Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“____” _____ 2022 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи CAN-мережі на основі технології CSDN”

Виконав здобувач вищої освіти
ІІ курсу, групи КІ-21М-1,4
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Олійник А.О.
«____» _____ 2022 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Доренський О.П.
«____» _____ 2022 р.
Рецензент _____
_____

м. Кропивницький

Центральноукраїнський національний технічний університет
Факультет *Механіко-технологічний*
Кафедра *Кібербезпеки та програмного забезпечення*
Рівень вищої освіти *магістр*
Галузь знань _12 _____*"Інформаційні технології"*_____
Спеціальність 123 *"Комп'ютерна інженерія"*
Освітньо-професійна (освітньо-наукова) програма _*"Комп'ютерна інженерія"*_

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
_____ Олексій СМІРНОВ
« 6 » вересня 2022 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Олійнику Артуру Олеговичу
(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи CAN-мережі на основі технології CSDN*

2. Керівник роботи *Доренський Олександр Павлович, канд. техн. наук, доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 19-13 від 17.08.2022 року

3. Строк подання студентом роботи до захисту *10.12.2022 р.*

4. Мета та завдання випускної кваліфікаційної роботи*: Метою розробки є дослідження та програмна реалізація системи CAN-мережі на основі технології CSDN*

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.	*6. Наукова новизна.*
2.Перегляд аналогічних існуючих систем.	*7. Економічна ефективність розробленої програми.*
3. Опис і обґрунтування проектних рішень.	*8. Заходи з охорони праці та техніки безпеки.*
4. Етапи програмування системи.	*9. Висновки.*
5. Впровадження системи в промислову експлуатацію	

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Наукова новизна	*1 аркуш*
Структурна схема системи	*1 аркуш*
Функціональна схема системи	*1 аркуш*
Діаграма процесів	*1 аркуш*
Блок-схема алгоритму роботи додатку	*2 аркуша*
Показники економічної ефективності	*1 аркуш*

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В.	05.10.2022	14.11.2022
Охорона праці	Оришака О.В.	06.10.2022	16.11.2022

7. Дата видачі завдання « 6 » вересня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2022 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2022 р.	
3.	Розробка моделі компонента	20.10.2022 р.	
4.	Розробка структур даних	25.10.2022 р.	
5.	Розробка алгоритмів зв’язку та відображення	30.10.2022 р.	
6.	Програмування алгоритмів	10.11.2022 р.	
7.	Розрахунок економічної ефективності	13.11.2022 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2022 р.	
9.	Оформлення ПЗ	17.11.2022 р.	
10.	Попередній захист роботи	10.12.2022 р.	

Дата видачі завдання
« 6 » вересня 2022 р.

Підпис керівника
_____ Доренський О.П.
(прізвище та ініціали)

Завдання прийнято до виконання
« 6 » вересня 2022 р.

Підпис здобувача
_____ Олійник А.О.
(прізвище та ініціали)

АНОТАЦІЯ

Олійник А.О. Дослідження та програмна реалізація системи CAN-мережі на основі технології CSDN. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2022.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи CAN-мережі на основі технології CSDN.

Метою розробки є дослідження та програмна реалізація системи CAN-мережі на основі технології CSDN.

Об'єктом дослідження є процес CAN-мережі на основі технології CSDN.

Предметом дослідження є методи CAN-мережі на основі технології CSDN.

Методи дослідження базуються на методах теорії побудови комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи CAN-мережі на основі технології CSDN.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ архітектури IBM PC з ОС Windows 10/11.

Програму розроблено в середовищі Delphi 10.4 Sydney.

Ключові слова: комп'ютерна інженерія, CAN-мережі, CSDN

ABSTRACT

Oliinyk A.O. Research and software implementation of the CAN network system based on CSDN technology. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2022.

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for the CAN network system based on CSDN technology.

The purpose of the development is research and software implementation of the CAN network system based on CSDN technology.

The object of the study is the CAN network process based on CSDN technology.

The subject of the study is CAN-network methods based on CSDN technology.

The research methods are based on the methods of the theory of building computer networks, the methods of mathematical statistics, and the methods of software development.

The result of the work is a software implementation of the CAN network system based on CSDN technology.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software tools are provided.

The program can be used on PCs of IBM PC architecture with Windows 10/11 OS.

The program was developed in the Delphi 10.4 Sydney environment.

Keywords: computer engineering, CAN networks, CSDN

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ЗІ	–	захист інформації
ІБ	–	інформаційна безпека
ІТ	–	інформаційна технологія
CAN	–	корпоративні інформаційні системи
ЛОМ	–	локальна обчислювальна мережа
НСД	–	несанкціонований доступ
ОС	–	операційна система
СГ	–	сегмент CAN
CAN	–	система захисту інформації
СЗІ	–	система запобігання вторгнень
IPS	–	Network Admission Control
NAC	–	система виявлення мережних вторгнень
NIDS		

ВСТУП

Актуальність теми. Новий етап, що наступив у розвитку обміну інформацією, який характеризується інтенсивним впровадженням сучасних інформаційних технологій, широким поширенням локальних, корпоративних і глобальних мереж, створює нові можливості і якість інформаційного обміну.

Корпоративні інформаційні системи (CAN) стають сьогодні одним з головних інструментів управління бізнесом, найважливішим засобом виробництва сучасного підприємства, вони використовуються в банківських, фінансової сферах, у сфері державного управління. CAN містить у собі інфраструктуру й інформаційні сервіси. Інфраструктура CAN (мережі, сервери, робочі станції, додатки) є географічно розподілені, її структурна одиниця – сегмент CAN (СГ CAN).

Однак застосування інформаційних технологій немислимо без підвищеної уваги до питань інформаційної (комп'ютерної) безпеки через наявність погроз захищеності інформації.

Для сучасного етапу розвитку теорії й, особливо, практики забезпечення захисту інформації (ЗІ) характерна парадоксальна ситуація: з одного боку, посилене увага до безпеки інформаційних об'єктів, істотне підвищення вимог по ЗІ, прийняття міжнародних стандартів в області інформаційної безпеки (ІБ), постійно зростаючі витрати на забезпечення захисту, з іншого боку – настільки ж неухильно зростаючий збиток, заподіюваний власникам і власникам інформаційних ресурсів, про що свідчать публікуємі регулярно дані про збиток світовій економіці від комп'ютерних атак.

Очевидно, що сучасні підходи до організації ЗІ не повною мірою забезпечують виконання вимог по захисту інформації. Основні недоліки СЗІ визначаються сформованими твердими принципами побудови архітектури й застосуванням в основному оборонної стратегії захисту від відомих

погроз. Критична ситуація в сфері ІБ збільшується у зв'язку з використанням глобальної мережі для зовнішніх і внутрішніх електронних транзакцій підприємства й появою невідомих раніше типів деструктивних інформаційних впливів.

Тому для успішного використання сучасних інформаційних технологій необхідно ефективно управляти не тільки мережею, але й СЗІ, при цьому на рівні СГ CAN автономно повинна працювати система, що реалізує управління складом подій інформаційної безпеки, планування модульного состава СЗІ й аудит. Оскільки об'єкт управління – СЗІ є досить складною організаційно-технічною системою, що функціонує в умовах невизначеності, суперечливості й неповноти знань про стан інформаційного середовища, управління такою системою повинне бути засноване на застосуванні системного аналізу, методів теорії прийняття рішень і необхідної інтелектуальної підтримки.

Разом з тим в області розробки методів і систем захисту інформації в цей час практично відсутні дослідження, спрямовані на забезпечення автоматизованої підтримки управління ЗІ для рішення проблеми забезпечення необхідного рівня захищеності інформації протягом усього періоду функціонування СЗІ.

Одним з варіантів рішення даної проблеми, розглянутим у магістерській роботі, є використання методів інтелектуальної підтримки управління ЗІ в сегменті корпоративної інформаційної системи, що у свою чергу, вимагає розробки на основі принципів системного аналізу й загальнонаукових підходів методологічних основ управління захистом інформації, що відповідають моделей, методів, алгоритмів і програмного забезпечення.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи CAN-мережі на основі технології CSDN.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем CAN-мережі на основі технології CSDN.

– Дослідження системи CAN-мережі на основі технології CSDN.

– Програмна реалізація системи CAN-мережі на основі технології CSDN.

Об'єктом дослідження є процес CAN-мережі на основі технології CSDN.

Предметом дослідження є методи CAN-мережі на основі технології CSDN.

Методи дослідження базуються на методах теорії побудови комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод CAN-мережі на основі технології CSDN.

– Розроблено вітчизняний продукт CAN-мережі на основі технології CSDN, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі CAN-мережі на основі технології CSDN.

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

Робота апробована на LVI Науково-технічній конференції здобувачів вищої освіти «Наука – виробництву», 2022, основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №13.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи CAN-мережі на основі технології CSDN, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Основні ідеї, декларуємі в роботі, у частині реалізації ефективних методів розробки й проектування технічних засобів захисту інформації від НСД у розподілених інформаційних системах з архітектурою «клієнт/сервер» (платформа Intranet) ґрунтуються на наступних положеннях:

1. Використання TCP/IP як базового протоколу транспортного й мережного рівня, як основи для забезпечення ефективності архітектури розподілених корпоративних додатків. У якості необхідних інформаційних служб, що забезпечують обробку інформації в корпоративній мережі, розглядаються служби telnet, ftp, smtp, snmp, http, а також служби SQL-запитів.

2. Структура корпоративної мережі містить у собі з'єднані за допомогою мережі передачі даних загального користування сегменти ЛОМ, окремі фізичні підмережі які не використовуються для надання служби доступу до інформаційних ресурсів різного рівня конфіденційності. У противному випадку окремі ЛОМ будуються на основі об'єднання мереж клієнтських станцій і інформаційних серверів, фізично відділених друг від друга.

3. Для забезпечення взаємодії процесів обробки даних використовуються стандартні протоколи. Модифікація відповідного програмного забезпечення, як клієнтського, так і серверного, не виробляється.

4. Використовується принцип доданого захисту, тобто спеціалізовані засоби не дублюють наявні механізми стандартного системного й прикладного програмного забезпечення.

5. Розподіл механізмів захисту по рівнях мережної взаємодії виконується відповідно до рекомендацій ISO/OSI.

6. Виділення мережної служби забезпечення захисту, побудованої на основі спеціалізованого прикладного протоколу, у рамках якого забезпечується:

– формування облікових записів у базі дані розмежування прав доступу;

– конфігурація режимів функціонування засобів захисту (сценарії реєстрації НСД, режими обслуговування вхідних запитів і т. д.);

– установлення віртуальних каналів у мережі управління, що захищаються із застосуванням процедур канального шифрування й вибіркового шифрування окремих полів інформаційних пакетів (наприклад, з метою забезпечення цілісності транспортних з'єднань);

– протокол розширеної автентифікації з технологією відкритого шифрування;

– діагностика й реєстрація стану ресурсних об'єктів, що захищаються.

7. Для мінімального зниження якості функціонування додатків забезпечується принцип ешелонованого (потокового) захисту. У рамках цього принципу виділяються окремі режими функціонування, що характеризуються, з одного боку, повнотою захисту, і витратами обчислювальних і мережних ресурсів, що вимагаються для виконання відповідних функцій, – з іншої сторони. Зміна режимів функціонування здійснюється відповідно до сигнальних повідомлень від інших засобів (процесів захисту) у форматі протоколів служби управління безпекою.

8. Дана мережа є віртуальною мережею щодо транспортної служби TCP.

9. Віртуальний простір інформаційних об'єктів і служб у рамках віддалених (різних за рівнем конфіденційності) сегментів ЛОМ, при якому неможливо пряме звертання до ресурсів корпоративної мережі, що захищаються.

10. Реєстрація подій, що входять у контекст безпеки, з можливістю наступного аналізу умов виникнення НСД.

1.2 Область застосування

У рамках реалізації основних принципів побудови систем захисту інформації – системності й комплексності – пропонується модель багаторівневого ієрархічного захисту. Комплексність захисту досягається тим, що модель несе в собі можливості захисту інформації від всієї сукупності погроз, в основу класифікації яких пропонується покласти можливість їхнього усунення. До погроз, що усуваються, відносимо помилки й закладки в програмному забезпеченні, до що не усуваються – погрози, пов'язані з необхідністю доступу до інформаційних ресурсів (доступ заборонити не можна, отже, неможливо усунути подібні погрози). Багаторівневість захисту складається у використанні додаткових технічних засобів з певними вимогами до архітектури, як виділених засобів захисту, так і у цілому системи, що захищається.

Принцип багаторівневого захисту покладений в основу здійсненої розробки ряду технологій і їхніх технічних засобів, що реалізують, захисту для корпоративних мереж. При цьому реалізовані наступні напрямки розробки:

– Розробка засобів захисту робочих станцій і серверів (у тому числі й інформаційних) корпоративної мережі, насамперед у частині захисту використовуваної на них платформи – операційної системи.

– Розробка протоколу встановлення захищеного з'єднання «клієнт-сервер» для реалізації віртуальної (або накладеної на мережний простір загального користування) мережі корпорації.

– Розробка виділених технічних засобів захисту ресурсів корпоративної мережі.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи CAN-мережі на основі технології CSDN, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Перехід від роботи на персональних комп'ютерах до роботи в мережі ускладнює захист інформації з наступних причин:

– велика кількість користувачів у мережі і їхній змінний состав. Захист на рівні імені й пароля користувача недостатній для запобігання входу в мережу сторонніх осіб;

– значна довжина мережі й наявність багатьох потенційних каналів проникнення в мережу;

– уже відзначені недоліки в апаратному й програмному забезпеченні, які найчастіше виявляються не на передпродажному етапі, називаному бета–тестуванням, а в процесі експлуатації.

У мережі є багато фізичних місць і каналів несанкціонованого доступу до інформації в мережі. Кожний пристрій у мережі є потенційним джерелом електромагнітного випромінювання через те, що відповідні поля, особливо на високих частотах, екрановані неідеально. Система заземлення разом з кабельною системою й мережею електроживлення може служити каналом доступу до інформації в мережі, у тому числі на ділянках, що перебувають поза зоною контрольованого доступу й тому особливо уразливих. Якщо паролі для входу в мережу стали відомі або підібрані, стає можливим несанкціонований вхід у мережу з файл-сервера або з однієї з робочих станцій. Нарешті можливий витік інформації по каналах, що перебувають поза мережею:

– сховище носіїв інформації;

– елементи будівельних конструкцій і вікна приміщень, які утворять

канали витоку конфіденційної інформації за рахунок мікрофонного ефекту;

– телефонні, радіо-, а також інші провідні й бездротові канали (у тому числі канали мобільного зв'язку).

Будь-які додаткові з'єднання з іншими сегментами або підключення до Інтернет породжують нові проблеми. Атаки на локальну мережу через підключення до Інтернету для того, щоб одержати доступ до конфіденційної інформації, останнім часом одержали широке поширення, що пов'язане з недоліками убудованої системи захисту інформації в протоколах TCP/IP. Мережні атаки через Інтернет можуть бути класифіковані в такий спосіб:

– Сніффер пакетів (sniffer – у цьому випадку в змісті фільтрація) – прикладна програма, що використовує мережну карту, що працює в режимі promiscuous (не робить розходження) mode (у цьому режимі всі пакети, отримані по фізичних каналах, мережний адаптер відправляє додатку для обробки).

– IP-спуфинг (spoof – обман, містифікація) – відбувається, коли хакер, що перебуває усередині корпорації або поза нею, видає себе за санкціонованого користувача.

– Відмова в обслуговуванні (Denial of Service – DoS). Атака DoS робить мережу недоступною для звичайного використання за рахунок перевищення припустимих меж функціонування мережі, операційної системи або додатка.

– Парольні атаки – спроба підбора пароля легального користувача для входу в мережу.

– Атаки типу in-the-Middle – безпосередній доступ до пакетів, переданим по мережі.

– Атаки на рівні додатків.

– Мережна розвідка – збір інформації про мережу за допомогою загальнодоступних даних і додатків.

– Зловживання довірою усередині мережі.

– Несанкціонований доступ (НСД), що не може вважатися окремим типом атаки, тому що більшість мережних атак проводяться заради одержання несанкціонованого доступу.

– Віруси й додатки типу "троянський кінь".

Класифікація засобів захисту інформації

Захист інформації в мережі може бути поліпшений за рахунок використання спеціальних генераторів шуму, що маскують побічні електромагнітні випромінювання й наведення, перешкодопридушуючих мережних фільтрів, пристроїв зашумлення мережі живлення, скремблерів (шифраторів телефонних переговорів), придушувачів роботи стільникових телефонів і т.д. Кардинальним рішенням є перехід до з'єднань на основі оптоволокна, вільним від впливу електромагнітних полів і що дозволяють виявити факт несанкціонованого підключення.

У цілому засоби забезпечення захисту інформації в частині запобігання навмисних дій залежно від способу реалізації можна розділити на групи:

1. Технічні (апаратні) засоби. Це різні по типі пристрої (механічні, електромеханічні, електронні й ін.), які апаратними засобами вирішують завдання захисту інформації. Вони або перешкоджають фізичному проникненню, або, якщо проникнення все-таки відбулося, доступу до інформації, у тому числі за допомогою її маскування. Першу частину завдання вирішують замки, ґрати на вікнах, захисна сигналізація й ін. Другу – згадувані вище генератори шуму, мережні фільтри, скануючі радіоприймачі й безліч інших пристроїв, потенційні канали, що перекривають витоки інформації або дозволяючі їх виявити. Переваги технічних засобів пов'язані з їхньою надійністю, незалежністю від суб'єктивних факторів, високою стійкістю до модифікації. Слабкі сторони – недостатня гнучкість, відносно великі обсяг і маса, висока вартість.

2. Програмні засоби включають програми для ідентифікації користувачів, контролю доступу, шифрування інформації, видалення залишкової (робочої) інформації типу тимчасових файлів, тестового контролю системи

захисту й ін. Переваги програмних засобів – універсальність, гнучкість, надійність, простота установки, здатність до модифікації й розвитку. Недоліки – обмежена функціональність мережі, використання частини ресурсів файл-сервера й робочих станцій, висока чутливість до випадкових або навмисних змін, можлива залежність від типів комп'ютерів (їхніх апаратних засобів).

3. Змішані апаратно-програмні засоби реалізують ті ж функції, що апаратні й програмні засоби окремо, і мають проміжні властивості.

4. Організаційні засоби складаються з організаційно-технічних (підготовка приміщень із комп'ютерами, прокладка кабельної системи з урахуванням вимог обмеження доступу до неї й ін.) і організаційно-правових (національні законодавства й правила роботи, установлювані керівництвом конкретного підприємства). Переваги організаційних засобів полягають у тому, що вони дозволяють вирішувати безліч різнорідних проблем, прості в реалізації, швидко реагують на небажані дії в мережі, мають необмежені можливості модифікації й розвитку. Недоліки – висока залежність від суб'єктивних факторів, у тому числі від загальної організації роботи в конкретному підрозділі.

