


Рис. 3.14 – Схема фрагмента комп'ютерної мережі підсистеми резервного копіювання

Такий вибір архітектури підсистеми резервного копіювання дозволив отримати стійку до різноманітних впливів, ІС в цілому. Ця схема практично унеможливлює одночасний вихід з ладу обох серверів і цим дає високі гарантії, що до збереження інформації, яка обробляється в системі.

Особливістю даної схеми є наявність окремого каналу резервного копіювання. Таке рішення дозволило позбутися вузького місця більшості систем резервного копіювання, пов'язаного із пропускною здатністю комп'ютерної мережі, забезпечивши максимальні швидкості передачі даних між серверами, як в режимі копіювання так і в режимі відновлення даних.

Ще однією особливістю організації роботи підсистеми резервного копіювання є розміщення бібліотеки резервних копій на комп'ютері, який в ІС виконує функцію

резервного сервера, що дозволяє, практично в режимі реального часу, перевірити працездатність отриманої копії БД.

Сам алгоритм резервного копіювання виконується в фоновому процесі відповідно до плану резервування, тому не потребує зупинки ІС, зрівнюючи дану ІС по цьому показнику із системами "високої доступності".

План резервування включає в себе процеси створення добових резервних копій, які додатково архівуються для зменшення займаного місця при їх зберіганні на диску. Також, створюються почасові, або з якимось іншим прийнятним періодом копії, які на протязі робочого дня і слугують джерелом даних резервного сервера. Ці копії для сервера створюються в режимі без використання компресії, але очисткою копії від видалених записів. Це дозволяє завжди мати готову для використання резервну копію БД, на яку перемкнеться резервний сервер у випадку аварії основного.

Для добових копій період зберігання встановлено три місяці, а для часових - три доби, після чого вони заміщаються більш актуальнішими, по мірі створення нових копій. Це дозволяє мати сталий об’єм бібліотеки резервних копій, уникаючи ситуації переповнення накопичувача.

Принцип конфіденційності, якому повинна відповідати система інформаційної безпеки, диктує вимогу забезпечення можливості отримання інформації лише легітимними користувачами.

Для безумовного виконання цього принципу в ІС для всіх АРМ було запроваджено систему криптографічного захисту, яка унеможливлює контроль мережевого трафіка обміну даними віддалених АРМ ІС із сервером БД. З цією метою використовується асиметрична криптосистема із парою ключів - відкритим ключем шифрування та закритим ключем для дешифрування (рис. 3.15).

Генерація таких пар ключів залежить від криптографічних алгоритмів, які ґрунтуються на односторонніх функціях. Для забезпечення безпеки передачі даних вимагається збереження таємниці закритого ключа, відкритий же ключ може відкрито розповсюджуватись без шкоди для безпеки.

Топологія комп'ютерної мережі, з використанням криптосистеми для передачі даних показана на рис. 3.16.

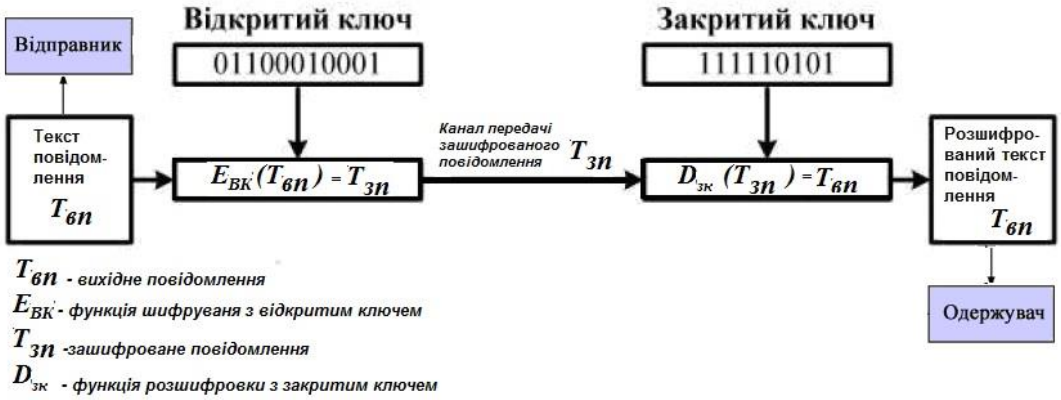


Рис. 3.15 – Асиметрична криптосистема з відкритим ключем шифрування

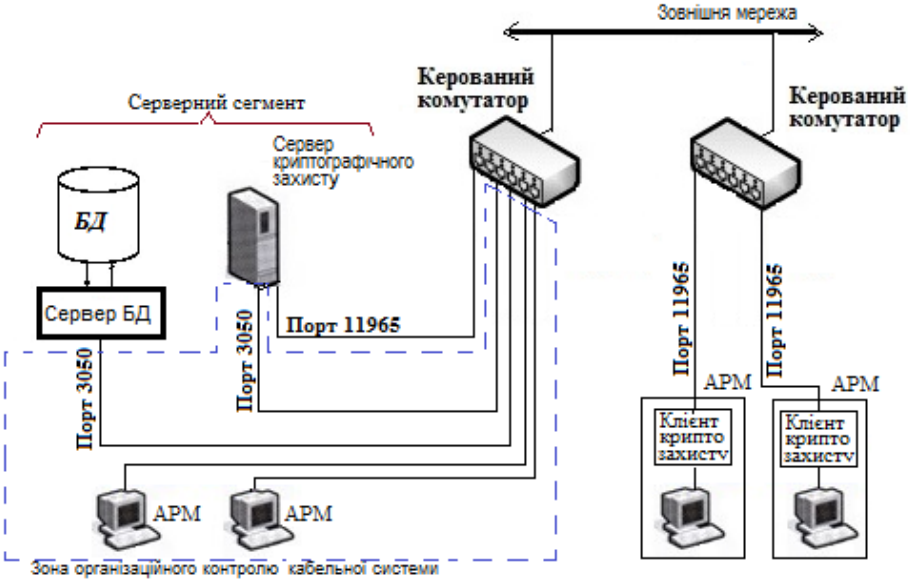


Рис. 3.16 – Схема топології комп’ютерної мережі ІС із забезпеченням криптографічного захисту інформації

Зображена на рис. 3.16 схема комп'ютерної мережі включає до свого складу сервер криптографічного захисту. На нього покладається завдання створення шифрованих тунелів обміну даними між сервером БД та віддаленими клієнтськими

АРМ ІС, кабельні та ефірні канали зв'язку з якими фізично не контролюються адміністраторами системи. Ті АРМ, що фізично знаходяться в зоні організаційного контролю кабельної системи, при обміні даними, засоби криптографічного захисту інформації не використовують, що збільшує продуктивність їх роботи і системи в цілому.

Клієнтські АРМ, які знаходяться поза цією зоною, обмін даними з сервером БД ведуть по захищеному каналу, загальна структура якого наведена на рис. 3.17.

Ініціатором встановлення з'єднання виступає віддалене АРМ. Сервер криптографічного захисту налаштований таким чином, що він постійно слухає свій порт, в даному випадку це порт 11965. При надходженні запиту на встановлення з'єднання від одного із віддалених АРМ ІС, сервер перевіряє його легітимність відповідно до параметрів налаштування таблиці зв'язків.

Якщо легітимність клієнта підтверджується, то сервер встановлює захищене з'єднання у вигляді шифрованого тунелю (рис. 3.17) із стискуванням даних, при їх передачі з використанням TCP і/або UDP протоколів. Застосування функції стискування даних дозволило підвищити пропускну здатність мережі.

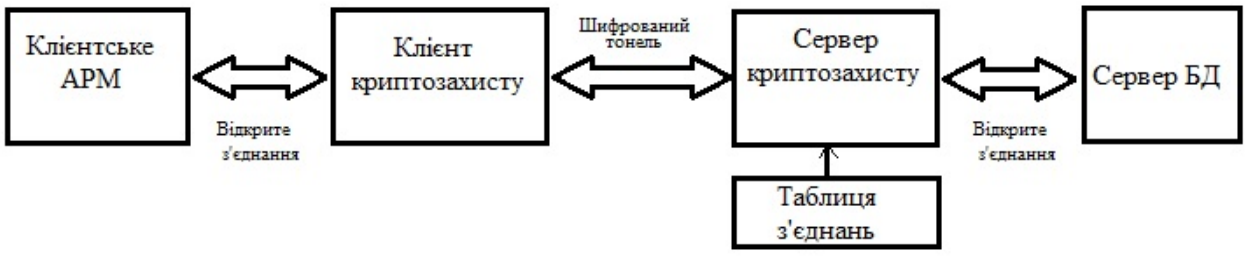


Рис. 3.17 – Загальна схема шифрованого каналу передачі даних сервером БД та клієнтським АРМ ІС

При цьому знімається загроза зі сторони 'sniffer'-ів (переглядачів пакетів). Шифрування пакетів йде з самого початку, тому дані з'єднання (username та інші)

разом з шляхом та ім'ям бази даних буде зашифровано, що значно ускладнює роботу потенційного зловмисника.

Як відомо, для успішної атаки на комп'ютерну мережу і ІС, що на неї базується, зловмиснику спочатку треба її вивчити, ведучи розвідку. Повне шифрування трафіка між клієнтським АРМ та сервером БД робить її практично не виконуваною задачею за прийнятний період часу, що в свою чергу не дозволить отримати службову інформацію, яка б дозволила почати атаку.

Таким чином, метод забезпечення захисту інформації в спеціалізованих ІТ згідно двофакторної перевірки легальності ПЗ користувача містить наступні кроки.

1. Створити базу еталонного ПЗ клієнтських АРМ.
2. Створити базу еталонних параметрів модулів ПЗ клієнтських АРМ та реєстраційний список фізичного розміщення АРМ з параметрами запуску та прив'язкою до IP-адреси.
3. На комп'ютер встановити ПЗ АРМ без параметрів запуску.
4. При спробі запуску ПЗ АРМ з клієнтської станції подати запит завантажувальнику, який знаходиться на сервері, на запуск з кодом реєстрації даного АРМ.
5. Програма завантажувальника перевіряє легітимність даної заявки відповідно до реєстраційного списку і, якщо дані запиту збігаються з очікуваними, то виконується віддалений запуск ПЗ АРМ на клієнтській станції. Якщо ні – то запуск не відбудеться, а сам факт спроби запуску буде зафіксовано в лог-файлі подій в ІС (рис. 3.23, подія 210551). Якщо зловмисник подасть заявку з правильними параметрами, змінивши адресу відправника в пакеті, то запуск ПЗ АРМ відбудеться, але не на його комп'ютері, а на штатному, відповідно до списку реєстрації, який знаходиться під контролем легітимного користувача.

Для перевірки розробленого методу забезпечення захисту інформації в спеціалізованих ІТ проведено два експерименти, які демонструють ефективність роботи засобів захисту інформації в ІС з реалізованим в них методом. Перший

експеримент полягав у перевірці реакції системи на відсутність активності оператора впродовж часу, що перевищує встановлений. Другий – перевірці реакції системи на спробу під'єднання до ІС нелегального ПЗ.

Для проведення першого експерименту було використано клієнтське АРМ №50. Змодельовано ситуацію відсутності активності оператора після його запуску. Результати експерименту наведені в табл. 3.1 – подія 210547 та 210548 та в фрагменті log-файлу (події 210547 та 210548), які зображені на рис. 3.18.

Таблиця 3.1

Результати експериментів з контролю роботи засобів захисту інформації в ІС

№ події в ІС	Розшифровка змісту події
210547	а) старт АРМ №50 (IP 192.168.168.10) в 8:12:05 з підключенням до сервера БД з IP адресою 192.168.168.1. б) припинена робота АРМ №50 через перевищення встановленого часу (15хв.) його знаходження в неактивному стані в 8:40:01 – код помилки 1000.
210547	о 8:42:57 АРМ №50 повторно запущено і працювало в штатному режимі до 9:21:08.
210551	В 8:52:02 фоновий процес контролю ПЗ АРМ зафіксував заявку на запуск від АРМ №103 з IP адреси 192.168.168.201. При перевірці параметрів заявки виявилось, що АРМ 103 в дійсності зв'язаний з IP адресою 192.168.168.4, а не 192.168.168.201. В запуску було відмовлено, а сам факт спроби нелегального запуску (або невірно налаштованого ПЗ деякого АРМ) було зафіксовано в log-файлі подій в ІС з кодом помилки 3060.

File0023.log [vk.com]

210545	149	23.09.2021 8:05:10				192.168.168.1	192.168.168.16
210546	2	23.09.2021 8:09:25	23.09.2021 17:01:26			192.168.168.1	192.168.168.4
210547	50	23.09.2021 8:12:05	23.09.2021 8:40:01	1000	Stop ..... Timed Out	192.168.168.1	192.168.168.10
210548	50	23.09.2021 8:42:57	23.09.2021 9:21:08			192.168.168.1	192.168.168.10
210549	3	23.09.2021 8:49:51	23.09.2021 13:00:55			192.168.168.1	192.168.168.5
210550	106	23.09.2021 8:51:07	23.09.2021 10:02:38			192.168.168.1	192.168.168.12
210551	103	23.09.2021 8:52:02		3060	Unknown program... Startup denied	192.168.168.1	192.168.168.201
210552	19	23.09.2021 8:55:08	23.09.2021 16:53:51			192.168.168.1	192.168.168.8
210553	20	23.09.2021 9:07:00	23.09.2021 16:40:35			192.168.168.1	192.168.168.5
210554	13	23.09.2021 9:08:01	23.09.2021 16:15:24			192.168.168.1	192.168.168.10
210555	3	23.09.2021 9:09:21	23.09.2021 17:01:44			192.168.168.1	192.168.168.5
210556	76	23.09.2021 9:09:34	23.09.2021 16:28:07			192.168.168.1	192.168.168.14

Рис.3.18 – Фрагмент Log-файлу подій в ІС пов’язаних з роботою засобів захисту інформації

Аналіз інтеграції в апаратні та програмні компоненти ІС засобів захисту інформації ІС, які у своїй сукупності утворюють комплексну підсистему захисту інформації, дозволив зробити висновки:

– сегментування мережі, виокремлення серверної частини в окрему під мережу з можливістю гнучкого налаштування політик безпеки для кожного сегмента підвищує стійкість ІС до атак ЗПЗ;

– застосування криптографічних засобів захисту гарантує відсутність несанкціонованого доступу до інформації в фізично неконтрольованих каналах передачі інформації, а також зменшує можливості ведення розвідки комп’ютерної мережі ІС;

– застосований метод запуску клієнтських АРМ з двофакторною автентифікацією ПЗ та користувача унеможливлює підключення нелегальних програм та незареєстрованих копій штатного ПЗ;

– нетривіальні процедури контролю за операціями маніпулювання даними, виконання всіх операцій з даними під управлінням транзакцій гарантують цілісність

та узгодженість інформації в ІС, а застосування контролю активності АРМ, гнучка дворівнева система управління наданням прав доступу до ресурсів ІС додатково зменшує вірогідність несанкціонованого доступу до інформації;

– автоматизована система резервного копіювання з територіальним розмежуванням місць зберігання копій та перевіркою їх працездатності унеможливлює не відновлювану втрату інформації.

Проведені експерименти підтверджують працездатність засобів захисту інформації ІС та зроблені висновки. Реакція ІС на впливи, змодельовані в обох експериментах була очікуваною і в межах встановлених часових меж.

Таким чином, перелічені заходи забезпечення захисту інформації ІС в поєднані із організаційно-правовими заходами, використані як єдиний комплекс, дозволяють отримати метод, застосування якого при розробці спеціалізованої ІС, гарантує високий рівень захищеності ресурсів від деструктивних впливів. Розроблений метод [76, 132] забезпечення захисту інформації спеціалізованих ІТ полягає в поєднані із організаційними заходами інтегрованих в ІТ механізмів сегментування мережі, криптографічного захисту, двофакторної автентифікації програмного забезпечення, хибних об’єктів атаки, резервного копіювання з територіальним розмежуванням місць зберігання копій, що дозволяє створювати засоби з покращеним рівнем захищеності інформації в умовах впливів ЗПЗ та комп’ютерних атак.

3.5. Висновки до третього розділу

Побудована узагальнена модель впливів зловмисного ПЗ на об’єкти та процеси комп’ютерних систем, яка дозволяє оцінити загрози для комп’ютерної системи зі сторони різноманітного зловмисного ПЗ, способи його проникнення в комп’ютерну систему, його прояви та масштаб.

Розроблено метод забезпечення живучості ІТ в умовах впливів зловмисного ПЗ з використанням параметричного контролю актуальності модулів ПЗ клієнтських АРМ та їх маскуванням.

Розроблений підхід до визначення ефективності ІТ на основі врахування кількісних величин, які характеризують живучість, та може бути розширений для врахування інших характеристичних величин. Для забезпечення живучості ІТ розроблено систему заходів в результаті виконання яких отримано ІТ вузькоспеціалізованого використання для різних сфер застосування, де супроводжуванні процеси відносяться до ірреального або нереального часу із досить високими параметрами живучості і, в той же час, прийнятним рівним фінансових витрат на її експлуатацію.

Розроблено метод забезпечення захисту інформації спеціалізованих ІТ, який полягає в поєднані із організаційними заходами інтегрованих в ІТ механізмів сегментування мережі, криптографічного захисту, двофакторної автентифікації програмного забезпечення, хибних об’єктів атаки, резервного копіювання з територіальним розмежуванням місць зберігання копій, що дозволяє створювати засоби з покращеним рівнем захищеності інформації в умовах впливів ЗПЗ та комп’ютерних атак.

Основні результати розділу опубліковані у працях [74, 76, 77, 129, 130, 132].

РОЗДІЛ 4.
МЕТОД ТА ЗАСОБИ ЗАБЕЗПЕЧЕННЯ ВІДМОВОСТІЙКОСТІ, ЖИВУЧОСТІ ТА ЗАХИСТУ ІНФОРМАЦІЇ СПЕЦІАЛІЗОВАНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В УМОВАХ ВПЛИВІВ ЗЛОВМИСНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1. Метод забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих ІТ

Забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих ІТ в умовах впливів ЗПЗ та комп'ютерних атак згідно розроблених методів дає змогу покращити стійкість щодо впливів в кожному окремому випадку. Але частина кроків трьох різних розроблених методів є збіжною, тому доцільним є поєднання трьох розроблених методів в один метод згідно спільних кроків та станів системи, в якій він буде реалізований. Тоді, в системі буде підсистема, яка реалізовуватиме метод забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих ІТ, що поєднуватиме всі три розроблені методи.

На рис. 4.1 зображено модель забезпечення відмовостійкості клієнтського АРМ. Тут вершини графа 1-4 представляють собою внутрішні впливи на програмне забезпечення всіх рівнів та апаратну платформу ІС, які направлені на зменшення відмовостійкості ІС. Вершина 1 фіксує ситуацію, коли через надвелику складність сучасних програмних систем, ПЗ клієнтського АРМ (рис. 4.1, вершина 5), як і ОС комп'ютерної станції (рис. 4.1, вершина 6), може самопошкодитись, роблячи шлях 5 - 9 на рис. 4.1 недоступним.

Служба контролю актуальності ПЗ (рис. 4.1, вершина 10) слідкує за станом ПЗ клієнтського АРМ і, у випадку виявлення його пошкодження, замінює його еталонним, відновлюючи його працездатність (рис. 4.1, вершина 9). Таким чином, забезпечується автоматичне відновлення доступності функцій ІС – шлях 5-9 на рис. 4.1 робиться доступним.

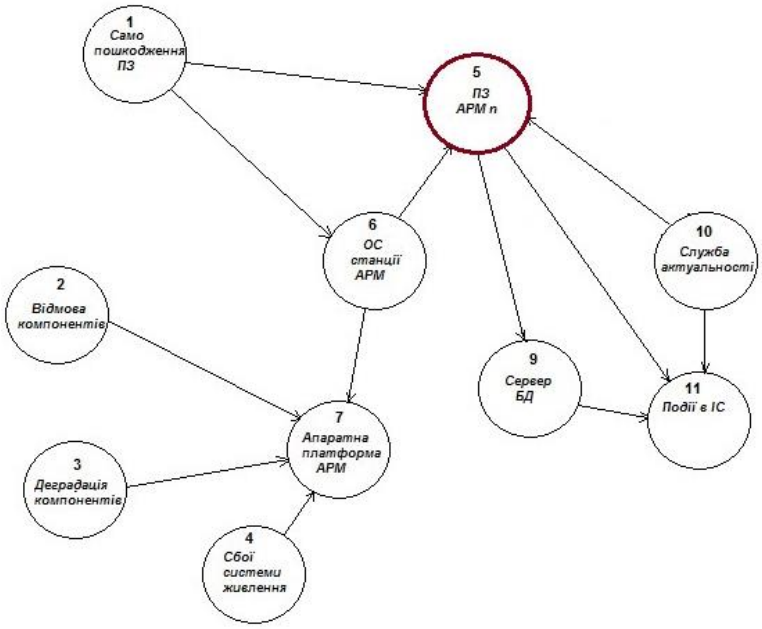


Рис. 4.1 – Графова модель забезпечення відмовостійкості

Недоступність функцій клієнтського АРМ може статись, також, і через вплив зовнішніх факторів, а саме через дії зловмисного ПЗ. Ця ситуація змодельована на рис. 4.2. Зовнішні впливи показані вершинами 1 та 3. Вони враховують різноманітні способи атаки на компоненти ІС. Вершина 1 моделює ситуацію, коли вплив ЗПЗ направлено безпосередньо на ПЗ клієнтського АРМ або на ОС, під управлінням якої воно працює.

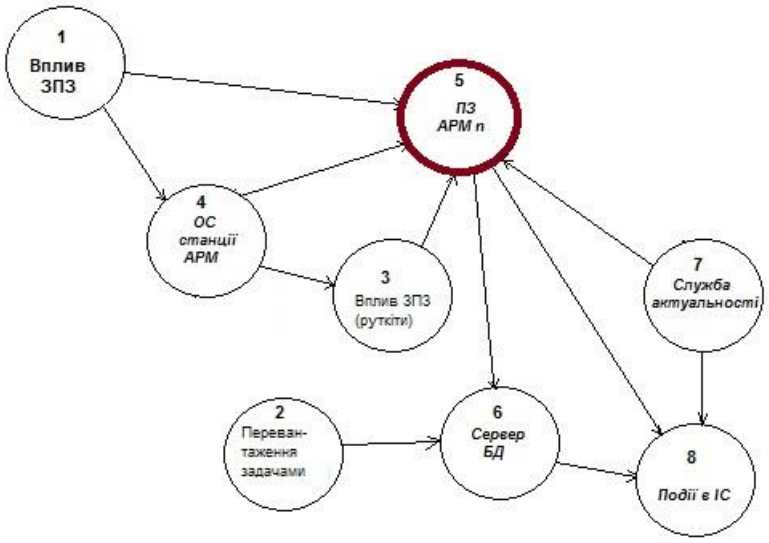


Рис. 4.2 – Графова модель забезпечення живучості

Вершина 3 моделює ситуацію, коли зловмисному ПЗ, в процесі атаки на ОС, вдалось створити замаскований тунель для подальшої атаки на інші компоненти ІС, в тому числі і на ПЗ клієнтського АРМ. Вершина 2 моделює ситуацію DOS-атаки на серверну частину ІС. Для протидії цим впливам, а саме їх атакам на найменш захищені її ланки - клієнтські станції, використовуються можливості розширеного функціоналу служби контролю актуальності ПЗ (вершина 7, рис.4.2), яка у випадку виявлення пошкодження ПЗ деякого АРМ (вершина 5, рис.4.2), виконує його заміну еталонним, відновлюючи шлях 5-6 графової моделі рис. 4.2, і таким чином відновлюючи доступність функцій ІС.

Зображена на рис. 4.3 модель, демонструє ситуації, в яких може перебувати система забезпечення захисту інформації ІС.

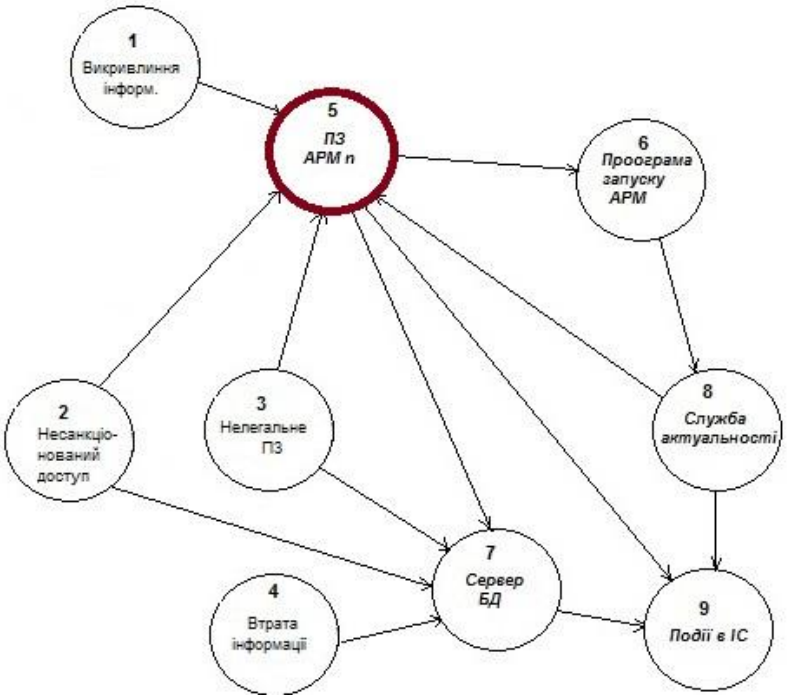


Рис. 4.3 – Графова модель забезпечення захисту інформації

В неї включені основні чинники, які різними способами загрожують інформації, що обробляється в ІС. Це викривлення, несанкціонований доступ та втрата інформації (вершини графа 1-4, рис. 4.3). Їх дія направлена проти клієнтських АРМ та сервера БД (вершини 5 та 7, рис. 4.3 відповідно). Особливо вразливою частиною ІС є

клієнтські АРМ, як і є найменш захищеними елементами системи. З метою посилення їх захисту в ІС використовується спеціальний механізм запуску ПЗ АРМ, в основі якого двох факторна перевірка легальності ПЗ, що робить спробу підключення до БД сервера. Відповідають за його підтримку програма запуску АРМ (вершина 6, рис. 4.3) та служба контролю актуальності ПЗ (вершина 8, рис.4.3), яка перевіряє відповідність ПЗ, яке намагається виконати з'єднання з БД параметрам, що зберігаються в реєстрі цієї служби. При виявлені неспівпадання в параметрах в з'єднані відмовляється, а сам факт несанкціонованого під'єднання фіксується в лог-файлі подій в системі (вершина 9 на графі модулі).