Шифрування даних являє собою різновид програмних засобів захисту інформації й має особливе значення на практиці як єдиний надійний захист інформації, переданої по протяжних послідовних лініях, від витоку. Шифрування утворить останній, практично непереборний "рубіж" захисту від НСД. Поняття "шифрування" часто вживається у зв'язку з більше загальним поняттям криптографії. Криптографія включає способи й засоби забезпечення конфіденційності інформації (у тому числі за допомогою шифрування) і автентифікації. Конфіденційність – захищеність інформації від ознайомлення з її змістом з боку осіб, що не мають права доступу до неї. У свою чергу автентифікація являє собою встановлення дійсності різних аспектів інформаційної взаємодії: сеансу зв'язку, сторін (ідентифікація), змісту (імітозахист) і джерела (установлення авторства с допомогою цифрового підпису).

Число використовуваних програм шифрування обмежено, причому частина з них є стандартами де-факто або де-юре. Однак навіть якщо алгоритм шифрування не являє собою секрету, зробити дешифрування (розшифрування) без знання закритого ключа надзвичайно складно. Ця властивість у сучасних програмах шифрування забезпечується в процесі багатоступінчастого перетворення вихідної відкритої інформації (plain text в англомовній літературі) з використанням ключа (або двох ключів – по одному для шифрування й дешифрування). В остаточному підсумку, будь-який складний метод (алгоритм) шифрування являє собою комбінацію щодо простих методів.

Вбудовані засоби захисту інформації в мережних ОС доступні, але не завжди, як ми вже відзначали, можуть повністю вирішити виникаючі на практиці проблеми. Система SFT (System Fault Tolerance – система стійкості до відмов) включає три основні рівні:

– SFT Level I передбачає, зокрема, створення додаткових копій FAT і Directory Entries Tables, негайну верифікацію кожного знову записаного на файловий сервер блоку даних, а також резервування на кожному жорсткому диску близько 2% від обсягу диска. При виявленні збоїв дані перенаправляються в зарезервовану область диска, а збійний блок позначається як "поганий" і надалі не використовується.

– SFT Level II містить додаткові можливості створення "дзеркальних" дисків, а також дублювання дискових контролерів, джерел живлення й інтерфейсних кабелів.

– SFT Level III дозволяє застосовувати в локальній мережі дубльовані сервери, один із яких є "головним", а другий, утримуючу копію всієї інформації, вступає в роботу у випадку виходу "головного" сервера з ладу.

Система контролю й обмеження прав доступу в мережах (захист від несанкціонованого доступу) також містить кілька рівнів:

– рівень початкового доступу (включає ім'я й пароль користувача, систему облікових обмежень – таких як явний дозвіл або заборона роботи,

припустима час роботи в мережі, місце на жорсткому диску, займане особистими файлами даного користувача, і т.д.);

– рівень прав користувачів (обмеження на виконання окремих операцій і/або на роботу даного користувача, як члена підрозділу, у певних частинах файлової системи мережі);

– рівень атрибутів каталогів і файлів (обмеження на виконання окремих операцій, у тому числі видалення, редагування або створення, що йдуть із боку файлової системи й дотичних всіх користувачів, що намагаються працювати з даними каталогами або файлами);

– рівень консолі файл-сервера (блокування клавіатури файл-сервера на час відсутності мережного адміністратора до уведення їм спеціального пароля).

Захист інформації – це тільки частина тих численних завдань, які вирішуються мережними ОС. Удосконалення однієї з функцій на шкоду іншим (при зрозумілих розумних обмеженнях на обсяг, займаний даної ОС на жорсткому диску) не може бути магістральним напрямом розвитку таких програмних продуктів загального призначення, якими є мережні ОС. У той же час у зв'язку з гостротою проблеми захисту інформації спостерігається тенденція інтеграції (вбудовування) окремих, що добре зарекомендували себе й стандартними засобів, що стали, у мережні ОС, або розробка власних "фірмових" аналогів відомим програмам захисту інформації.

Спеціалізовані програмні засоби захисту інформації від несанкціонованого доступу володіють у цілому кращими можливостями й характеристиками, чим убудовані засоби мережних ОС. Крім програм шифрування й криптографічних систем, існує багато інших доступних зовнішніх засобів захисту інформації. З найбільше, що часто згадуються рішень, слід зазначити наступні дві системи, що дозволяють обмежити й контролювати інформаційні потоки.

1. Firewalls – брандмауери (дослівно firewall – вогненна стіна). Між локальною й глобальною мережами створюються спеціальні проміжні сервери,

які інспектують і фільтрують весь минаючий через них трафік мережного/транспортних рівнів. Це дозволяє різко знизити погрозу несанкціонованого доступу ззовні в корпоративні мережі, але не усуває цю небезпеку повністю. Більше захищений різновид методу – це спосіб маскараду (masquerading), коли весь вихідний з локальної мережі трафік посилає від імені firewall-сервера, роблячи локальну мережу практично невидимою.

2. Proxy-servers (proxy – доручення, довірена особа). Весь трафік мережного/транспортного рівнів між локальною й глобальною мережами забороняється повністю – маршрутизація як така відсутня, а обіг з локальної мережі в глобальну відбуваються через спеціальні сервери-посередники. Очевидно, що при цьому обіг із глобальної мережі в локальну стає неможливим в принципі. Цей метод не дає достатнього захисту проти атак на більше високих рівнях – наприклад, на рівні додатка (віруси, код Java і JavaScript).

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Embarcadero Delphi, раніше Borland Delphi і Codegear Delphi, – інтегроване середовище розробки ПЗ для Microsoft Windows, Mac OS, iOS і Android мовою Delphi (що раніше носила назву Object Pascal), створена спочатку фірмою Borland і на даний момент приналежна й розроблювальна Embarcadero Technologies. Embarcadero Delphi є частиною пакета Embarcadero RAD Studio і поставляється в чотирьох редакціях: Community (поширюється безкоштовно й має обмежену ліцензію на використання в комерційних цілях), Professional, Enterprise і Architect.

Delphi 10.4 Sydney

Випущено 26 травня 2020 року. RAD Studio Delphi 10.4 забезпечує значно поліпшену високопродуктивну нативну підтримку Windows, кращу продуктивність розробки, миттєві підказки code completion, прискорення

виконання коду із синтаксисом керованих записів, поліпшення виконання паралельних завдань на сучасних багатоядерних CPU, а також містить більш 1000 виправлень багів, поліпшення продуктивності середовища й бібліотек і багато чого крім того.

Основні можливості Delphi 10.4.1:

− Істотні розширення для Windows: поліпшення для застосунків на моніторах 4K High DPI, інтеграція з новим WebView2 на базі Chromium, використання розширених title bars, таких же, як в Office, Explorer, Google Chrome.

− Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовучи класичну реалізацію керування пам'яттю об'єктів.

− Істотне поліпшення Delphi Code Insight (без можливого блокування IDE – в окремому процесі), що допоможе при роботі з великими проектами.

− Тип даних Delphi «record» тепер підтримуть довільні ініціалізацію, фіналізацію й операції копіювання.

− Розширена підтримка бібліотек C++: ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode.

− Відладник Win 64 (на LLDB) і збирач для C++.

− Поліпшення для C++: Включена велика кількість поліпшень STL з Dinkumware.

− Підтримка Metal Driver GPU для macOS і iOS.

− Вбудований Fmxlinux.

− Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API. Реалізація компонента Media Player для macOS тепер використовує Avfoundation. Реалізований заново стилізуємий FMX компонент TMemo на платформі Windows значно поліпшений і тепер має відмінну підтримку IME.

− Численні поліпшення швидкості й стабільності роботи нашої бібліотеки The Parallel Programming Library (PPL).

− Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.

− Клієнтські бібліотеки HTTP і REST Client розширені застосунковими можливостями роботи з HTTPS. Також були розширені можливості підтримки Amazon AWS services

− У технологію Visual LiveBindings внесена безліч поліпшень, у тому числі швидкодії, що стосуться, застосунків на VCL і FireMonkey

RAD Studio 10.4 Короткий огляд:

– Істотні розширення для Windows. Створення застосунків, що чудово виглядають, із чіткими елементами інтерфейсу на 4k моніторах High DPI за допомогою нової гнучкої підтримки стилів елементів керування на екрані. Інтеграція із сучасними, безпечними web-технологіями від Microsoft – новим WebView2 на базі Chromium. Використання сучасних розширених title bars, таких же, як в Office, Explorer, Google Chrome, у своїх проектах. Істотні поліпшення надійності налагодження в новому відладнику для C++ Windows 64-bit.

– Зросла продуктивність розробки. Ріст продуктивності за рахунок миттєвої реакції підказок code completion у середовищі IDE. Краща сумісність із уже наявною кодовою базою, і спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю. Швидке зв'язування даних і візуальних елементів за допомогою розширеної технології Visual LiveBindings з підвищеною швидкодією. Просте використання розповсюджених бібліотек C++, наприклад, ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode. Оновлена підтримка Amazon AWS cloud.

– Поліпшення швидкодії і якості. Більш 1000 поліпшень швидкодії і якості. Краща ефективність коду за допомогою нового синтаксису custom managed records. Більш швидке виконання паралельних завдань на сучасних багатоядерних CPU. Переконаєтеся в прискоренні відображення на екрані з підтримкою Metal API на macOS і iOS. Краща сумісність із уже наявною кодовою базою й спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю.

Істотне поліпшення Delphi Code Insight

Як найбільше й головне поліпшення інструментів програмування Delphi за багато років, в 10.4 Delphi Code Insight реалізований через Language Server Protocol (LSP). LSP – це технологія генерації результатів для code completion, навігації й інших сервісів в окремому процесі. Це значить, що code completion і Code Insight одержать більш точні результати без блокування IDE. 10.4 забезпечує набагато більш високу продуктивність розроблювачів, які працюють із більшими проектами, що містять мільйони рядків коду.

Delphi Custom Managed Records

Ключове розширення мови Delphi: тип даних Delphi «record» тепер підтримуть довільні ініціалізацію, фіналізацію й операції копіювання. Управляйте тем, як ці структури створюються, копіюються й звільняються з допомогу вашого коду, який буде виконуватися у відповідний момент.

Це розширює потужність конструкцій records в Delphi, які використовуться щоб одержати більшу ефективність у порівнянні із класами.

Єдине керування пам'яттю

Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовучи класичну реалізацію керування пам'яттю об'єктів.

У порівнянні з Automatic Reference Counting (ARC), це дає кращу сумісність із існучим кодом і спрощує написання компонентів, бібліотек і застосунків.

ARC модель керування пам'яттю model залишилася для керування рядками й посиланнями на тип інтерфейсу на всіх платформах. Для C++ це означає, що при створенні й звільненні Delphi-style класів в C++ використовується звичайне керування пам'яттю, як у будь-якого heap-allocated класу C++, що значно знижує складність коду.

Розширена підтримка бібліотек C++

В 10.4 ми портували багато популярних бібліотек C++ у C++Builder.

Забезпечивши оптимізовану підтримку бібліотек ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode, поряд із уже підтримуваними Boost і Eigen, які можуть бути додані за допомогою менеджера пакетів Getit.

Win 64-відладник і збирач для C++

В 10.4 з'явився новий відладник C++ для Windows 64-bit. Відладник заснований на LLDB і показує значне збільшення стабільності при налагодженні 64-bit застосунків поряд з новими відладочними можливостями, такими як перегляд і інспекція типів начебто рядків C++ і Delphi, а також колекцій STL, включаючи std::vector, std::map і інших. Крім того, згенерована для застосунку відладочна інформація має інший внутрішній формат, сприяючи більш стабільному й багатому на можливості процесу налагодження, більш докладним перегляду й інспекції в debug-time.

Підвищення якості й швидкодії інструментів

- Велика кількість поліпшень STL від Dinkumware.
- Поліпшені деякі найважливіші методи й області RTL, на базі поліпшень сумісності з популярними бібліотеками C++.
- Поліпшена підтримка Cmake.
- Велика кількість виправлень для підвищення стабільності і якості.
- Відновлення Windows API – Обновлено й додали безліч декларацій API щоб добитися ще більшої інтеграції із платформою Windows.
- Загальні вдосконалення в бібліотеці доступу до БД FireDAC, включаючи оновлені драйвера для FireBird, PostgreSQL і SQLite. Вибір статичного або динамічного підключення SQLite до застосунку.

Змінені стилі VCL для High DPI

В 10.4, архітектура стилізації VCL була суттєво розширена для підтримки High DPI і 4K моніторів. Тепер усі елементи UI на формі VCL автоматично масштабуться під відповідне до монітора дозвіл для показу форми. Був оновлений API стилізації для підтримки стилів high DPI.

Кожний графічний елемент UI може бути обраний з наборів різних масштабів і масштабований до потрібного DPI, що дає чітке зображення елементів UI на всіх моніторах.

Нові High DPI стилі й стилізація окремих VCL компонент

Обновлено велике число вбудованих і преміальних VCL стилів для підтримки нового режиму стилізації High-dpi. Це дозволяє вам створювати застосунку з відмінним дизайном для всіх моніторів.

Розроблювачі VCL застосунків тепер можуть використовувати трохи VCL стилів на різних формах в одному застосунку або в різних компонентів на одній формі. Це також включає стилізацію компонентів загальною темою для платформи. Крім застосункової гнучкості використання стилів, це дозволяє використовувати нестилізуємі компоненти із зовнішніх бібліотек в VCL застосунках, що використовуть стиль.

Поліпшена кроссплатформеність

- Додана підтримка Metal Driver GPU для macOS і iOS.
- Крім підтримки останнього iOS SDK, в RAD Studio 10.4 розроблювачі можуть задовольнити нові вимоги Apple до набору стартових екранів.
- Реалізований заново стилізуємий FMX компонент TMemo на платформі Windows значно поліпшений і тепер має відмінну підтримку IME.
- Користувачам редакцій Enterprise або Architect доступна повна інтеграція Fmxlinux з IDE для створення клієнтських застосунків Linux з GUI.
- Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.
- Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

Оновлений менеджер пакетів Getit

Менеджер пакетів Getit в IDE був значно вдосконалений.

Дати випуску релізів пакетів тепер видні, і можливе сортування списку по цих датах; відбір тільки встановлених пакетів, контенту, доступного тільки при наявності підписки, багато чого іншого.

Універсальний інсталятор для установки Online і Offline

В 10.4 включений новий універсальний інсталятор, який використовує технологію на базі Getit. Цей інсталятор підтримує як online, так і offline (з ISO) варіанти установки.

Тепер обоє варіанта установки дозволяють вам указати початковий набір можливостей RAD Studio для установки, наприклад, свою комбінацію мов програмування й цільових платформ, мов інтерфейсу, і додавати до нього або видаляти непотрібне в будь-який момент.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускну кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи CAN-мережі на основі технології CSDN.

В процесі розробки випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які

забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформувати висновки про виконаний обсяг робіт та одержані результати.

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Опис архітектури Cisco Self-Defending Network

Частково завдяки діяльності Cisco, що представляє стратегію мережі, яка самозахищається Cisco (Self-Defending Network) (CSDN), багато хто починає усвідомлювати необхідність інтегрованих засобів мережного захисту.

Механізми забезпечення мережної безпеки еволюціонували від незалежно використовуваних «точкових» продуктів, таких як міжмережні екрани або засоби виявлення вторгнень, в область інтегрованих і цілісних рішень. Cisco Systems є провідною компанією по розробці технології, що дозволяє зробити мережі, що самозахищаються, реальністю.

Ідея рішення досить просте: призначення IT-інфраструктури полягає в створенні систем, що надають можливість виявлення порушень безпеки й захисту від несанкціонованого доступу з одночасним наданням оперативного доступу легальним користувачам. Проста відмова в доступі вже не є підходящою реакцією на атаку – сучасні мережі повинні реагувати на атаки, зберігаючи свою доступність, надійність і працездатність. У багатьох відносинах, метою забезпечення безпеки стає підвищення ступеня відказостійкості мереж. Замість того, щоб ставати жертвами, мережі повинні стати здатними «поглинати» атаки й зберігати працездатність, подібно імунній системі людини, що дозволяє організму функціонувати при наявності в ньому вірусів і бактеріальних інфекцій.

Розвиток ситуації в сфері безпеки

За останні три роки технології забезпечення безпеки змінилися більше, ніж за все попереднє десятиліття. Обсяг і темп цих змін ускладнили покладену на IT-Фахівців завдання підтримки належного рівня захищеності. Перед тим, як

продовжити розповідь про Cisco SDN, необхідно одержати поданя про суть цих змін:

– Захист периметра мережі. Мабуть, найбільш істотним фактором, що вплинув на зміну підходу до забезпечення безпеки мереж, стала зміна самої сутності мережі. Після того, як корпорації стали консолідувати центри обробки даних, використовувати конверговані внутрішні мережі й активно використовувати мережу Інтернет, уже не можна забезпечити безпеку мережі тільки за рахунок організації захисту її периметра. Середовище, що раніше вважалося ізольованим і контрольованим, тепер є напіввідчиненим за рахунок, наприклад, мереж "екстранет", підключень пунктів роздрібного продажу, надомних працівників та ін. Розширення корпоративної мережі, таким чином, приводить до необхідності взаємодії через ненадійні проміжні мережі й неконтрольовані середовища. Пристрої, що підключаються до корпоративної мережі через ці проміжні мережі, найчастіше не відповідають вимогам корпоративних політик безпеки. Пристрої, їм відповідні, часто використовуються для доступу до інших неконтрольованих мереж до з'єднання з корпоративною мережею. У результаті, пристрої, підключені до зовнішніх мереж, можуть стати «перевалочним пунктом» для атак і пов'язаних з ними несанкціонованих дій.

– Бездротові мережі й мережі мобільного зв'язку. Прив'язані до поняття периметра захисту бездротові мережі й мережі мобільного зв'язку підприємств тепер забезпечують підтримку ноутбуків, кишенькових комп'ютерів (PDA) і мобільних телефонів, які підключені до декількох мереж. Ці пристрої з декількома мережними інтерфейсами підтримують можливість установлення одноранговых бездротових з'єднань для роботи в мережі "точка-точка". Крім того, пакети можуть ефективно передаватися між пристроями на прикладному рівні. У результаті поняття границь мережі стає усе більше розмитим і для забезпечення безпеки компаніям необхідно мати можливість управління такими мобільними пристроями.

– Електронна комерція, мережі "екстранет" і проведення ділових операцій у глобальній мережі. Поява загальних прикладних інтерфейсів на основі протоколів передачі повідомлень, таких як XML і SOAP, зробило благотворний вплив на електронну комерцію й продуктивність роботи підприємств. Але, як і в більшості випадків появи нових технологій, їхня поява привела до виникнення зовсім нових уразливістей і джерел атак, з якими доводиться боротися. Дані, які раніше передавалися за допомогою множини мережних протоколів і проходили фільтрацію на міжмережних екранах, тепер передаються за допомогою декількох або всього одного транспортного протоколу (наприклад, HTTP з використанням порту 80 TCP). У результаті, більша частина даних, що раніше містилася в заголовках пакетів, тепер розташовується в тілі пакетів. Це істотно полегшує зловмисникові завдання обходу класичної системи захисту мережі. Більше того, для забезпечення конфіденційності й цілісності корпоративних даних всі частіше використовується шифрування трафіка прикладного рівня за допомогою протоколів SSL/TLS і HTTPS. При цьому виникає побічний ефект, пов'язаний з ускладненням контролю доступу на границі мережі через неможливість перевірки пакетів у зашифрованих потоках даних.

– Віруси, Інтернет-хробаки й швидкість їхнього поширення. Кількість і різноманіття вірусів, що з'явилися за останні три роки, і Інтернет-хробаків саме по собі є застрашливим. Дивовижний вплив цих Інтернет-хробаків і вірусів на мережі підприємств і їхня продуктивність було обумовлено наявністю двох факторів: короткого проміжку часу між виявленням уразливості й появою атаки з її використанням, а також швидкості, з якою більшість атак поширювалося по мережі. При цьому число порушень роботи мереж досягало неприпустимого рівня, а для усунення наслідків доводилося йти на незаплановані витрати людських, тимчасових і матеріальних ресурсів.

– Дотримання встановлених норм. Факти, що одержали широкий розголос, порушень і неправомірні дії усередині корпорацій підштовхнули керуючі органи багатьох галузей до створення норм по регулюванню ризиків

відносно корпоративної інформації. У США ці норми, найбільш відомими з яких є закон Сарбейнса-Окслі, закон Гремма-Ліча-Блілі й закон про дотримання конфіденційності інформації про охорону здоров'я й особистих даних пацієнтів (HIPAA), привели до корінних змін способів організації корпоративних мереж, серверів, баз даних і хостів. Аналогічна тенденція спостерігається й в Україні.

Хоча багато організацій думають, що дотримання норм забезпечує більш надійний захист їхньої інфраструктури, дане думка найчастіше є помилковим. Сам процес проходження встановленим нормам може привести до виникнення нових уразливістей. Наприклад, Інтернет-хробаки й віруси можуть більш ефективно поширюватися в мережі, що підтримує наскрізні VPN-з'єднання, у зв'язку з тим, що минаючі по них потоки даних є невидимими для проміжних вузлів. Такі потоки даних можуть переносити Інтернет-хробаків на критично важливі корпоративні сервери за допомогою надійно зашифрованих пакетів. Крім того, що на виявлення такої атаки йде багато часу, наскрізні VPN-з'єднання ускладнюють процес усунення її наслідків.

Принципи побудови сучасних безпечних мереж

Корпорації не можуть нескінченно додержуватися напрямків в області безпеки, що змініються. В ідеалі, удосконалювання системи безпеки повинне впливати на існуючу інфраструктуру маршрутизації й комутації, методи розмежування й контролю доступу й суміжних організаційних структур, що забезпечують підтримку цих систем. У цьому розділі ми опишемо основні елементи мережі, що самозахищається, **Cisco Self-Defending Network**:

– Присутність. Фундаментальним поняттям захищеної системи є поняття контрольних точок, що ми визначимо як присутність. Подібно імунній системі людини, що заснована на розосереджені по всьому тілу людини й виконуючі функції виявлення інфекції й виконання відповідних дій клітки, мережа покладається на наявність певних можливостей в окремих вузлів. До таких можливостей відносяться класичні методи ідентифікації, контролю доступу, перевірки даних і захисту взаємодії, а також нові можливості аналізу дій клієнтів

файлообмінних мереж, web-сервісів, голосових сервісів і сервісів передачі динамічного контенту по мобільних мережах.

– Контекст. При вході користувача в систему мережа запитує й одержує доступ до набору реквізитів доступу користувача й хоста, що представляють собою кінцеву сутність. Повноваження можуть змінюватися із часом залежно від дій підключеного до мережі хоста. Сукупність цих даних і являє собою контекст. На відміну від існуючих систем мережної безпеки, у яких велика увага приділяється тільки перевірці повноважень користувача при вході в мережу, мережа, що самозахищається Cisco Self-Defending Network ухвалює рішення щодо наданя або скасування повноважень на основі змін поводження й відповідного йому контексту за увесь час з'єднання користувача з мережею. Наприклад, якщо мережа виявляє, що хост заражено вірусом (при цьому користувач може мати всі повноваження на доступ), вона ізолює цей хост у карантинний сегмент мережі. Оскільки дані можуть бути підмінені, у процесі забезпечення безпеки системи може знадобитися одержання контексту від інших систем для точного й своєчасного визначення прав хоста й привілеїв у конкретний момент часу.

– Взаємозв'язок. Взаємозв'язки між окремими пристроями дозволяють обмінюватися контекстом і створювати «систему». Традиційно, взаємозв'язки між пристроями мережі встановлювалися за допомогою протоколів маршрутизації, такими як протокол BGP. Для того щоб протистояти найсучаснішим видам погроз і несанкціонованих дій, тепер необхідно розширювати ці взаємозв'язки по всьому маршруті від джерела до одержувача мережного трафіка. Крім того, через зростаюче число мобільних пристроїв, взаємозв'язки вийшли за межі границь, які донедавна розглядалися як зовнішні границі мереж у традиційному розумінні. Привілеї, які пристрій одержує при доступі до мережі й характер їхньої зміни в процесі сеансу роботи визначаються на основі контексту цього пристрою і його взаємозв'язків у мережі.

– Довіра. Безпека системи визначається безпекою вступної у неї інформації; система функціонує набагато ефективніше, якщо в ній присутні довірчі відносини. Раніше ступінь довіри визначалася головним чином на основі ідентифікації пристрою або користувача. Результати останніх досліджень показали, що в концепцію захищених систем повинні бути включені поняття стану й місця розташування пристрою.

По багатьом параметрам дії, виконувані користувачами або пристроями в мережі, можна зрівняти з управлінням автомобілем. Подібно тому, як людина дістає водійські права, що дозволяють йому управляти певним класом транспортних засобів, користувачі повинні мати деяку ідентифікаційну інформацію для входу в мережу. Крім того, у кожного автомобіля є ідентифікаційний номер, що повинен бути зареєстрований у місцевих органах управління – мережі й кінцеві вузли незабаром будуть мати цифрові сертифікати, створювані під час випуску й потребуючі виконання певного типу реєстрації при використанні в рамках компанії. Але оскільки пристроям не завжди вдається вчасно надавати ідентифікаційні дані, мережі, що самозахищаються Cisco Self-Defending Network використовує передові методи непрямої довіри й максимальних зусиль для автентифікації й авторизації сутностей. Мережа, що самозахищається, Cisco Self-Defending Network повинна як мінімум уміти запитувати ідентифікаційні дані кожного пристрою й користувача, виконувати аналіз стану пристрою й установлювати місце розташування пристрою в мережі. Технологія, що дозволяє реалізувати ці можливості, буде повсюдно поширена й задіяна за допомогою чітко певних стандартних форматів повідомлень і протоколів, таких як протокол 802.1x і протокол автентифікації EAP.

Саме по собі кожне із цих понять не дуже примітно. Але вони здобувають силу при об'єднанні в мережі, що самозахищається, Cisco Self-Defending Network. У частині, що залишилася, даного огляду описуються деякі способи

використання цих понять у рамках мережі, що самозахищається Cisco Self-Defending Network.

Елементи й побудова мережі

Оскільки одночасна перебудова всіх підсистем без порушення цілісності IT-сервисів може виявитися складним завданням, більшість споживачів не зможе впровадити всі компоненти стратегії Cisco SDN одночасно. Крім того, деякі споживачі можуть баритися з передачею функцій контролю безпеки автоматизованій системі доти, поки вони не переконаються в надійності роботи рішення. Стратегія мережі, що самозахищається, Cisco Self-Defending Network дозволяє таким компаніям здійснювати поступовий перехід до Cisco SDN за рахунок наданя продуктів, які можуть використовуватися незалежно друг від друга. Таким чином, має сенс розглянути наступні основні етапи проектування мережі, що самозахищається, Cisco Self-Defending Network.

Захист кінцевих вузлів. Віруси й Інтернет-хробаки, що заражають кінцеві вузли, часто приводять і до побічного ефекту – перевантаженню мережі, що є наслідком їхнього швидкого поширення.

Cisco пропонує засіб запобігання вторгнень на кінцеві вузли Cisco Security Agent, що дозволяє вирішити обидві проблеми. Використовувані в Cisco Security Agent передові методи захисту на основі аналізу поводження дозволяють виявляти віруси й Інтернет-хробаки, а також запобігати їхнє проникнення на кінцеві системи й поширення по мережі. Фактично, Cisco Security Agent є першою лінією оборони для запобігання поширення вірусів і Інтернет-хробаків.

Другим очевидним аргументом на користь застосування Cisco Security Agent є те, що він використовується на кінцевих вузлах і дозволяє створити ланцюг відповідної реакції між кінцевим вузлом і мережею. У результаті виходить мережа, здатна швидко адаптуватися до виникаючих погроз.

Контроль доступу. Однією з найбільш важливих можливостей мережі, що самозахищається, Cisco Self-Defending Network є механізм контролю доступу до мережі Cisco Network Admission Control (NAC).

NAC дозволяє вирішити, який рівень доступу варто надати кінцевому вузлу, виходячи з відповідності стану вузла політиці безпеки компанії, обумовленого шляхом аналізу стану безпеки операційної системи й установлених додатків. На додаток до функцій контролю й розмежування доступу NAC надає IT-адміністраторам можливість автоматичного перекладу в карантин і лікування кінцевих вузлів, що не пройшли перевірку відповідності. Перевірка відповідності є ефективною другою лінією оборони для запобігання поширення вірусів і Інтернет-хробаків. NAC можна також розглядати як інструментальний засіб аналізу уразливістей і управління установкою «латок» на вимогу.

Відмінною рисою NAC є наданя як клієнтського, так і адміністративного інтерфейсу AAA, що дозволяють споживачам додатково встановлювати продукти великої кількості розроблювачів засобів захисту.

У цей час більше 250 лідируючих на ринку розроблювачів інтенсивно впроваджують або вже впровадили у свої продукти механізми NAC.

Важливо надати можливість використання NAC у системах малих і середніх підприємств. Для цього Cisco кілька років назад придбала корпорацію Perfigo, областю діяльності якої є розробка комплексних рішень контролю доступу до мережі. Основними функціями рішень є аналіз політик кінцевих вузлів, перевірка відповідності стану вузлів установленим вимогам і забезпечення працездатності засобів контролю й розмежування доступу. Тепер у рамках ініціативи Network Admission Control компанія Cisco пропонує рішення за назвою Cisco NAC Appliance (Cisco Clean Access).

Обмеження області зараження. Посилені політики доступу не є панацеєю й не усувають необхідність моніторингу пристроїв після їхнього входу в мережу. Кваліфіковані зловмисники в стані обійти практично будь-яку перевірку прав доступу, а мережі не можуть постійно покладатися на заражений елемент або довіряти йому. Пристрої, що пройшли перевірку відповідності, також можуть бути інфіковані за допомогою різноманітних джерел зараження після входу в мережу – наприклад, зараження з USB-накопичувача.

Мережа, що самозахищається Cisco Self-Defending Network спроектована для виконання перевірок безпеки не тільки під час одержання вузлом доступу до мережі, але й протягом усього сеансу з'єднання. Крім того, мережа, що самозахищається Cisco Self-Defending Network може покладатися на інші елементи мережі, включаючи кінцеві вузли для визначення компрометації інших вузлів, за аналогією з тим, як поліція контролює рівень злочинності шляхом аналізу дзвінків на номер 911. Cisco розглядає засоби обмеження області зараження як третю лінію оборони для запобігання поширення вірусів і Інтернет-хробаків.

На жаль, протоколи автентифікації, що існують, не розроблялися для роботи після початкового обміну інформацією. Таким чином, мережа, що самозахищається Cisco Self-Defending Network повинна забезпечувати нові способи обміну інформацією про стан пристроїв (контекст), а також способи оцінки вірогідності цієї інформації на основі як непрямого, так і прямої довіри. Наприклад, адміністратор повинен мати можливість створювати правило, відповідно до якого повідомлення, отримане від кінцевого вузла із установленим агентом Cisco Security Agent, заслуговує більшої довіри, чим повідомлення, що прийшло від незахищеного кінцевого вузла. У результаті компанія Cisco почала розробку нових механізмів кореляційного аналізу й відповідної реакції на основі непрямих атрибутів.

Інтелектуальні засоби кореляційного аналізу й реагування на інциденти

Для забезпечення ефективної роботи методів відповідної реакції, швидкої оцінки впливу, вибору конкретної дії й визначення найкращого засобу захисту необхідно, щоб мережа, що самозахищається, Cisco Self-Defending Network надавала сервіси кореляційного аналізу подій у сфері безпеки в режимі реального часу. Для рішення цього завдання компанія Cisco придбала компанію Protego Networks, що розробила сімейство продуктів MARS, що надають методи зв'язування відповідної реакції від різних мережних пристроїв (міжмережні екрани, системи виявлення вторгнень, маршрутизатори, комутатори й хости) з

контекстом, одержуваним у результаті вивчення топології мережі на рівні 2 і 3. Це дозволяє групі реакції на порушення в сфері безпеки швидко визначити місце появи атак у мережі.

Інтегровані системи виявлення вторгнень і механізми виявлення аномалій. Проектування ефективних систем виявлення мережних вторгнень (NIDS) завжди було важливим напрямком в області постійно, що ведуться досліджень, і розробок Cisco. Одним з перших нововведень Cisco у цій області було впровадження NIDS у маршрутизатори й комутатори. Але для того щоб система NIDS мала повну функціональність, її необхідно перетворити в систему запобігання вторгнень (IPS) з убудованими можливостями фільтрації трафіка, що дозволяє відкидати непотрібні пакети за допомогою підсистем, що набудовуються тонко, класифікації трафіка.

На жаль, більшість NIDS видають занадто багато помилкових спрацьовувань і не можуть надійно виконувати завдання запобігання атак при установці системи на проміжному пристрої. Почасти проблема полягає в необхідності збору й обробки великого обсягу інформації (контексту) протягом досить короткого проміжку часу. Особливо це, до речі, стосується додатків, які дуже чутливі до затримок передачі (наприклад, IP-телефонія). Для рішення цього завдання Cisco розробляє кілька методів, що забезпечують високоякісну й ефективну обробку й класифікацію контрольованого трафіка.

Багато легальних дій можуть бути помилково сприйняті мережею як аномальні; головним чином, це стосується мереж зі значним числом змінних факторів. У результаті компанія Cisco стала випливати консервативному поетапному підходу до виявлення аномалій, починаючи з Cisco Security Agent, оскільки було встановлено, що операційні системи моделювати простіше, ніж мережні середовища. Після цього компанією Cisco була придбана ефективна система запобігання вторгнень Riverhead, що характеризується низьким числом помилкових спрацьовувань за рахунок чіткого поділу дій, спрямованих на проведення атак типу «відмова в обслуговуванні», і іншої мережної активності.

Безпека додатків і захист від шкідливих програм (Anti-X). За останні кілька років з'явилися нові мережні додатки, що забезпечують захист від нових

видів погроз, включаючи віруси, Інтернет-хробаків, спам, шпигунські програми, зловмисне використання web-сервісів і засобів IP-телефонії, а також несанкціоноване використання клієнтів файлообмінних мереж, – захист від яких не забезпечувалася повною мірою класичними міжмережними екранами й продуктами NIDS. З метою захисту від цих погроз фахівцями Cisco були розроблені сервіси захисту нового покоління, що виконують перевірку заголовків пакетів і їхнього вмісту. Це дозволяє забезпечити ретельну перевірку трафіка в критично важливих точках мережі й обробляти зловмисний трафік до влучення в корпоративну мережу.

Структурна схема мережі, що самозахищається, Cisco Self-Defending Network наведена на рисунку 3.1.

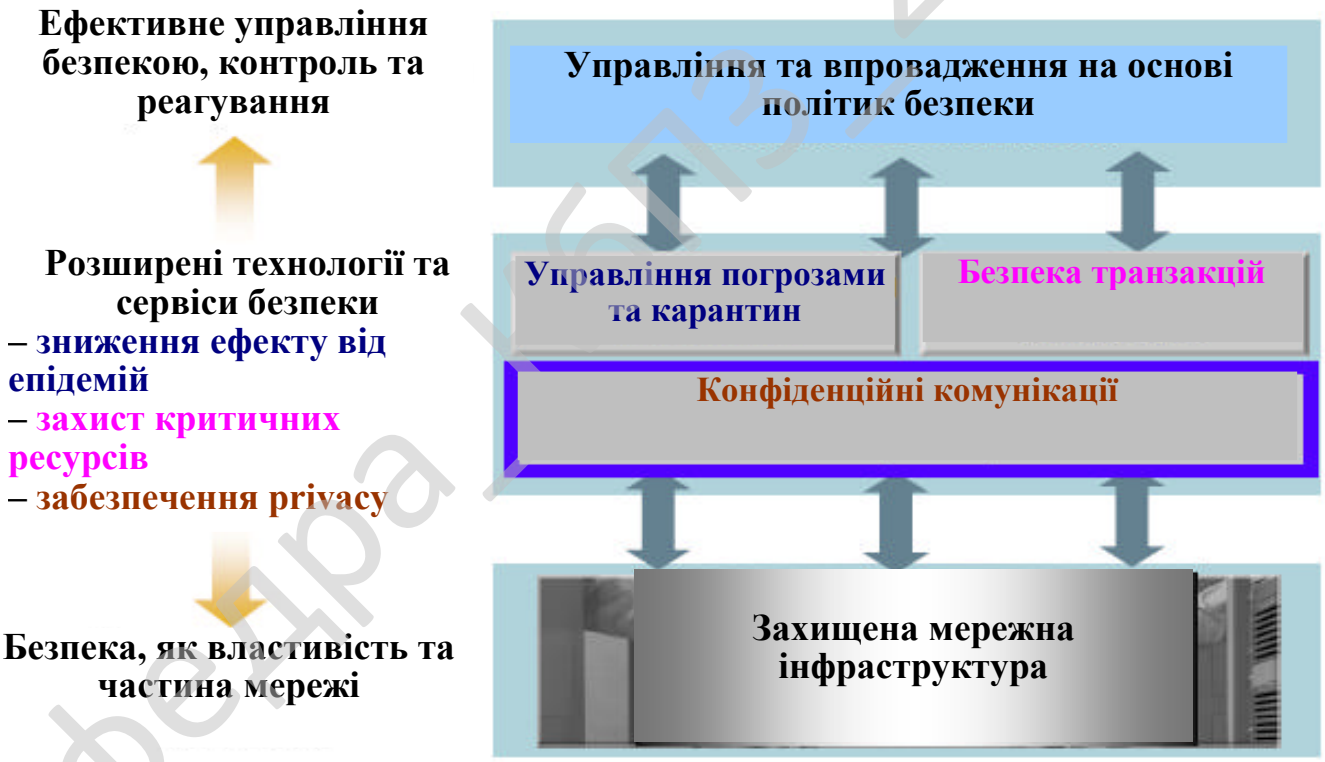

Рисунок 3.1 – Структурна схема мережі, що самозахищається, Cisco Self-Defending Network

Об'єднання цих сервісів у багатофункціональні платформи дозволяє розширити можливості розроблювачів, а також знизити сукупну вартість

володіння для споживача. Крім того, інтеграція цих механізмів дозволить розширити можливості мережі, що самозахищається, Cisco Self-Defending Network по контролі додатків. Якщо в додатках використовується наскрізне шифрування, мережа, що самозахищається Cisco Self-Defending Network може збирати інформацію з кінцевих вузлів, компенсуючи втрати, пов'язані з неможливістю контролю даних на границі мережі.