Аналіз впливів, направлених на ІС і приведених на рис. 4.1, 4.2 та 4.3 показав, що викликані вони різними причинами, але направлені практично на одні і ті ж компоненти системи. Як видно з представленої на рис. 4.4 результуючої моделі, такими компонентами є клієнтські АРМ ІС та сервер БД (вершини 5 та 7, рис. 4.4 відповідно).

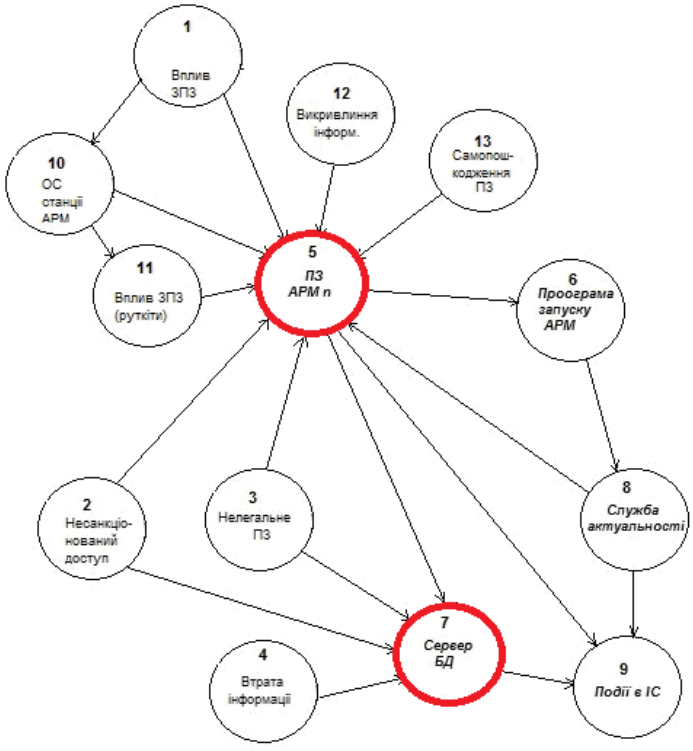


Рис. 4.4 – Результуюча графова модель відмовостійкої ІС

При цьому ПЗ клієнтських АРМ є найбільш вразливим від дії всіх приведених впливів. Для підвищення його стійкості проти всіх видів збурень в ІС включені засоби автоматичного відновлення працездатності ПЗ АРМ. І, як видно з приведених моделей (рис. 4.1, 4.2 та 4.3), протидію різним типам впливів виконують одні і ті ж компоненти. Ними є служба контролю актуальності ПЗ (вершина 8 рис. 4.4) та програма, яка контролює запуск ПЗ АРМ (вершина 6 рис. 4.4). Алгоритм її роботи побудований таким чином, що для неї не є важливим, що стало причиною порушення роботи ПЗ АРМ, головним є можливість виявлення відхилення його параметрів від очікуваних та його заміна еталонним.

Така організація протидії зловмисним впливам є досить ефективною з точки зору витрат на неї, що в багатьох проєктах може бути вирішальним аргументом.

Згідно представлених графових моделей забезпечення стійкості ІС до різноманітних негативних впливів розроблено моделі методів забезпечення відмовостійкості, живучості ПЗ клієнтських АРМ (рис. 4.5) та захисту інформації (рис. 4.6).

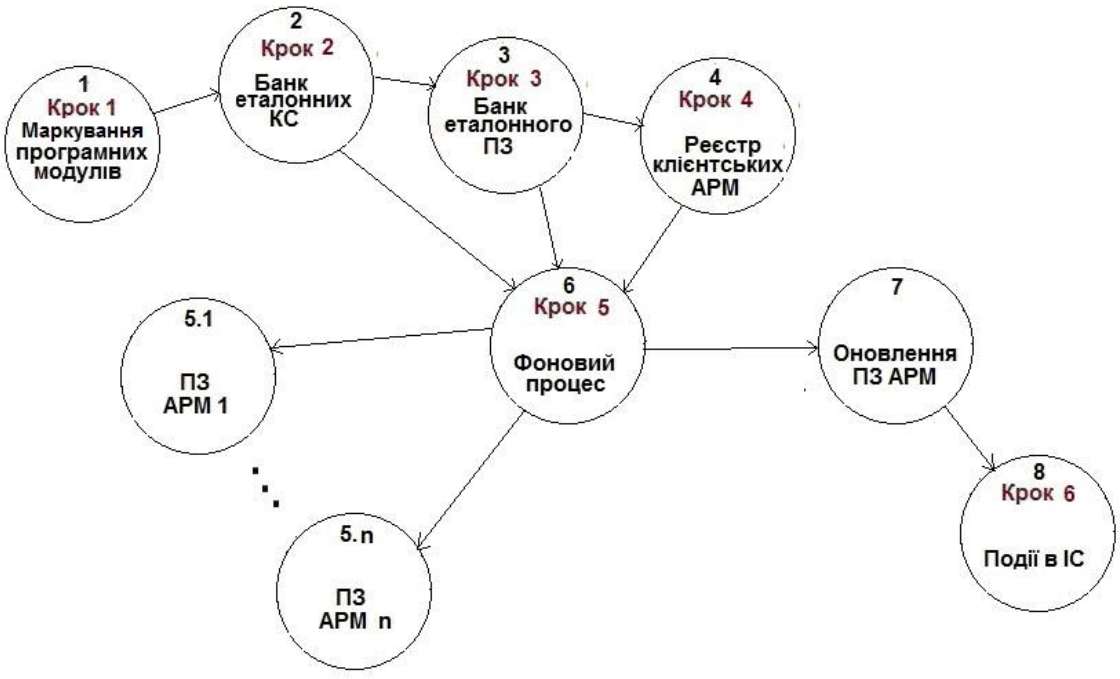


Рис.4.5 – Графова модель методу забезпечення живучості клієнтських АРМ ІС

Зображена на рис. 4.5 модель реалізує основні кроки методу забезпечення живучості ПЗ клієнтських АРМ, які включають такі етапи як маркування програмних модулів (крок 1 – вершина 1, рис.4.5), які входять до складу файлів без сталої контрольної суми, підрахунок контрольних сум кожного модуля (крок 2 – вершина 2, рис.4.5), формування бази еталонного ПЗ клієнтських АРМ (крок 3 – вершина 3, рис.4.5), підготовка списку значень маркерів та контрольних сум маркованих ними модулів (крок 4 – вершина 4, рис. 4.5). Після виконання підготовчих кроків 1-4 методу забезпечення живучості спеціалізованих ІТ, які моделюються, відповідно, вершинами 1-4 графа рис. 4.5, запускається фоновий процес служби контролю актуальності ПЗ АРМ (вершина 6, рис. 4.5), який є четвертим кроком методу. Якщо фоновим процесом буде виявлено відхилення параметрів деякого АРМ (вершини графа 5.1 – 5.n рис. 4.5) від очікуваних, то буде виконано відновлення його ПЗ (вершина 7, рис.4.5), що потягне за собою виконання ще одного кроку – останнього, згідно якого в лог-файлі подій в ІС буде зафіксовано факт відновлення пошкодженого ПЗ (вершина 8, рис. 4.5). В результаті виконання всіх кроків методу забезпечення живучості ПЗ клієнтських АРМ (шлях по графу моделі 1-2-3-6-7-8 або 1-2-4-6-7-8 рис.4.5) буде відновлено доступність функцій ІС. При цьому слід зазначити, що кроки 3 та 4 можуть виконуватись послідовно, або паралельно, як це зображено на моделі рис.4.5.

Модель процесу забезпечення захисту інформації ІС, в основі якого лежить запуск ПЗ клієнтського АРМ із двофакторною перевіркою показано на рис. 4.6. Основні його кроки змодельовані вершинами представленої графової моделі. В своїй роботі метод охоплює три можливі ситуації. Перша передбачає режим запуску легального ПЗ. Від деякого АРМ (вершина 3.1 - 3.n) поступає заявка на запуск. Програма віддаленого запуску клієнтських програм (крок 4 - вершина 4 рис. 4.6) у взаємодії із фоновим процесом (вершина 8 рис. 4.6) перевіряє достовірність параметрів і, якщо розбіжностей не виявлено, віддалено запускає ПЗ АРМ, що звернулось із заявкою. Друга і третя ситуації пов’язані із спробую підключення до ІС

нелегального ПЗ (вершина 6 рис. 4.6). У випадку, коли заявка на запуск поступає від нелегальної копії ПЗ у запуску буде відмовлено, через відсутність інформації про неї в реєстрі клієнтського ПЗ (вершини 1 та 2 рис. 4.6).

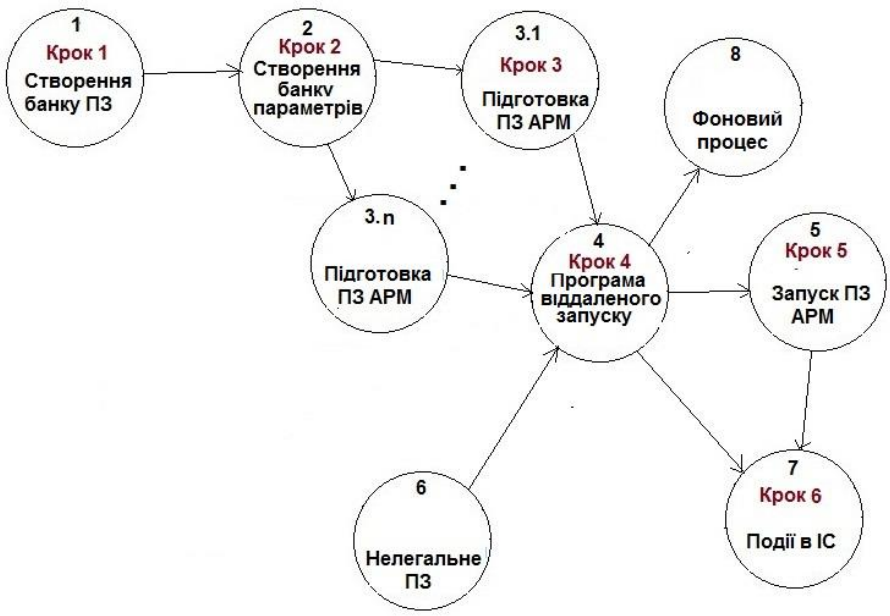


Рис. 4.6 – Графова модель методу забезпечення захисту інформації в ІС

В ситуації, коли нелегальна копія ПЗ подасть заявку із параметрами, що відповідають одному із легальних АРМ ІС, то ПЗ деякого АРМ буде запущено, але не на комп'ютері зловмисника, а на комп'ютері легального оператора, що стане сигналом тривоги.

Як видно з опису всіх моделей, ІС не містить в собі спеціального механізму запуску засобів забезпечення відмостійкості, живучості та захисту інформації. Ним слугують виявленні в ході фонового контролю ПЗ АРМ розбіжності між вирахуваними параметрами та еталонними, що зберігаються в банку служби контролю актуальності ПЗ АРМ, при цьому причини, що їх викликали не встановлюються, а лише ліквідується наслідок їх деструктивних проявів, а сам факт такої події фіксується в лог-файлі для подальшого аналізу. В результаті доступність функцій ІС відновлюється, не зважаючи, що стало причиною їх втрати.

Аналіз моделей, представлених на рис. 4.1 та 4.2 показав, що загалом їх різнять деструктивні впливи, які на них направлені. Всі ці впливи умовно можна поділити на дві групи – до однієї включимо негативні зовнішні впливи, а до другої впливи, що такими не є. Засоби ж протидії залишись тими ж самими. Це пояснює, чому кроки методів забезпечення відмовостійкості та живучості є однаковими – фактично це випливає з означень цих понять. Їх спільна графова модель показана на рис. 4.5.

Тепер порівняємо моделі представлені на рис. 4.5 та 4.6, де представлено кроки методів відмово стійкості та живучості з однієї сторони і методи забезпечення захисту інформації з другої.

Проведений аналіз показав, що послідовність кроків у обох методів дещо різниться, але їх наповненість багато в чому співпадає. Це дає підставу розглянути варіант їх поєднання в один узагальнений метод. Як видно з графових моделей методів, їх підготовчі кроки (кроки 2-4 рис. 4.5 та кроки 1,2 рис. 4.6) практично співпадають – по суті вони є частинами, що пересікаються одного і того ж процесу створення банків еталонного ПЗ та банків параметрів, які ним спільно використовуються, зрозуміло кожним методом по своєму.

Ще одна спільність полягає, в тому, що виконавчі кроки обох методів (крок 5 рис. 4.5 та кроки 4,5 рис. 4.6) забезпечуються фактично однією і тією ж службою контролю актуальності ПЗ клієнтських АРМ ІС. Різниця в тому, механізмом забезпечення відмово стійкості та живучості виступає фоновий процес (вершина 4 рис. 4.5), а за забезпечення захисту інформації відповідає програма віддаленого запуску (вершина 4 рис. 4.6), які є невід'ємними взаємодіючими частинами цієї служби.

Також, спільним для обох методів є крок, що забезпечує документування подій у системі. Спільність всіх методів проявляється також і в об'єкті впливу – він для всіх один – ПЗ клієнтських АРМ.

В результаті виконання операції інтегрованого поєднання графів представлених на рис. 4.5 та 4.6 було отримано граф узагальненого методу (рис. 4.7), який об’єднує кроки трьох методів забезпечення відмовостійкості, живучості та захисту інформації.

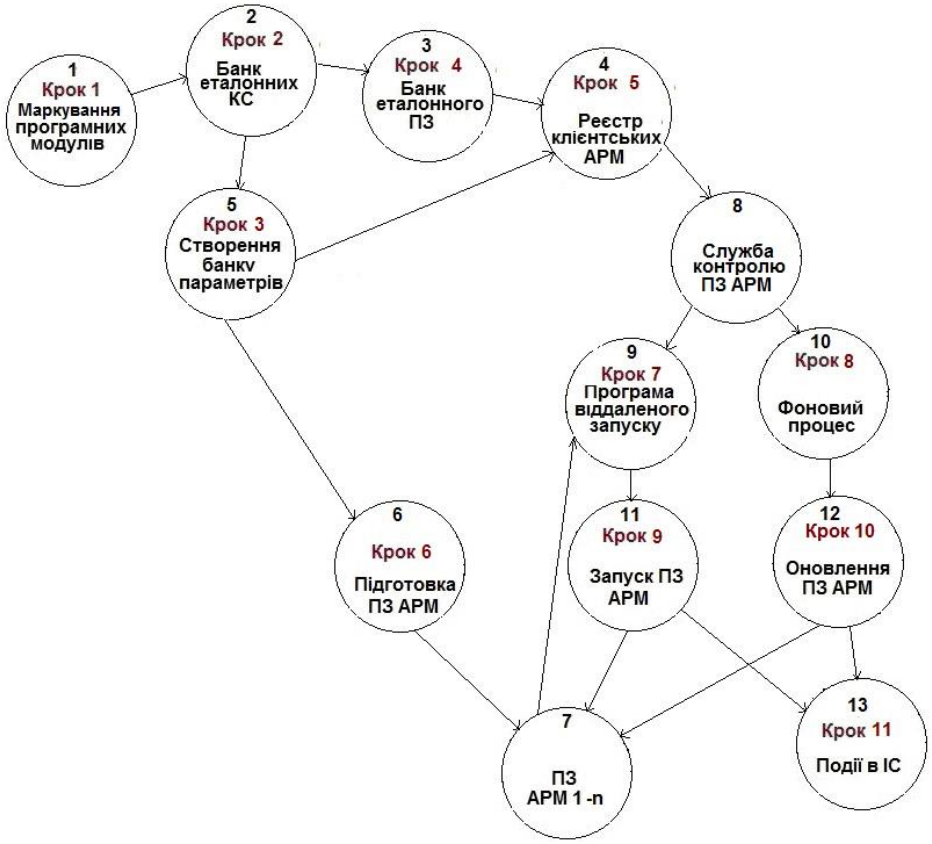


Рис. 4.7 – Графова модель інтеграції методів забезпечення відмовостійкості, живучості та захисту інформації ІС

Як видно, модель узагальненого методу, включає в себе 11 кроків. Слід зауважити, що деякі кроки (кроки 2, 3,5 рис. 4.7) тут показані лише з метою унаочнення зв’язку отриманої моделі з моделями попередницями, і можуть бути поглинуті більш ємким кроком підготовки реєстра АРМ та їх параметрів, як це має місце насправді.

Загалом всі кроки узагальненого методу розділимо на чотири групи:

– підготовчі кроки 1 – 6;

- кроки забезпечення відмовостійкості та живучості: 8,10;
- кроки забезпечення захисту інформації: 7, 9;
- кроки документування подій в ІС: 11.

Впровадження узагальненого методу [76] забезпечення відмовостійкості, живучості та захисту інформації дозволив спростити технологію впровадження підсистем захисту, дозволив спільне використання одних і тих же ресурсів, що сприяло підвищенню ефективності роботи підсистем забезпечення відмовостійкості, живучості та захисту інформації.

4.2. Архітектура системи і засоби забезпечення відмовостійкості, живучості та захисту інформації спеціалізованої ІТ

4.2.1. Загальна архітектура спеціалізованої ІТ

В основі вирішення задачі побудови спеціалізованої ІС з підвищеними засобами забезпечення відмовостійкості, живучості та захисту інформації є архітектурний підхід. Його ключовим аспектом є створення архітектури системи, яка може адаптуватись під вимоги конкретної спеціалізованої ІС. Сам же процес розробки архітектури ІС буде включає виконання двох етапів. Перший – розробка конструктивної частини системи, другий – її адаптація під конкретну функціональність системи за рахунок ітеративності самого процесу. Такий підхід до розробки ІС дозволив забезпечити втілення таких загальних вимог до побудови систем як системність, відкритість, сумісність, уніфікація та ефективність. В результаті отримано архітектуру ІС в якій досягнуто раціональне співвідношення між затратами і цільовими ефектами.

Узагальнена модель макроархітектури ІС зображена на рис. 4.8. Вона включає в себе клієнтську та серверну частини, кожна з яких реалізується кількома рівнями своїх компонентів. Її особливістю є включення в компоненти всіх рівнів архітектури

засобів, які відповідають за відмовостійкість, живучість та захист інформації з урахуванням загальної ефективності ІС.

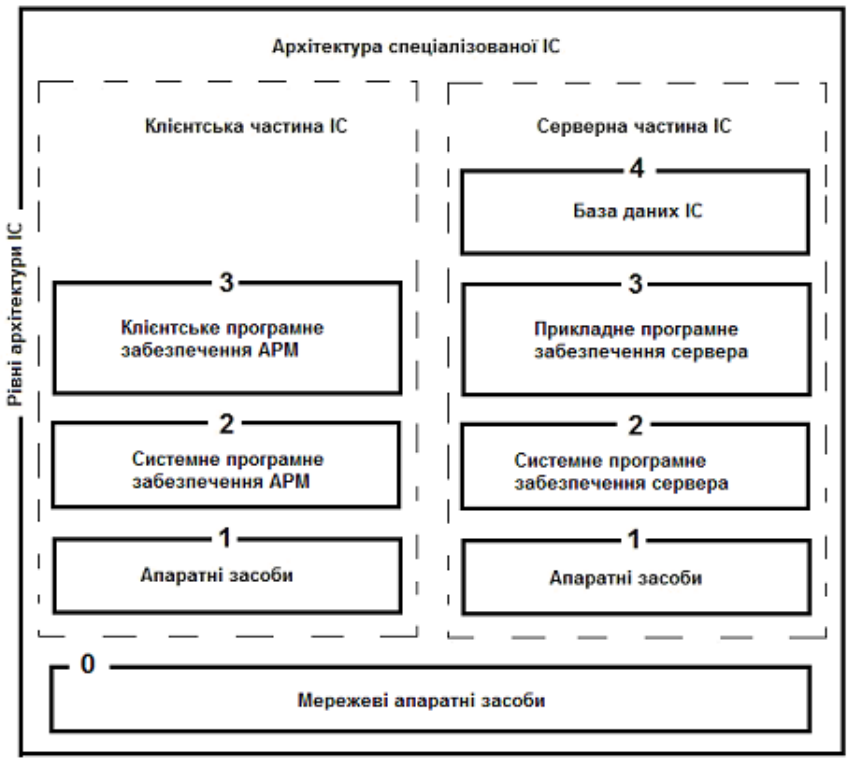


Рис. 4.8 – Узагальнена модель архітектури ІС

Клієнтська частина архітектури ІС включає в себе три стандартних рівня, де саме високе положення займає ПЗ клієнтського АРМ. Ця частина ІС масштабується до необхідної кількості робочих місць з урахуванням необхідної функціональності та об’єму виконуваних робіт по обробці інформації.

Серверна частина включає в себе чотири архітектурних рівні, де найвище положення займає рівень реалізації структур даних, які обробляються в ІС.

Ще один, нульовий рівень макроархітектури ІС, приведеної на рис. 4.8, включає в себе мережеве апаратне забезпечення яке слугує для об’єднання клієнтської та серверної частин в єдину систему.

4.2.2. Засоби забезпечення відмовостійкості, живучості та захисту інформації спеціалізованої ІТ в архітектурі клієнтської частини

Для більш детального огляду архітектури клієнтської частини перейдемо до рівня мікроархітектури, яка дозволяє її бачити на рівні компонентів. На рис. 4.9 зображено архітектуру апаратної частини типового клієнтського АРМ. Червоними прямокутниками виокремлено компоненти, що в тій чи іншій мірі забезпечують його відмовостійкість, живучість та захист інформації.

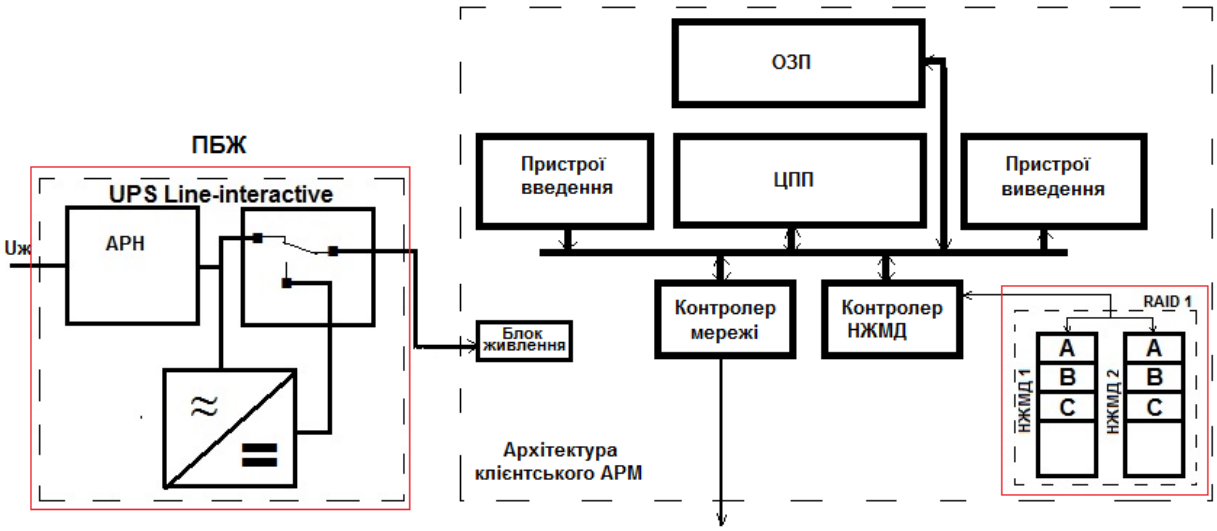


Рис. 4.9 – Мікроархітектура апаратної платформи типового АРМ клієнтської частини ІС (рівень 1)

Особливу роль в забезпечені живучості та захисту інформації відіграє пристрій безперебійного живлення (ПБЖ рис. 4.9) виконаний по технології Line-interactive, яка забезпечує високі параметри системи живлення в частині стабільності параметрів та часу реакції на зникнення напруги живлення Uж яка вкладається в 2-4 mS.

Ще одним компонентом, який суттєво впливає на відмовостійкість та захист інформації в системі, є застосування RAID – масиву накопичувачів на жорстких магнітних дисках в якості постійної пам’яті. Він складається з двох накопичувачів (НЖМД 1, НЖМД2 рис. 4.9) включених по схемі RAID 1, яка передбачає дзеркальне

збереження інформації на обох накопичувачах. Таким чином, у випадку виходу з ладу одного із них, інший із них продовжить свою роботу не допускаючи втрати інформації.

Таким чином, за рахунок включення до архітектури першого рівня клієнтської частини АРМ компонентів, що є надлишковими в звичайній роботі системи, ми отримали досить стійку до відхилень апаратну платформу з прийнятними параметрами ефективності через їх невисоку вартість.

Мікроархітектура системного програмного забезпечення клієнтського АРМ, яка зображена на рис. 4.10, включає компоненти операційної системи на прикладі ОС MS Windows та антивірусні засоби загально системного рівня. Червоним виділено компоненти, які задіяні в забезпечені живучості та захисту інформації.