3.2 Розробка структурної схеми

Ґрунтуючись на принципах системного аналізу, що являє собою теорію й практику поліпшуючого втручання в проблемну ситуацію, пропонується варіант декомпозиції проблеми дозволу наявних протиріч в області забезпечення безпеки інформації.

На підставі системного підходу видно, що модель проблемної ситуації в області захисту інформації містить сукупність трьох взаємодіючих систем:

– проблемоутримуючої СЗІ,

– проблемодозволяючої керуючої системи, що розробляється для того, щоб проблема зникла або ослабнула, що оточує;

– істотного середовища, з якої взаємодіє СЗІ, під якою розуміється безліч потенційна можливих погроз інформаційної безпеки.

Вимога постійно наростаючої деталізації приводить до побудови моделі состава проблемоутримуючої системи, моделі об'єкта захисту й моделі погроз.

Відзначається, що основною проблемою при побудові керуючої системи є розробка моделі погроз, що зв'язано зі специфічністю взаємодії об'єкта управління – СЗІ з навколишнім середовищем. У зв'язку із цим пропонується концепція побудови моделі погроз безпеки інформації, що базується на розроблювальній класифікаційній схемі навмисних цілеспрямованих погроз інформаційному середовищу корпоративної інформаційної системи. Показано доцільність побудови сукупності моделей:

– функціональної, на основі опису послідовності дій зловмисника (порушника) за допомогою дерев погроз;

– просторової графової, систематизованих у форматі інтегральної структурної моделі каналів несанкціонованого доступу, витоку й деструктивних впливів, що дозволяє провести всебічний аналіз реальних погроз, підвищити адекватність моделі погроз для конкретного об'єкта захисту.

На основі аналізу принципів управління в умовах невизначеності пропонується узагальнена архітектура системи управління захистом інформації в сегменті корпоративної інформаційної системи. Проаналізуємо основні функції управління, обґрунтовується доцільність варіанта побудови системи, що включає дві функціональні підсистеми:

– підсистему організаційно-технічного управління;

– і підсистему оперативного управління в реальному масштабі часу.

Відповідно до вимоги кількісної оцінки характеристик систем, висунутим системотехнікою, у якості керованої змінної введемо показник – рівень захищеності, необхідна значення якого залежить від максимального рівня критичності оброблюваної в даний період часу інформації.

У контурі організаційно-технічного управління створюються механізми управління захистом інформації при зміні інфраструктури, бізнес-додатків, планів обробки інформації й відповідних їм вимог до рівня захищеності інформації. Контур включає: систему інтелектуальної підтримки прийняття рішень на вибір стратегії захисту, систему оцінки рівня захищеності (ризику), що управляє вплив реалізується співробітниками відділу інформаційної безпеки. Командна інформація формується в ході планування – цілеспрямованого вибору раціонального комплексу засобів захисту.

У контурі оперативного управління формується оперативна командна інформація, що доводить до об'єкта управління адміністратором безпеки або автоматично за допомогою засобів реалізації керуючих впливів на убудовані в засоби захисту керуючі модулі.

У системі управління, що має таку архітектурну побудову, ефективні рішення вибираються й приймаються як на основі відомостей про технічні характеристики засобів захисту, так і на основі аналізу контрольованого простору.

Структурна схема системи управління захистом інформації в сегменті корпоративної інформаційної системи показана на рисунку 3.2.

На основі аналізу можливостей удосконалювання управління захистом інформації за рахунок застосування нових методів рішення завдань управління й скорочення тривалості циклу управління розробляється функціональна модель системи управління в стандарті IDEF0, що дозволяє наочно й ефективно відобразити механізм управління загрозами, виявити процеси, для реалізації яких необхідна розробка автоматизованої системи інтелектуальної підтримки управління.

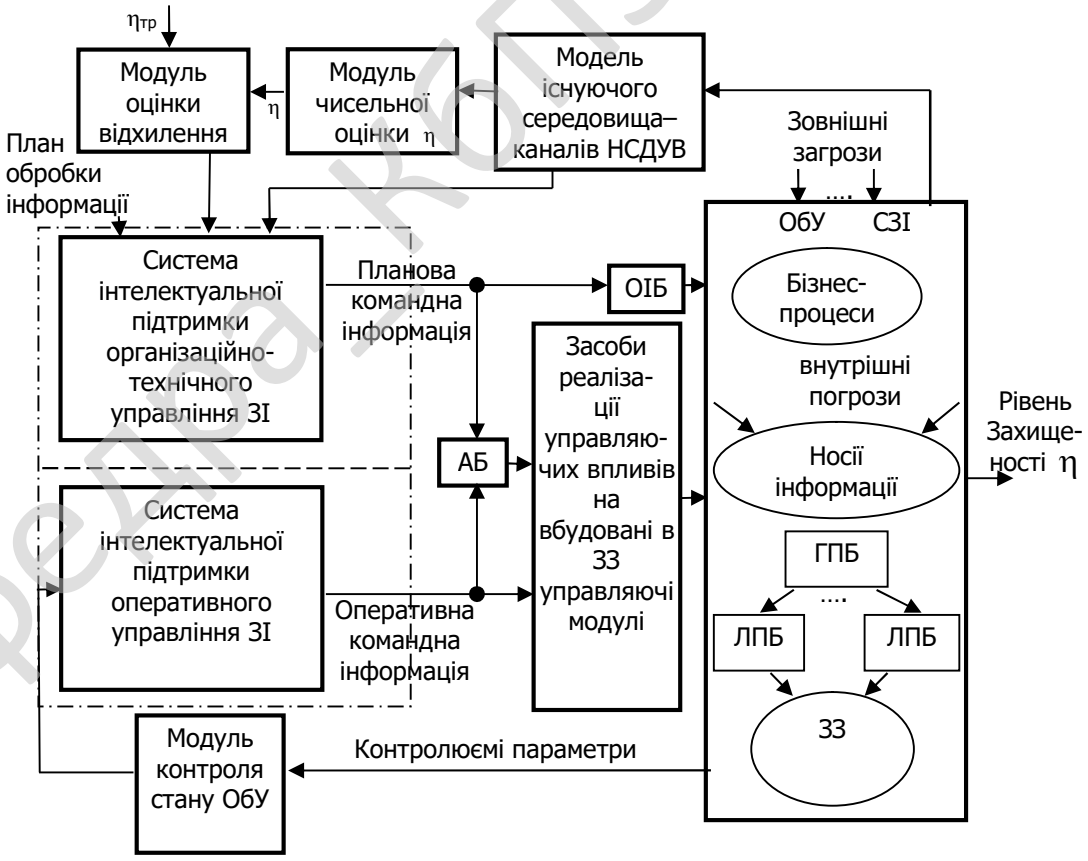

Рисунок 3.2 – Структурна схема системи управління захистом інформації в сегменті корпоративної інформаційної системи

У структурній схемі застосовуються наступні скорочення:

– АБ – адміністратор безпеки;

– ОІБ – співробітники відділу інформаційної безпеки;

– Обу (СЗІ) – об'єкт управління;

– ГПБ, ЛПБ – глобальна, локальні політики безпеки;

– НСДУВ – несанкціонований доступ, витік, деструктивний вплив;

– ЗЗ – засоби захисту;

– $\eta_{тр}$ – необхідне значення рівня захищеності.

3.3 Розробка функціональної схеми

Для подолання труднощів у слабоформалізованих ситуаціях більше високий якісний рівень оперативного управління припускає забезпечення необхідної й достатньої інтелектуальної підтримки. Запропонована в роботі функціональна схема системи інтелектуальної підтримки (СІП) оперативного управління наведена на рисунку 3.3.

У системі інтелектуальної підтримки оперативного управління пропонується використовувати інтелектуальні технології:

– механізм нечіткого логічного виводу для чисельної оцінки ймовірності атаки;

– організоване впорядкування інформації про події в базі знань;

– моделі протидії погрозам;

– прийняття рішень на вибір раціонального варіанта реагування на події безпеки.

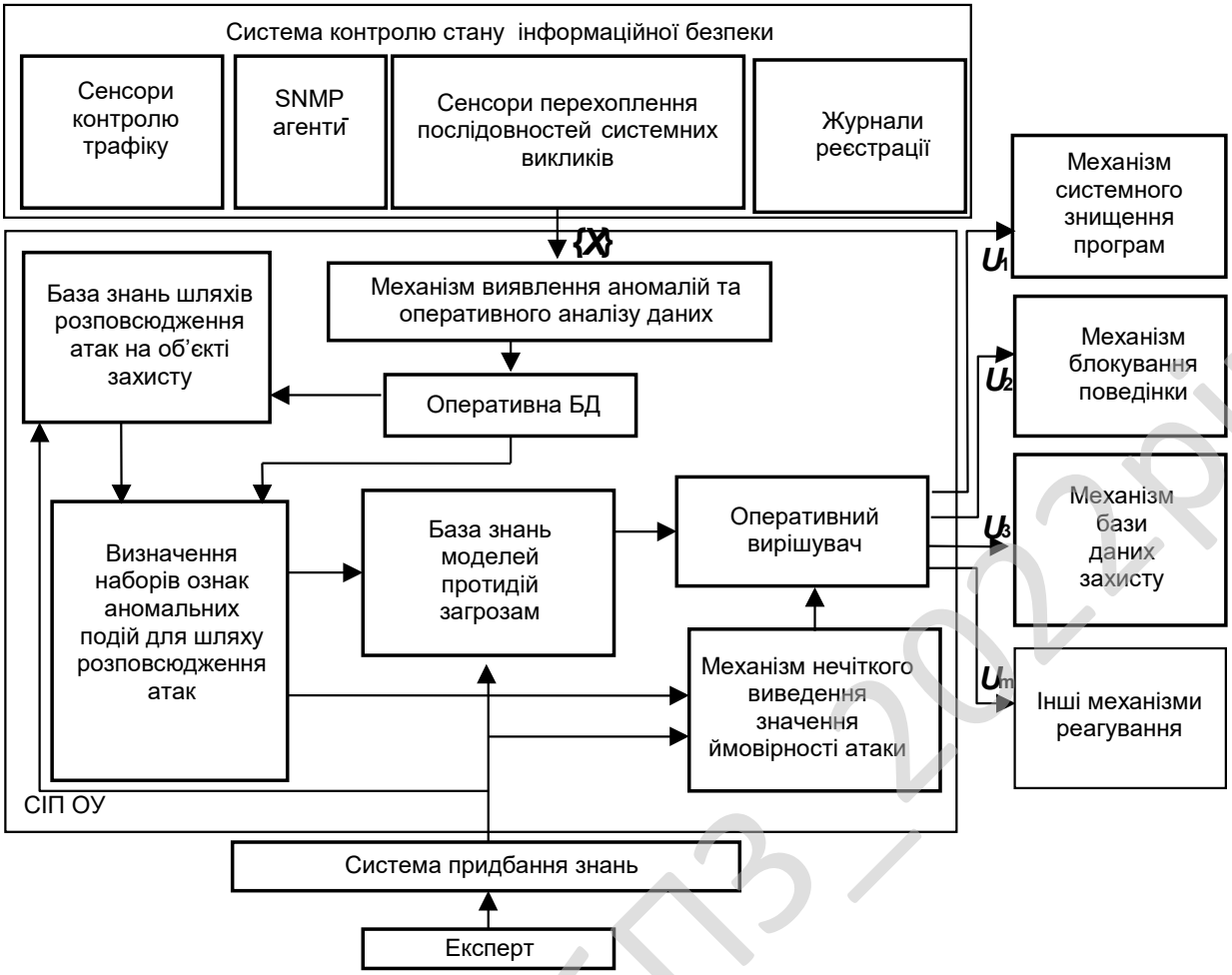

Рисунок 3.3 – Функціональна схема системи інтелектуальної підтримки оперативного управління ЗІ

Функціональна схема системи інтелектуальної підтримки організаційно-технічного управління ЗІ, у якій реалізується метод формування раціонального комплексу засобів захисту, представлена на рисунку 3.4.

Через необхідність максимальної структуризації розроблювальної системи й рішень, пропонується трьохрубіжна модель захисту, що щонайкраще задовольняє всієї сукупності умов її розробки, експлуатації й удосконалення. Трьохрубіжна модель захисту – неформалізований опис комплексу програмно-апаратних засобів захисту, що є основою для розробки системи захисту:

– перший рубіж – периметр об'єкта захисту – набір функціональних підсистем, що включають засоби захисту від зовнішніх вторгнень зловмисника й потенційно можливих погроз віддаленого користувача;

– другий рубіж – набір засобів захисту мережного сегмента від віддалених і локальних мережних вторгнень;

– третій рубіж містить у собі набір засобів захисту окремого персонального комп'ютера або сервера.

У процесі організаційно-технічного управління, планування ЗІ як функція управління являє собою процес послідовного зняття невизначеності щодо структури й состава засобів захисту в СЗІ. Процес планування $Р_{пл}$ раціональних наборів ЗЗ характеризується за допомогою вираження:

$$P_{\text{пл}} = \Phi \to S_r,$$

де Ф – множина функціональних підсистем для рубежу захисту;

S_r – обраний набір засобів захисту.

На першому етапі задається множина функціональних підсистем для рубежів захисту, результатом планування є командна інформація, що містить конкретні дані по розподіляються ресурсам, що, що направляється на досягнення цільового стану СЗІ.

Процес ухвалення рішення про вибір раціонального варіанта набору ЗЗ для рубежу захисту – це функція перетворення змісту інформації про вимоги, запропонованих до засобів захисту, що входить у набір, про характеристики засобів захисту, у підмножину найкращих варіантів набору $S' \subseteq S$. Множина варіантів набору:

$$S = \{S_1, \ldots, S_r, \ldots, S_R\},$$

де R – число варіантів альтернативних наборів, з яких здійснюється вибір.

Для вибору раціонального варіанта набору засобів захисту використовується цільова функція J:

$$S_r = J(S).$$

Сукупність відомостей, що дозволяють зіставляти варіанти наборів, це характеристики засобів захисту функціональних підсистем для рубежу – множина W, що включає в себе дві підмножини:

$$W_{\text{зщ}_l} \subset W_l \text{ і } W_{\text{и}_l} \subset W_l,$$

де $W_{\text{зщ}_l}$ – показник засобів захисту «захищеність інформації»;

$W_{\text{и}_l}$ – показник засобів захисту «витрати» для l-ої функціональної підсистеми.

На основі морфологічного підходу модель прийняття рішень на вибір раціонального варіанта набору може бути представлена у вигляді кортежу:

$$\text{ПР: } \langle \text{Ц}, \Phi, \Pi_s, S, W_l, J, S_r(S') \rangle,$$

де Ц – ціль ухвалення рішення;

Ф – вихідні дані для породження варіантів набору засобів захисту:

$$\Phi = \{\Phi_1, \Phi_2, \ldots, \Phi_l, \ldots, \Phi_L\};$$

Π_s – правило породження варіантів набору, що може бути представлене в аналітичному виді як векторний добуток множин:

$$S = \Phi_1 \times \Phi_2 \times \ldots \times \Phi_l \times \ldots \times \Phi_L,$$

де Φ_l – множина, що складається із засобів захисту l-ої функціональної підсистеми:

$$\Phi_l = \{A_{l1}, A_{l2}, \ldots A_{lm} \ldots, A_{lK_l}\};$$

S – множина породжених варіантів набору;

W_l – дані для вибору раціональних варіантів;

J – цільова функція для вибору раціонального набору засобів захисту(правило вибору);

S_r – раціональний набір засобів захисту.

Відзначається, що в умовах автоматизованого управління й при використанні експертної інформації в процесі ухвалення рішення можна говорити (навіть у випадку формалізованого правила вибору) про раціональне, а не оптимальне рішенні.

Відповідно до трьохрубіжної моделі захисту, основою планування раціонального модульного состава СЗІ є функціональні вимоги до наборів ЗЗ для кожного рубежу, які формулюються на основі нормативної документації, відповідно до рівня критичності оброблюваної інформації. Альтернативні засоби захисту для кожної функціональної підсистеми набору засобів захисту вибираються з урахуванням цих вимог. Варіантів наборів, сертифікованих по необхідному класі захищеності, може бути багато. Порівняння варіантів наборів засобів захисту пропонується робити по кількісній мері.

Для рішення завдання вибору раціональних варіантів наборів засобів захисту для рубежів захисту розробляється метод обробки знань, що використовує неформалізуємий досвід експерта в області ЗІ, що забезпечує перетворення відомостей про характеристики засобів захисту з бази знань і вивід рішення в аналітичній формі – метод формування раціонального комплексу засобів захисту для СЗІ.

1. Розробляються варіанти набору ЗЗ. Множина можливих варіантів рішення завдання вибору задається морфологічною матрицею. Розробляються морфологічні матриці засобів захисту для трьох рубежів.

2. Заповнюються допоміжні матриці, у яких відзначаються сумісні один з одним програмно-апаратні засоби. Допоміжна квадратна матриця сумісних рішень заповнюється в такий спосіб: для кожної пари засобів захисту різних функціональних підсистем визначається, чи сумісні вони, і результат заноситься в таблицю. Якщо ЗЗ сумісні, то функція сумісності $s(A_{lm}, A_{pr}) = 1$, у противному випадку $s(A_{lm}, A_{pr}) = 0$.

3. Генерується безліч рішень на вибір варіантів набору ЗЗ із усіканням цієї множини до підмножини варіантів набору із сумісних між собою програмно-апаратних продуктів.

Множина $S = \{S_1, ...S_r, ..., S_R\}$, що складається із всіх можливих варіантів побудови набору ЗЗ для рубежу, є декартовим добутком множин альтернатив (рядків морфологічної матриці).

Елемент множини:

$$S_r = \left\{ \left(A_{1i}, A_{2j}, \ldots A_{lm}, \ldots A_{Ln} \right) : A_{lm} \in \Phi_l, \forall l = \overline{1, L} \right\},$$

де L – число функціональних підсистем для рубежу;

A_{lm} – засіб захисту для реалізації l-ой функціональної підсистеми.