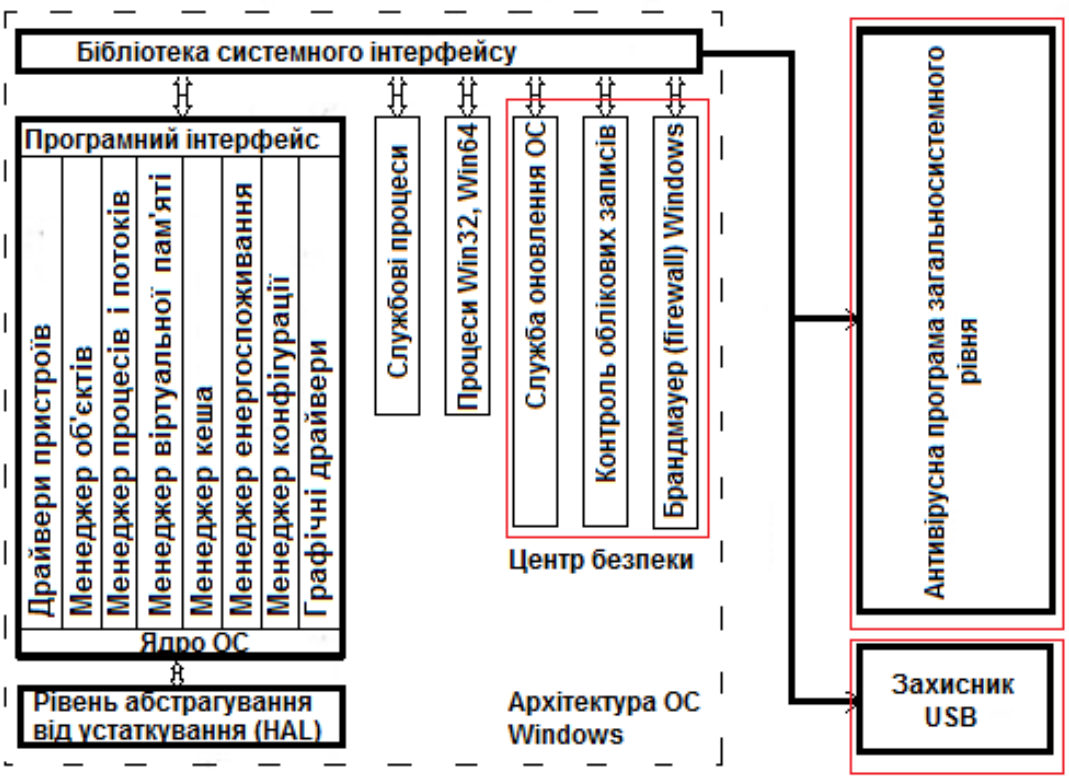


Рис. 4.10 – Мікроархітектура ОС та загальносистемні антивірусні засоби типового АРМ клієнтської частини ІС (рівень 2)

До неї відноситься і центр безпеки ОС, який включає в себе такі компоненти:

– служба оновлення – забезпечує підтримку ПЗ ОС в актуальному стані;

– служба контролю облікових записів – забезпечує контрольований доступ до комп'ютера відповідно до повноважень. Сприяє захисту інформації від оволодіння нею сторонніми особами;

– брандмауер (firewall) Windows забезпечує контроль над вхідними та вихідними з'єднаннями. Підключення, які не відповідають вимогам безпеки, автоматично блокуються. Цим він захищає інтернет-трафік, перешкоджаючи проникненню на комп'ютер зловмисного ПЗ, що загалом підвищує живучість ІС.

Підвищенню живучості системи сприяють такі програмні компоненти як загальносистемний антивірус та захисник USB.

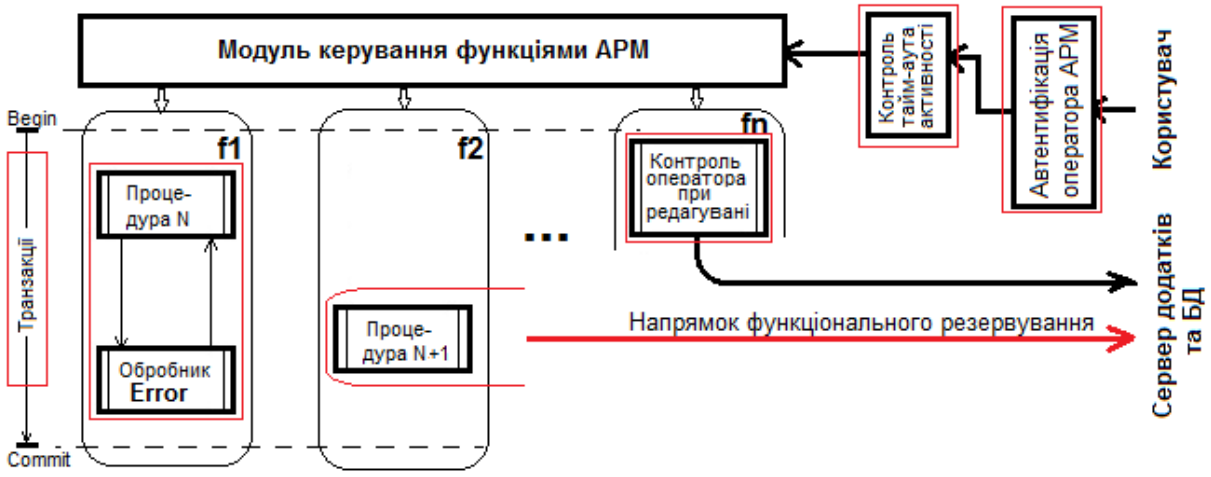


Рис. 4.11 – Мікроархітектура клієнтського АРМ (рівень 3)

Завершується архітектура клієнтської частини третім рівнем, який являє собою пласт прикладного ПЗ, що реалізує функції користувацького АРМ. Модель його мікроархітектури приведена на рис. 4.11. Дана архітектура є типовою для всіх АРМ ІС. Її особливістю є впровадження принципів уніфікації та відкритості як на рівні самого АРМ так і на рівні його компонентів. Ще однією особливістю реалізації

користувацького АРМ з такою архітектурою є застосування засобів відмовостійкості, живучості та захисту інформації. Вони є невід'ємними частинами практично всіх компонентів АРМ, що разом із засобами двох нижніх рівнів дозволило отримати клієнтське АРМ з підвищеними параметрами відвостійкості, живучості та захищеності оброблюваної ним інформації.

Кожне АРМ представляє собою набір функцій f1 – fn (рис. 4.11), кожна з яких побудована з використанням типової скелетної частини, реалізуючи принцип уніфікації. Скелетні частини включають засоби відмовостійкості та живучості: процедури з використанням двох взаємодіючих процесів, один із яких є нетривіальним обробником помилок (f1 рис. 4.11), функціональну надмірність, з можливістю управління завантаженістю АРМ (f2 рис. 4.11). Захист інформації в АРМ покладено на цілу низку програмних компонентів. До них відносяться нетривіальні редактори даних з елементами контролю цілісності даних (fn рис. 4.11), виконання всіх маніпуляцій з даними під управлінням транзакцій, а також компоненти автентифікації користувача та його активності в ІС.

4.2.3. Засоби забезпечення відмовостійкості, живучості та захисту інформації спеціалізованої ІТ в архітектурі серверної частини

Архітектура апаратної платформи серверної частини, яка умовно відноситься до першого рівня (рис. 4.8), має набагато складнішу компонентну будову в порівняні із архітектурою клієнтського АРМ. Це пов'язано з тим, що вона є центром, круг якого вибудовується ІС, а вона сама слугує платформою для розміщення БД та ПЗ, яке управляє її роботою. Її компонентний склад наведено на рис. 4.12.

Центральним компонентом приведенної архітектури апаратної платформи є основний сервер, всі компоненти якого мають серверне виконання, що само по собі задає підвищений рівень відмовостійкості та живучості. Додатково, архітектура дискової підсистеми виконана по схемі RAID-масиву типу 1, який забезпечує

стовідсоткове оперативне резервування накопичувачів. Така архітектура дискової підсистеми не тільки задає високі параметри відмовостійкості та живучості, но і гарантує захист інформації від втрати.

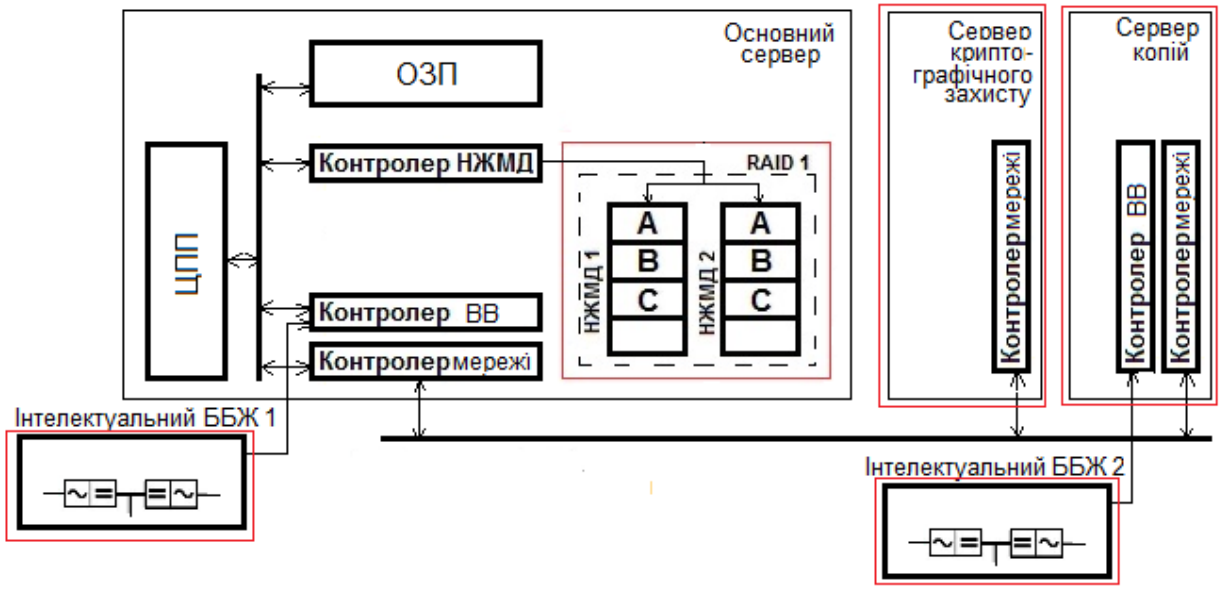


Рис. 4.12 – Архітектура апаратної платформи серверного сегмента ІС (рівень 1)

В великій мірі стабільна робота сервера залежить від роботи підсистеми живлення. Для цього архітектура серверної частини включає в себе такий компонент як блок безперебійного живлення ББЖ1 (рис. 4.12), особливістю якого є інтелектуальність. Його призначення не тільки забезпечувати сервер живленням в заданих параметрах, але і слідкувати за власним станом. У випадку якихось критичних відхилень, наприклад зниження нижче заданого рівня заряду акумуляторної батареї, подати команду серверу на його зупинку. Це не тільки розширює параметри живучості системи, але і гарантує захист інформації БД, яка дуже чутлива до збоїв.

Окрім основного сервера, приведена на рис. 4.12 модель архітектури серверної частини ІС включає в себе і сервер копій. Він має ту ж мікроархітектуру, що і основний. В поточній роботі він зберігає копії БД основного сервера, а у випадку його відмови бере на себе функції основного сервера, чим надійно забезпечує параметри

живучості, відмовостійкості та захисту інформації ІС. При цьому основний і резервний сервер територіально розмежовані, що додатково підвищує живучість системи в цілому.

Також до архітектури серверної частини входить сервер криптографічного захисту. Його призначення в забезпечені гарантованого захисту інформації в ІС, яка передається по неконтрольованих інформаційних каналах, а також протидія розвідці мережі, тобто забезпечує одночасно і параметри живучості ІС.

Архітектура ІС включає в себе окрім архітектури апаратної платформи архітектуру ПЗ (рівні 2, 3 та 4 рис. 4.8), яка є її невід'ємною частиною. Більш детальніше представлення архітектури ПЗ ІС показано на рис. 4.13.

Основними компонентами представленої на рис. 4.13 моделі архітектури, являються основний та резервний сервери БД.

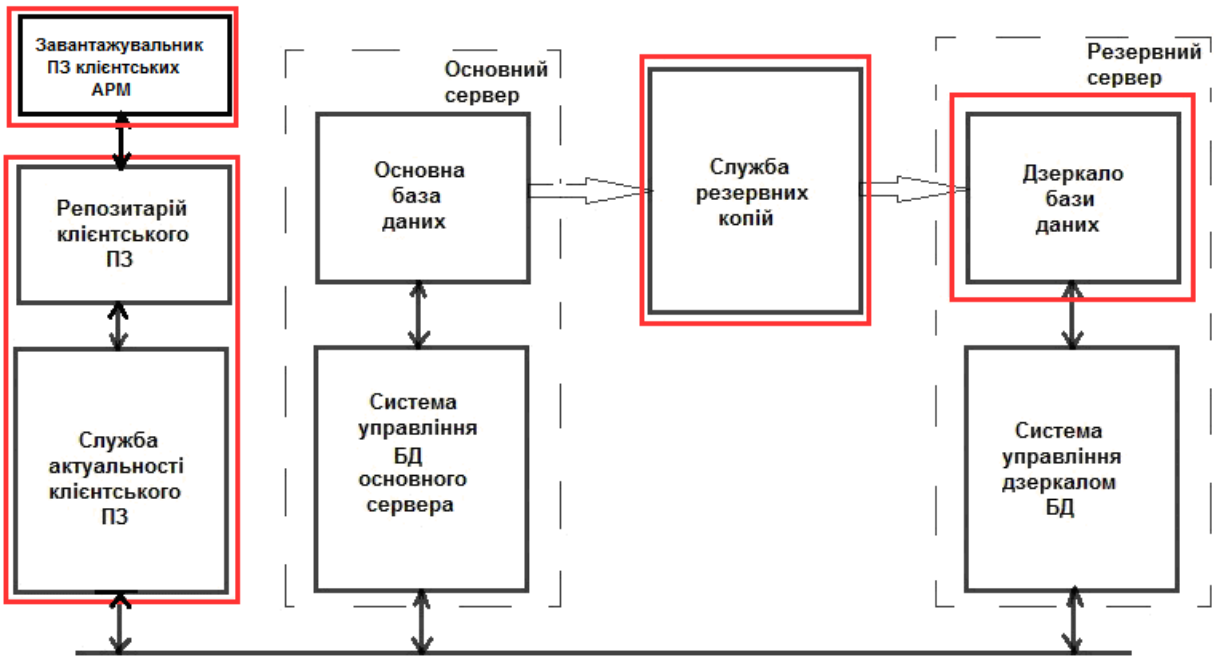


Рис. 4.13 – Архітектура ПЗ серверного кластера ІС (рівень 2, 3 та 4).

Навколо основного сервера групуються всі інші компоненти архітектури, які забезпечують стабільність його роботи і ІС в цілому. Особливо велику роль відіграє

резервний сервер із дзеркалом БД. Саме він є тим компонентом, від якого в дуже великій мірі одночасно залежать відмовостійкість, живучість за захист інформації ІС в цілому. Актуальність БД-дзеркала забезпечує такий компонент архітектури, як служба резервних копій.

Ще одним важливим компонентом архітектури ПЗ є служба актуальності клієнтського АРМ. Від роботи цього компонента залежить в великій мірі відмовостійкість, живучість ПЗ клієнтських АРМ. В своїй роботі він опирається на такий компонент як репозитарій клієнтського ПЗ. На цей же компонент опирається і робота загружчика ПЗ. Його роль полягає в недопущені до роботи з інформацією БД нелегального ПЗ, що в великій мірі впливає на захищеність інформації в ІС.

4.2.4. Засоби забезпечення відмовостійкості, живучості та захисту інформації спеціалізованої ІТ в архітектурі мережевої складової

Архітектура мережевої складової включає в себе керований комутатор та кабельні лінії передачі даних (рис. 4.14). В даній архітектурі використовуються лінії двох типів. Перший тип – лінії які знаходяться під постійним контролем і допускають в силу цього передачу даних в відкритому вигляді, що дозволяє добитись максимальної продуктивності роботи.

Другий тип – неконтрольовані. По них дані передаються в режимі криптографічного захисту. Це зменшує продуктивність роботи ІС, але гарантує захищеність даних.

Як видно з рис. 4.14, застосування керованого комутатора дає можливість створювати віртуальні локальні мережі з власними політиками безпеки. Застосування комутаторів з можливістю реалізації такої архітектури в поєднані з криптографічним захистом мережевого трафіка суттєво підвищує параметри живучості ІС та захисту інформації в ІС, надійно перешкоджаючи веденню розвідки мережі, позбавляючи зловмисників інформації, необхідної для початку атаки.

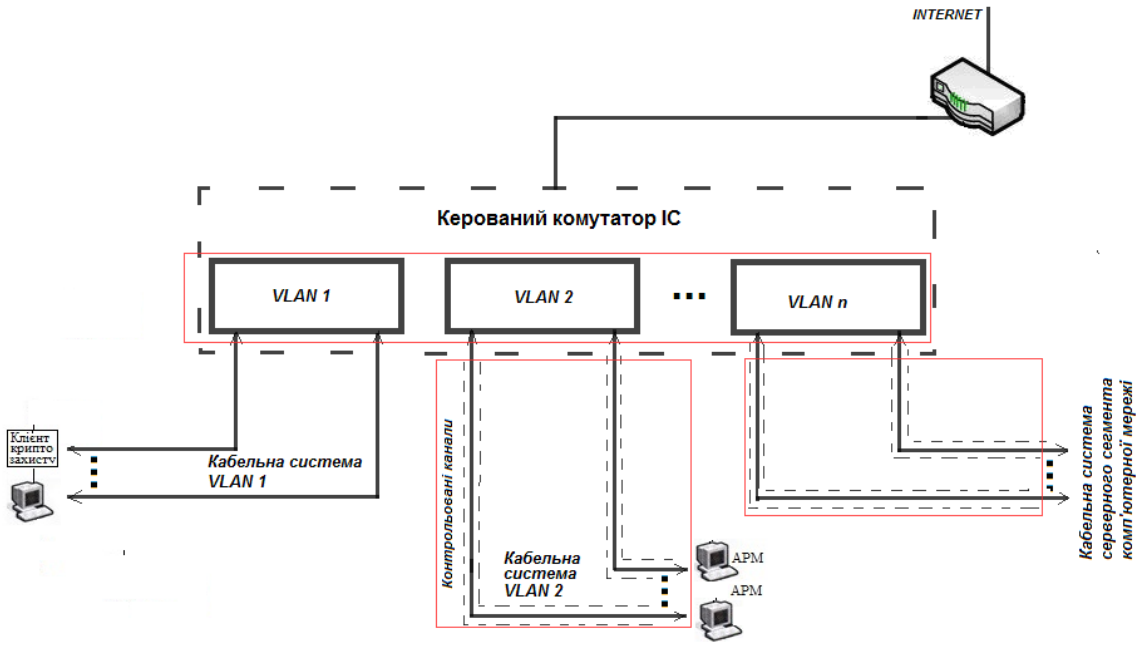


Рис. 4.14 –Архітектура мережевої апаратної платформи (рівень 0)

Зображена [127-131, 134] архітектура ІС має покращені параметри відмовостійкості, живучості та захисту інформації як для серверної та і для клієнтської частини. Це стало можливим завдяки підходу до її побуди, що базується на привнесенні деяких надмірностей в як компоненти апаратної платформи та і в програмне забезпечення ІС. Таким чином застосування всіх доступних видів резервування на всіх рівнях архітектури ІС з одночасним дотриманням принципів уніфікації та ефективності дало змогу отримати технологію побудови ІС з прийнятними вартісними характеристиками та підвищеними параметрами відмовостійкості, живучості та захисту інформації.

4.3. Ефективність методу забезпечення відмовостійкості, живучості та захисту інформації в умовах впливів ЗПЗ

ІС працює в умовах постійних різноманітних впливів на свої складові. В таких умовах виростає роль підсистеми документування подій в ІС. Основне її завдання

полягає в фіксації в режимі реального часу подій, що відбулись в різних частинах ІС з необхідним ступенем деталізації.

Для забезпечення її функціонування до службової БД включені необхідні інформаційні ресурси. Особливістю устрою підсистеми документування є той факт, що внесення інформації в БД з метою її захисту, виконується через посередництво спеціальної програмної процедури. Її виклики включені до складу всіх програмних компонентів ІС в тому числі і до засобів забезпечення відмовостійкості, живучості та захисту інформації. Таким чином забезпечується автоматичний збір інформації про критичні для функціонування ІС та її компонентів події. Візуалізація накопиченої інформації виконується у вигляді лог-файлу. Його формат приведений на рис. 4.15.

Номер події в системі	Номер робочого місця	Дата запуску АРМ або дата події в форматі ДД.ММ.РРРР гг.мм.сс	Дата завершення роботи АРМ	Код помилки	Змістовна складова критичної події	IP-адреса сервера БД	IP-адреса АРМ
NPP	ARM	PVR	KVR	Error	NAMERROR	IP_BD	IP_ARM
210195	3	13.09.2021 12:06:25				192.168.168.1	192.168.168.5
210197	76	13.09.2021 12:31:55	13.09.2021 12:48:30			192.168.168.1	192.168.168.14

Рис. 4.15 – Структура даних лог-файлу.

Як видно з рис. 4.15, структура даних лог-файлу дозволяє отримати інформацію про подію в ІС, яка включає відповіді на такі питання: коли і в якому місці ІС сталась подія, характер помилки та що до інформації про параметри мережевого з'єднання.

Якщо проаналізувати вміст інформації включеної до лог-файлу, то її можна умовно віднести до кількох блоків. До першого блоку відноситься інформація про поточну роботу компонентів ІС. В ньому накопичується інформація про те, які клієнтські АРМ і в які проміжки часу підключались до БД, а також інформація про характер та об'єми виконуваних ними операцій (рис. 4.16).

File0013.log [vk.com]

	NPP	ARM	PVR	KVR	ERROR	Content	IP_BD	IP_ARM
	200744	19	04.01.2021 9:21:45	04.01.2021 16:41:39		Вставлено:17, Оновлено:92, Проглянуто звітів:27, Видалено:1	192.168.168.1	192.168.168.8
	200745	16	04.01.2021 9:23:51	04.01.2021 16:53:19		Вставлено:3, Оновлено:82, Проглянуто звітів:8, Видалено:1	192.168.168.1	192.168.168.10

Рис. 4.16 – Збір даних про роботу клієнтських АРМ

Аналіз інформації цього блоку дозволяє бачити реальні об’єми виконуваних операцій в системі, продуктивність роботи окремих АРМ. Це дозволяє ліквідувати вузькі місця в роботі системи, отримувати інформацію щодо напрямків подальшого удосконалення ПЗ ІС.

Другий блок включає в себе інформацію про критичні ситуації зафіксовані в роботі ІС пов’язані із забезпеченням захисту інформації. Сюди відносяться аварійні завершення роботи окремих АРМ, наприклад подія 210195 рис. 4.8. Зафіксована ситуація зняття програми АРМ №3 з виконання з невідомої причини.

Часті такі події, пов’язані з якимось АРМ, говорять про проблеми з його ПЗ. Також сюди відносяться події, пов’язані з помилками обробки інформації, які призвели до відкату транзакції.

Скелетною частиною розрахункових процедур, які реалізують функціонал клієнтського АРМ є спеціальна програмна структура (рис.3.15 стр. 143), яка відноситься до їх підсистем забезпечення відмовостійкості та живучості. Вона передбачає розділення тіла процедури на марковані блоки. У випадку виникнення помилки, з якою не зміг справитись обробник помилок, в лог-файл вноситься інформація про код помилки та додається маркер блока в якому вона виникла (рис. 4.17). Це дозволяє в процесі аналізу помилок, що виникли при роботі клієнтських АРМ однозначно ідентифікувати місце їх виникнення і оперативно удосконалювати їх ПЗ.

File0013.log [vk.com]

NPP	ARM	PVR	KVR	Error	NAMERROR	IP_BD	IP_ARM
210617	16	24.09.2021 8:55:13	24.09.2021 9:00:35	-		192.168.168.1	192.168.168.10
210618	20	24.09.2021 9:11:17	24.09.2021 15:04:27	302-12		192.168.168.1	192.168.168.5

Рис. 4.17 – Приклад документування фатальної помилки в АРМ при виконані деякої функції

File001.log [vk,com]

NPP	ARM	PVR	KVR	Error	NAMERROR	IP_BD	IP_ARM
210546	2	23.09.2021 8:09:25	23.09.2021 17:01:26			192.168.168.1	192.168.168.4
210547	50	23.09.2021 8:12:05	23.09.2021 8:40:01	1000	Stop ..... Timed Out	192.168.168.1	192.168.168.10

Рис. 4.18 – Реакція ІС на тайм-аут АРМ №50

На рис.4.18 продемонстровано реакцію системи на відсутність активності оператора клієнтського АРМ №50. З метою недопущення неконтрольованого доступу до інформації ІС шляхом використання можливо неконтрольованого оператором АРМ.

File0007.log [vk.com]

NPP	ARM	PVR	KVR	Error	NAMERROR	IP_BD	IP_ARM
210551	103	23.09.2021 8:52:02		3060	Unknown program... Startup denied	192.168.168.1	192.168.168.201
210552	19	23.09.2021 8:55:08	23.09.2021 16:53:51			192.168.168.1	192.168.168.8

Рис. 4.19. Реакція ІС на спробу запуску нелегального ПЗ

Реакція ІС на спробу запуску ПЗ незареєстрованого в банку еталонного ПЗ служби контролю актуальності клієнтського ПЗ. На рис.4.12 показано зафіксовану в лог-файлі подію 210551. З неї видно, що у вказаний момент часу була зроблена спроба запустити ПЗ АРМ №103 з комп'ютерної станції з IP адресою 192.168.168.201. Оскільки це суперечить еталонним параметрам реєстра з'єднань, то запуск цього АРМ не дозволено. Таким чином ІС не допускає можливості отримання нелегального доступу до інформації шляхом використання нелегальних копій ПЗ клієнтських АРМ,

припиняючи атаку зсередини системи. Зовнішня ж атака на сервер практично неможлива через засекречений трафік і сегментування мережі із чітко виписаними політиками безпеки, через що він із зовнішньої мережі невидимий і при скануванні портів не проявляється.

Третій блок подій зв'язаний із ситуаціями недоступності сервера БД та процедурою переадресації клієнтських АРМ на резервний сервер.

File0031.log [vk.com]

NPP	ARM	PVR	KVR	ERROR	NAMERROR	IP_BD	IP_ARM
210304	3	16.09.2021 8:52:47		3050	The server is not responding	192.168.168.1	192.168.168.5
210309	3	16.09.2021 8:55:13		3050	The server is not responding	192.168.168.1	192.168.168.5
210313	3	16.09.2021 8:58:06	16.09.2021 10:08:27		redirect to backup server	192.168.168.2	192.168.168.5

Рис. 4.20 – Реакція клієнтського АРМ на недоступність основного сервера

На рис. 4.20 відображено реакції ПЗ клієнтського АРМ на недоступність ресурсів ІС. Важливо відмітити два можливих сценарія. Перший - втрата доступу до сервера в момент виконання поточної задачі - в цьому випадку транзакція , що охоплює цю операцію, буде відкочена. В іншому випадку операція просто не почнеться, до тої пори, поки не стане доступним один із серверів ІС.

На рис. 4.21 приведена послідовність подій зафіксованих в лог-файлі ІС, що ілюструють процес втрати доступу до основного сервера і перемикання клієнтських АРМ на резервний. Спеціальна процедура, яка моніторить доступність основного сервера з періодом одна хвилина, перемикає клієнтські АРМ на резервний сервер після трьох підтверджень недоступності основного. Це необхідно для відвернення випадкового перемикання серверів.

File0032.log [vk.com]

	NPP	ARM	PVR	KVR	ERROR	NAMERROR	IP_BD	IP_ARM
	210305	50	16.09.2021 8:52:54		3050	server IP 192.168.168.1 is not available	192.168.168.1	192.168.168.10
	210306	50	16.09.2021 8:53:56		3050	server IP 192.168.168.1 is not available	192.168.168.1	192.168.168.10
	210307	50	16.09.2021 8:54:12		3050	server IP 192.168.168.1 is not available	192.168.168.1	192.168.168.10
	210308	50	16.09.2021 8:54:38			blocking connection to SQL server reserve IP 192.168.168.2	192.168.168.2	192.168.168.10
	210310	50	16.09.2021 8:55:17			change address bar for clients to IP 192.168.168.2	192.168.168.2	192.168.168.10

Рис. 4.21 – Реакція ІС при недоступності основного сервера

File0039.log [vk.com]

	NPP	ARM	PVR	KVR	ERROR	NAMERROR	IP_BD	IP_ARM
	210563	16	23.09.2021 9:50:48	23.09.2021 10:57:49			192.168.168.1	192.168.168.13
	210564	50	23.09.2021 10:04:05	23.09.2021 10:06:29	3081	Software Recovery ... Module 3	192.168.168.1	192.168.168.10
	210565	149	23.09.2021 10:04:43	23.09.2021 11:57:50			192.168.168.1	192.168.168.16

Рис. 4.22 – Оновлення ПЗ АРМ через його невідповідність еталонним параметрам

В фоновому режимі служба контролю виконує перевірку актуальності ПЗ клієнтських робочих місць. На рис. 4.15 приведена подія 210564 лог-файлу, яка зафіксувала факт невідповідності параметрів програмних модулів АРМ №3 еталонним. Причиною цього може бути атака ЗПЗ або самопошкодження, в результаті чого контрольна сума модуля змінилась і його прийшлось оновити з банку ПЗ системи.

Згідно наведених прикладів фрагментів лог-файлів (рис. 4.15-4.22) встановлено, що засоби забезпечення відмовостійкості, живучості та захисту інформації ІС мають достатній рівень селективності до причин збурень у системі, надаючи її персоналу широкий спектр інформації для подальшого аналізу.

Наведено графіки (рис. 3.6), отримані за розрахунками за формулою (3.12) для результатів живучості (формула (3.13)).

Оціночні значення відмовостійкості та живучості ІТ зображено на рис. 4.23 при імплементації в ІС розробленого методу забезпечення відмовостійкості, живучості та захисту інформації ІТ, які розраховано за формулою (3.12), і відображають належний рівень стійкості щодо впливів ЗПЗ та комп'ютерних атак в процесі активізації підсистем забезпечення відмовостійкості та живучості в ІС, який становить не менше 67%.

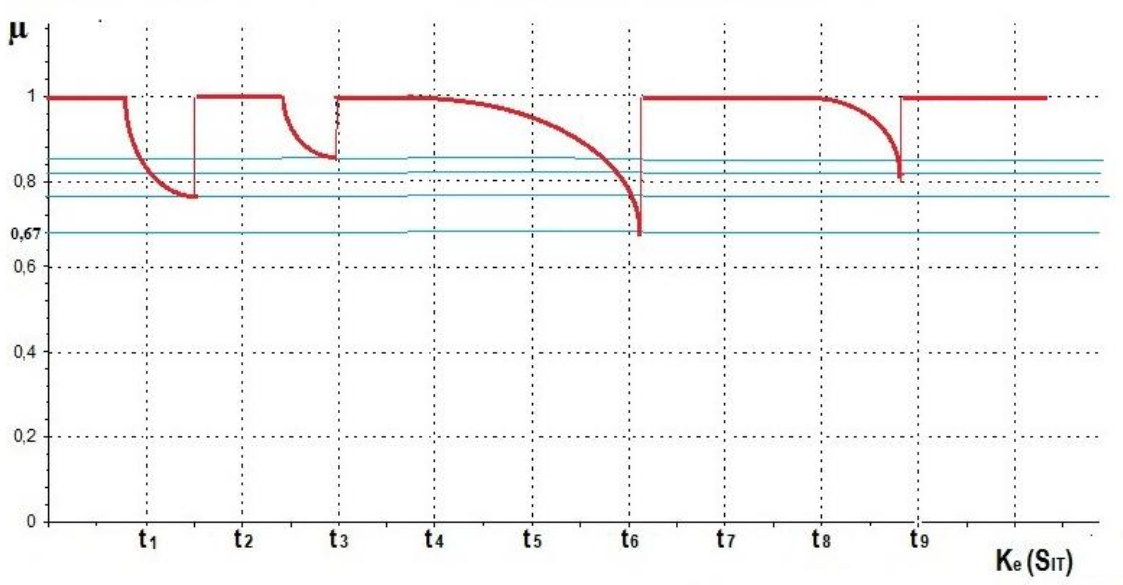


Рис. 4.23 – Оцінювання значень рівнів відмовостійкості та живучості ІС

Для перевірки ефективності розроблених засобів застосовано методику, яка представлена в п.3.3, але з врахуванням її застосування як до відмовостійкості так і до живучості одночасно. Результати дослідження підтверджують високий рівень стійкості та живучості в корпоративних комп'ютерних мережах, який становить понад 67% [74, 76, 77].

4.4 Висновки до четвертого розділу

Таким чином, розроблений метод забезпечення відмовостійкості, живучості та захисту інформації ІТ [74, 76, 77, 127-131, 134] полягає в поєднані та інтегруванні в ІТ

механізмів забезпечення відмовостійкості, живучості та захисту інформації згідно їх збігів в станах при реагуванні на впливи ЗПЗ та комп'ютерних атак, що надало змогу створювати спеціалізовані ІС стійкі до цих впливів.

Згідно розробленого методу забезпечення відмовостійкості, живучості та захисту інформації ІТ запропоновано архітектуру засобів, в які він імплементований, на основі якої створено ІС для проведення експериментальних досліджень щодо запропонованого рішення з покращення відмовостійкості, живучості та захисту інформації спеціалізованих ІТ при впливах ЗПЗ та комп'ютерних атак.

В результаті використання перелічених заходів було отримано ІС вузькоспеціалізованого використання для різних сфер застосування [131], де супроводжуванні процеси відносяться до ірреального або нереального часу із покращеними параметрами відмовостійкості, живучості та захисту інформації. Результати проведених досліджень з розробленою ІС та застосування методики оцінювання ефективності ІТ підтверджують покращений рівень стійкості та живучості в корпоративних комп'ютерних мережах, який становить понад 67%, для ІТ в які імплементовано метод забезпечення відмовостійкості, живучості та захисту інформації ІТ [74, 76, 77].

Основні результати розділу опубліковані у працях [74, 76, 77, 127-134].

ВИСНОВКИ

У результаті виконання дисертаційного дослідження було розв'язано актуальну науково-прикладну задачу розроблення методів забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих ІТ в умовах впливів ЗПЗ та комп'ютерних атак, а також розроблено відповідні засоби.

У роботі отримано наступні наукові та практичні результати:

1. Розроблений новий метод забезпечення відмовостійкості спеціалізованих ІТ згідно інтеграції в ньому резервування та надмірностей, який на відміну від відомих методів, надає змогу розширити можливості ІТ в частині її адаптивності та відповідно автоматичної зміни апаратно-програмної конфігурації, що дозволив створювати відмовостійкі спеціалізовані ІТ щодо впливів ЗПЗ та комп'ютерних атак.

2. Розроблений новий метод забезпечення живучості спеціалізованих ІТ згідно аналізу маркерів та збереженої інформації для самодослідження, який на відміну від відомих методів, зберігає інформацію про ключові процеси та здійснює їх самоаналіз, що дало можливість покращити забезпечення живучості ІТ в умовах впливів ЗПЗ та комп'ютерних атак.

3. Розроблений новий метод забезпечення захисту інформації спеціалізованих ІТ, який на відміну від відомих, полягає в поєднані із організаційними заходами інтеграцію в спеціалізовані ІТ методів сегментування мережі, криптографічного захисту, двофакторної автентифікації програмного забезпечення, створення хибних об'єктів атаки, резервного копіювання з територіальним розмежуванням місць зберігання копій, що дозволило створювати засоби з покращеним захистом інформації в умовах впливів ЗПЗ та комп'ютерних атак.

4. Розроблений новий метод забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих ІТ, який на відміну від відомих, полягає в інтеграції в спеціалізовані ІТ методів забезпечення відмовостійкості, живучості та захисту інформації згідно їх збігів в станах при реагуванні на впливи ЗПЗ та комп'ютерних

атак, що надало змогу створювати спеціалізовані ІС з покращеними характеристиками відмовостійкості, живучості та захисту інформації до цих впливів.

5. Практичне значення отриманих результатів полягає у розробленні алгоритмів та підсистеми забезпечення відмовостійкості, живучості та захисту інформації спеціалізованих ІТ, в яких здійснено інтеграцію методів забезпечення відмовостійкості, живучості та захисту інформації згідно їх збігів в станах при реагуванні на впливи ЗПЗ та комп'ютерних атак. Це дало змогу створювати спеціалізовані ІС з покращеними характеристиками відмовостійкості, живучості та захисту інформації до таких впливів. В результаті проведених експериментальних досліджень з засобами, в які імплеиентовано розроблені методи, отримано покращення відмовостійкості, живучості та захисту інформації засобів до впливів ЗПЗ та комп'ютерних атак, оціночні значення яких становлять 67%.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Adel, O.I.; Hussain, A.M.; Oligeri, G.; Di Pietro, R. Key is in the Air: Hacking Remote Keyless Entry Systems. Security and Safety Interplay of Intelligent Software Systems. CSITS ISSA 2018, Lecture Notes in Computer Science; Hamid, B., Gallina, B., Shabtai, A., Elovici, Y., Garcia-Alfaro, J., Eds.; Springer, Cham, 2019; vol 11552, pp 125–132. https://doi.org/10.1007/978-3-030-16874-2_9

2. Albuquerque, A.; Caixinha, D. Mastering Elixir: Build and scale concurrent, distributed, and fault-tolerant applications. 1st edition; Publishing, 2018; p 574. ISBN: 1788472675

3. Anjankar, S.C.; Pund, A.M.; Junghare, R.; Zalke, J. Real-Time FPGA-Based Fault Tolerant and Recoverable Technique for Arithmetic Design Using Functional Triple Modular Redundancy (FRTMR). Proceedings of the Second International Conference on Computational Intelligence and Informatics. Advances in Intelligent Systems and Computing, July 24, 2018; Bhateja, V., Tavares, J., Eds, Springer; Singapore, 2018; vol 712; pp 491–499 https://doi.org/10.1007/978-981-10-8228-3_45

4. Asharina, I.V. The Concept of Failure- and Fault-Tolerance on Base of the Dynamic Redundancy for Distributed Control Systems of Spacecraft Groups. Robotics: Industry 4.0 Issues & New Intelligent Control Paradigms. Studies in Systems, Decision and Control, Kravets, A. Eds, Springer, Cham, 2020; vol 272, pp 181–191. https://doi.org/10.1007/978-3-030-37841-7_15

5. Avizienis, A.; Kopetz, H.; Jean-Claude L. The Evolution of Fault-Tolerant Computing In the Honor of William C. Carter. Series: Dependable Computing and Fault-Tolerant Systems. New York, 1987; X, p 465.

6. Avoine, G.; Carpent, X.; Kordy, B.; Tardif, F. How to Handle Rainbow Tables with External Memory. ACISP 2017: Information Security and Privacy. Lecture Notes in Computer Science, May 31, 2017; Pieprzyk, J., Suriadi, S. Eds.; Springer, Cham, 2017; vol 10342, pp 306–323. https://doi.org/10.1007/978-3-319-60055-0_16

7. Banatre, M.; Lee, P.A. Papers of the workshop on hardware and software architectures for fault tolerance: Experiences and perspectives. Springer, 2014; p 332. ISBN-13: 978-3662189030

8. Bao, Z.; Dinur, I.; Guo, J.; Leurent, G.; Wang L. Generic Attacks on Hash Combiners, Journal of Cryptology 2020, 33, pp 742–823. https://doi.org/10.1007/s00145-019-09328-w

9. Barabady, J. Improvement of System Availability Using Reliability and Maintainability Analysis. Division of Operation and Maintenance Engineering Lulea University of Technology. Sweden, Lulea University of Technology 2005; p 98.

10. Blanke, M.; Kinnaert, M.; Lunze, J.; Staroswiecki, M. Diagnosis and Fault-Tolerant Control., 3rd ed., Springer-Verlag: Berlin Heidelberg, 2016; XX, p 695. DOI 10.1007/978-3-662-47943-8

11. Volkanov, D.Y. Method for Choosing a Balanced Set of Fault-Tolerance Techniques for Distributed Computer Systems. Automatic Control and Computer Sciences 2018, 51, pp 539–550.

12. Boranbayev, A.; Boranbayev, S.; Yersakhanov, K.; Nurusheva, A.; Taberkhan, R. Methods of Ensuring the Reliability and Fault Tolerance of Information Systems. Information Technology - New Generations. Advances in Intelligent Systems and Computing. April 13, 2018; Latifi, S. Eds.; Springer, Cham, 2018; vol 738, pp 729–730. https://doi.org/10.1007/978-3-319-77028-4_93

13. Botnet Wiki: "Botnet detection". WIRED. Retrieved 2017-05-24. http://jpdias.me/botnet-lab//countermeasures/detection.html, (accessed on Maj 29, 2022)

14. Bozzano, M.; Cimatti, A.; Griggio, A.; Jonas, M. Efficient Analysis of Cyclic Redundancy Architectures via Boolean Fault Propagation. Tools and Algorithms for the Construction and Analysis of Systems. TACAS 2022. March 30, 2022; Fisman, D., Rosu, G. Eds.; Springer, Cham, 2020; vol 13244 pp 273–291. https://doi.org/10.1007/978-3-030-99527-0_15

15. BROADCOM. Support and Services\Symantec Security Centerhttps. https://www.broadcom.com/support/security-center (accessed on Maj 29, 2022).

16. Buke, A.O.; Yongcai, W.; Lu, Y.; Brooks, R.R.; Iyengar S.S. On Precision Bound of Distributed Fault-Tolerant Sensor Fusion Algorithms. ACM Computing Surveys 2017, vol.49, №1, pp 1 -23. https://doi.org/10.1145/2898984

17. Cheng, X.; Guan, Y.; Zhang, Y. Design and Implementation of Dynamic Memory Allocation Algorithm in Embedded Real-Time System. Data Science. ICPCSEE 2018. Communications in Computer and Information Science. Sept 9, 2018; Zhou, Q., Gan, Y., Jing, W., Song, X., Wang, Y., Lu, Z. Eds.; Springer, Singapore, 2018; vol 901, pp 539–547. https://doi.org/10.1007/978-981-13-2203-7_43

18. Common Weakness Enumeration. "2019 CWE Top 25 Most Dangerous Software Errors». https://cwe.mitre.org/top25/archive/2019/2019_cwe_top25.html (accessed Maj 25, 2022).

19. Croft, R.; Xie, Y.; Zahedi, Babar, M.A.; Treude, C. An empirical study of developers' discussions about security challenges of different programming languages. Empir Software Eng. 2022, vol. 27, Article number 27. https://doi.org/10.1007/s10664-021-10054-w

20. De Nicola, R. Adaptive and Autonomous Systems and Their Impact on Us. Software, Services, and Systems. Lecture Notes in Computer Science. Serbedzija, N., Hennicker, R. Eds.; Springer, Cham, 2015; vol 8950, pp 662–675. https://doi.org/10.1007/978-3-319-15545-6_37

21. Dongarra J.; Herault1, T.; Robert, Y. Fault tolerance techniques for high-performance computing. Series: Computer Communications and Networks. 2015; IX, 320p. https://www.netlib.org/lapack/lawnspdf/lawn289.pdf (accessed Maj 25, 2022)

22. Du, D.; Xu, S.; Cocquempot V. Observer-Based Fault Diagnosis and Fault-Tolerant Control for Switched Systems. Series: Studies in Systems, Decision and Control. Springer: Singapore, 2021; vol 280, p 81. DOI 10.1007/978-981-15-9073-3

23. Du, D.; Ren, X.; Wu, Y.; Chen, J.; Ye, W.; Sun, J.; Xi, X.; Gao, O.; Zhang S. Refining Traceability Links Between Vulnerability and Software Component in a Vulnerability Knowledge Graph. Web Engineering. ICWE 2018. May 20, 2018; Mikkonen, T., Klamma, R., Hernández, J. Eds,; Springer, Cham, 2018; vol 10845, pp 33–49 https://doi.org/10.1007/978-3-319-91662-0_3

24. Erlank, A.O.; Bridges C.P. A hybrid real-time agent platform for fault-tolerant, embedded applications. Autonomous Agents and Multi-Agent Systems 2017; vol 32, pp 252–274. https://doi.org/10.1007/s10458-017-9378-4

25. Erickson, J. Buffer overflow. The Art of Exploitation. 2nd Edition, Amazon, 2010; p 681. ISBN-13 978-1593271442

26. Erickson, K. Hacking for Beginners: Step By Step Guide to Cracking Codes Discipline, Penetration Testing, and Computer Virus. Learning Basic Security Tools On How To Ethical Hack And Grow. Francesco Cammardella, 2020; p 156.

27. Gabsi, W.; Zalila, B.; Jmaiel, M. Extension of the Ocarina Tool Suite to Support Reliable Replication-Based Fault-Tolerance. Reliable Software Technologies – Ada-Europe 2016. May 31, 2016; Bertogna, M., Pinho, L., Quiñones, E. Eds.; Springer, Cham, 2016; vol 9695, pp 129–144. https://doi.org/10.1007/978-3-319-39083-3_9

28. Garcia-Alfaro, J. Security and Privacy in Communication Networks. 17th EAI International Conference, SecureComm 2021. September 6–9, 2021; Li S., Poovendran,R., Debar, H., Yung, M. Eds.; Springer Cham, 2021; vol 399, Part II. https://doi.org/10.1007/978-3-030-90022-9

29. GeeksforGeeks. Dangling, Void, Null and Wild Pointers. https://www.geeksforgeeks.org/dangling-void-null-wild-pointers/ (accessed Maj 25, 2022).

30. Geetha, D.M.; Muthusundar, S.K.; Subramaniam, M.; Ayyaswamy, K. Temporary Redundant Transmission Mechanism for SCTP Multihomed Hosts., The Scientific World Journal, vol 2015, Hindawi Publishing Corporation, 2015, pp 1-10. DOI:10.1155/2015/158697

31. Gregory, P.; Kruchten, P. Agile Processes in Software Engineering and Extreme Programming – Workshops. XP 2021 Workshops, Virtual Event, June 14–18, 2021; Springer Cham, 2021; XIII, p 231. https://doi.org/10.1007/978-3-030-88583-0

32. Grusho, A.A.; Grusho, N.A.; Zabezhailo, M.I.; Timonina, E.E. Use of Contradictions in Data for Finding Implicit Failures in Computer Systems. Automatic Control and Computer Sciences 2022, vol 55, pp 1115–1120. https://doi.org/10.3103/S0146411621080149

33. Gunawi, H.S.; Do, T.; Joshi, P.; Hellerstein, J.M.; Arpaci-Dusseau, A.C.; Arpaci-Dusseau, R.H.; Sen, K. Towards Automatically Checking Thousands of Failures with Micro - specifications. https://www.usenix.org/legacy/events/hotdep10/tech/full_papers/Gunawi.pdf (accessed on Maj 25, 2022).

34. Habibian H.; Patooghy A. Fault-tolerant routing methodology for hypercube and cube-connected cycles interconnection networks. The Journal of Supercomputing 2017, vol 73, pp 4560–4579. https://doi.org/10.1007/s11227-017-2033-7

35. Habli, I.; Sujan, M.; Gerasimou, S.; Schoitsch, E.; Bitsch, F. Computer Safety, Reliability, and Security. SAFECOMP 2021 Workshops. DECSoS, MAPSOD, DepDevOps, USDAI, and WAISE, York, UK, Sept 7, 2021; Springer Cham, 2021; XV, p 324. https://doi.org/10.1007/978-3-030-83906-2

36. Haqiq A.; Bounabat B. The First International Conference on Intelligent Computing in Data Sciences Towards Integration of Fault Tolerance in Agent-based Systems. Procedia Computer Science 2018, vol 127, pp 264-273. https://doi.org/10.1016/j.procs.2018.01.122

37. Hatami, E.; Arasteh, B. An efficient and stable method to cluster software modules using ant colony optimization algorithm. J Supercomput. 2020, vol 76, pp 6786–6808. https://doi.org/10.1007/s11227-019-03112-0

38. IANA. Internet Assigned Numbers Authority. Service Name and Transport Protocol Port Number Registry. http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml, (accessed Maj 26, 2022).

39. Jain, T.; Yame, J.J.; Sauter, D. Active Fault-Tolerant Control Systems: A Behavioral System Theoretic Perspective Series: Studies in Systems, Decision and Control. Springer; 2018; p 167. ISBN-13: 978-3319688275

40. Jahankhani, H.; Jamal, A.; Lawson S. Cybersecurity, Privacy and Freedom Protection in the Connected World. Proceedings of the 13th International Conference on Global Security, Safety and Sustainability, London, January 2021; Springer Cham, 2021; XI, pp 469 - 475. https://doi.org/10.1007/978-3-030-68534-8

41. Java & point. «Dangling Pointers in C». https://www.javatpoint.com/dangling-pointers-in-c (accessed Maj 26, 2022).

42. Ji, H.; Wu, G.; Zhao, Y.; Wei, L.; Wang G.; Fan Y. A fault-tolerant optimization mechanism for spatiotemporal data analysis in flink. World Wide Web, [Online] 2022; https://doi.org/10.1007/s11280-022-01006-5 (accessed Maj 26, 2022).

43. Kalinin, M.O.; Pavlenko, E.Y. Increasing the fault tolerance and availability of software defined networks using network equipment control based on multiobjective optimization by service quality parameters. Aut. Control Comp. 2015; vol 49, issue 8, pp 673–678. https://doi.org/10.3103/S014641161508026X

44. Khan, M.I.; Foley, S.N.; O'Sullivan B. Database Intrusion Detection Systems (DIDs): Insider Threat Detection via Behaviour-Based Anomaly Detection Systems - A Brief Survey of Concepts and Approaches. Nov 2021, Conference paper: EISA 2021: Emerging Information Security and Applications. Eprint arXiv, 2022; pp 178–197. https://doi.org/10.48550/arXiv.2011.02308

45. Khurana, M.; Yadav, R.; Kumari, M. Buffer Overflow and SQL Injection: To Remotely Attack and Access Information. Cyber Security. Advances in Intelligent Systems and Computing. Bokhari, M., Agrawal, N., Saini, D. Eds.; Springer, Singapore, 2018; vol 729, pp. 301–313. https://doi.org/10.1007/978-981-10-8536-9_30

46. Koiso, K.; Sakamoto, N.; Nonaka, J.; Shoji, F. A Transfer Entropy Based Visual Analytics System for Identifying Causality of Critical Hardware Failures Case Study: CPU Failures in the K Computer. Methods and Applications for Modeling and Simulation of Complex Systems. AsiaSim 2018. Communications in Computer and Information Science. Li, L., Hasegawa, K., Tanaka, S. Eds.; Springer, Singapore, 2018; vol 946, pp 563–574. https://doi.org/10.1007/978-981-13-2853-4_44

47. Kolhar, M.; Alameen, A.; Gulam, M. Performance evaluation of framework of VoIP/SIP server under virtualization environment along with the most common security threats. Neural Computing and Applications 2018, vol 30, pp 2873–2881. https://doi.org/10.1007/s00521-017-2886-y

48. Linger, R.C.; Mead, N.R.; Lipson, H.F. Requirements Definition for Survivable Network Systems. Proceedings of IEEE International Symposium on Requirements Engineering: RE '98. Colorado Springs, CO, USA, April 10-10 1998; IEEE, 2002; 5912305, pp 1 - 10. DOI: 10.1109/ICRE.1998.667804

49. Li, Z. Optimization of Rainbow Tables for Practically Cracking GSM A5/1 Based on Validated Success Rate Modeling. Topics in Cryptology - CT-RSA 2016. CT-RSA 2016. Sako, K. Eds.; Springer, Cham. 2016; vol 9610, pp 359–377. https://doi.org/10.1007/978-3-319-29485-8_21

50. Lu, Xh.; Zeng, Lf.; Huang, Hh.; Yan, Wh. Data Scheduling Method of Social Network Resources Based on Multi-Agent Technology. Multimedia Technology and Enhanced Learning. ICMTEL 2020. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Zhang, YD., Wang, SH., Liu, S. Eds.; Springer, Cham, 2020; vol 326, pp 148–158. https://doi.org/10.1007/978-3-030-51100-5_13

51. Ludwig, M. The Giant Black Book of Computer Viruses. Amazon, Paperback, 2019; p 468. ISBN-13 : 978-1441407122

52. Miele, A. Buffer overflow vulnerabilities in CUDA: a preliminary analysis. Journal of Computer Virology and Hacking Techniques, 2016, vol 12, pp 113–120. https://doi.org/10.1007/s11416-015-0251-1

53. Monnappa, K.A. Learning Malware Analysis. Published by Packt Publishing Ltd, 2018; p 501. ISBN 978-1-78839-250-1

54. Mustafa, O.; Lockard, R.P. Oracle Database Application Security: With Oracle Internet Directory, Oracle Access Manager, and Oracle Identity Manager. Apress, 2019; XVII, p 341. ISBN: 9781484253670

55. Nguyen, V.T.; Tuan, C.N.; Dung, L.T.; Hai, V.M.; Nguyen, T.T. Computer Virus Detection Method Using Feature Extraction of Specific Malicious Opcode Sets Combine with aiNet and Danger Theory. Future Data and Security Engineering. FDSE 2016. Dang, T., Wagner, R., Küng, J., Thoai, N., Takizawa, M., Neuhold, E. Eds.; Springer, Cham, 2016; vol 10018, pp 199–208. https://doi.org/10.1007/978-3-319-48057-2_14

56. Nguyen, V.T.; Hien, V.T.; Tuan, L.D.; Tiep, M.V.; Anh, N.H.; Vuong, P.T. A Computer Virus Detection Method Based on Information from PE Structure of Files Combined with Deep Learning Models. Future Data and Security Engineering. Big Data, Security and Privacy, Smart City and Industry 4.0 Applications. FDSE 2020. Communications in Computer and Information Science. Dang, T.K., Küng, J., Takizawa, M., Chung, T.M. Eds.; Springer, Singapore, 2020; vol 1306, pp 120–129. https://doi.org/10.1007/978-981-33-4370-2_9

57. Ni, C.; Liu, WS.; Chen, X.; Gu, Q.; Chen, D.X.; Huang, Q.G. A Cluster Based Feature Selection Method for Cross-Project Software Defect Prediction. J. Comput. Sci. Technol. 2017, vol 32, pp 1090–1107. https://doi.org/10.1007/s11390-017-1785-0

58. Nogueras, R.; Cotta, C. Studying Fault-Tolerance in Island-Based Evolutionary and Multimemetic Algorithms. J Grid Computing 2015, vol 13, pp 351–374. https://doi.org/10.1007/s10723-014-9315-6

59. OWASP Cheat Sheet Series. Cross Site Scripting Prevention Cheat Sheet. https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html (accessed Maj 26, 2022).