Генерація множин рішень на вибір варіантів набору, що складаються із сумісних між собою ЗЗ, здійснюється в такий спосіб.

Відбувається ітераційний синтез варіантів набору, що складаються із сумісних ЗЗ: на першому кроці перебираються послідовно варіанти засобів захисту для першої підсистеми, після вибору альтернативи A_{1i} здійснюється перехід до другого кроку. На другому кроці виконується послідовний перебір варіантів засобів захисту другої підсистеми, але вибір здійснюється тільки для таких альтернатив A_{2j}, для яких функція сумісності $s(A_{1i}, A_{2j}) = 1$ і т.д. При виборі альтернатив з l-ої підсистеми вибір здійснюється тільки з таких альтернатив A_{lm}, для яких функції сумісності дорівнюють одиниці: $s(A_{l-1,m}, A_{lm}) = 1$, $s(A_{2j}, A_{lm}) = 1$, $s(A_{1i}, A_{lm}) = 1$... Таким, образом, вибір ЗЗ із кожного рядка морфологічної матриці (по одній з кожного рядка) для формування варіанта набору здійснюється тільки із сумісних між собою програмно-апаратних продуктів.

4. Подальше усікання множини S виконується методом повного перебору по заданій цільовій функції. Як цільова функція для вибору варіанта набору $S_r = \left\{ A_{1i}, A_{2j}, \ldots A_{lm}, \ldots A_{Ln} \right\}$, застосовується функція:

$$J = \max_r \frac{W_{K_{\text{зщ}}^1}^{A_{1i}} + \ldots + W_{K_{\text{зщ}}^1}^{A_{lm}} + \ldots + W_{K_{\text{зщ}}^1}^{A_{Ln}}}{W_{K_{\text{и}}^1}^{A_{1i}} + \ldots + W_{K_{\text{и}}^1}^{A_{lm}} + \ldots + W_{K_{\text{и}}^1}^{A_{Ln}}},$$

де $W_{K_{\text{зщ}}^1}^{A_{lm}}$ – значення показника «захищеність»;

$W_{K_{\text{и}}^1}^{A_{lm}}$ – значення показника «витрати» засобу захисту A_{lm}.

Для оцінки засобів захисту різних функціональних підсистем наборів розробляються ієрархічні структури узагальнених критеріїв якості засобів захисту: показник «захищеність» і показник «витрати».

Критерії якості засобів захисту по ієрархії «захищеність» діляться на дві групи: показники забезпечення ефективності оперативних методів захисту й показники функціональної придатності. Критерії якості по ієрархії «витрати» діляться також на дві групи: у першу включена вартість відповідного засобу захисту, число користувачів по однієї ліцензії й інші можливі економічні витрати; до другої групи витрат ставляться функціональні витрати, такі, наприклад, як падіння продуктивності інформаційної системи при використанні даного засобу захисту.

Оцінка засобів захисту й критеріїв здійснюється попарним порівнянням по методу Т. Саати, результати приводяться в числовому виді. З використанням ієрархічних структур критеріїв якості ЗЗ обчислюються нормовані значення власних векторів засобів захисту за всіма критеріями до показників «захищеність» $K^1_{\text{зщ}}$ і «витрати» $K^1_{\text{и}}$ на підставі обробки всіх матриць попарних порівнянь із урахуванням зв'язків критеріїв.

Після вибору раціональних наборів засобів захисту для рубежів захисту отриманий раціональний модульний состав цілісного комплексу засобів захисту об'єкта, що задовольняє вимозі

$$J \rightarrow \max.$$

5. Оцінюється, чи задовольняє сформований комплекс засобів захисту вимозі:

$$C_{\Sigma} \le C$$

де C_{Σ} – сумарні витрати на реалізацію комплексу ЗЗ;

$C_{\text{доп}}$ – виділені на реалізацію комплексу грошові ресурси.

При цьому C_{Σ} обчислюється за допомогою наступного вираження:

$$C_{\Sigma} = \sum_{S} \left(\sum i_S + \sum C^{\text{с}}_{j_S} + \sum C^{\text{в}}_{k_S} + C_{\text{сегм}} \right) + C_{\text{пр}},$$

де S – число мережних сегментів;

$C^{\text{Б}}_{i_S}$ – вартість набору засобів захисту хоста, на якому обробляється інформація базового рівня критичності;

$C^{\text{С}}_{j_S}$ – вартість набору засобів захисту хоста, на якому обробляється інформація середнього рівня критичності;

$C^{\text{В}}_{k_S}$ – вартість набору засобів захисту хоста, на якому обробляється інформація високого рівня критичності;

$C_{\text{сегм}_S}$ – вартість набору засобів захисту на границі s-го мережного сегмента;

$C_{\text{пр}}$ – вартість наборів засобів захисту периметра.

Вибір комплексу засобів захисту для СЗІ досягається ітераційно шляхом наближення до раціонального состава, що задовольняє вимогам до припустимих витрат на його реалізацію.

У системі інтелектуальної підтримки раціональні рішення пропонується вибирати на основі використання експертних знань; у ній реалізується механізм придбання знань у процесі заповнення полів знань експертом при взаємодії його з автоматизованою системою, виконується сукупність процедур над проблемною областю з використанням багатокритеріального порівняльного аналізу для виявлення в заданому експертом множини підмножини найкращих за критеріями переваги варіантів наборів, з яких формується раціональний комплекс засобів захисту.

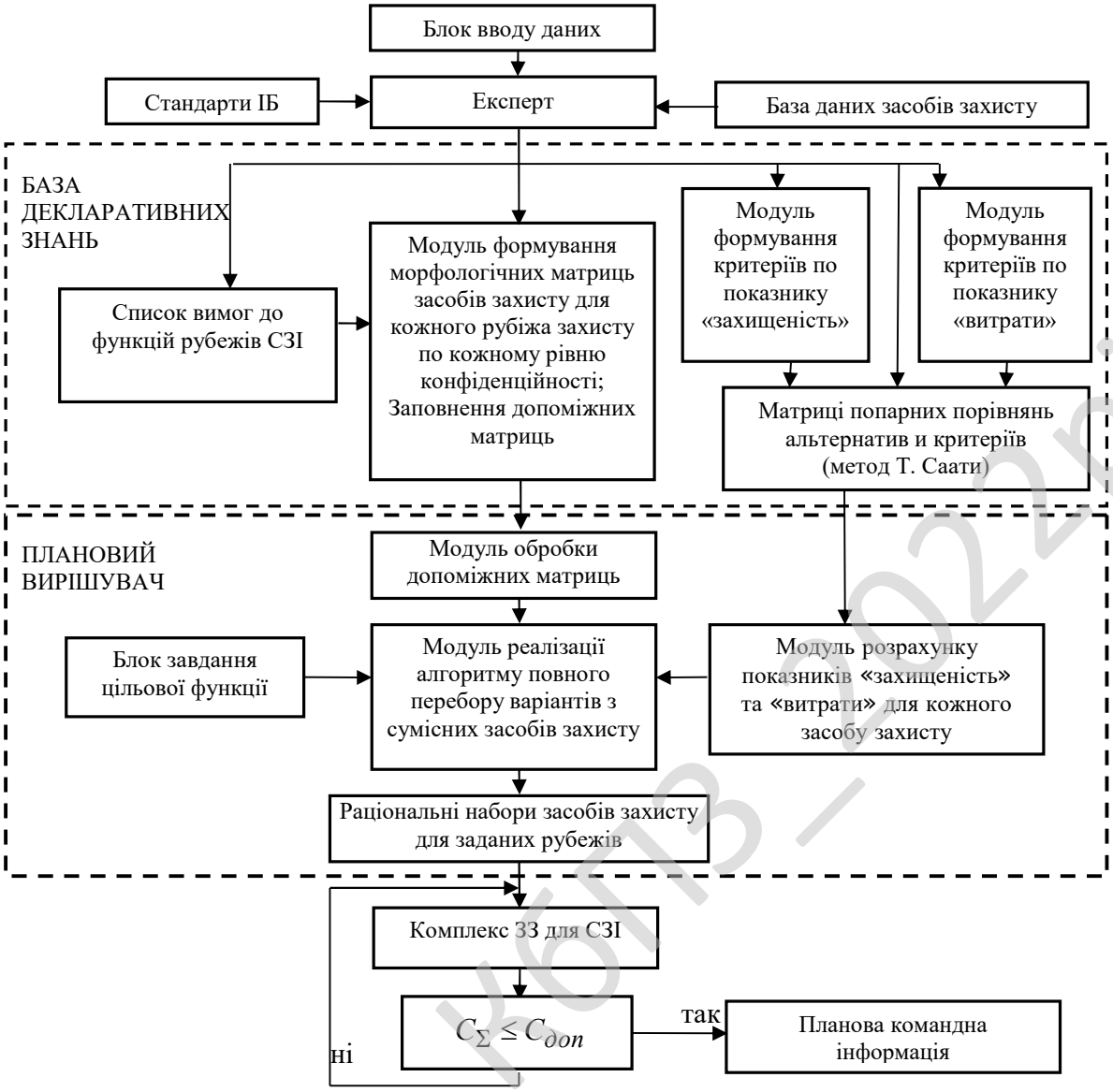

Рисунок 3.4 – Функціональна схема системи інтелектуальної підтримки прийняття рішень по організаційно-технічному управлінню ЗІ

3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.5. З діаграми взаємодії процесів ми бачимо, що у системі відбуваються наступні процеси.

Робота системи починається з запуску процесу початку/кінця роботи, який взаємодіє з наступними процесами:

– Процес введення даних.

– Процес вибору комплексу засобів захисту для систем захисту інформації.

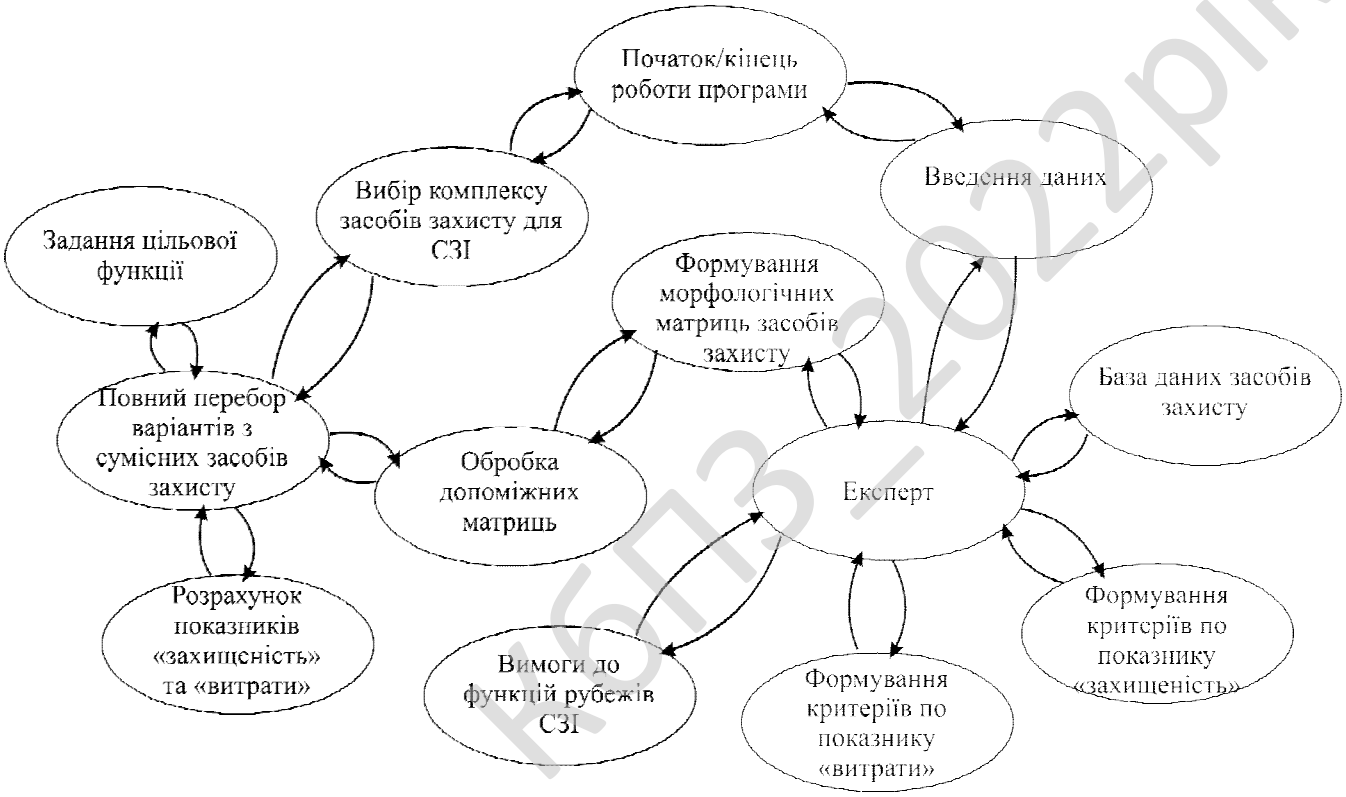

Рисунок 3.5 – Діаграма процесів системи

Процес введення даних взаємодіє з процесом запуску експертної системи.

Цей процес взаємодіє з наступними процесами:

– Процес заповнення та роботи з базою даних засобів захисту.

– Процес формування критеріїв по показнику «захищеність».

– Процес формування критеріїв по показнику «витрати».

– Процес визначення вимог до функцій рубежів системи захисту інформації.

– Процес формування морфологічних матриць засобів захисту.

Процес формування морфологічних матриць засобів захисту взаємодіє з процесом обробки допоміжних матриць.

Останній процес у свою чергу взаємодіє з процесом запуску повного перебору варіантів з сумісних засобів захисту.

Процес запуску повного перебору варіантів з сумісних засобів захисту взаємодіє з наступними процесами:

– Процесом розрахунку показників «захищеність» та «витрати».

– Процесом задання цільової функції.

– Процесом вибору комплексу засобів захисту для системи захисту інформації.

Процес вибору комплексу засобів захисту для системи захисту інформації є кінцевим у системі й взаємодіє з процесом початку/кінця роботи програмного продукту.

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Блок-схема програмного забезпечення розробленої системи наведена на рисунку 4.1.

З нього ми бачимо, що програма починається з виведення заставки.

Після цього відбувається виведення вікна вибору критеріїв виявлення загроз роботі мережі.

Якщо натискається кнопка «Запуск», то відбувається виконання наступних кроків:

– Запускається експертна система оцінки стійкості, у плані інформаційної безпеки, комп'ютерної мережі.

– Виводяться результати аналізу мережі на основі технології Cisco Self-Defending Network.

– Виводяться рекомендації по забезпеченню безпечної роботи комп'ютерної мережі.

– Виводиться перелік засобів захисту комп'ютерної мережі.

Якщо ж користувач не натискає кнопку «Запуск», то він може перейти у режим додавання критеріїв до бази знань.

Якщо користувач не бажає додавати критерії у базу знань, то він може перейти у режим додавання загроз безпеці інформації, яка циркулює у комп'ютерній мережі, до бази знань.

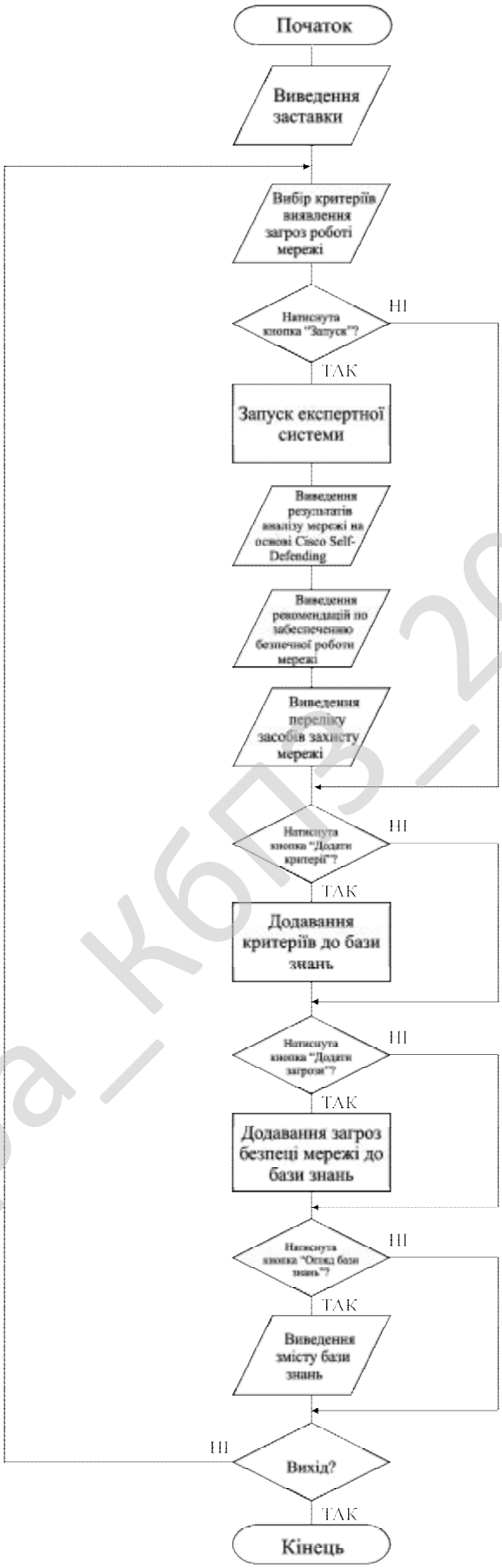

Рисунок 4.1 – Блок-схема алгоритму роботи основної програми

Якщо користувач не бажає додавати загрози безпеці мережної інформації, то він може перейти у режим «Огляд бази знань», який дозволяє оглянути наступні дані:

– Перелік загроз інформації.

– Перелік засобів захисту.

– Перелік критеріїв.

– Перелік рекомендацій по підвищенню безпеки комп'ютерної мережі.

Якщо користувач не обирає ні одну з вищеперерахованих дій, то він покидає систему захисту інформації в корпоративній мережі на основі технології Cisco Self-Defending Network.

На рисунку 4.2 зображена блок-схема алгоритму роботи підпрограми редагування бази знань.

Вона працює наступним чином.

Спершу відбувається виведення списку загроз інформаційній безпеці комп'ютерної мережі.

Після цього відбувається виведення списку критеріїв.

Наступним кроком є виведення додаткової інформації.

Якщо необхідно додати нові загрози безпеці інформації у базу знань, то відбуваються наступні дії:

– Виведення назви загрози безпеці комп'ютерній мережі.

– Вибір критеріїв, за допомогою яких дану загрозу можна виявити.

– Введення стислої інформації про загрозу.

– Введення методів та засобів захисту протидії загрозі.

– Додавання загрози безпеці мережі до бази знань.

У іншому випадку відбувається перевірка на видалення загрози із бази знань.

Якщо загрозу потрібно видалити, то відбуваються наступні дії:

– Виділяється загроза, яку необхідно видалити.

– Видаляється вибрана загроза безпеці мережі з бази знань.

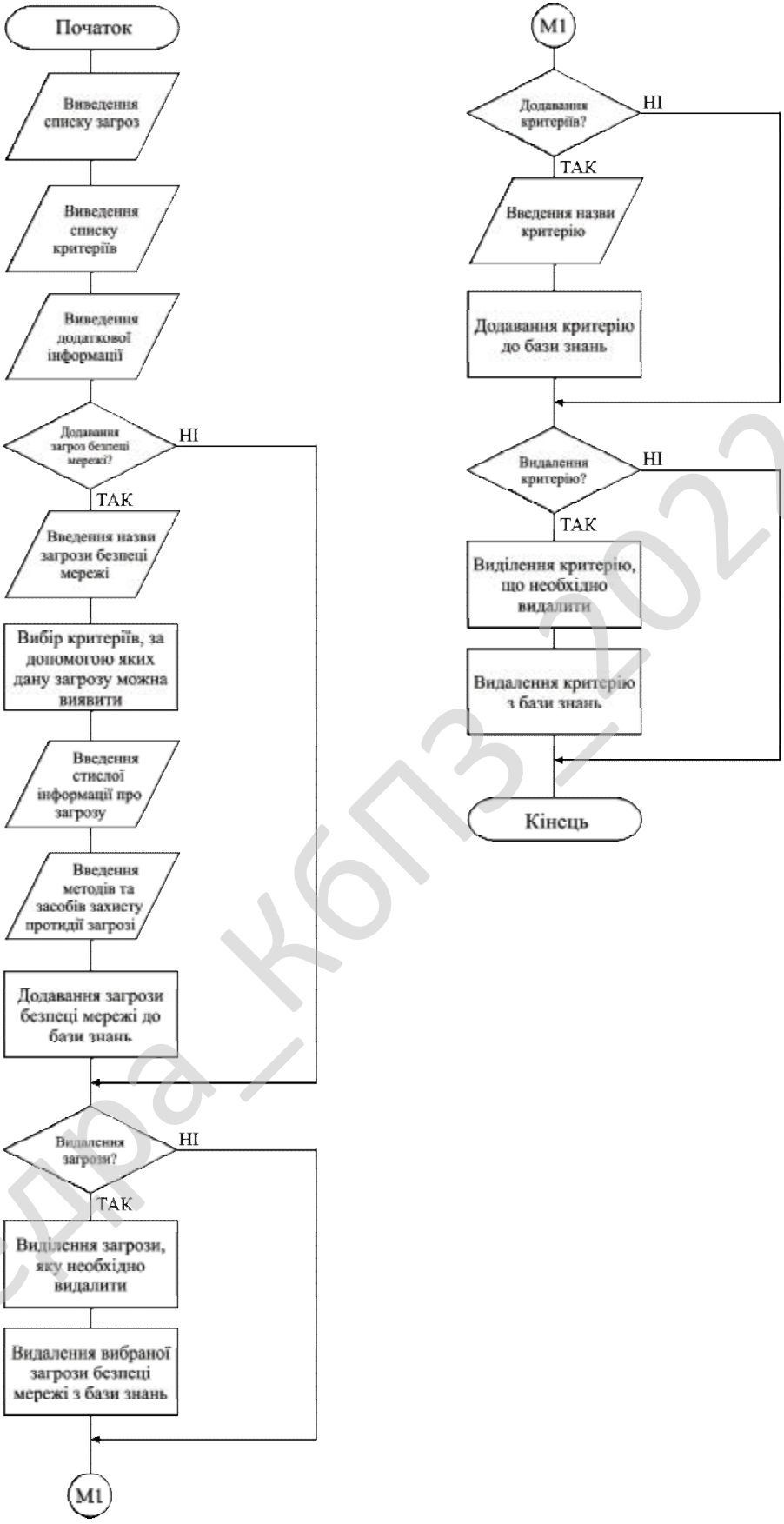

Рисунок 4.2 – Блок-схема алгоритму роботи підпрограми редагування бази знань

Наступним вибором, який може здійснити користувач є обирання, додавати критерії, або ні.

Якщо необхідно додавати критерії, то відбувається наступна послідовність ітерацій:

– Вводяться назви критерію.

– Відбувається додавання критерію до бази знань.

Якщо необхідно видалити критерій, то відбувається наступна послідовність дій:

– Виділяється критерій, який необхідно видалити.

– Відбувається видалення критерію з бази знань.

На цьому підпрограма редагування бази знань закінчує свою роботу.

Нижче наведемо деякі процедури, які реалізовані у програмному продукті.

```
begin
//Вивід списку критеріїв
  Query1.SQL.Clear;
  Query1.SQL.Add('select * from krt order by Name asc');
  Query1.Open;
  Query1.First;
  while not Query1.Eof do begin
    CheckListBox1.Items.Add(Query1.Fields[1].AsString);
    Query1.Next;
  end;
end;
//Процедура для StatusBar
procedure TFormMain.ShowHint(Sender: TObject);
begin
  StatusBar1.SimpleText:=Application.Hint;
end;
//Процедура для StatusBar
procedure TFormMain.FormShow(Sender: TObject);
begin
  Application.OnHint:=ShowHint;
end;
//Натиснення кнопки "Додати критерії"
procedure TFormMain.Image3Click(Sender: TObject);
begin
  FormAddKr.ShowModal;
```

```
end;
procedure TFormMain.Image6Click(Sender: TObject);
type
  Zagrzes=record
    name:string;
    probability:integer;
    all_kr:integer;
  end;
var str, prob:string;
  probab,probab2,p1,p2:real;
  i,j,count_krt,kol,num_zagr:integer;
  a:array[0..128] of integer;
  zagr:array[0..128] of Zagrzes;
  temp: Zagrzes;
begin
  Image6.Visible:=False;
  count_krt:=0;
  kol:=0;
  num_zagr:=0;
  for i:=0 to 128 do a[i]:=0;
  for i:=0 to 128 do zagr[i].probability:=0;
  //Очищення Memo1, Memo2
  Memo1.Clear;
  Memo2.Clear;
  //Формування списку номерів вибраних критеріїв
  for i:=0 to CheckListBox1.Items.Count-1 do begin
    if CheckListBox1.Checked[i] then begin
      str:=CheckListBox1.Items[i];
      Query1.First;
      while not Query1.Eof do begin
        if Query1.Fields[1].AsString=str then begin
          a[count_krt]:=Query1.Fields[0].AsInteger;
          count_krt:=count_krt+1;
        end;
        Query1.Next;
      end;
    end;
  end;
  //Відправляємо запит в БД загроз безпеці мережі
  Query1.Active:=False;
  Query1.SQL.Clear;
  Query1.SQL.Add('select * from Zagrzesses order by Name asc');
```

```
  Query1.Open;
  Query1.First;
  while not Query1.Eof do begin
    zagr[num_zagr].all_kr:=0;
    for i:=2 to 11 do begin
      if Query1.Fields[i].AsInteger<>0 then
zagr[num_zagr].all_kr:=zagr[num_zagr].all_kr+1;
      for j:=0 to count_krt do begin
        if (Query1.Fields[i].AsInteger=a[j]) and (Query1.Fields[i].AsInteger<>0)
then begin
          //Memo1.Lines.Add('OK - '+IntToStr(a[j]));
zagr[num_zagr].probability:=zagr[num_zagr].probability+1;
          zagr[num_zagr].name:=Query1.Fields[1].AsString;
        end;
      end;
    end;
    Query1.Next;
    num_zagr:=num_zagr+1;
  end;
  //Сортування
{   for j:=0 to 128 do begin
   for i:=0 to 128 do begin
    probab:=zagr[i].probability/zagr[i].all_kr;
    if zagr[i+1].probability<>0 then begin
      probab2:=zagr[i+1].probability/zagr[i+1].all_kr;
      if probab<probab2 then begin
        temp.name:=zagr[i+1].name;
        temp.probability:=zagr[i+1].probability;
        temp.all_kr:=zagr[i+1].all_kr;
        zagr[i+1].name:=zagr[i].name;
        zagr[i+1].probability:=zagr[i].probability;
        zagr[i+1].all_kr:=zagr[i].all_kr;
        zagr[i].name:=temp.name;
        zagr[i].probability:=temp.probability;
        zagr[i].all_kr:=temp.all_kr;
      end;
    end;
    end;
   end;
}
  //Вивід результату
  for i:=0 to 128 do begin
```

```
    if zagr[i].probability<>0 then begin
      probab:=(zagr[i].probability/zagr[i].all_kr)*100;
      Memo1.Lines.Add(zagr[i].name + '   (Вірогідність:
'+FloatToStrF(probab,ffNumber,3,1)+'%)');
    end;
  end;
  //Вивід детальної інформації
  for i:=0 to 128 do begin
    if zagr[i].probability<>0 then begin
      Query1.SQL.Clear;
      Query1.SQL.Add('select * from about where Name="'+zagr[i].name+'"');
      Query1.Open;
      Query1.First;
      Memo2.Lines.Add('ЗАГРОЗА: '+Query1.Fields[0].AsString);
      Memo2.Lines.Add('       '+Query1.Fields[1].AsString);
      Memo2.Lines.Add(' ');
      Memo2.Lines.Add('КРИТЕРІЇ ВИЯВЛЕННЯ ЗАГРОЗИ БЕЗПЕКИ ІНФОРМАЦІЇ У МЕРЕЖІ:');
      Memo2.Lines.Add('       '+Query1.Fields[2].AsString);
      Memo2.Lines.Add(' ');
      Memo2.Lines.Add('МЕТОДИ ТА ЗАСОБИ ЗАПОБІГАННЯ ДАНІЙ ЗАГРОЗІ БЕЗПЕКИ
ІНФОРМАЦІЇ У МЕРЕЖІ:');
      Memo2.Lines.Add('       '+Query1.Fields[3].AsString);
      Memo2.Lines.Add(' ');
      Memo2.Lines.Add('-------------------------------------------------------
-------------------------------------------------------------------------------
----------------------------------------');
      Memo2.Lines.Add(' ');
    end;
  end;
  //Перемотування Memo на початок
  Memo1.SelStart := 0;
  Memo1.Perform(EM_SCROLLCARET, 0, 0);
  Memo2.SelStart := 0;
  Memo2.Perform(EM_SCROLLCARET, 0, 0);
  Query1.SQL.Clear;
  Query1.SQL.Add('select * from krt order by Name asc');
  Query1.Open;
  Query1.First;
  Image6.Visible:=True;
end;
```

Оцінювання рівня захищеності інформації здійснюється на основі одного з основних положень уніфікованої концепції захисту – вимоги науково обґрунтованого підходу до оцінки (бажано в кількісному вираженні) необхідного рівня захищеності при проектуванні й у процесі експлуатації СЗІ.

У процесі аналізу й оцінювання ризиків установлюється ступінь адекватності використовуваних або планованих наборів засобів захисту (ЗЗ) існуючим погрозам. Властивість «захищеність інформації» кожного ЗЗ, що входить у СЗІ, у сукупності визначає захищеність інформації в СЗІ в цілому. Наявність уразливості ЗЗ може привести до порушення захищеності, тобто здійсненню погрози, тому при рішенні завдань захисту інформації першорядне значення має кількісна оцінка уразливістей засобів захисту. Оскільки вплив на інформацію різних деструктивних факторів значною мірою є випадковим, то як кількісну міру уразливості найбільше доцільно застосувати ймовірність порушення захищеності інформації.

Неясність способу визначення значень ймовірностей погроз і уразливістей є основною проблемою при одержанні кількісної оцінки ризику порушення інформаційної безпеки. Відомо, що застосування методів класичної теорії ймовірностей припустимо при повторюваності дослідів і однаковості умов. Це вимога в складних системах, якими є СЗІ, звичайно не виконується. Відповідно до одному із принципів системного аналізу – принципу невизначеності, у процесі дослідження системи необхідний облік невизначеностей і випадків. Оскільки складні відкриті системи не підкоряються імовірнісним законам, у них варто оцінити найгірші ситуації відповідно до методу гарантованого результату, що пропонується використовувати при оцінці ймовірностей погроз.

Приймається, що значення показника m-го ЗЗ захищеність інформації $P_{бт}$ – це суб'єктивна ймовірність виявлення й блокування засобом захисту несанкціонованих дій, тобто теоретична очікувана ефективність бар'єра.

Очевидно, що ймовірність порушення захищеності $P_{бm}^{н}$ доповнює $P_{бm}$ до одиниці, тобто:

$$P_{бm}^{н} = 1 - P_{бm} ,$$

де $P_{бm}^{н}$ – імовірність порушення захищеності інформації, або ймовірність уразливості m-го ЗЗ (імовірність подолання бар'єра).

Пропонується ймовірносно-статичний підхід, при якому не враховується динаміка зміни значень ймовірностей погроз і уразливістей у часі, оцінюються апріорні очікувані значення ймовірностей порушення захищеності інформації.

Особливістю пропонованого в магістерській роботі підходу є одержання чисельних значень суб'єктивних ймовірностей на основі використання як приватні показники захищеності технічних характеристик і можливостей засобів захисту, декларуємих розроблювачами. Вирішується завдання одержання чисельної оцінки узагальненого показника якості засобу захисту.

Пропонується для одержання чисельної оцінки узагальненого показника якості засобу захисту захищеність інформації використовувати теорію нечітких множин. Для оцінки засобів захисту за кожним критерієм нижнього ієрархічного рівня формуються функції приналежності. При цьому використовуються методи побудови функцій приналежності, засновані на формалізації й інтеграції нечітких даних, сформованих експертом у процесі оцінювання параметрів реальних засобів захисту. Формулюються відповідні продукційні правила, що дозволяють обробляти складні з'єднання. Достоїнство способу – відносно висока об'єктивність.

Метод оцінювання рівня захищеності інформації базується на трьохрубіжній моделі захисту, він розроблений для об'єкта захисту, архітектура якого відповідає основним принципам безпеки, що рекомендуються.

Відомо, що рівень захищеності й відносний ризик доповнюють один одного до одиниці. Пропонується розраховувати рівень захищеності η за формулою:

$$\eta = 1 - \overline{R} = 1 - \sum_{s} \frac{C_s}{C_\Sigma} \cdot P_s ,$$

де $\overline{R}$ – відносний ризик;

C_s – частка вартості інформаційних ресурсів, що захищаються, у сегменті s;

s – номер сегмента;

S – число сегментів;

P_s – результуюча ймовірність погроз інформаційному середовищу сегмента;

C_Σ – сумарний неприйнятний збиток;

$\frac{C_s}{C_\Sigma}$ – коефіцієнт небезпеки сукупності погроз в s-ому сегменті, обумовлений як частка вартості інформації, що захищається, об'єкта захисту, оброблюваного в сегменті.

Таким чином, для оцінки рівня захищеності потрібна кількісна оцінка ймовірностей реалізації каналів несанкціонованого доступу.

Для оцінки ймовірності порушення захищеності підмножиною порушників $\{K'\}$ по підмножині можливих каналів несанкціонованого одержання інформації $\{J'\}$ для сегмента s використовується співвідношення:

$$P_{s\{J'\}\{K'\}} = 1 - \prod_{J'} \left(1 - P_{sjk}^{(б)}\right) \prod_{K'} \left(1 - P_{sjk}^{(б)}\right),$$

у якому приймається:

$$P_{sjk}^{(б)внш} \subset P_{sjk}^{(б)}, P_{sjk}^{(б)вн} \subset P_{sjk}^{(б)},$$

де $P_{sjk}^{(б)вн}, P_{sjk}^{(б)внш}$ – імовірність несанкціонованого одержання інформації, оброблюваної в s-му сегменті, відповідно, внутрішнім і зовнішнім порушником (зловмисником) для об'єкта захисту, що має точки виходу в глобальну мережу, зовнішні виділені канали зв'язку, для якого можливі віддалені атаки через периметр.

З обліком прийнятої трьохрубіжної моделі захисту $P_{sjk}^{(б)внш}$ обчислюється за формулою:

$$P_{sjk}^{(б)внш} = 1 - \prod_{l=1}^{3} (1 - P_{sjk\ l}^{внш}),$$

де $P_{sjk\,l}^{внш}$ – імовірність несанкціонованого одержання інформації, оброблюваної в *s*-му сегменті, зловмисника, зовнішнього порушника у випадку подолання відповідного рубежу захисту *l*.

Імовірність $P_{sjk\,l}^{внш}$ залежить від чотирьох факторів і визначається залежністю:

$$P_{sjkl}^{внш} = P_{skl}^{д} \cdot P_{sjkl}^{н} \cdot P_{sjl}^{к} \cdot P_{sjl}^{и},$$

де $P_{sk\,l}^{д}$ – імовірність спроби доступу зловмисника або зовнішнього порушника-користувача до *l*-му рубежу захисту;

$P_{sjk\,l}^{н}$ – імовірність подолання зловмисником або зовнішнім порушником *l*-го рубежу захисту;

$P_{sj\,l}^{к}$ – імовірність наявності трафіка із сегмента s через l-*й* рубіж захисту, залежить від технології обробки інформації на об'єкті захисту, імовірність можна прийняти рівній частоті роботи каналу;

$P_{sj\,l}^{и}$ – імовірність наявності інформації, що захищається, *s*-го сегмента в трафіку в момент подолання зовнішнім порушником *l*-го рубежу захисту, залежить від технології обробки інформації на об'єкті захисту.

Внутрішній порушник у процесі реалізації каналів несанкціонованого доступу повинен перебороти два рубежі захисту.

Тоді ймовірність несанкціонованого одержання інформації, оброблюваної в сегменті s, внутрішнім порушником обчислюється за формулою:

$$P_{sj}^{(б)вн} = 1 - \prod_{l=1}^{2} (1 - P_{sjl}^{вн}),$$

де $P_{sl}^{\text{вн}}$ – імовірність несанкціонованого доступу до інформації, оброблюваної в *s*-му сегменті, внутрішнього порушника у випадку подолання відповідного рубежу захисту *l*.

З перерахованих ймовірностей, що входять у формули для розрахунку $P_{sjl}^{\text{вн}}$ й $P_{sjkl}^{\text{внш}}$, одна з ймовірностей, а саме $P_{sjkl}^{\text{н}}$, залежить від якості використовуваних у системі засобів захисту й кількості бар'єрів на рубежі захисту. Якщо порушникові необхідно перебороти *M* бар'єрів на рубежі захисту, то ймовірність його вдалої атаки визначається як добуток:

$$P_{sjkl}^{\text{н}} = \prod_{m=1}^{M} P_{\text{б}m}^{\text{н}} = \prod_{m=1}^{M} (1 - P_{\text{б}m}).$$

На основі пропонованого методу оцінки ризику порушення інформаційної безпеки розробляються алгоритм і функціональна модель прогнозування рівня захищеності інформації за допомогою IDEF 0-технології.

Автоматизація прийняття рішень по управлінню ЗІ відповідає високому рівню моделі зрілості процесів управління, забезпечує обґрунтованість і раціональність рішень на основі використання математичного апарата, знижує працєвитрати на виконання обчислень.