60. Panda\mediacentr. What Is a Heuristic Virus and How to Remove It. https://www.pandasecurity.com/en/mediacenter/security/heuristic-virus/ (accessed Maj 26, 2022).

61. Patranabis, S.; Mukhopadhyay, D. Fault Tolerant Architectures for Cryptography and Hardware Security. Series: Computer Architecture and Design Methodologies. Springer: Singapore, 2018; XII, p 240.

62. Paul, C. Software Security - Exploits and Privilege Escalation. Computer Security and the Internet. Information Security and Cryptography. Springer, Cham, 2020; pp 155–182. https://doi.org/10.1007/978-3-030-33649-3_6

63. Pavesi, J.; Villegas, T.; Perepechko, A.; Aguirre, E.; Galeazzi, L. Validation of ICS Vulnerability Related to TCP/IP Protocol Implementation in Allen-Bradley Compact Logix PLC Controller. Telematics and Computing. WITCOM 2019. Communications in Computer and Information Science. Nov 4-8, 2019; Merida, Mexico. Mata-Rivera, M., Zagal-Flores, R., Barría-Huidobro, C. Eds.; Springer, Cham, 2019; vol 1053, pp 355–364. https://doi.org/10.1007/978-3-030-33229-7_30

64. Phillip, L.W.; Crawley, K. The Pentester BluePrint: Starting a Career as an Ethical Hacker. Wiley, 2020; p 192. ISBN-13 : 978-1119684305

65. Quora. «What is the difference between a wild and a dangling pointer in C?». https://www.quora.com/What-is-the-difference-between-a-wild-and-a-dangling-pointer-in-C (accessed Maj 26, 2022).

66. TechTarget. RAID (redundant array of independent disks) https://www.techtarget.com/searchstorage/definition/RAID (accessed Maj 26, 2022).

67. Rawat, A.; Sushil, R.; Agarwal, A.; Sikander, A.; Bhadoria, R.S. A New Adaptive Fault Tolerant Framework in the Cloud. IETE Journal of Research, 2021, pp 113-117. DOI: 10.1080/03772063.2021.1907231

68. Sakamoto, J.; Hayashi, S.; Fujimoto, D.; Matsumoto, T. Constructing software countermeasures against instruction manipulation attacks: an approach based on vulnerability evaluation using fault simulator. Cluster Comput. [Online] 2021; pp 1-15. https://doi.org/10.1007/s10586-021-03438-6

69. Sanders C. Practical Packet Analysis. Using Wireshark to Solve Real-World Network Problems. 3rd Edition. William Pollock: San Francisco, 2017; p 491. ISBN: 9781593278021

70. Sniatala, P.; Amini, M.H.; Boroojeni K.G. Fundamentals of Brooks–Iyengar Distributed Sensing Algorithm. Trends, Advances, and Future Prospects. 2020; p 230. https://doi.org/10.1007/978-3-030-33132-0

71. Sniatala, P.; Iyengar, S.S.; Sanjeev Kaushik Ramani S.K. Evolution of Smart Sensing Ecosystems with Tamper Evident Security. Springer, 2021; p 255. ISBN-13 : 978-3030777630

72. Sreenivasulu, G.; Srinivas, P.V.S.; Goverdhan, A. A Distributed Fault Analysis (DFA) Method for Fault Tolerance in High-Performance Computing Systems. ; Jaipur, India, Nov 9-10, 2020; Bansal, J., Gupta, M., Sharma, H., Agarwal, B. Eds.; Communication and Intelligent Systems. ICCIS 2019. Springer: Singapore, 2020; vol 120, pp 61–76. https://doi.org/10.1007/978-981-15-3325-9_5

73. Steinberg, J. Cybersecurity For Dummies. Computer Science. Publisher Wiley, 2019: p 345. ISBN 9781119560340

74. Stetsyuk, M.; Bedratyuk, L.; Savenko, B.; Stetsyuk, V.; Savenko O. Providing the Resilience and Survivability of Specialized Information Technology Across Corporate Computer Networks. 1st International Workshop on Intelligent Information Technologies & Systems of Information Security. Khmelnytskyi, Ukraine, June 10-12, 2020; CEUR Workshop Proceedings, 2020; vol 2623, pp 219-238. (*Scopus*)

75. Stetsiuk, M.; Nicheporyk, A.; Savenko, B. Ensuring the Fault Tolerance And Survivability of Specialized Information Technologies in Corporate Computer Networks Under the Influence of Malicious Software. Proceedings of VII International conference

"Information Technology and Interactions" (IT&I-2020), Taras Shevchenko National University, Kyiv, December 02-04, 2020; Snytyuk, V., Anisimov, A., Krak, I., Nikitchenko, M. Eds.; pp 105-106.

76. Stetsiuk, M.V.; Kashtalian, A.S. The methods of ensuring fault tolerance, survivability and protection of information of specialized information technologies under the influence of malicious software. Computer Systems And Information Technologies (Комп'ютерні системи та інформаційні технології) 2022, №1, pp 36 - 44. http://csitjournal.khmnu.edu.ua/index.php/csit/article/view/126/78

77. Stetsyuk, M.V.; Stetsyuk, V.M.; Savenko, B.O.; Savenko, O.S.; Dobrowolski M. Implementation of contropl by parameters of client automated workplaces of specialized information systems for neutralization malware. 2st International Workshop on Intelligent Information Technologies & Systems of Information Security. Khmelnytskyi, Ukraine, March 24-26, 2021; CEUR Workshop Proceedings, 2021; 2853, pp 340–352. (*Scopus*)

78. Steven, X.D. Advanced methods for fault diagnosis and fault-tolerant control. Springer-Verlag: Berlin Heidelberg, 2021; p 658. https://doi.org/10.1007/978-3-662-62004-5

79. Tang, X.; Zhai, J.; Yu, B.; Chen, W.; Zheng W. Self-Checkpoint: An In-Memory Checkpoint Method Using Less Space and Its Practice on Fault-Tolerant HPL. ACM SIGPLAN Notices, 2017, vol 52, Issue 8, pp 401–413. https://doi.org/10.1145/3155284.3018745

80. Karatos. The only way for C ++ programmers-dangling pointers and wild pointers. https://titanwolf.org/Network/Articles/Article?AID=93b3cb9f-74e8-4d2f-a295-e69edffb9f37#gsc.tab=0 (accessed Maj 26, 2022).

81. Turner, S.; Security vulnerabilities of the top ten programming languages: C, Java, C++, Objective-C, C#, PHP, Visual Basic, Python, Perl, and Ruby. Journal of Technology Research 2014, p 16.

82. Varadharajan, V.; Bansal, S. Data Security and Privacy in the Internet of Things (IoT) Environment. Connectivity Frameworks for Smart Devices. Computer

Communications and Networks. Mahmood, Z. Eds.; Springer, Cham, 2016; pp 261–281. https://doi.org/10.1007/978-3-319-33124-9_11

83. Verwcode. State of Software Security. https://www.veracode.com/state-of-software-security-report (accessed Maj 26, 2022).

84. Virus a retrospective. How Anti-Virus Software Works. https://cs.stanford.edu/people/eroberts/cs181/projects/2000-01/viruses/anti-virus.html (accessed Maj 27, 2022).

85. Volkanov, D.Y. Method for Choosing a Balanced Set of Fault-Tolerance Techniques for Distributed Computer Systems. Automatic Control and Computer Sciences 2017, vol 51, pp 539–550. https://doi.org/10.3103/S0146411617070239

86. Wang, J.; Kang, J.; Hou, G. Real-Time System Fault-Tolerant Scheme Based on Improved Chaotic Genetic Algorithm. Wireless and Satellite Systems. WiSATS 2019. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. May 07, 2019; Jia, M., Guo, Q., Meng, W. Eds.; Springer, Cham. 2019; vol 281, pp. 145–156. https://doi.org/10.1007/978-3-030-19156-6_14

87. Wang, T.; Zhang, Z.; Zhao, M.; Liu, K.; Jia, Z.; Yang, J.; Wu, Y. H_2-RAID: A Novel Hybrid RAID Architecture Towards High Reliability. Algorithms and Architectures for Parallel Processing. ICA3PP 2018. Vaidya, J., Li, J. Eds.; Springer, Cham. 2018; vol 11337, pp. 617–627. https://doi.org/10.1007/978-3-030-05063-4_48

88. WhiteSource. What are the most secure Programming languages? How Do the Top Programming Languages Measure Up When it Comes to Security? https://www.whitesourcesoftware.com/most-secure-programming-languages (accessed Maj 27, 2022).

89. Winderickx, J.; Braeken, A.; Singelée, D.; Mentens, N. In-depth energy analysis of security algorithms and protocols for the Internet of Things. J Cryptogr 2022, 12, pp 137–149. https://doi.org/10.1007/s13389-021-00274-7

90. Wu, J. Revelation of the Heterogeneous Redundancy Architecture. In: Cyberspace Mimic Defense. Wireless Networks. Springer Nature Switzerland AG. 2020; pp 207–271. https://doi.org/10.1007/978-3-030-29844-9_6

91. Wu, Y.; Liu, D.; Chen, X.; Ren, J.; Liu, R.; Tan, Y.; Zhang Z. MobileRE: A replicas prioritized hybrid fault tolerance strategy for mobile distributed system. Journal of Systems Architecture, 2021, vol 118, N102217. https://doi.org/10.1016/j.sysarc.2021.102217

92. Xiao, J.; Liao, L.; Hu, J.; Chen, Y.; Hu, R. Exploiting global redundancy in big surveillance video data for efficient coding. Cluster Comput, 2015, vol 18, pp 531–540. https://doi.org/10.1007/s10586-015-0434-z

93. Yang, M.; Hua, G.; Feng, Y.; Gong, J. Fault-Tolerance Techniques for Spacecraft Control Computers. Wiley, 2017; p 352. ISBN 111910727X

94. Zegzhda, P.D.; Alekseev, I.V. Specification-Based Classification of Network Protocol Vulnerabilities. Automatic Control and Computer Sciences, 2021; vol 54, pp. 922–929. https://doi.org/10.3103/S0146411620080040

95. Zhilenkov, A.A.; Chernyi, S.G. Enhanced Fault Tolerance in Software and Hardware Network Control Systems Using Soft Cloud Storage. Automatic Documentation and Mathematical Linguistics, 2020, vol 54, pp 36 - 42. https://doi.org/10.3103/S0005105520010021

96. Билаш, А.А.; Белобородов, А.Ю.; Бохан, К.А. Технологии Web, Grid, Cloud для гарантоспособных ИТ-инфраструктур. Харків, Нац. аэрокосм. ун-т им. Н.Е. Жуковского, ХАИ. 2013; с 868

97. Боровська Т.М. Математичні моделі функціонування і розвитку виробничих систем на базі методології оптимального агрегування: монографія. Вінниця: ВНТУ 2018; с 308. ISBN 978–966–641–731–5.5.

98. Грищенко І. В. Живучість інформаційних систем. Стан, досягнення та перспективи інформаційних систем і технологій. Матеріали XVI Всеукр. наук.-техн.

конф. молодих вчен., асп. та студ., Одеса, 25-26 квіт. 2015; Одес. нац. акад. харч. Технологій, Одеса: ОНАХТ, 2016; с 83-84.

99. Додонов, О.Г.; Ланде, Д.В. Мережева модель структурної живучості. Реєстрація, зберігання і обробка даних 2021, том 23, № 1, с 15-21.

100. Додонов, А.Г.; Ландэ, Д.В. Живучесть информационных систем. Наукова думка: Київ, 2011; с 256. ISBN 978-966-00-0973-9

101. Додонов, А.Г.; Флейтман, Д.В. Корпоративные информационные системы: обеспечение живучести. Математичні машини і системи 2005, № 4, с 118-130.

102. ДСТУ ISO/IEC 2382-14:2005 Інформаційні технології. Словник термінів. Частина 14. Безвідмовність, ремонтопридатність і готовність.

103. ДСТУ ISO/IEC 15288:2005 Інформаційні технології. Процеси життєвого циклу системи (ISO/IEC 15288:2002, IDT).

104. ДСТУ ISO/IEC 13335-1:2004 Інформаційні технології. Методи захисту. Керування інформацією й безпекою технології комунікацій. Частина 1. Поняття й моделі для інформації й керування безпекою технології комунікацій.

105. ДСТУ ISO/IEC 2382-18:2005 Інформаційні технології. Словник термінів. Частина 18. Розподілене оброблення даних.

106. ДСТУ ISO/IEC 2382:2017 (ISO/IEC 2382:2015, IDT) Інформаційні технології. Словник термінів.

107. ДСТУ. Надійність техніки. Терміни та визначення. ДСТУ 2860-94.

108. ЕЖНВ: Крупнейшие кибератаки против Украины с 2014 года. Инфографика. Новое время [online]; 2017, 3193. https://nv.ua/ukraine/events/krupnejshie-kiberataki-protiv-ukrainy-s-2014-goda-infografika-1438924.html (дата звернення Травень 27, 2022)

109. Закон України: Про захист інформації в інформаційно телекомунікаційних системах №80 від 05.07.1994. Відомості Верховної Ради України. 1994, №31, зі змінами та доповненнями.

110. Использование уязвимостей для сброса базы данных. 2017. https://coderlessons.com/articles/programmirovanie/ispolzovanie-uiazvimostei-sql-inektsii-dlia-sbrosa-bazy-dannykh (дата звернення Травень 27, 2022)

111. Інформаційні технології. Словник термінів. Частина 14. Безвідмовність, ремонтопридатність та готовність (ISO/IEC 2382 – 14:1997, IDT): ДСТУ 2668 – 2005. Держспоживстандарт України: Київ, 2007; с 20.

112. Комп'ютерні віруси та їх основна характеристика. https://sites.google.com/site/diresideinaction/komp-uterni-virusi-ta-ieh-osnovna-harakteristika (дата звернення Травень 27, 2022)

113. Корченко, А.О. Методи ідентифікації аномальних станів для систем виявлення вторгнень. Автореферат дисертації д-ра техн. наук: 05.13.21, Національний авіаційний університет, Київ, 2019, с 40.

114. Ланде, Д.В. Методи підвищення живучості інформаційної складової корпоративних інформаційно-аналітичних систем підтримки прийняття рішень. Реєстрація, зберігання і обробка даних 2012, том 14, №2, с 48-58.

115. Лукова-Чуйко, Н.В. Методологічні основи забезпечення функціональної стійкості розподілених інформаційних систем до кібернетичних загроз: автореферат дисертації д-ра техн. наук: 05.13.06, Державний університет телекомунікацій, Київ, 2018, с. 40.

116. Мудла, Б.Г.; Єфімова, Т.І.; Рудько, Р.М. Гарантоздатність як фундаментальний узагальнюючий та інтегруючий підхід. Математичні машини і системи 2010, № 2, с 148 – 165.

117. Мухін, В.Є.; Ткач, М.М.; Корнага, Я.І.; Мостовий, Є.О.; Герасименко О.Ю. Структурна модель інтелектуального агента для підтримки захищеної обробки даних в гетерогенних розподілених системах. Наукові записки Українського науково-дослідного інституту зв'язку 2016, №2, с 37-43.

118. Науменко, Т.О.; Черномаз, В.С. Аналіз застосування детектору SQL ін'єкцій побудованого на основі штучного інтелекту у безсерверній архітектурі. Кібернетика та комп'ютерні технології: Зб. наук. пр. 2021, №2, с 85-89.

119. Нічепорук, А.О.; Стецюк, М.В.; Сорочинський, О.Ю.; Шаповалов Ф.В. Система для виявлення шкідливого програмного забезпечення на основі дослідження структурних особливостей виконуваних файлів. Матеріали Міжнародної науково-практичної конференції «Інформаційні технології та взаємодії», Київ, Україна, листопад 20-21, 2018, с 291-292.

120. Недашківський, О.М. Планування та проєктування інформаційних систем. Київ, 2014, с 215.

121. Одарченко, Р.С.; Самойлик, Є.О.; Сімахін, В.М.; Боровик, В.О.; Тимчишин, Р.М. Криптосемантична система захисту текстової інформації. Control systems & computers 2020, № 1, с 35-46.

122. Петрівський, В.Я.; Шевченко, В.Л.; Бичков, О.С.; Сініцин, І.П. Інформаційна технологія забезпечення живучості сенсорних мереж. Проблеми програмування 2021, № 4, с 62-69.

123. Пітух, І.Р.; Возна, Н.Я. Способи організації руху моніторингових, інтерактивних і діалогових даних у структурах розподілених комп'ютерних систем. Науковий вісник НЛТУ України 2021, т. 31, № 3, с 101-108.

124. Пунда, С.Ю. Системи збереження даних для IT інфраструктури. Проблеми програмування 2020, № 2-3, с 82-93.

125. Романов, В.О.; Галелюка, І.Б.; Остапенко, В.О. Вимоги до забезпечення функціональної та інформаційної безпеки бездротових сенсорних мереж. Комп'ютерні засоби, мережі та системи 2017, №16, с 106-117.

126. Komar, M.; Golovko, V.; Sachenko A.; Bezobrazov, S. Development of neural network immune detectors for computer attacks recognition and classification / 2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2013; p 665-668, doi: 10.1109/IDAACS.2013.6663008.

127. Стецюк, М.В.; Стецюк, В.М.; Савенко, О.С. Модель архітектури автоматизованих інформаційних систем супроводу фінансово-господарських процесів та підтримки управлінських рішень в закладах вищої освіти. Вимірювальна та обчислювальна техніка в технологічних процесах 2019, №2, с 91 - 98. http://elar.khnu.km.ua/jspui/handle/123456789/9220

128. Стецюк, М.В.; Стецюк, В.М.; Савенко, О.С. Модель архітектури автоматизованих інформаційних систем супроводу фінансово-господарських процесів у корпоративних мережах в умовах впливу зловмисних дій / Актуальні проблеми комп'ютерних наук. Збірник наукових праць за матеріалами XII всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2020», Хмельницький: ХНУ, 2020, с 288-291.

129. Стецюк, М.В.; Каштальян, А.С.; Грибинчук, В.І. Архітектура спеціалізованих інформаційних систем з врахуванням вимог живучості та відмовостійкості в умовах впливів зловмисного програмного забезпечення. Вимірювальна та обчислювальна техніка в технологічних процесах 2020, №2, с 69-77. https://doi.org/10.31891/2219-9365-2020-66-2-12

130. Стецюк, М.В.; Горошко, А.В.; Савенко, Б.О. Модель забезпечення живучості та відмовостійкості спеціалізованих інформаційних технологій в умовах руйнуючого впливу зловмисного програмного забезпечення. Вимірювальна та обчислювальна техніка в технологічних процесах 2020, №1, с. 97-103. https://doi.org/10.31891/2219-9365-2020-65-1-15

131. Стецюк, М.В.; Савенко, О.С.; Стецюк, В.М. Модель архітектури автоматизованих інформаційних систем супроводу фінансово-господарських процесів та підтримки управлінських рішень в закладах вищої освіти. Тези II Всеукраїнської науково-практичної конференції здобувачів вищої освіти й молодих учених "Комп'ютерна інженерія і кібербезпека: досягнення та інновації", м. Кропивницький, Україна, листопад 25–27, 2020; Кропивницький: ЦНТУ, 2020; с 34 - 35.

132. Стецюк, М.В. Метод забезпечення захисту інформації в спеціалізованих інформаційних технологіях при впливах зловмисного програмного забезпечення. Вимірювальна та обчислювальна техніка в технологічних процесах 2021, №2, с 57-68. https://doi.org/10.31891/2219-9365-2021-68-2-7

133. Стецюк, М.В.; Каштальян, А.С. Абстрактна модель впливів зловмисного програмного забезпечення та метод забезпечення відмовостійкості спеціалізованих інформаційних технологій. Вісник Хмельницького національного університету 2022, №1, с 30-42. https://doi.org/10.31891/2307-5732-2022-305-1-31-42

134. Стецюк, М.В.; Стецюк, В.М.; Нічепорук, А.А.; Савенко, Б.О. А. с. 112335, Україна. Комп'ютерна програма «Розподілена інформаційна система з підсистемами забезпечення відмовостійкості, живучості та захисту інформації». Дата реєстрації 14.03.2022.

135. Сучасні антивірусні програми та принцип їх роботи – Безпечний Інтернет. https://sites.google.com/site/bezpecnijinternet1999/sucasni-antivirusni-programi-ta-princip-ieh-roboti (дата звернення Травень 29, 2022)

136. Федухин, А.В.; Пасько, В.П.; Муха, А.А. К вопросу моделирования надежности восстанавливаемой квазимостиковой структуры с учетом тренда параметров надежности составных частей. Математичні машини і системи 2016, № 1, с 158–167.

ДОДАТОК А.
СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

Наукові праці, в яких опубліковані основні наукові результати дисертації

1. Стецюк, М.В.; Стецюк, В.М.; Савенко, О.С. Модель архітектури автоматизованих інформаційних систем супроводу фінансово-господарських процесів та підтримки управлінських рішень в закладах вищої освіти. Вимірювальна та обчислювальна техніка в технологічних процесах 2019, №2, с 91 - 98. http://elar.khnu.km.ua/jspui/handle/123456789/9220

2. Стецюк, М.В.; Горошко, А.В.; Савенко, Б.О. Модель забезпечення живучості та відмовостійкості спеціалізованих інформаційних технологій в умовах руйнуючого впливу зловмисного програмного забезпечення. Вимірювальна та обчислювальна техніка в технологічних процесах 2020, №1, с. 97-103. https://doi.org/10.31891/2219-9365-2020-65-1-15

3. Стецюк, М.В.; Каштальян, А.С.; Грибинчук, В.І. Архітектура спеціалізованих інформаційних систем з врахуванням вимог живучості та відмовостійкості в умовах впливів зловмисного програмного забезпечення. Вимірювальна та обчислювальна техніка в технологічних процесах 2020, №2, с 69-77. https://doi.org/10.31891/2219-9365-2020-66-2-12

4. Стецюк, М.В. Метод забезпечення захисту інформації в спеціалізованих інформаційних технологіях при впливах зловмисного програмного забезпечення. Вимірювальна та обчислювальна техніка в технологічних процесах 2021, №2, с 57- 68. https://doi.org/10.31891/2219-9365-2021-68-2-7

5. Стецюк, М.В.; Каштальян, А.С. Абстрактна модель впливів зловмисного програмного забезпечення та метод забезпечення відмовостійкості спеціалізованих інформаційних технологій. Вісник Хмельницького національного університету 2022, №1, с 30 - 42. https://doi.org/10.31891/2307-5732-2022-305-1-31-42

6. Stetsiuk, M.V.; Kashtalian, A.S. The methods of ensuring fault tolerance, survivability and protection of information of specialized information technologies under the influence of malicious software. Computer Systems And Information Technologies (Комп'ютерні системи та інформаційні технології) 2022, №1, pp 36 - 44. http://csitjournal.khmnu.edu.ua/index.php/csit/article/view/126/78

Праці, які засвідчують апробацію матеріалів дисертації

1. Stetsyuk, M.; Bedratyuk, L.; Savenko, B.; Stetsyuk, V.; Savenko O. Providing the Resilience and Survivability of Specialized Information Technology Across Corporate Computer Networks. 1st International Workshop on Intelligent Information Technologies & Systems of Information Security. Khmelnytskyi, Ukraine, June 10-12, 2020; CEUR Workshop Proceedings, 2020; vol 2623, pp 219-238. (*Scopus*)

2. Stetsyuk, M.V.; Stetsyuk, V.M.; Savenko, B.O.; Savenko, O.S.; Dobrowolski M. Implementation of contropl by parameters of client automated workplaces of specialized information systems for neutralization malware. 2st International Workshop on Intelligent Information Technologies & Systems of Information Security. Khmelnytskyi, Ukraine, March 24-26, vol 2021; CEUR Workshop Proceedings, 2021; 2853, pp 340–352. (*Scopus*)

3. Нічепорук, А.О.; Стецюк, М.В.; Сорочинський, О.Ю.; Шаповалов Ф.В. Система для виявлення шкідливого програмного забезпечення на основі дослідження структурних особливостей виконуваних файлів. Матеріали Міжнародної науково-практичної конференції «Інформаційні технології та взаємодії», Київ, Україна, листопад 20-21, 2018, с 291-292.

4. Стецюк, М.В.; Стецюк, В.М.; Савенко, О.С. Модель архітектури автоматизованих інформаційних систем супроводу фінансово-господарських процесів у корпоративних мережах в умовах впливу зловмисних дій / Актуальні проблеми комп'ютерних наук. Збірник наукових праць за матеріалами XII

всеукраїнської науково-практичної конференції «Актуальні проблеми комп’ютерних наук АПКН-2020», Хмельницький: ХНУ, 2020, с 288-291.

5. Stetsiuk, M.; Nicheporyk, A.; Savenko, B. Ensuring the Fault Tolerance And Survivability of Specialized Information Technologies in Corporate Computer Networks Under the Influence of Malicious Software. Proceedings of VII International conference “Information Technology and Interactions” (IT&I-2020), Taras Shevchenko National University, Kyiv, December 02-04, 2020; Snytyuk, V., Anisimov, A., Krak, I., Nikitchenko, M. Eds.; pp 105-106.

6. Стецюк, М.В.; Савенко, О.С.; Стецюк, В.М. Модель архітектури автоматизованих інформаційних систем супроводу фінансово-господарських процесів та підтримки управлінських рішень в закладах вищої освіти. Тези II Всеукраїнської науково-практичної конференції здобувачів вищої освіти й молодих учених “Комп’ютерна інженерія і кібербезпека: досягнення та інновації”, м. Кропивницький, Україна, листопад 25–27, 2020; Кропивницький: ЦНТУ, 2020; с 34 - 35.

Публікації, які додатково відображають наукові результати дисертації

1. Стецюк, М.В.; Стецюк, В.М.; Нічепорук, А.А.; Савенко, Б.О. А. с. 112335, Україна. Комп'ютерна програма «Розподілена інформаційна система з підсистемами забезпечення відмовостійкості, живучості та захисту інформації». Дата реєстрації 14.03.2022.

ДОДАТОК Б.
АКТИ ВПРОВАДЖЕННЯ

«Затверджую»

Проректор з науково-педагогічної роботи,

_____ Лопатовський В.Г

«9» травня 2022 р.

АКТ

про впровадження результатів дисертаційної роботи

Стецюка Миколи Васильовича

«Методи та засоби забезпечення відмовостійкості та живучості спеціалізованих інформаційних технологій в умовах впливів зловмисного програмного забезпечення»

Ми, комісія в складі: декана факультету інформаційних технологій д.т.н., проф. Савенка О.С., завідувача кафедри комп'ютерної інженерії та інформаційних систем д.т.н., проф. Говорущенко Т.О., професора кафедри комп'ютерної інженерії та інформаційних систем д.т.н., проф. Лисенка С.М. склала акт про те, що результати дисертаційної роботи Стецюка М.В. впроваджені та використовуються в навчальному процесі на кафедрі комп'ютерної інженерії та інформаційних систем для спеціальності 123 «Комп'ютерна інженерія» галузі знань 12 «Інформаційні технології», зокрема, при викладанні освітніх компонентів «Технічна діагностика і надійність комп'ютерних пристроїв та систем» та «Безпека та захист комп'ютерних систем».

При викладанні цих дисциплін використовувалися наступні матеріали досліджень, отримані Стецюком М.В. особисто:

1) алгоритми забезпечення відмовостійкості, живучості та захисту інформації в спеціалізованих інформаційних технологій (ІТ), в яких поєднані та інтегровані механізми забезпечення відмовостійкості, живучості та захисту інформації згідно їх збігів в станах при реагуванні на впливи зловмисного програмного забезпечення (ЗПЗ) та комп'ютерних атак;

2) метод забезпечення відмовостійкості, живучості та захисту інформації в спеціалізованих ІТ, який ґрунтується на поєднанні та інтегруванні в ІТ механізмів забезпечення відмовостійкості, живучості та захисту інформації згідно їх збігів в станах при реагуванні на впливи ЗПЗ та комп'ютерних атак.

Отримані матеріали досліджень надали можливість розробити лабораторні практикуми з використанням розроблених алгоритмів для забезпечення відмовостійкості, живучості та захисту інформації в спеціалізованих ІТ.

_____ Савенко О.С.

_____ Говорущенко Т.О.

_____ Лисенко С.М.

«Затверджую»

Проректор з науково-педагогічної роботи
Хмельницького національного університету
доктор економічних наук
професор Віктор НИЖНИК

«6» травня 2022 р.

АКТ

про впровадження результатів дисертаційної роботи

Стецюка Миколи Васильовича

«Методи та засоби забезпечення відмовостійкості та живучості спеціалізованих інформаційних технологій в умовах впливів зловмисного програмного забезпечення»

Практичні результати дисертаційної роботи аспіранта кафедри комп'ютерної інженерії та інформаційних систем Хмельницького національного університету Стецюка М.В. впроваджені в бухгалтерії Хмельницького національного університету.

В процесі впровадження були використанні наступні результати дисертаційної роботи, одержані особисто Стецюком М.В.:

1) метод забезпечення захисту інформації в спеціалізованих інформаційних технологіях, заснований на поєднані із організаційними заходами інтегрованого в інформаційних технологіях залучення механізмів сегментування мережі, криптографічного захисту, двохфакторної автентифікації програмного забезпечення, хибних об'єктів атаки, резервного копіювання з територіальним розмежуванням місць зберігання копій;

2) метод забезпечення відмовостійкості, живучості та захисту інформації в спеціалізованих інформаційних технологіях, заснований на поєднані та інтегруванні в інформаційні технології механізмів забезпечення відмовостійкості, живучості та захисту інформації згідно їх збігів в станах при реагуванні на впливи зловмисного програмного забезпечення та комп'ютерних атак.

Впровадження в бухгалтерії ХНУ результатів дисертаційної роботи надало можливість покращити стійкість інформаційних технологій, що використовуються нею, до впливів комп'ютерних атак та зловмисного програмного забезпечення.

Начальник ЦЦТ _____ к.т.н. Кльоц Ю.П.

«Затверджую»

Директор ТОВ «ІТТ»

_____ В.С. Сімогук

«11» травня 2022 р.

АКТ

про впровадження результатів дисертаційної роботи

Стецюка Миколи Васильовича

«Методи та засоби забезпечення відмовостійкості та живучості спеціалізованих інформаційних технологій в умовах впливів зловмисного програмного забезпечення»

Комісія в складі:

Технічний директор	Веремеєнко В. А.
Начальник відділу продажу комп'ютерної техніки	Вінтер Ю.Г.
Адміністратор	Ладунець В.І.

склала акт про впровадження результатів дисертаційної роботи аспіранта кафедри комп'ютерної інженерії та інформаційних систем Хмельницького національного університету Стецюка М.В. на «ІТТ», в тому, що він проводив роботу по впровадженню методів та засобів забезпечення відмовостійкості та живучості спеціалізованих інформаційних технологій (ІТ) в умовах впливів зловмисного програмного забезпечення (ЗПЗ).

В процесі вирішення науково-практичної задачі забезпечення відмовостійкості та живучості спеціалізованих інформаційних технологій в умовах впливів зловмисного програмного забезпечення, яке грунтується на використанні основних положень абстрактної алгебри, теорії комп'ютерних мереж, теоретичних основ інформаційних технологій, методів захисту інформації в комп'ютерних системах, методів проектування інформаційних систем, Стецюком М.В.. було особисто отримано і використано на «ІТТ» такі результати:

1) метод забезпечення відмовостійкості інформаційних технологій згідно інтегрованого залучення компонентів резервування та надмірностей; метод забезпечення

живучості спеціалізованих ІТ згідно аналізу маркерів та збереженої інформації для самодослідження; метод забезпечення захисту інформації в спеціалізованих ІТ, суть якого полягає в поєднані із організаційними заходами інтегрованого в ІТ залучення механізмів сегментування мережі, криптографічного захисту, двохфакторної автентифікації програмного забезпечення, хибних об'єктів атаки, резервного копіювання з територіальним розмежуванням місць зберігання копій; метод забезпечення відмовостійкості, живучості та захисту інформації в спеціалізованих ІТ, суть якого полягає в поєднані та інтегруванні в ІТ механізмів забезпечення відмовостійкості, живучості та захисту інформації згідно їх збігів в станах при реагуванні на впливи ЗПЗ та комп'ютерних атак.

2) проведені експериментальні дослідження за участі Стецюка М.В. надали можливість перевірити ефективність розроблених алгоритмів та підсистем забезпечення відмовостійкості, живучості та захисту інформації в спеціалізованих ІТ, в яких поєднані та інтегровані механізми забезпечення відмовостійкості, живучості та захисту інформації згідно їх збігів в станах при реагуванні на впливи ЗПЗ та комп'ютерних атак;

3) експериментальні дослідження показали, що засоби, в які імплементовано розроблені методи, надають можливість покращити стійкість засобів до впливів ЗПЗ та комп'ютерних атак.

Отримані результати дозволили покращити стійкість засобів до впливів ЗПЗ та комп'ютерних атак.

Веремеєнко В. А.

Вінтер Ю.Г.

Ладунець В.І.

«Затверджую»
Директор ТОВ «Деймос»
Пантелєєв В.І.
«29» 04 2022 р.

АКТ

про впровадження результатів дисертаційної роботи

Стецюка Миколи Васильовича

«Методи та засоби забезпечення відмовостійкості та живучості спеціалізованих інформаційних технологій в умовах впливів зловмисного програмного забезпечення»

Результати дисертаційної роботи аспіранта кафедри комп'ютерної інженерії та інформаційних систем Хмельницького національного університету Стецюка М.В. впроваджені на ТОВ «Деймос».

В процесі впровадження методів та засобів забезпечення відмовостійкості та живучості спеціалізованих інформаційних технологій (ІТ) в умовах впливів зловмисного програмного забезпечення (ЗПЗ) були використані на ТОВ «Деймос» результати, які одержані Стецюком М.В. особисто:

1) метод забезпечення відмовостійкості інформаційних технологій згідно інтегрованого залучення компонентів резервування та надмірностей; метод забезпечення живучості спеціалізованих ІТ згідно аналізу маркерів та збереженої інформації для самодослідження; метод забезпечення захисту інформації в спеціалізованих ІТ, заснований на поєднані із організаційними заходами інтегрованого в ІТ залучення механізмів сегментування мережі, криптографічного захисту, двохфакторної автентифікації програмного забезпечення, хибних об'єктів атаки, резервного копіювання з територіальним розмежуванням місць зберігання копій; метод забезпечення відмовостійкості, живучості та захисту інформації в спеціалізованих ІТ, заснований на поєднанні та інтегруванні в ІТ механізмів забезпечення відмовостійкості, живучості та захисту інформації згідно їх збігів в станах при реагуванні на впливи зловмисного програмного забезпечення (ЗПЗ) та комп'ютерних атак;

2) підсистеми забезпечення відмовостійкості, живучості та захисту інформації в спеціалізованих ІТ, в яких поєднані та інтегровані механізми забезпечення відмовостійкості, живучості та захисту інформації згідно їх збігів в станах при реагуванні на впливи ЗПЗ та комп'ютерних атак, що надають можливість покращити стійкість засобів до впливів ЗПЗ та комп'ютерних атак.

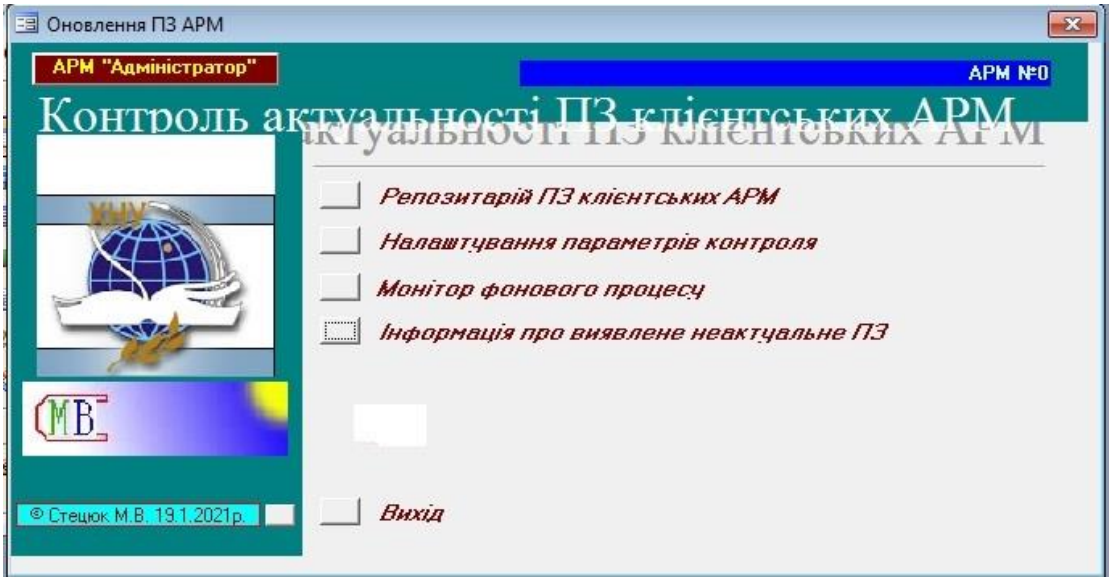
Отримані результати дозволили покращити стійкість програмних засобів, використовуваних на ТОВ «Деймос», до впливів ЗПЗ та комп'ютерних атак. Результати роботи використано для організації захисту використовуваних на ТОВ «Деймос» інформаційних технологій від впливів ЗПЗ та комп'ютерних атак.

Цей акт не є підставою для фінансових розрахунків.

Шимко І.О.

ДОДАТОК В.
ЛІСТИНГ ПРОГРАМНОГО КОДУ

Комп'ютерна програма контролю актуальності програмного забезпечення клієнтських автоматизованих робочих місць в умовах впливів зловмисного програмного забезпечення.




```
'Модуль ініціалізації
Option Compare Database
Option Explicit
Public soob1 As String
Public baza1 As Database
Public tabl1 As DAO.Recordset, tabl2 As DAO.Recordset,
                                tabl3 As DAO.Recordset, tabl4 As DAO.Recordset
Public zap1 As QueryDef, zap2 As QueryDef, zap3 As QueryDef
Public z_sql As QueryDef, z_sql2 As QueryDef, z_sql3 As QueryDef, z_sql4 As
QueryDef
Public z_SB1 As QueryDef, z_SB2 As QueryDef
Public ZapP As QueryDef, ZapT As DAO.Recordset
Public ZapQ As QueryDef
```

```
Public FB_ODBC As Workspace, FB_ODBC_SB As Workspace
Public StrConnect As String, StrConnectSB As String, FPKBD As Boolean
Public User_Rol As String
Public zzap As Connection, zsbaz As Connection
Public StatArm As Integer
Public KodRobM As Integer
Public NOrg1 As String, KNOrg2 As String
Public KOrg3 As String
Public NGRozK As String
Public NomVid As Integer
Public ZT1 As Boolean, ZT2 As Boolean, ZT3 As Boolean
Global Const MSG_T = "Увага!"
Global Const ERR_VIP = "Трапилась помилка!@@Операція не виконана!"
Global Const MSG_DR = "Розробник  Стецюк Микола Васильович."
Global Const MSG_ERR = "Система контролю виявила пошкодження даної програми.@  Причиною пошкодження " _
& "могли бути нестабільність напруги в мережі живлення, некорректне завершення роботи ком'ютера, помилка в " _
& "роботі операційної системи, помилкові дії оператора системи. Подальше використання даної програми " _
& "пов'язане з ризиком втрати або пошкодження таблиць бази даних.@  Для відновлення її роботи зверніться " _
& "до розробника:"
Global Const MSG_AVT =  "Контроль програмного забезпечення клієнтських робочих місць.@"
```

Entry point:

```
Public Function Init(pUID As String, pPWD As String, pRole As String) As Boolean
On Error GoTo Init_Err
```

```
Init = False
Set baza1 = CurrentDb
Set zap1 = baza1.CreateQueryDef("")
Set zap2 = baza1.CreateQueryDef("")
Set zap3 = baza1.CreateQueryDef("")
DoCmd.Hourglass True
StrConnect = "ODBC;DATABASE=;UID=" & pUID & ";PWD=" & pPWD & ";ROLE=" & User_Rol & ";DRIVER = IscDbc;" _
    & "CHARSET=WIN1251;READONLY = 0;NOWAIT = 0;DIALECT = 3;QUOTED = 0;DSN=OPLATANAV" & IIf(Forms!Login!pDSN, "1", "")

Set FB_ODBC = CreateWorkspace("NewSeans1", " & pUID & ", " & pPWD & ", dbUseODBC)
Set zzap = FB_ODBC.OpenConnection("HNU", , False, StrConnect)
Set ZapP = zzap.CreateQueryDef("")
Set z_sql = zzap.CreateQueryDef("")
Set z_sql2 = zzap.CreateQueryDef("")
Set z_sql3 = zzap.CreateQueryDef("")
Set z_sql4 = zzap.CreateQueryDef("")
Set ZProc = zzap.CreateQueryDef("")

FPKBD = False
If Forms!Login!pSBAZ Then
   StrConnectSB = "ODBC;DATABASE=;UID=SWM;PWD=12SWM12;ROLE=SINHRO_BD;DRIVER = IscDbc;" _
      & "CHARSET=WIN1251;READONLY = 0;NOWAIT = 0;DIALECT = 3;QUOTED = 0;DSN=SBORD"
```

```
    Set FB_ODBC_SB = CreateWorkspace("NewSeans2", "", "", dbUseODBC)
    Set zsbaz = FB_ODBC_SB.OpenConnection("SBD", , False, StrConnectSB)
    Set z_SB1 = zsbaz.CreateQueryDef("")
    Set z_SB2 = zsbaz.CreateQueryDef("")
FPKBD = True
End If

Dim ConInf As String 'Ініціалізація таблиць
Dim NamT As String, NewT As String
Dim ttabl1 As TableDef
zap1.SQL = "SELECT BSQL,PutF,NTabl,NomT,PPK FROM RabTabl WHERE Mid(NTabl,1,2)<>'ZZ';"
Set tabl1 = zap1.OpenRecordset(dbOpenSnapshot)
Do While Not tabl1.EOF
ConInf = Nz(tabl1!PutF, "")
NamT = tabl1!NTabl
NewT = CStr("T" & tabl1!NomT)
For Each ttabl1 In baza1.TableDefs
If ttabl1.Name = NewT Then
GoTo NEWTABL
End If
Next ttabl1
Set ttabl1 = baza1.CreateTableDef(NewT)
    If tabl1!BSQL = -1 Then
    ttabl1.Connect = StrConnect
    Else
    ttabl1.Connect = ";DATABASE=" & ConInf
    End If
```

```
        ttabl1.SourceTableName = NamT
        baza1.TableDefs.Append ttabl1
        If tabl1!BSQL = -1 And Len(Nz(tabl1!PPK, "")) > 0 Then
        CurrentDb.Execute "CREATE UNIQUE INDEX PRIMARYKEY ON " & NewT
& " (" & tabl1!PPK & ")"
        End If
      NEWTABL: tabl1.MoveNext
      Loop
      tabl1.Close

            zap1.SQL = "SELECT NTabl,NomT FROM RabTabl WHERE
Mid(NTabl,1,2)='ZZ' AND BSQL=-1;"
            Set tabl1 = zap1.OpenRecordset(dbOpenSnapshot)
            Do While Not tabl1.EOF
              NamT = tabl1!NTabl
              NewT = CStr("Z" & tabl1!NomT)
                For Each ZapQ In baza1.QueryDefs
                If ZapQ.Name = NewT Then
                GoTo NEWZAPR
              End If
           Next ZapQ
      Set ZapQ = baza1.CreateQueryDef(NewT)
      NEWZAPR: ZapQ.Connect = StrConnect
      ZapQ.ReturnsRecords = True
      tabl1.MoveNext
      Loop
      tabl1.Close
```

```
PotDat = Date
z_sql.SQL = "SELECT CURRENT_DATE AS PD FROM RDB$DATABASE;"
Set tabl1 = z_sql.OpenRecordset
PotDat = tabl1!PD
tabl1.Close
StatArm = 2
Init = True
soob1 = ""
DoCmd.Hourglass False
Exit Function

Init_Err:   On Error Resume Next
            DoCmd.Hourglass False
            Init = False
tabl1.Close
End Function



Function ObrErr(MyVal As Variant) As
On Error GoTo FDataErr
ObrErr = CCur(MyVal)
Exit Function
FDataErr:
ObrErr = 0
End Function
```

```
Function ZapKljucha(pnorg1 As String) As Long
ZapKljucha = 0
ZapP.SQL = "SELECT GEN_ID(" & pnorg1 & ", 1) AS ZN_GEN FROM RDB$DATABASE;"
Set ZapT = ZapP.OpenRecordset
If Not ZapT.EOF Then ZapKljucha = ZapT!ZN_GEN
ZapT.Close
End Function

Function ZamApostrof(nstr1 As String) As String
   Dim Cikl As Integer
   For Cikl = 1 To Len(nstr1)
   If Mid(nstr1, Cikl, 1) = "" Then
   ZamApostrof = ZamApostrof & """
   ElseIf Mid(nstr1, Cikl, 1) = "," Then
   ZamApostrof = ZamApostrof & "."
   ElseIf Mid(nstr1, Cikl, 1) = """" Then
   ElseIf AscB(Mid(nstr1, Cikl, 1)) = 10 Or AscB(Mid(nstr1, Cikl, 1)) = 13 Then
   Else
   ZamApostrof = ZamApostrof & Mid(nstr1, Cikl, 1)
   End If
   Next Cikl
End Function

Public Sub Zagr_SQL(p_nam As String, p_sql As String)
For Each ZapQ In baza1.QueryDefs 'Цикл по запитах
If ZapQ.Name = p_nam Then
ZapQ.SQL = p_sql
```

```
ZapQ.ReturnsRecords = True
Exit Sub
End If
Next ZapQ
End Sub
Private Sub btnCancel_Click()
Application.QUIT acQuitSaveAll
End Sub

Function ZapKDat(pnorg1 As Date, pKMis As Integer) As Date 'Çàïèò ê³íöåâî¿ äàòè
On Error Resume Next
ZapKDat = 0
ZapP.SQL = "SELECT ADDMONTH('" & pnorg1 & "', " & pKMis & ") AS KD FROM RDB$DATABASE;"
Set ZapT = ZapP.OpenRecordset
If Not ZapT.EOF Then ZapKDat = ZapT!KD
ZapT.Close
End Function

Public Function SelectDay(Optional CurrentDate As Variant) As String
On Error Resume Next
  SelectDay = ""
  If IsMissing(CurrentDate) Then CurrentDate = Date
  DoCmd.OpenForm "Calendar", , , , , acDialog, CurrentDate
  SelectDay = Form_Calendar.Tag
End Function
```

```
‘Модуль LOGIN
Private Sub btnOK_Click()
On Error Resume Next
  If Init(UCase(Me.txtUID), Nz(Me.txtPWD, "-"), UCase(User_Rol)) = False Then
  DoCmd.Hourglass False
    If Len(soob1) > 0 Then
    MsgBox "Помилка підключення до сервера!@" & soob1 & "@Зверніться до адміністратора!", , MSG_T
    Else
    MsgBox " Помилка підключення до сервера!@@Перевірте правильність введення імені та пароля!", , MSG_T
    End If
  Else
    DoCmd.OpenForm "GlavMenu"
  End If
End Sub
Private Sub Form_Activate()
DoCmd.Restore
End Sub
Private Sub Form_Load()
Call pDSN_AfterUpdate

  If Me.OpenArgs <> "" Then
    Me.txtUID = Me.OpenArgs
    Me.txtPWD.SetFocus
  Else
    Dim zap1 As QueryDef
    Dim tabl1 As Recordset
```

```
    Set baza1 = CurrentDb

    Set zap1 = baza1.CreateQueryDef("")
    zap1.SQL = "SELECT Ar3,Ar4 FROM ElKnF WHERE SID=1 AND INum=0;"
    Set tabl1 = zap1.OpenRecordset 'Çàíåñåíÿ äàíèõ ïðîãðàìè
    Me.txtUID = Nz(tabl1!Ar3, "-")
    User_Rol = Nz(tabl1!Ar4, "-")
    tabl1.Close
    Me.txtPWD = ""
    Me.txtUID.SetFocus
  End If
End Sub
Private Sub pDSN_AfterUpdate()
If Me.pDSN Then
    Me.nDSN.Caption = "Копія"
Else
    Me.nDSN.Caption = "Робоча"
End If
End Sub
```