Розроблено інструментальні програмні комплекси для автоматизованої системи інтелектуальної підтримки організаційно – технічного й оперативного управління ЗІ.

На основі програмного засобу «Прийняття рішень в умовах ризику», що реалізує метод вибору раціонального варіанта реагування на події безпеки, розроблений модуль «Оперативний вирішувач» системи інтелектуальної підтримки оперативного управління. Результати моделювання роботи модуля наведені в таблиці 4.1.

Таблиця 4.1 – Результати моделювання роботи модуля «Оперативний вирішувач»

Види погроз	Параметри інформаційного середовища / Варіанти реагування		
Локальне мережне вторгнення порушника	$A = 2, B = 3, P_a = 0{,}57$	$A = 1, B = 1, P_a = 0{,}238$	
	Завершення сесії з атакуючим вузлом	Відсилання попередження користувачеві	
Зовнішнє вторгнення по радіоканалу (Wi Fi, Wi Max)	$A = 3, C = 3,$ $P_a = 0{,}678$	$A = 1, З = 2, P_a = 0{,}4$	$A = 1, З = 1, P_a = 0{,}3$
	Блокування ТД	DOS-атака на атакуючу станцію	Відсутність реагування
Віддалена атака через периметр по лінії зв'язку	$A = 3, B = 4, C = 2,$ $P_a = 0{,}742$	$A = 1, B = 1, C = 1,$ $P_a = 0{,}238$	$A = 1, P_a = 0{,}08$
	Блокування доступу до сервісу в мережі	Переконфігурування сервісів безпеки з метою блокування IP	Відсилання попередження на IP-Адресу

Досвід розробки, впровадження й супроводи інформаційної системи показує, що якщо вона забезпечує бізнес-процеси досить великої організації, те практично ніколи не буває статичної. Як наслідок, рішення по безпеці, прийняті при проектуванні СЗІ, дуже швидко втрачають відповідність тій системі, ресурси якого вони покликані захищати. Тому в такій системі важливим є підтримка заданого рівня захищеності в процесі експлуатації постійно, що змінюється системи.

Інструментальний програмний комплекс «Система підтримки прийняття рішень по управлінню захистом інформації на об'єкті інформатизації», призначений для обґрунтованого вибору раціонального комплексу засобів

захисту при проектуванні СЗІ й у ході планування в процесі експлуатації, розроблений для системи інтелектуальної підтримки організаційно – технічного управління ЗІ. Для реалізації механізму придбання знань у рамках розробленого програмного комплексу організується взаємодію експерта з автоматизованою системою, у процесі якого експерт заповнює запропоновані йому розроблені поля знань. У плановому вирішувачі реалізований алгоритм пошуку шляхів від вхідних даних до вихідних – раціональним набором засобів захисту для заданих рубежів. Створений програмний продукт дозволяє не тільки одержати результат на основі уведених даних і знань, але й увести базу знань (створення, динамічне розширення, повторне використання).

Програмний комплекс «Система підтримки прийняття рішень по управлінню захистом інформації на об'єкті інформатизації» був використаний для рішення завдання забезпечення інформаційної безпеки в мережі транспортного підприємства – сегменті корпоративної інформаційної системи. Необхідність модернізації СЗІ виникла у зв'язку з недостатньою пропускною здатністю віртуальних каналів VPN, побудованих на базі програмного комплексу VIPNet Custrom 3.

Для збільшення швидкості обміну інформацією до 40 Мбіт/с у зв'язку зі зрослим обсягом трафіка, переданого по корпоративних каналах, розглядається можливість і економічна доцільність переходу із програмного рішення VPN на програмно – апаратний модуль NME – RVPN VIPNet (S – Terra CSP і Інфотекс) для маршрутизаторів Cisco 2811, що реалізує апаратне шифрування трафіка зі швидкістю 40 Мбіт/с.

Дане рішення, у зв'язку з необхідністю заміни маршрутизаторів, з одного боку, зажадає значних витрат, а з іншого боку – не приведе до комплексного рішення ЗІ. За допомогою програмного комплексу «Система підтримки прийняття рішень по управлінню захистом інформації на об'єкті інформатизації» вирішене завдання вибору раціонального модульного состава СЗІ для використання наявних у мережі маршрутизаторів Cisco (їхня заміна не буде

потрібна). Комплекс засобів захисту містить у собі сумісні з наявною апаратурою сертифіковані ДСТЗІ СБУ програмно – апаратні засоби вітчизняних і закордонних виробників.

Застосування запропонованого підходу до побудови комплексної СЗІ дозволяє зменшити витрати на захист на 44% у порівнянні з альтернативним комплексом, що забезпечує той же клас захищеності інформаційної системи 1Г.

Розроблений програмний комплекс для одержання чисельної оцінки рівня захищеності (ризику порушення інформаційної безпеки) «Розрахунок чисельного значення ризику порушення інформаційної безпеки (InfoRisk)» який дозволяє оцінити рівень захищеності на технічному рівні забезпечення ІЗ із використанням моделі погроз.

Програмний засіб має наступні можливості: дозволяє оцінювати рівень захищеності інформації на об'єкті захисту, що складається з множини сегментів, у яких обробляється інформація різного рівня критичності; дозволяє задавати вихідні дані по кількості сегментів, застосовуваних або планованих засобів захисту, по рівнях критичності інформаційних ресурсів; застосуємих на стадії розробки СЗІ й на стадії експлуатації системи; забезпечує оперативність оцінювання; дозволяє проводити порівняльний аналіз різних комплексів засобів захисту в ході управління ризиками; дозволяє однозначно враховувати специфіку функціонування конкретного об'єкта захисту, реальні погрози для конкретних ключових ресурсів; розрахунок ризику проводиться з мінімальним залученням експертів.

Знання експерта використовуються лише для побудови функцій приналежності й складаня продукційних правил при використанні інструментарію нечіткої логіки Fuzzy Toolbox програмного продукту Matlab, у якому обчислюються показники ЗЗ «захищеність інформації». Оцінювання даного показника здійснюється на основі використання відомостей про технічні характеристики засобів захисту.

Показник ризику формується на основі заданих значень цінностей потребуючі захисти ресурсів. Адекватність методу оцінювання ризику порушення ІБ, реалізованого в програмному комплексі InfoRisk, не залежить від наявності або відсутності достовірних статистичних даних по інцидентах ІБ, тим більше, що на етапі проектування СЗІ такі дані відсутні.

Наведено результати розрахунку прогнозованого ризику в СЗІ з раціональним модульним составом і до модернізації системи захисту. При проведенні розрахунків ураховувалася можлива наявність зловмисника, що реалізує віддалене вторгнення через периметр, наявність зовнішніх і внутрішніх користувачів – порушників і інсайдера, що має високі привілеї й порушує політику безпеки. Розрахункове прогнозоване значення ризику склало 1,84%, що в 6 разів менше значення ризику в існуючої СЗІ. Розрахунок проводився на основі методу гарантованого результату, для найгіршої ситуації.

У цілому, на основі проведених досліджень можна констатувати ефективність використаного системного підходу й запропонованих моделей, методів і алгоритмів для управління ЗІ у сегменті корпоративної інформаційної системи.

4.2 Захист розробленого програмного забезпечення

Дані в програмі захищаються за допомогою використання алгоритму Md5. Він отримує на вході повідомлення довільної довжини і створює на виході дайджест повідомлення довжиною 128 біт. Алгоритм складається з наступних кроків:

1. Додавання недостаючих біт. Повідомлення доповнюється так, щоб його довжина стала рівна 448 по модулю 512 (довжина 448 mod 512). Це означає, що довжина доданого повідомлення на 64 біта менше, ніж число, кратне 512. Додавання проводиться завжди, навіть якщо повідомлення має потрібну довжину. Наприклад, якщо довжина повідомлення 448 біт, воно доповнюється 512 бітами

до 960 біт. Таким чином, число біт, що додаються, знаходиться в діапазоні від 1 до 512.

Додавання складається з одиниці, за якою слідує необхідна кількість нулів.

2. Додавання довжини. 64-бітове представлення довжини початкового (до додавання) повідомлення в бітах приєднується до результату першого кроку. Якщо первинна довжина більша, ніж 264, то використовуються тільки останні 64 біта. Таким чином, поле містить довжину початкового повідомлення по модулю 264.

В результаті перших двох кроків створюється повідомлення, довжина якого кратна 512 бітам. Це розширене повідомлення представляється як послідовність 512-бітових блоків Y_0, Y_1,..., Y_{l-1}, при цьому загальна довжина розширеного повідомлення рівна L*512 бітам. Таким чином, довжина отриманого розширеного повідомлення кратна шістнадцяти 32-бітовим словам.

3. Ініціалізація MD-буфера. У алгоритмі Md5 використовується 128-бітовий буфер для зберігання проміжних і остаточних результатів хеш-функції. Буфер може бути представлений як чотири 32-бітові регістри (*A*, *B*, *C*, *D*). Ці регістри ініціалізувалися наступними шістнадцятковими числами:

$$A = 01234567$$

$$B = 89abcdef$$

$$C = Fedcba98$$

$$D = 76543210$$

4. Обробка послідовності 512-бітових (16-словних) блоків. Основою алгоритму Md5 є модуль, що складається з чотирьох циклічних обробок, позначений як Hmd5. Чотири цикли мають схожу структуру, але кожен цикл використовує свою елементарну логічну функцію, *ff*, що позначається, *fg*, *fh* і *fi* відповідно.

Кожен цикл приймає на вхід поточний 512-бітовий блок Y_q, що обробляється в даний момент, і 128-бітове значення буфера ABCD, яке є проміжним значенням дайджесту, і змінює вміст цього буфера. Кожен цикл також

використовує четвертую частина 64-елементної таблиці *T*[1 . 64], побудованої на основі функції sin. *i*-ий елемент *T*, *T*[i], що позначається, має значення, рівне цілій частині від 232 * abs (sin (*i*)), і задане в радіанах. Оскільки abs (sin (*i*)) є числом між 0 і 1, кожен елемент *T* є цілим, яке може бути представлене 32 бітами. Таблиця забезпечує "випадковий" набір 32-бітових значень, які повинні ліквідовувати будь-яку регулярність у вхідних даних.

Для отримання *Mdq*+1 вихід чотирьох циклів складається по модулю 232 з *Mdq*. Складання виконується незалежно для кожного з чотирьох слів в буфері.

5) Вихід Md5. Після обробки всіх *L* 512-бітових блоків виходом *L*-ої стадії є 128-бітовий дайджест повідомлення.

Детальніше логіку кожного з чотирьох циклів виконання одного 512-бітового блоку розглянуто нижче. Кожен цикл складається з 16 кроків, що оперують з буфером *ABCD*.

$$A \quad B + \text{Clss}\,(A + f\,(B, C, D) + X\,[k] + T\,[i]),$$

A, *B*, *C*, *D* – чотири слова буфера; після виконання кожного окремого кроку відбувається циклічне зрушення вліво на одне слово.

f – одна з елементарних функцій *ff*, *fg*, *fh*, *fi*.

CLSs – циклічне зрушення вліво на s біт 32-бітового аргументу.

X [*k*] – *M* [*q* * 16 + *k*] – кодує 32-бітове слово в *q*-ому 512 блоці повідомлення.

T [*i*] – *i*-е 32-бітове слово в матриці T.

+ – складання по модулю 232.

На кожному з чотирьох циклів алгоритму використовується одна з чотирьох елементарних логічних функцій. Кожна елементарна функція отримує три 32-бітові слова на вході і на виході створює одне 32-бітове слово. Кожна функція є безліччю побітових логічних операцій, тобто *n*-ий біт виходу є функцією від n-ого біта трьох входів. Елементарні функції наступні:

$$ff = (B \,\&\, C) \quad (\text{not } B \,\&\, D)$$

$$fg = (B \,\&\, D)\ \text{V}\ (C \,\&\, \text{not } D)$$

$$fh = B \quad C \quad D$$

$$fi = C \quad (B \ \& \ \text{not} \ D)$$

Масив з 32-бітових слів X [0..15] містить значення поточного 512-бітового вхідного блоку, який обробляється зараз. Кожен цикл виконується 16 разів, а оскільки кожен блок вхідного повідомлення обробляється в чотирьох циклах, то кожен блок вхідного повідомлення обробляється по схемі 64 рази. Якщо представити вхідний 512-бітовий блок у вигляді шістнадцяти 32-бітових слів, то кожне вхідне 32-бітове слово використовується чотири рази, по одному разу в кожному циклі, і кожен елемент таблиці T, що складається з 64 32-бітових слів, використовується тільки один раз. Після кожного кроку циклу відбувається циклічне зрушення вліво чотирьох слів A, B, C і D. На кожному кроці змінюється тільки одне з чотирьох слів буфера $ABCD$. Отже, кожне слово буфера змінюється 16 разів, і потім 17-й раз в кінці для отримання остаточного виходу даного блоку.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розроблене програмне забезпечення реалізує систему CAN-мережі на основі технології CSDN.

Програмно-апаратні вимоги:

– Загальний обсяг ОЗП: 512 Мбайт.

– Вільний простір на жорсткому диску: 15 Мбайт.

– Операційна система Microsoft Windows 10/11.

Програма має простий та інтуїтивно зрозумілий інтерфейс, який зображений на рисунку 5.1.

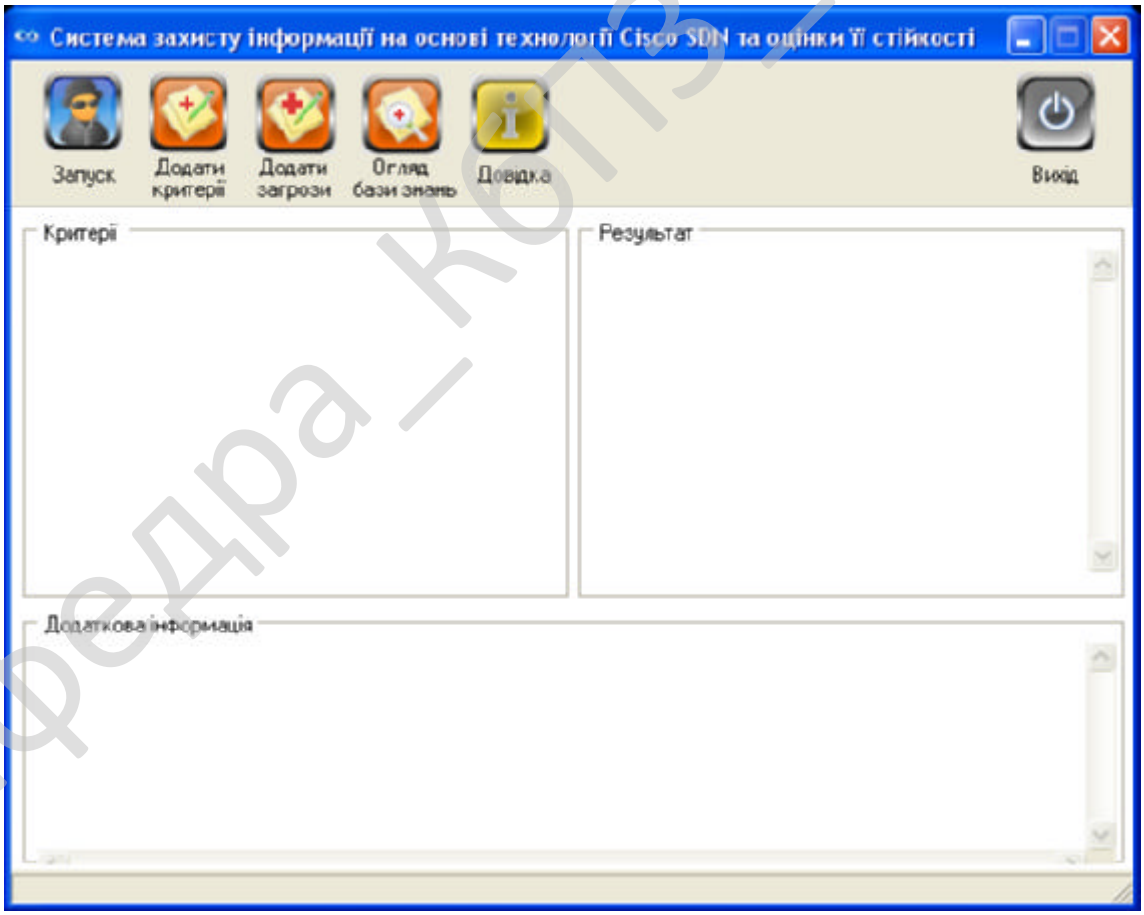

Рисунок 5.1 – Основне вікно програми

Програма складається з таких частин:

– головне вікно (рисунок 5.1);

– додавання критеріїв до бази знань (рисунок 5.2);

– додавання загрози до бази знань (рисунок 5.3);

– огляд бази знань;

– додаткова інформація про програму (рисунок 5.4).

Спершу висвітлюється головна форма програми, після чого відбуваються дії у наступній послідовності:

– переглядається наявність у базі існуючих критеріїв, даних про загрози та даних про детальну інформацію про загрози;

– згідно з вибором однієї з дій, а саме:

а) додавання критеріїв до бази знань;

б) додавання загрози до бази знань;

в) огляд бази знань;

г) додаткова інформація про програму;

відбувається перехід до відповідної процедури програми, яка виконує вищеперераховані дії. При цьому у першому випадку, робиться запит до бази даних критеріїв (krt.db), та робиться запис до відповідних полей бази даних, про новий симптом. У другому випадку до бази даних загроз (zagrzesses.db) відбувається запис найменування нової загрози, а до відповідних полів, записуються критерії, які притаманні даній конкретній загрозі. У третьому випадку, при визначені загрози, по установленим критеріям, експерту з інформаційної безпеки виводиться уся інформація про дану загрозу, тобто, найменування загрози, критерії визначення загрози, процес протидії даній загрозі, та які методи, технології й інструментарій, потрібно використовувати при цій загрозі. У останньому випадку виводиться така інформація як:

– інформація про авторські права;

– інформація про керівника магістерської роботи;

– інформація про навчальний заклад, у якому виконана та відбувається захист магістерської роботи.

На рисунках 5.2, 5.3 приведені деталі інтерфейсу програмного забезпечення виконаної магістерської роботи.