Модуль головного меню

```
Private Sub Form_Open(Cancel As Integer)
DoCmd.Close acForm, "Login", acSaveYes
    Me.Filter = "[INum] = 0 AND [Ar1] = '1'"
    Me.FilterOn = True
End Sub
Private Sub Form_Current()
    Me.Caption = Nz(Me![TextP], "")
```

```
        L1.Caption = Nz(Me![TextP2], "")
        L2.Caption = Nz(Me![TextP2], "")
        nn1.Caption = "ÀÐÌ ¹" & KodRobM
        FillOptions
    End Sub
    Private Sub FillOptions()
        Const KolKn = 8
        Dim dbs As Database
        Dim rst As Recordset
        Dim strSQL As String
        Dim intOption As Integer
        Me![kn1].SetFocus
        For intOption = 2 To KolKn
            Me("kn" & intOption).Visible = False
            Me("n" & intOption).Visible = False
        Next intOption

        Set dbs = CurrentDb()
        strSQL = "SELECT * FROM [ElKnF] WHERE [INum] > 0 AND [SID]=" &
Me![SID]
        strSQL = strSQL & " ORDER BY [INum];"
        Set rst = dbs.OpenRecordset(strSQL)
        If (rst.EOF) Then 'True
            Me![n1].Caption = "Елементи кнопкової форми відсутні"
        Else
            While (Not (rst.EOF))
                Me("kn" & rst![INum]).Visible = True
                Me("n" & rst![INum]).Visible = True
```

```
            Me("n" & rst![INum]).Caption = Nz(rst![TextP], "")
            rst.MoveNext
        Wend
    End If
    rst.Close
    dbs.Close
End Sub
Private Function HandleButtonClick(intBtn As Integer)
    Const ZapKnF = 1
    Const zapF = 2
    Const ZapZw = 3
    Const ZapExit = 4
    Const ZapMacro = 5
    Const ZapProc1 = 6
    Const ZapProc2 = 7
    Const ZapProc3 = 8
    Const ZapCode = 9
    Const KodErr = 2501

    Dim dbs As Database
    Dim rst As Recordset
On Error GoTo W_Err
    Set dbs = CurrentDb()
    Set rst = dbs.OpenRecordset("ElKnF", dbOpenSnapshot)
    rst.FindFirst "[SID]=" & Me![SID] & " AND [INum]=" & intBtn
    If (rst.NoMatch) Then
        MsgBox "Помилка читання елементів головної форми!"
        rst.Close
```

```
    dbs.Close
    Exit Function
End If

Select Case rst![KCom]
    Case ZapKnF
        Me.Filter = "[INum] = 0 AND [SID]=" & rst![Ar1]
    Case zapF
        DoCmd.OpenForm rst!Ar1
    Case ZapZw
        DoCmd.OpenReport rst!Ar1, acPreview
    Case ZapProc1
        Call RozrZal
    Case ZapProc2
        Call NewRik
    Case ZapProc3
        Call StRik
    Case ZapExit
        Call Form_Close
        Application.QUIT acQuitSaveAll
    Case ZapMacro
        DoCmd.RunMacro rst![Ar1]
    Case ZapCode
        Application.Run rst![Ar1]
    Case Else
        MsgBox "Невідома команда!"
End Select
rst.Close
```

```
    dbs.Close
W_Exit:    Exit Function
W_Err:
    If (Err = KodErr) Then
        Resume Next
    Else
        MsgBox "Помилка при виконанні команди", vbCritical
        Resume W_Exit
    End If
End Function



Private Sub kn20_Click()
MsgBox MSG_AVT & Chr(13) & MSG_DR, , "Від автора"
End Sub

Модуль параметрів
Option Compare Database
Option Explicit
Public ZRik As Integer, DataSinhro As Date

Private Sub Form_Close()
On Error GoTo ErrZF
DoCmd.Close acForm, "UPR_SBAZ_PZ", acSaveYes
ExitZF:   Exit Sub
ErrZF:    MsgBox Err.Description
          Resume ExitZF
End Sub
```

```
Private Sub Form_Open(Cancel As Integer)
On Error GoTo Err_OF

z_sql.SQL = "SELECT P10,P4 FROM PAROTKR WHERE KROBM=1000;"
Set tabl1 = z_sql.OpenRecordset
If Not tabl1.EOF Then
ZRik = Nz(tabl1!p10, 0)
DataSinhro = Nz(tabl1!p4, CDate("1.1." & ZRik))
tabl1.Close
End If

Me.np1.Caption = "Дисковий шлях до репозитарію"
Me.np2.Caption = "Дискретність ітерацій фонового процесу, ms"
Me.np3.Caption = "Тайм-аут операцій копіювання, ms"
Me.np4.Caption = "Кількість спроб повторення операцій"
Me.np5.Caption = "Включити документування помилок"
Me.np6.Caption = "Хеш-фунція підрахунку CRC"

Exit_OF: Exit Sub
Err_OF:  MsgBox Err.Description
        Resume Exit_OF
End Sub
Private Sub kn3_Click()
On Error GoTo Err_kz1

Dim ZTranz As Boolean, ZTranzSB As Boolean
Dim FlS As Integer
```

```
soob1 = ""

If Len(Nz(Me.p1, 0)) > 0 Then soob1 = "Дисковий шлях по умовчанню до репозитарія не вказаний!"
If Nz(Me.p2, 0) = 0 Then soob1 = "Невідома дискретність фонового процесу!"
If Nz(Me.p3, 0) = 0 Then soob1 = "Не встановлено обмеження по тайм-ауту операцій копіювання!"
If Nz(Me.p4, 0) = 0 Then soob1 = "Íå âêàçàíà ê³ëüê³ñòü ñïðîá ïîâòîðåííÿ ïîâ³äîìëåíü!"
If Len(Nz(Me.p6, 0)) > 0 Then soob1 = "Íå âêàçàíà õåø-ôóíêö³ÿ ï³äàõóíêó CRC!"
If Len(Nz(Me.p6, 0)) > 6 Then soob1 = "Íåâ³äîìà õåø-ôóíêö³ÿ ï³äàõóíêó CRC!"

If Not FPKBD Then soob1 = "Íå âñ³ ïàðàìåòðè ôîíîâîãî ïðîöåñó âêàçàí³!"
If Len(soob1) > 0 Then GoTo SOOB_ERR

Dim ZTranz As Boolean, ZTranzSB As Boolean
Dim FlS As Integer

FB_ODBC.BeginTrans
ZTranz = True
z_SB1.SQL = "SELECT KZ,RSP,KPIDR,KERP FROM GLOBDAN WHERE DATIZM>='" & Me.pNS & "';"
Set tabl1 = z_SB1.OpenRecordset()
  With tabl1
Do While Not .EOF
  z_sql.SQL = "UPDATE GLOBDAN SET KERP='" & ZamApostrof(Nz(!KERP, " -")) & "',DATIZM='" & !DATIZM & "' WHERE KZ=" & !KZ & ";"
  z_sql.Execute
  If z_sql.RecordsAffected = 0 Then
```

```
        z_sql.SQL = "INSERT INTO GLOBDAN (KZ,RSP,KPIDR,KERP,DZAJ,DATIZM) VALUES (" _
            & !KZ & "," & !RSP & "," & !KPIDR & ",'" & ZamApostrof(Nz(!KERP, "-")) & "','" & !DZAJ & "','" & !DATIZM & "');"
        z_sql.Execute
        End If
      .MoveNext
      Loop
      .Close
        End With
      FB_ODBC.CommitTrans
      ZTranz = False

      Exit_kz1: Exit Sub
      Err_kz1: If ZTranz Then FB_ODBC.Rollback
            ZTranz = False
            If ZTranzSB Then FB_ODBC_SB.Rollback
            ZTranzSB = False
            MsgBox Err.Description
            Resume Exit_kz1

      End Sub

      Модуль елементів репозитарію файлів
      Option Compare Database
      Option Explicit
      Dim DatDocum As Date
      Dim KodBOp As Integer 'Код бухгалтерської операції
```

```
Dim BP0 As Integer, BP1 As Integer, BP2 As String, BP3 As String, BP4 As String, BP5 As Integer, BP6 As Boolean, _
  BP7 As Integer, BP8 As String, BP9 As String, BP10 As String
Private Sub Form_Open(Cancel As Integer)
On Error GoTo ErrVF
Me.InsideHeight = 7500
Me.InsideWidth = 11150

Call kn0_Click
Call Zagr_SQL("Z1", "SELECT NVB,NAMV FROM VIDDIL ORDER BY NAMV;")
NomVid = Me.pVd
ZT1 = True
ExitVF:   Exit Sub
ErrVF:    MsgBox Err.Description
        Resume ExitVF
End Sub
Private Sub Form_Timer()
On Error GoTo ErrVF
If ZT1 Then
ZT1 = False
NomVid = Me.pVd
Call Zagr_SQL("Z2", "SELECT KROBM,NVB,NAMRM,STARM,P14P,P14AI,P14NI,P14V,P14AR3,P14AR4,P14AR5,P14SP,PAS1,PAS2,PAS3,PAS4 FROM PAROTKR WHERE NVB=" & NomVid & ";")
Me.RecordSource = "Z2"
End If
ExitVF:   Exit Sub
```

```
ErrVF:    MsgBox Err.Description
          Resume ExitVF
End Sub
Private Sub kn0_Click()
On Error Resume Next
   BP0 = 0
Me.pp0 = BP0
   BP1 = 0
Me.pp1 = BP1
   BP2 = ""
Me.pp2 = BP2
   BP3 = ""
Me.pp3 = BP3
   BP4 = ""
Me.pp4 = BP4
   BP5 = 0
Me.pp5 = BP5
   BP7 = 0
Me.pp7 = BP7
   BP8 = ""
Me.pp8 = BP8
   BP9 = ""
Me.pp9 = BP9
   BP10 = ""
Me.pp10 = BP10
End Sub

Private Sub Kn1_Click()
```

```
On Error GoTo ErrKn1
Dim ZTranz As Boolean, RegRab As Boolean
Dim SQL1 As String

soob1 = ""
If Nz(Me.pp0, 0) = 0 Then soob1 = "Не вибрано АРМ!"
If Nz(Me.pVd, 0) = 0 Then soob1 = "Не вибрано відділ ІНФОРМАЦІЙНОЇ СИСТЕМИ!"
If Len(soob1) > 0 Then GoTo SoobErr

Me.pp3 = Left(Nz(Me.pp3, ""), 16)
Me.pp4 = Left(Nz(Me.pp4, ""), 16)
If Nz(Me.pp5, 0) = 0 Then Me.pp5 = 0
Me.pp8 = Left(Nz(Me.pp8, ""), 100)
Me.pp9 = Left(Nz(Me.pp9, ""), 25)
Me.pp10 = Left(Nz(Me.pp10, ""), 160)

   SQL1 = ""
   If Me.pp3 <> BP3 Then  'Користувач
      If Len(Me.pp4) > 0 Then
      SQL1 = ",P14AR3='" & ZamApostrof(Me.pp3) & "'"
      Else
      SQL1 = ",P14AR3=NULL"
      End If
   End If

   If Me.pp4 <> BP4 And Len(SQL1) > 0 Then 'Файл
      If Len(Me.pp4) > 0 Then
```

```
    SQL1 = SQL1 & ",P14AR4='" & ZamApostrof(Me.pp4) & "'"
    Else
    SQL1 = SQL1 & ",P14AR4=NULL"
    End If
ElseIf Me.pp4 <> BP4 Then
    If Len(Me.pp4) > 0 Then
    SQL1 = ",P14AR4='" & ZamApostrof(Me.pp4) & "'"
    Else
    SQL1 = ",P14AR4=NULL"
    End If
End If

If Me.pp5 <> BP5 And Len(SQL1) > 0 Then
  If Me.pp5 > 0 Then
  SQL1 = SQL1 & ",P14AR5=" & Me.pp5
  Else
  SQL1 = SQL1 & ",P14AR5=NULL"
  End If
ElseIf Me.pp5 <> BP5 Then
  If Me.pp5 > 0 Then
    SQL1 = ",P14AR5=" & Me.pp5
  Else
    SQL1 = ",P14AR5=NULL"
  End If
End If
If Me.pp8 <> BP8 And Len(SQL1) > 0 Then 'Файл
    If Len(Me.pp8) > 0 Then
    SQL1 = SQL1 & ",P14AI='" & ZamApostrof(Me.pp8) & "'"
```

```
    Else
    SQL1 = SQL1 & ",P14AI=NULL"
    End If
ElseIf Me.pp8 <> BP8 Then
    If Len(Me.pp8) > 0 Then
    SQL1 = ",P14AI='" & ZamApostrof(Me.pp8) & "'"
    Else
    SQL1 = ",P14AI=NULL"
    End If
End If
If Me.pp9 <> BP9 And Len(SQL1) > 0 Then 'Файл
    If Len(Me.pp9) > 0 Then
    SQL1 = SQL1 & ",P14NI='" & ZamApostrof(Me.pp9) & "'"
    Else
    SQL1 = SQL1 & ",P14NI=NULL"
    End If
ElseIf Me.pp9 <> BP9 Then
    If Len(Me.pp9) > 0 Then
    SQL1 = ",P14NI='" & ZamApostrof(Me.pp9) & "'"
    Else
    SQL1 = ",P14NI=NULL"
    End If
End If
If Me.pp10 <> BP10 And Len(SQL1) > 0 Then 'Призначення
    If Len(Me.pp10) > 0 Then
    SQL1 = SQL1 & ",P14P='" & ZamApostrof(Me.pp10) & "'"
    Else
    SQL1 = SQL1 & ",P14P=NULL"
```

```
            End If
         ElseIf Me.pp10 <> BP10 Then
            If Len(Me.pp10) > 0 Then
            SQL1 = ",P14P='" & ZamApostrof(Me.pp10) & "'"
            Else
            SQL1 = ",P14P=NULL"
            End If
         End If
      If Len(SQL1) > 0 Then
         z_sql.SQL = "UPDATE PAROTKR SET STARM=" & Me.pp1 & SQL1 & "
WHERE KROBM=" & Me.pp0 & ";"
         FB_ODBC.BeginTrans
         ZTranz = True
            z_sql.Execute
         FB_ODBC.CommitTrans
         ZTranz = False
         Me.RecordSource = Me.RecordSource
      End If

      Call kn0_Click

      Exit Sub
      SoobErr: MsgBox soob1 & "@@ ", , MSG_T
                  Exit Sub
      ErrKn1:    soob1 = "Операція не виконана!"
         If ZTranz Then FB_ODBC.Rollback
         ZTranz = False
         Resume SoobErr
```

```
End Sub

Private Sub kn4_Click()
On Error GoTo Errkn4
Dim ZTranz As Boolean
Dim MestoErr As Integer

   Dim wsp As Workspace 'Ініціалізація системних зміних
   Dim TablName As TableDef, TablName1 As TableDef
   Dim db2 As Database

MestoErr = 0
z_sql.SQL = "SELECT KROBM,NAMRM,P14AR3,P14AR4,P14AR5,P14AI,P14NI,P14P FROM PAROTKR " _
   & "WHERE  P14V = -1 And STARM >0  And NVB NOT IN( 54) ORDER BY P14SP;"
Set tabl1 = z_sql.OpenRecordset()
  With tabl1
Do While Not .EOF
If Len(Nz(!P14AR4, "")) < 2 Or Len(Nz(!P14AI, "")) < 2 Or Len(Nz(!P14NI, "")) < 5 Or Len(Nz(!P14P, "")) < 5 Then GoTo SL_I1
DoCmd.Echo True, "Інсталюється файл " & !P14NI & " АРМ №" & !KROBM & " -оператор " & Nz(!NAMRM, "-")

   MestoErr = 1
   Set wsp = DBEngine.Workspaces(0)    ' Возвращает ссылку на заданную по умолчанию рабочую область.
```

```
        Set db2 = wsp.OpenDatabase(!P14AI & !P14NI)     ' Возвращает ссылку на файл
бази
        For Each TablName In db2.TableDefs 'Цикл по іменах таблиць бази даних
            If TablName.Name = "ElKnF" Then
              Set TablName1 = baza1.CreateTableDef("T100")
              TablName1.Connect = ";DATABASE=" & !P14AI & !P14NI 'Під'єднання
таблиці локальної бази даних
              TablName1.SourceTableName = "ElKnF"
      SL_I2:  baza1.TableDefs.Append TablName1
                zap1.SQL = "UPDATE T100 SET Ar2 =" & !KROBM & ",Ar3 =" &
IIf(Len(Nz(!P14AR3, "")) > 0, "'" & Nz(!P14AR3, "-") & "'", "NULL") _
                & ",Ar4 ='" & !P14AR4 & "'" & IIf(Nz(!P14AR5, 0) > 0, ",Ar5 =" &
!P14AR5, "") & " WHERE SID=1 AND INum=0;"
                zap1.Execute 'Внесення даних настройки АРМ
              baza1.TableDefs.Delete "T100" 'Відключення таблиці
              GoTo SL_I3
            End If
          Next TablName

      SL_I3:   db2.Close 'Закриття БД, звільнення файлу для копіювання
              MestoErr = 3
              Kill !P14P & !P14NI ' Удаляет файл.
      SL_I4:   MestoErr = 4
              FileCopy !P14AI & !P14NI, !P14P & !P14NI

              MestoErr = 5 'Зняття відмітки після успішної інсталяції
              z_sql2.SQL = "UPDATE PAROTKR SET P14V=0,P14SP=0 WHERE
KROBM=" & !KROBM & ";"
```

```
                FB_ODBC.BeginTrans
                ZTranz = True
               z_sql2.Execute
                FB_ODBC.CommitTrans
                ZTranz = False

        If MestoErr = 100 Then
        SL_I5:      MestoErr = 6
                z_sql2.SQL = "UPDATE PAROTKR SET P14SP=P14SP+1 WHERE
KROBM=" & !KROBM & ";"
                FB_ODBC.BeginTrans
                ZTranz = True
               z_sql2.Execute
                FB_ODBC.CommitTrans
                ZTranz = False
        End If

        SL_I1: .MoveNext
        Loop
        .Close
          End With
        MestoErr = 10

        Me.RecordSource = Me.RecordSource
        Exit Sub

        SoobErr: MsgBox soob1 & "@@ ", , MSG_T
                 Exit Sub
```

```
Errkn4:
   If ZTranz Then FB_ODBC.Rollback
   ZTranz = False
   Select Case Err.Number
   Case 53 'Файл для видалення не знайдено
        If MestoErr = 3 Then Resume SL_I4
        GoTo ERRORIN
   Case 70 'APM працює
        If MestoErr = 3 Then Resume SL_I5
        GoTo ERRORIN
   Case 3012 'обэкт існує
        baza1.TableDefs.Delete "T100"
        Resume SL_I2
   Case 3024 'Файл не знайдено
        Resume SL_I1
   Case Else 'Інша помилка
ERRORIN:  MsgBox Err.Number & "  - " & Err.Description
          Select Case MestoErr
          Case 1 To 9
          tabl1.Close
          Case Else
          End Select
          soob1 = "Операція не виконана!"
          Resume SoobErr
   End Select
End Sub
Private Sub p0_DblClick(Cancel As Integer)
On Error Resume Next
```

```
    BP0 = Me.p0
Me.pp0 = BP0
    BP1 = Me.p1
Me.pp1 = BP1
    BP2 = Nz(Me.p2, "")
Me.pp2 = BP2
    BP3 = Nz(Me.p3, "")
Me.pp3 = BP3
    BP4 = Nz(Me.p4, "")
Me.pp4 = BP4
    BP5 = Nz(Me.p5, 0)
Me.pp5 = Me.p5
    BP7 = Me.P7
Me.pp7 = BP7
    BP8 = Nz(Me.p8, "")
Me.pp8 = BP8
    BP9 = Nz(Me.p9, "")
Me.pp9 = BP9
    BP10 = Nz(Me.p10, "")
Me.pp10 = BP10
End Sub

Private Sub p6_DblClick(Cancel As Integer)
On Error GoTo Err_kz1
Dim ZTranz As Boolean

z_sql.SQL = "UPDATE PAROTKR SET P14V=ABS(P14V)-1 WHERE KROBM="
& Me.p0 & ";"
```

```
FB_ODBC.BeginTrans
ZTranz = True
z_sql.Execute
FB_ODBC.CommitTrans
ZTranz = False
'ZT1 = True
Me.RecordSource = Me.RecordSource
Exit_kz1: Exit Sub
Err_kz1: If ZTranz Then FB_ODBC.Rollback
        ZTranz = False
        MsgBox Err.Description
        Resume Exit_kz1
End Sub

Private Sub pp8_DblClick(Cancel As Integer)
On Error Resume Next
Dim strDate As String
  strDate = SelectDay(Nz(Me.pp8, ""))
  If Len(strDate) > 0 Then
  Me.pp8 = CDate(strDate)
  Me.pp8.Requery
  End If
End Sub
Private Sub pVd_AfterUpdate()
On Error Resume Next
ZT1 = True
End Sub
```

ДОДАТОК Г.

ТАБЛИЦІ ВЗАЄМОЗВ'ЯЗКІВ

Таблиця Г.1. Впливи ЗПЗ на обладнання клієнтських АРМ (прояви живучості)

Впливи ЗПЗ		Стани складових ІС							
		Центральний процесорний пристрій	Оперативна пам'ять	Постійна пам'ять	Дискові накопичувачі	Засоби живучості	засоби захисту інформації	Мережеві пристрої	Периферійні пристрої
1	Контроль над КС. Наприклад, КС стала вузлом бот-мережі.	+ \ 1, 6	+ \ 7	-	+ \ 7	+ \ 6	+ \ 6	+ \ 1, 7	-
2	Атака та її типи.								
2.1	*Віддалене проникнення (Зомбі віруси,рекламні віруси, руткіти)*	+ \ 1	-	-	+ \ 7	-	-	+ \ 1	+ \ 1

2.2	*Локальне проникнення (файлові віруси)*	+ \ 1	+ \ 7	-	+ \ 7	-	-	-	+ \ 1
2.3	*Віддалена відмова (DOS-DDOS-атаки)*	+ \ 1,7	-	-	-	+ \ 2	+ \ 2	+ \ 1,4,7	-
2.4	*Локальна відмова (червяки, файлові віруси)*	+ \ 2	-	-	+ \ 7	+ \ 4	+ \ 2	+ \ 1	-
3	Виконання деструктивних дій ЗПЗ.	+ \ 1,2	+ \ 1,2,7	+ \ 7,8	+ \ 1,7,8	+ \ 4,6	+ \ 4	+ \ 1,4,7	+ \ 1,4,7
4	Виконання розмноження ЗПЗ.	+ \ 1	+ \ 7	-	+ \ 7	-	+ \ 4,5	-	-
5	Комбінації 1-4, що посилюють вплив на КС.	1; 3 \ 1,2,6	1; 3 \ 1,2,6	-	1; 3 \ 1,2,6	2.3; 3 \ 2,4,6	2.3; 3 \ 2,4,6	1; 2.3; 3 \ 1,4,7	-

Таблиця Г.2. Впливи ЗПЗ на обладнання сервера (прояви живучості)

Впливи ЗПЗ		Апаратні складові частини сервера ІС					
		Оперативна пам’ять	Мережеві модулі	Дискова підсистема	Підсистема переривання	BIOS система	Система живлення зі зворотним зв’язком
1	*Мережева розвідка, сканування портів*	-	+ \ Можливі наслідки 1	-	+ \ Можливі наслідки 1	-	-
2	*Віддалена відмова (DOS-DDOS-атаки)*	+ \ Можливі наслідки 1; 7	+ \ Можливі наслідки 1;3;4	+ \ Можливі наслідки 1;7	+ \ Можливі наслідки 6	-	-
3	*Віддалене проникнення (брутфорс - атака)*	+ \	+ \ Можливі наслідки-1	+ \ Можливі наслідки- 7	+ \ Можливі наслідки 6	-	-

		Можливі наслідки-1, 3, 6					
4	*Виконання деструктивних дій ЗПЗ.*	+ \ Можливі наслідки 5; 7	+ \ Можливі наслідки 1;3;4	+ \ Можливі наслідки 1;3;7;8;9	+ \ Можливі наслідки 4;6	+ \ Можливі наслідки 9	+ \ Можливі наслідки 4

Де наслідки впливів ЗПЗ в таблиці 1 та в таблиці 2:

1- зниження продуктивності

2- взаємоблокування при змаганні процесів за ресурси (пам’ять, реакція на події,)

3 - блокування доступу до ресурсів (пам’ять, периферія, мережа, ...)

4 - блокування запуску, роботи

5 - інфікування програмних файлів АРМ

6 - зменшення часу реакції на події

7 - неефективне використання ресурсів системи

8 - прискорення деградації

9 - тимчасове виведення з ладу

Таблиця Г.3. Впливи ЗПЗ на обладнання клієнтських АРМ (прояви відмовостійкості)

		Стани ІС							
Впливи ЗПЗ		старт	виконання завдань	запуск підсистем	запуск засобів відмовостійкості	запуск засобів живучості	запуск засобів захисту інформації	коректне завершення роботи	некоректне завершення роботи
			Основна подія (робота АРМ на якомусь часовому відрізку складається із множини виконаних або невиконаних завдань – функцій)	Ця подія на рівні АРМ відсутня.	Тут можуть бути 4 варіанти: А- ручний запуск, Б-автоматизований, В-автоматичний (А-програмно керований, Б-апаратнокерований)	Така подія, як запуск засобів живучості та захисту інформації в клієнтському АРМ відсутня, оскільки вони ніколи не виокремлюються в окремі підсистеми. Вони існують тільки в уяві для зручності оперування. Самі заходи (засоби) інтегровані в алгоритми виконуваних функцій АРМ (для кожної в рамках		Для ІС ці поняття розглядаються на рівні окремої функції, множина яких являє функціонал АРМ. Запуск потрібної функції (завдання) виконується під управлінням транзакції запущеної на сервері ІС.

						необхідного для її успішного виконання) і окремо не існують.			
1	Контроль над КС. Наприклад, КС стала вузлом бот-мережі.	+ Можливі наслідки-4	+\- Все залежить від закладених деструкцій. На практиці можливі наслідки – 1,2,3		А - + Б - + \ - ВА - + \ - ВБ - +	Як для стану «виконання завдань»	Як для стану «виконання завдань»	+	+ але це призведе до відкату транзації і некоректний результат буде анульовано.
2	Атака та її типи.				А - + (Можливі наслідки – немає) Б - + \ - (Можливі наслідки – 4,6) ВА- + \ -(Можливі наслідки – 4,6)				
2.1	*Віддалене проникнення (Зомбі віруси,рекламні віруси, руткіти)*	+\- Можливі наслідки-4	+ Можливі наслідки-1,2					+ Можливі наслідки-немає	
2.2	*Локальне проникнення (файлові віруси)*	+\- Можливі наслідки-4	+ Можливі наслідки-1,2,3,5						+ Можливі наслідки-2,3
2.3	*Віддалена відмова (DOS- DDOS-атаки)*	+\- Можливі наслідки-4	+ Можливі наслідки-1,3						+ Можливі наслідки-2,3

2.4	*Локальна відмова (червяки, файлові віруси)*	+\- Можливі наслідки-4	+ Можливі наслідки-1,2,4,5		ВБ - + (Мож. Наслідки – немає)				+ Можливі наслідки-2,3
3	Виконання деструктивних дій ЗПЗ.	+ Можливі наслідки – 1,2,3,4	+ Можливі наслідки – 1,2,3		А - + (Можливі наслідки – немає) Б - + \ - (Можливі наслідки – 4,6) ВА - + \ - (Можливі наслідки – 4,6) ВБ - + (Можливі наслідки – немає)			+ Якщо характер деструкції не пересікається з потребами виконуваної функції то прийдемо до успішнього завершення Можливі наслідки – немає	+ Якщо характер деструкції пересікається з потребами виконуваної функції то успішне завершення не наступить Можливі наслідки – 2,3
4	Виконання розмноження ЗПЗ.	+ Можливі наслідки – 5	+ Можливі наслідки – немає		А - + Б - + ВА - + ВБ - +			+ Можливі наслідки – немає	- Можливі наслідки – 1,2,3
5	Комбінації 1-4, що	Особливо важкі комбінації	Особливо важкі комбінації впливів: 1,3		Особливо важкі комбінації впливів: А – немає Б – 1,3			Особливо важкі комбінації впливів: 1,3	Особливо важкі комбінації впливів: 1,3

	посилюють вплив на КС.	впливів: 1,3			ВА – 1,3 ВБ – немає				

Таблиця Г.4. Впливи ЗПЗ на обладнання сервера (прояви відмовостійкості)

Впливи ЗПЗ		Складові частини сервера ІС			
2	Типи атак вірусів	SQL сервер	WEB сервер	Файловий сервер копій	
2.1	*Мережева розвідка, сканування портів*	Можливі наслідки-7	Можливі наслідки-7	Можливі наслідки - 7	
2.2	SQL Injection	Можливі наслідки-1, 3, 6	Можливі наслідки-1, 3, 6	Можливі наслідки-немає	
2.3	*Локальне проникнення (файлові віруси)*	Маловірогідна ситуація	Маловірогідна ситуація	Маловірогідна ситуація	
2.4	*Віддалена відмова (DOS-DDOS-атаки)*	Можливі наслідки - 1,3	Можливі наслідки - 1,3	Можливі наслідки - 1	
2.5	*HRS (HTTP Resource Splitting)*	Можливі наслідки-немає	Можливі наслідки - 3	Можливі наслідки-немає	
2.6	*Fishing (заманювання) - атака*	Можливі наслідки-немає	Можливі наслідки-немає	Можливі наслідки-немає	
2.7	*Scam - атака*	Можливі наслідки - 8	Можливі наслідки - 8	Можливі наслідки - 8	

Де наслідки впливів ЗПЗ в таблиці 3 та таблиці 4:

1- зниження продуктивності

2- взаємоблокування при змаганні процесів за ресурси (пам’ять, реакція на події,)

3 - блокування доступу до ресурсів (принтери, мережі, ...)
4 - блокування запуску
5 - інфікування програмних файлів АРМ
6 - зменшення часу реакції на події
7 - майбутня загроза 1, 3
8 - захоплення чужих прав доступу

ДОДАТОК Д.
БЛОК-СХЕМИ АЛГОРИТМІВ

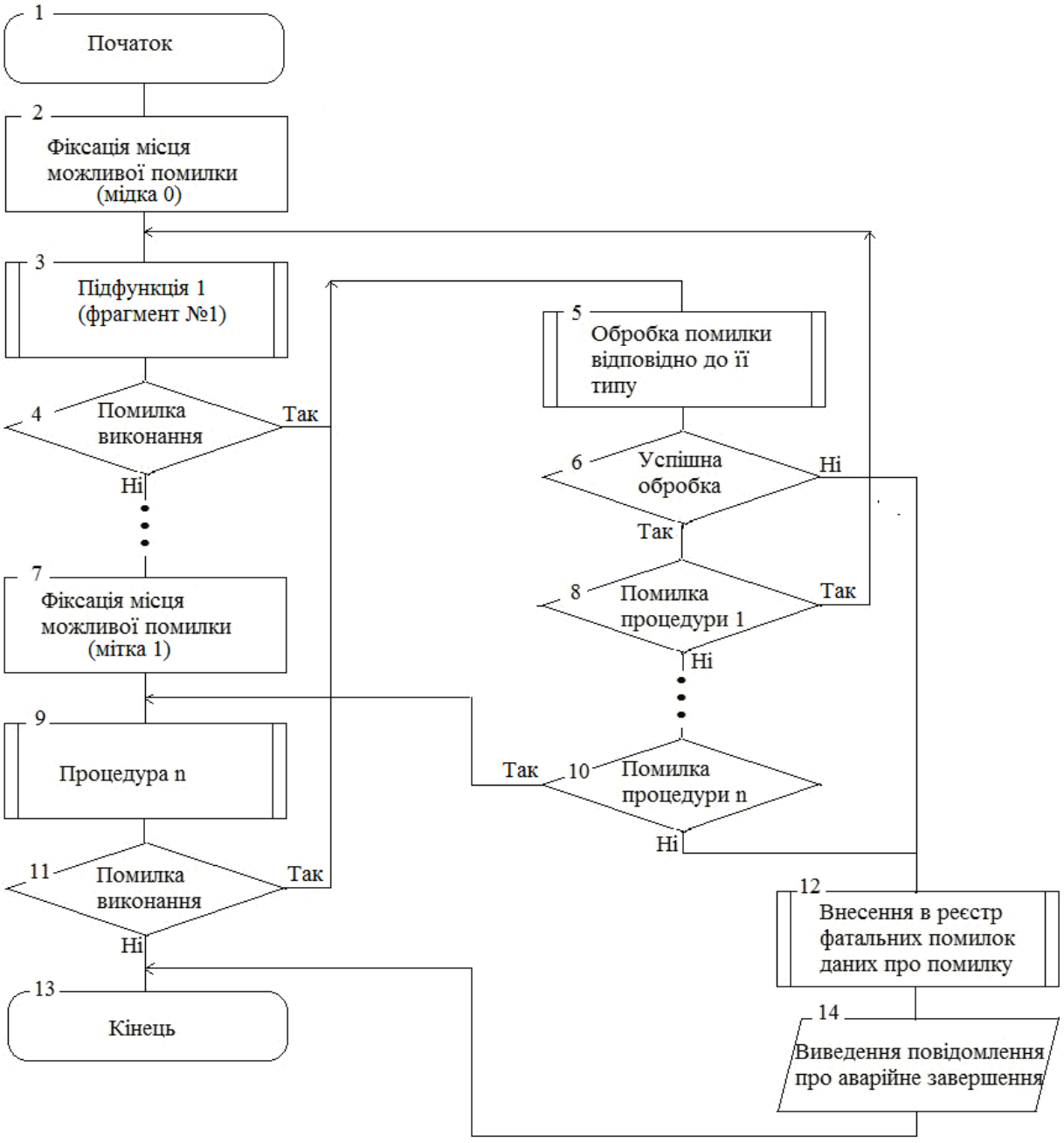


Рис. Д.1 – Блок-схема алгоритму типового програмного модуля з обробником помилок підсистеми відмовостійкості

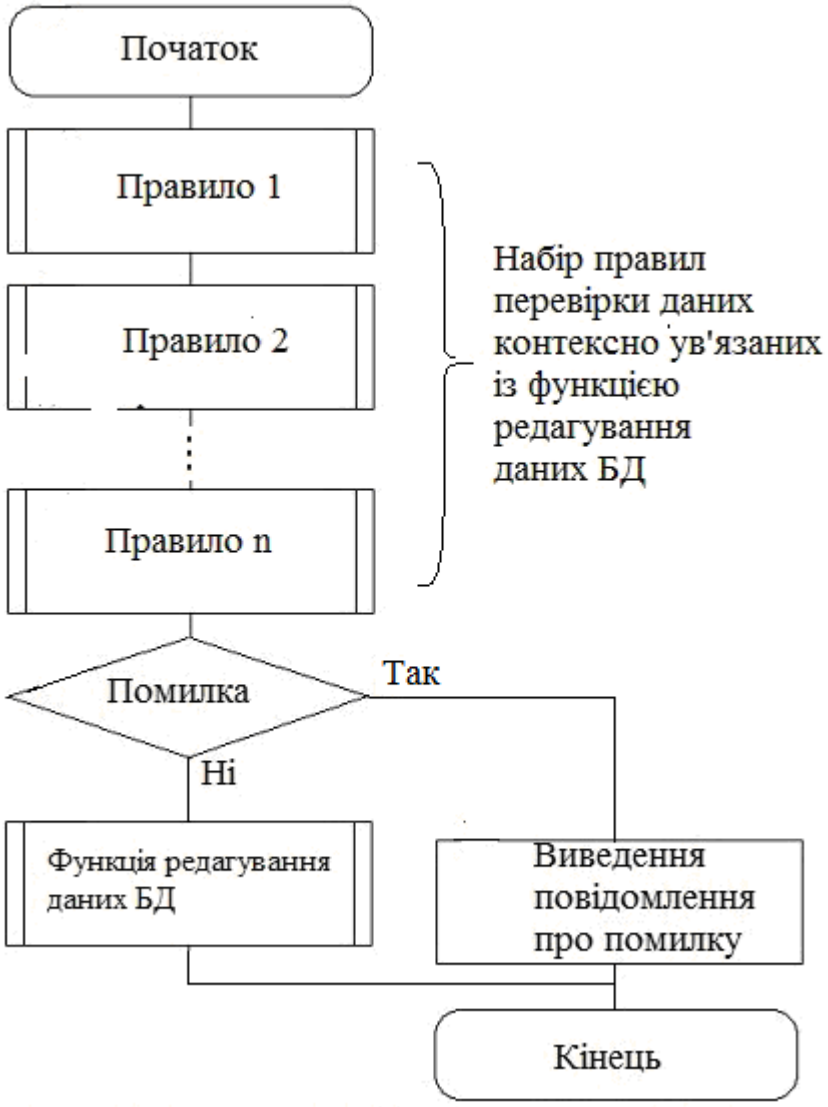


Рис. Д.2 – Блок - схема шаблона реалізації функцій редагування даних БД

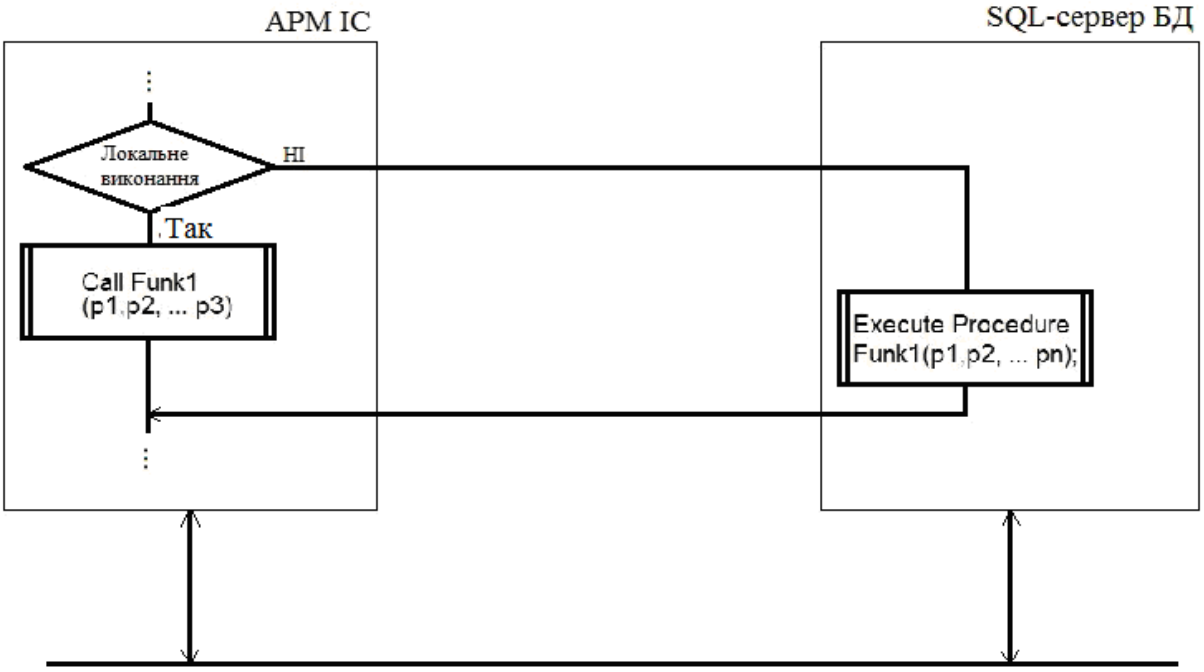


Рис. Д.3 – Модель застосування функціонального резервування розрахункових функцій ІС в середовищі клієнт- серверної архітектури

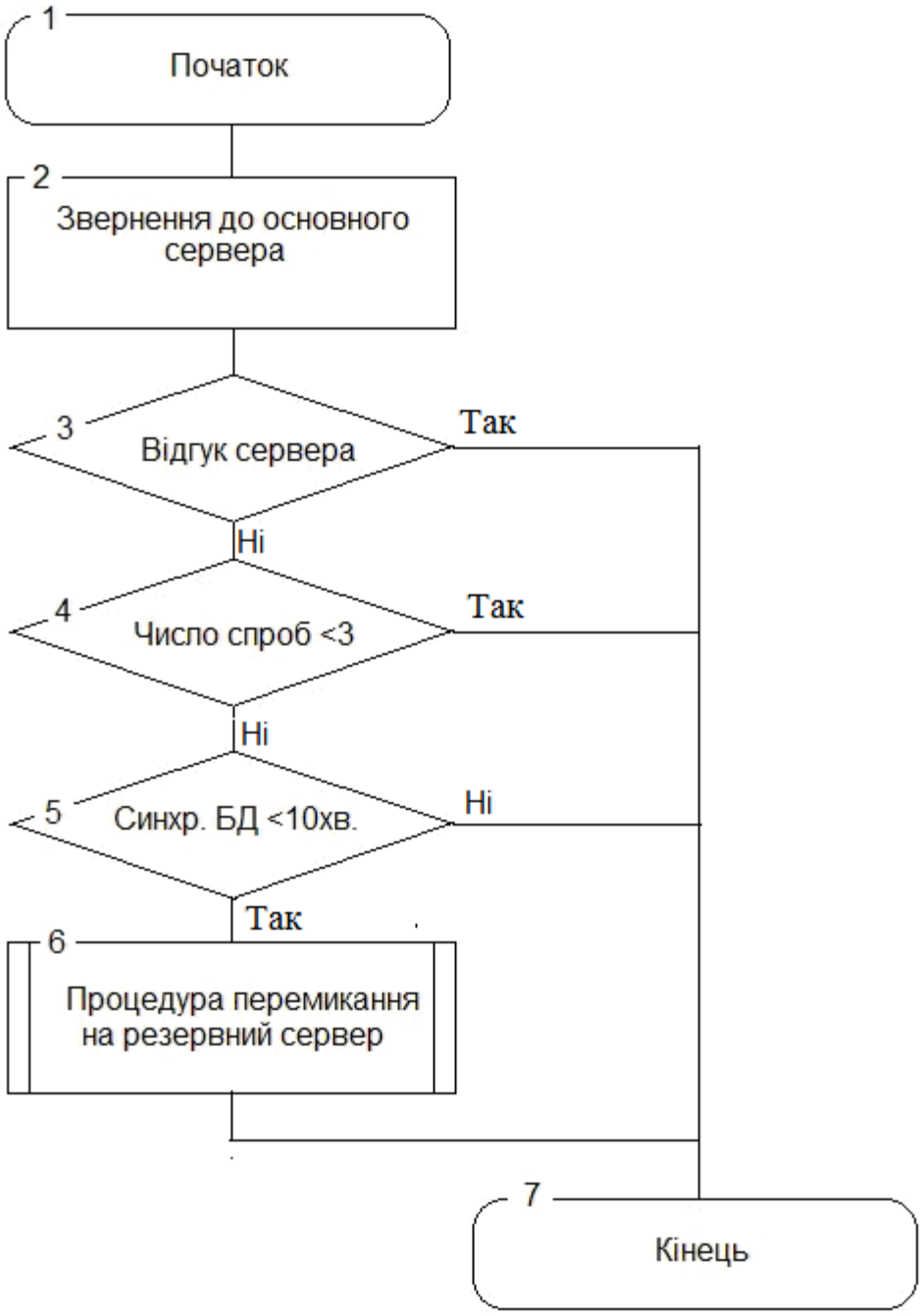


Рис. Д.4 – Алгоритм фонового процеса АРМ «Адміністратор ІС»

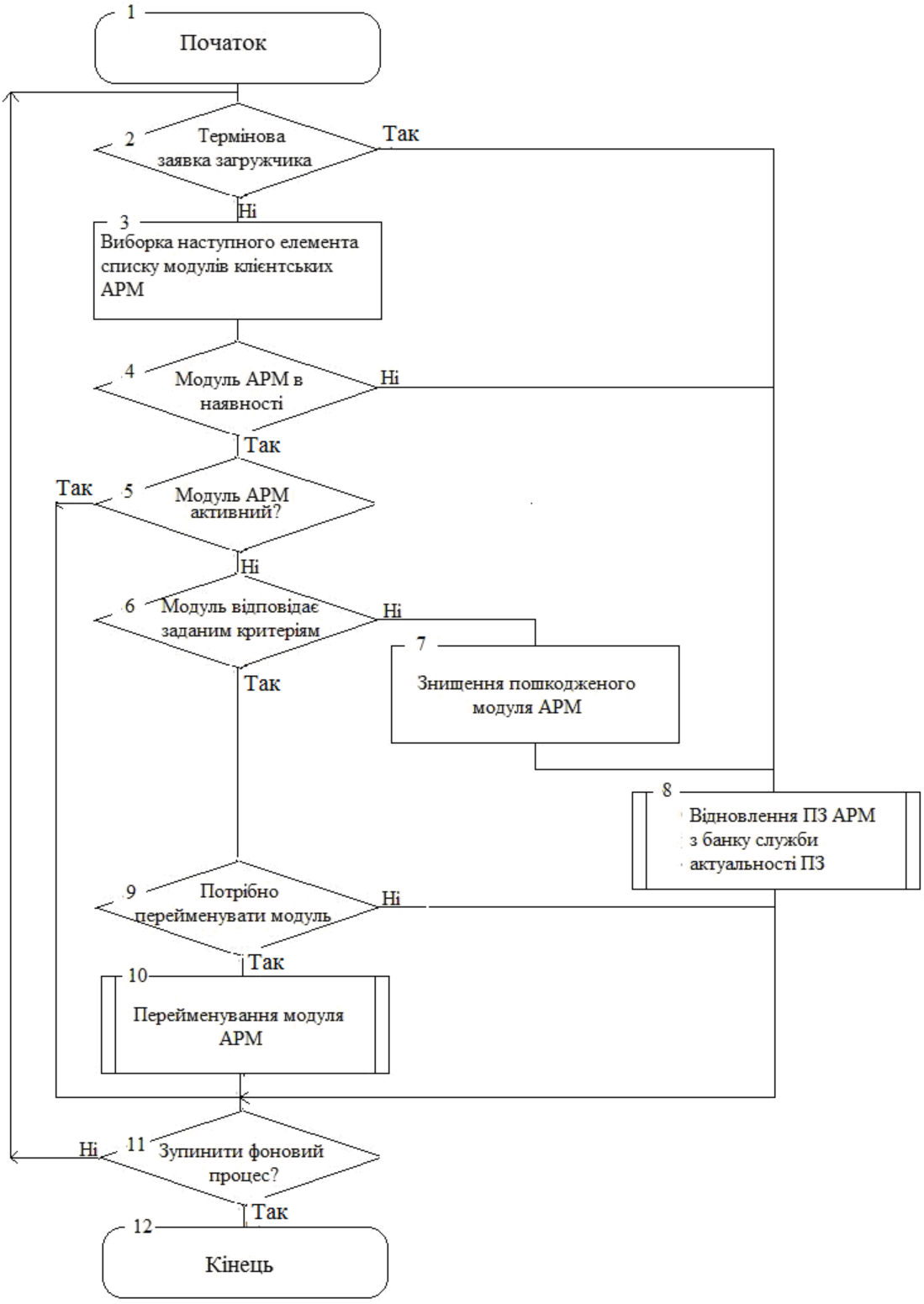


Рис. Д.5 – Алгоритм фонового процесу служби підтримки актуальності ПЗ в частині забезпечення живучості клієнтської частини спеціалізованої ІС в умовах впливів ЗПЗ

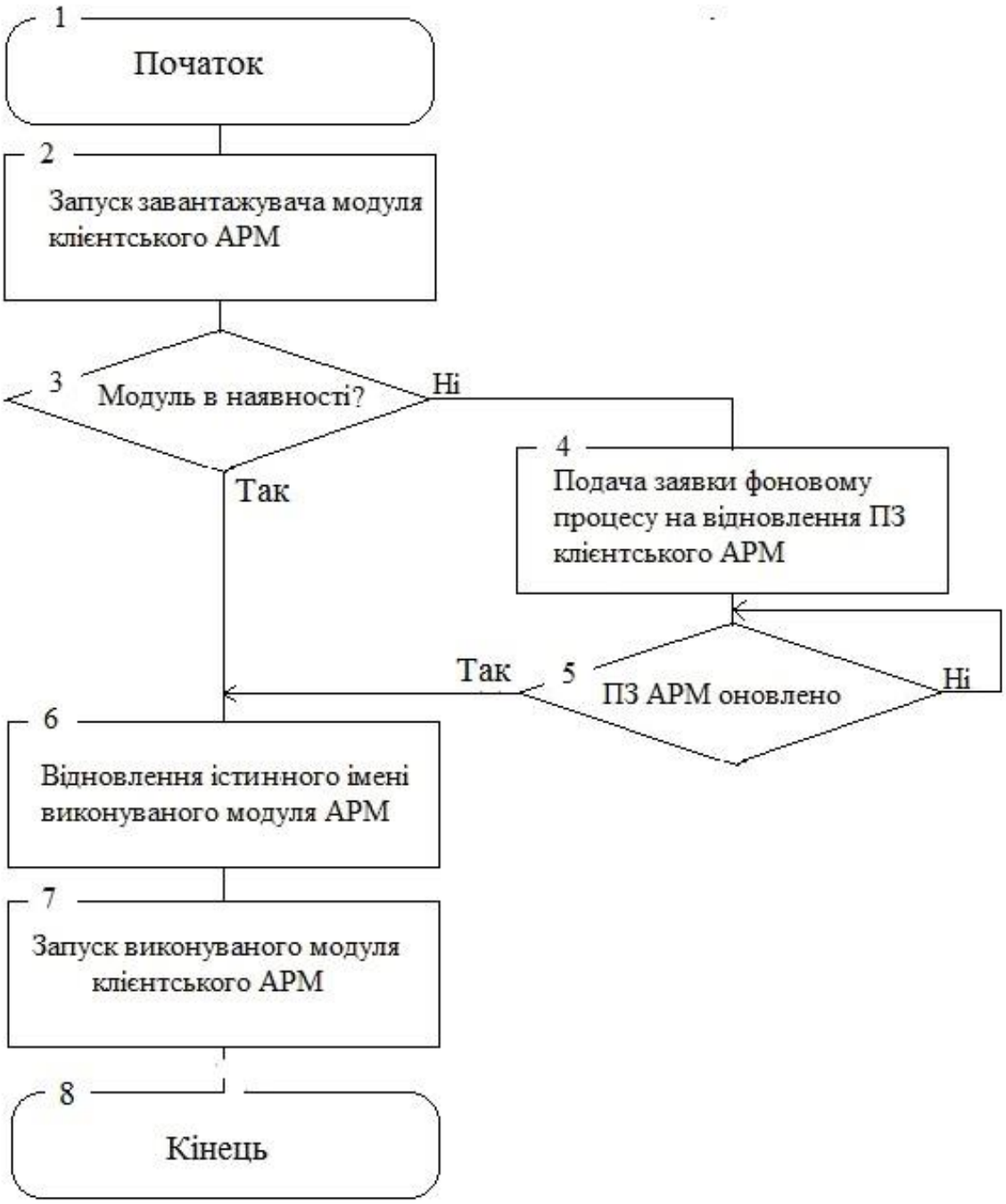


Рис. Д.6 – Алгоритм роботи завантажувача модулів АРМ служби підтримки актуальності клієнтського ПЗ

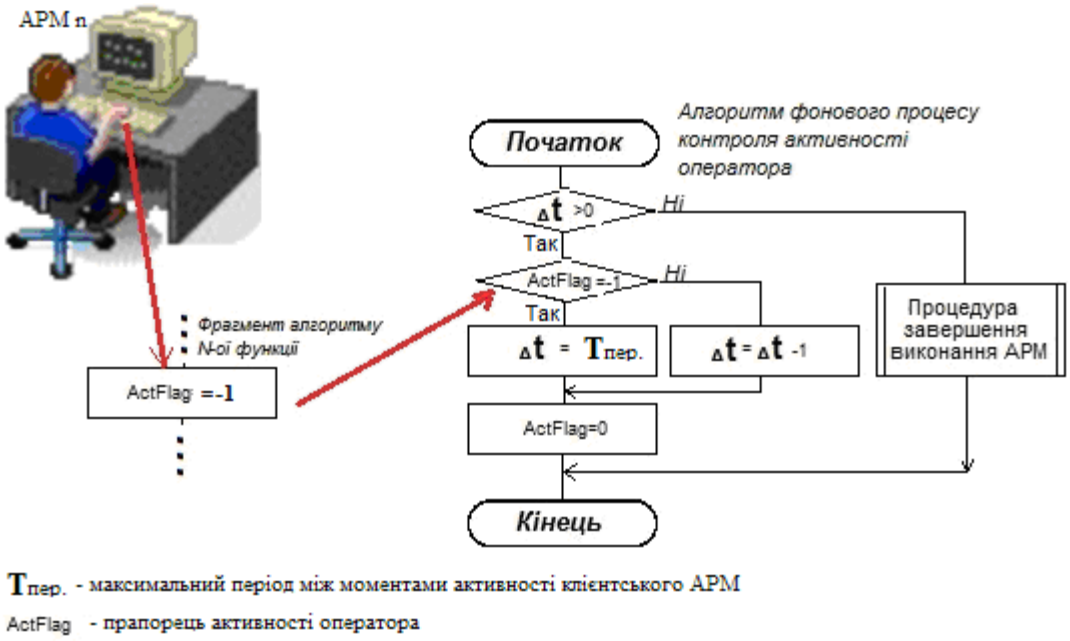


Рис. Д.7 – Схема взаємодії програмних модулів клієнтських АРМ по контролю за активністю оператора

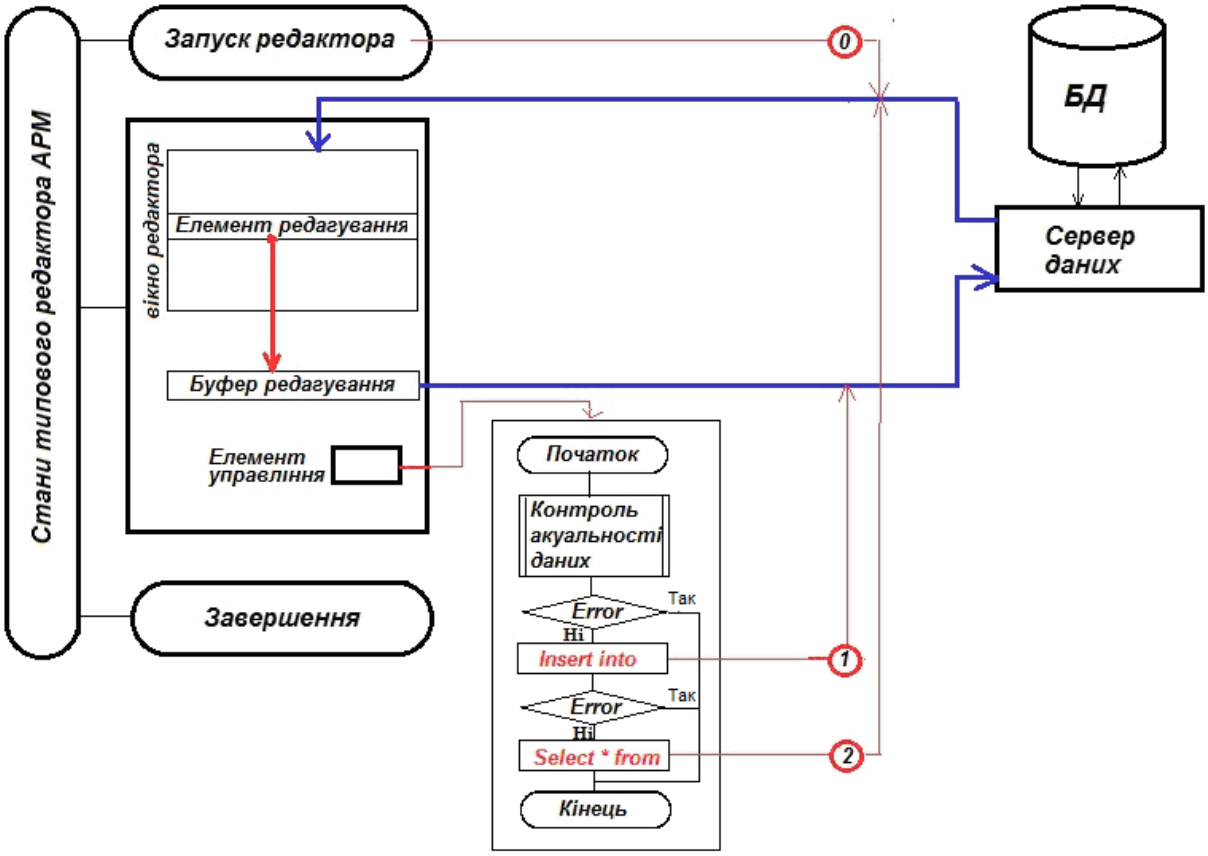


Рис. Д.8 – Схема контролю ПЗ клієнтського АРМ за діями оператора