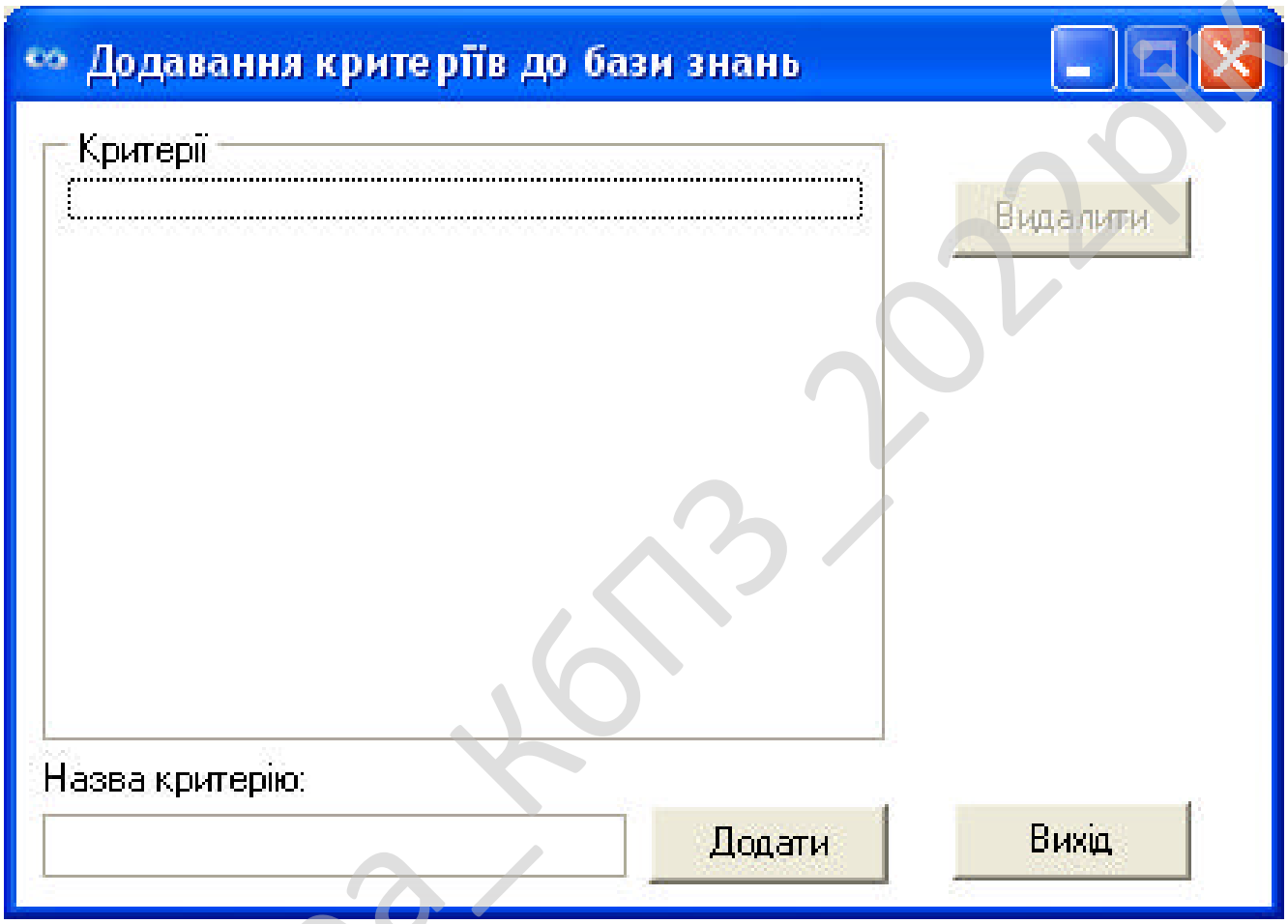

Рисунок 5.2 – Додавання критеріїв до бази знань

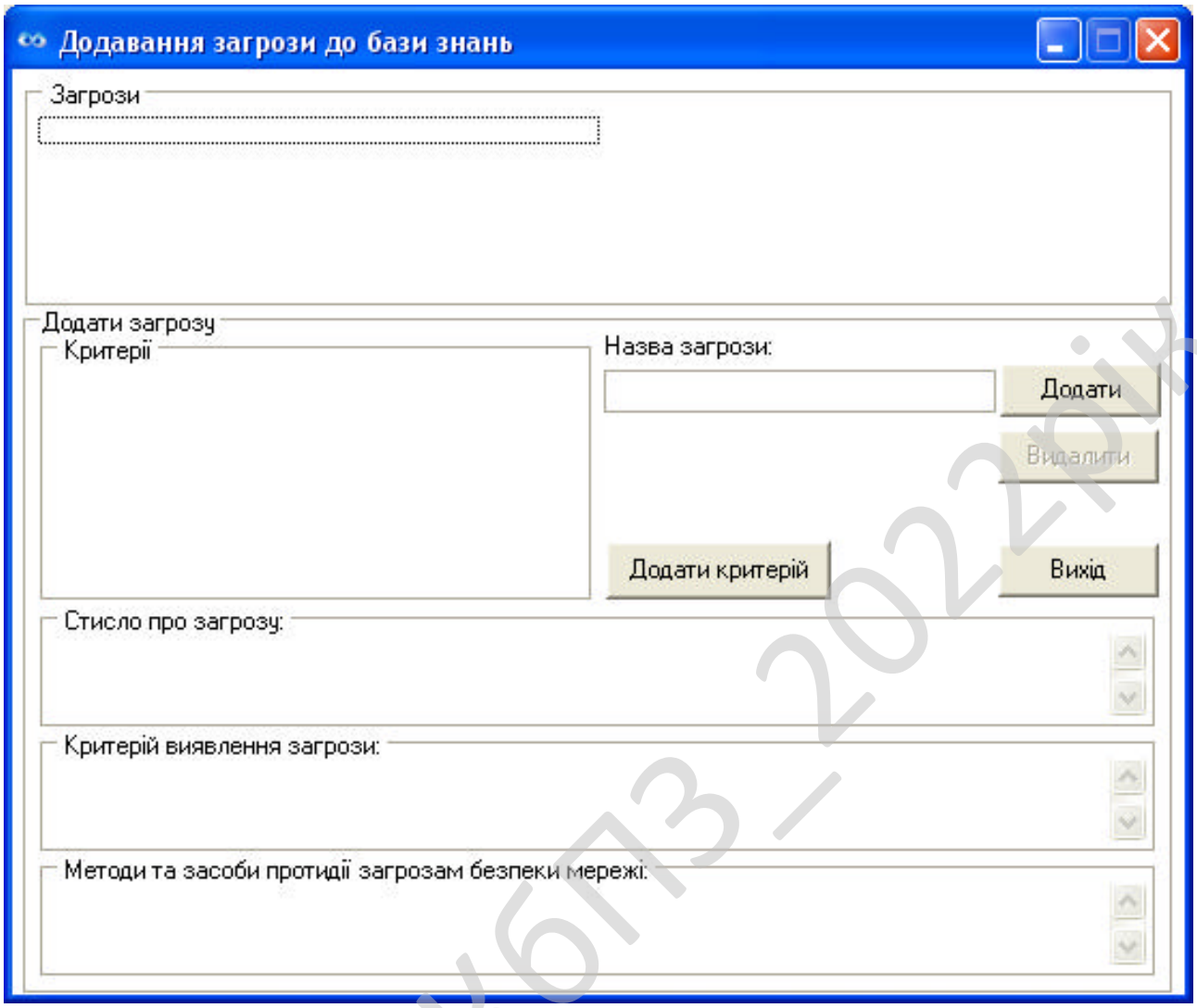

Рисунок 5.3 – Додавання загрози до бази знань

Для перегляду короткої довідки про програму слід натиснути на основному вікні кнопку **«Про програму…»**, після чого на екрані з’явиться вікно показане на рисунку 5.4.

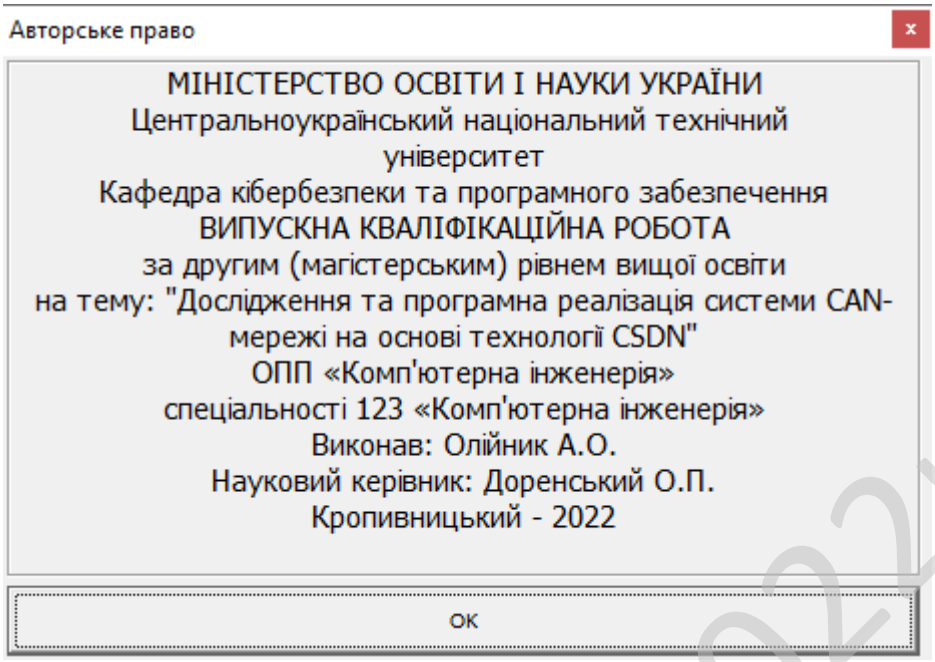

Рисунок 5.4 – Довідка

Інструкція користування програмою:

1. Запуск експертної системи

– Виберіть критерії.

– Натисніть кнопку «Запуск».

2. Додавання даних до бази знань:

– Натисніть кнопку «Добавити загрозу».

– В полі «Назва загрози» введіть назву загрози.

– З списку «Критерії» виберіть необхідні критерії.

– Заповніть додаткові поля: «Стисло про загрозу», «Критерії, протікання загрози», «Методи протидії».

– Натисніть кнопку «Додати».

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи CAN-мережі на основі технології CSDN.

Метою розробки є дослідження та програмна реалізація системи CAN-мережі на основі технології CSDN.

Об'єктом дослідження є процес CAN-мережі на основі технології CSDN.

Предметом дослідження є методи CAN-мережі на основі технології CSDN.

Методи дослідження базуються на методах теорії побудови комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод CAN-мережі на основі технології CSDN.

– Розроблено вітчизняний продукт CAN-мережі на основі технології CSDN, який має більш широкі можливості, на відміну від існуючих аналогів.

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

7.1 Техніко-економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Після ознайомлення з підприємством та засобами розробки програмної продукції був розроблений план розробки програми. Був підрахований необхідний час для розробки та впровадження програми. Цей час склав 60 днів (три місяці).

В магістерській роботі було проведене дослідження та виконана програмна реалізація системи CAN-мережі на основі технології CSDN.

Розроблене програмне забезпечення має достатню надійність і задовольняє усім поставленим умовам, а саме:

а) невеликий розмір;

б) невеликі системні потреби;

в) незалежність від встановлених на комп’ютері баз даних;

г) зручність у користуванні та надійність.

Таблиця 7.1 – Початкові дані

Показники	Позначення	Характеристика або величина
1	2	3
1. Кількість розроблених програм період, шт.	N	1
2. Кількість екземплярів програм, шт.	Ne	280
3. Запланований термін розробки, днів	Fpq	60 (3 місяці)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Б
6. Складність алгоритму (1, 2, 3)	–	2

Продовження таблиці 7.1

1	2	3
7. Кількість макетів вхідної інформації	–	3
8. Кількість форм вихідної інформації.	–	4
9. Мова програмування (1-6)	–	1
10. Попередній досвід (1-6)	–	3
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	2
13. Рівень спрацьованості колективу (1-6)	–	2
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	2
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	2
17. Складність кінцевого програмного продукту (1-6)	–	2
18. Необхідний рівень забезпечення повторного використання (1-6)	–	2
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	2
20. Вимоги до швидкодії ПП (1-6)	–	2
21. Обмеження на розміри основного сховища даних (1-6)	–	2
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	2
23. Професійний рівень аналітиків (1-6)	–	2
24. Професійний рівень програмістів (1-6)	–	2
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	2
27. Досвід роботи з обчислювальною платформою (1-6)	–	2

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	28000
33. Норматив додаткової зарплати, % :	$Н_д$	10
34. Норматив відрахувань у соціальні фонди, %	Нс	22
35. Норматив загальногосподарських витрат, %	Нг	15
36. Норматив витрат на освоєння нових мов програмування, %	Нп	15
37. Рівень рентабельності програмної продукції, %	Ре	50
38. Ставка податку на додану вартість, %	Ндв	20

7.2 Розрахунок трудомісткості розробки програмної продукції

Значення трудомісткості розробки програмного забезпечення для стадій ТЗ, ЕК, ТП та ВП визначаємо по типовим нормам часу приведеним в додатках МВ. Стадія РП є найбільш тривалою і трудомісткою, що робить значний вплив на інші стадії проекту.

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

$$T_{ном} = A \, \text{Size}^B, \qquad (7.1)$$

де: A – коефіцієнт Боема, A = 2,45;

Size – загальний об'єм відлагодженого програмного коду, тис. рядків;

B – показник ступеня, що визначається співвідношенням:

$$B = 1{,}01 + 0{,}001\sum W_i, \qquad (7.2)$$

де: *Wi* – сумарне значення п'яти показників (МВ, додаток 2), що відображають особливості розробки проекту програмного продукту (ПП) і колективу розробників.

$$B = 1{,}01+0{,}001(2{,}43+3{,}64+3{,}38+3{,}95+2{,}73) = 1{,}027.$$

$$T_{ном} = 2{,}45 \cdot 2{,}7^{1{,}026} = 6{,}78 \text{ люд-міс.}$$

Визначаємо уточнені (з урахуванням приведених в МВ додатку 3 сімнадцяти додаткових коефіцієнтів) трудовитрати, люд-міс.:

$$T_{уточн} = T_{ном}\, ПВ_j, \qquad (7.3)$$

де: $ПВ_j$ – добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

$$T_{уточн} = 6{,}78 \cdot (0{,}88 \cdot 0{,}93 \cdot 0{,}88 \cdot 0{,}91 \cdot 0{,}95 \cdot 1 \cdot 1 \cdot 0{,}87 \cdot 1{,}22 \cdot 1{,}16 \cdot 1{,}1 \cdot 1{,}1 \cdot$$
$$\cdot 1{,}12 \cdot 1{,}1 \cdot 1 \cdot 1{,}1 \cdot 1{,}1) = 9{,}37 \text{ люд-міс.}$$

Ці коефіцієнти дозволяють диференційовано оцінювати результати роботи програмістів, беручи до уваги швидкодію програми, використання різноманітних обчислювальних платформ і інструментів розробки, взаємодію декількох серверів, вимоги до об'ємів баз даних і ін.

Визначаємо підсумкові трудовитрати по стадії робочий проект, люд-дні:

$$T_{РП} = 0{,}3 C T_{уточн}^{0{,}33+0{,}2(B-1{,}01)} S, \qquad (7.4)$$

де: *C* – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4);

S – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПЗ згідно встановленим вимогам. Вибираємо в межах (25…350)%.

$$T_{РП} = 0{,}3 \cdot 3{,}23 \cdot 9{,}37^{0{,}33+0{,}2(1{,}026-1{,}01)} \cdot 60 = 123 \text{ люд/день.}$$

Для зручності визначення загальної трудомісткості на розробку програмного забезпечення результати розрахунків по стадіям зводимо до таблиці 7.2.

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

Стадії розробки	Трудомісткість за типовими нормами та розрахунками	
	Величина, люд/дні	Підстава
Технічне завдання	9	Д5
Ескізний проект	10	Д6
Технічний проект	9	Д7
Робочий проект	123	Ф 7.1-7.4
Впровадження	13	Д13
Всього	164	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

Чисельність ставок інженерів-програмістів для розробки програмного забезпечення визначається за формулою:

$$Ч = \frac{T_{пз} N}{F_{рq} - H_{ев}}, \tag{7.5}$$

де: $F_{рq}$ – плановий фонд робочого часу одного спеціаліста, днів;

$T_{пз}$ – трудомісткість розробки програмного забезпечення люд-дні.

$$Ч = \frac{164 \cdot 1}{60 - 5} = 3 \text{ ставки.}$$

Чисельність інженерів-електронщиків для проведення технічного обслуговування та ремонту комп'ютерних мереж визначається в залежності від наявності технічних засобів і норм витрат часу на виконання профілактичних робіт на протязі року.

Визначаємо затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за період розробки. Результати розрахунку зводимо до таблиці 7.3.

Таблиця 7.3 – Затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за розрахунковий період

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на один. обл.	Кількість облад-нання	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	90	7	630	10,5
Монітор	60	7	420	7
Клавіатура	30	7	210	3,5
Маніпулятор «мишка»	30	7	210	3,5
Принтер матричний	60	0	0	0,0
Принтер лазерний	120	1	120	2
Принтер струминний	60	1	60	1
Сканер	20	1	20	0,33
Концентратор-маршрутизатор	30	1	30	0,5
Кабельні господарства ЛОМ на 1 м.п.	2,5	250	625	10,42
Копіювальний апарат	140	1	140	2,33
Усього за рік:			$З_ч$	41,08

Час на профілактику обладнання в загальному балансі робочого часу інженерів-електронщиків неповинен складати більше 10%.

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

$$Ф_{др}^{с} = \frac{З_ч \cdot n_{тіс}}{1,2}, \quad (7.6)$$

$$Ф_{др}^{с} = \frac{41 \cdot 3}{1,2} = 102,5 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

$$Ч_{ел} = \frac{Ф_{др}^{с}}{F_{др} \cdot T_{зм}}, \quad (7.7)$$

$$Ч_{ел} = 102,5/(60 \cdot 8) = 0,2 \text{ ставки.}$$

Для забезпечення нормального технічного обслуговування засобів ТО та мереж, необхідно прийняти найбільше ціле значення розрахункової чисельності інженерів-електронщиків.

Таблиця 7.4 – Розрахунок чисельності штатного персоналу сектору системного та адміністративного обслуговування засобів ОТ та комп'ютерних мереж

Посада	Вид роботи	Час	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (ОС FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 2019, серверу доступу ADSL (ОС Linux), налаштування ADSL, VPN PPPoE, Frame Relay, Wi-Fi	2	0,5
	Налаштування і конфігурування базової станції безпровідного зв'язку (CMTS)	0,5	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,5	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	1	
Всього		4	

Продовження таблиці 7.4

Посада	Вид роботи	Час	К-ть штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	1	0,25
	Підтримка постійних клієнтів	0,5	
	Оформлення договорів, ведення тендерів	0,25	
	Контроль взаєморозрахунків з постачальниками	0,25	
Всього		2	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	1	0,25
	Створення графічних і стилістичних елементів сайту	0,5	
	Розміщення графіки і контенту на Інтернет сторінках	0,5	
Всього		2	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	1	0,25
	Верстка друкованих видань	0,5	
	Додрукова підготовка макетів	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	

Чисельність інженерів-системотехніків, адміністраторів мережі, дизайнерів WEB вузлів, системних програмістів (аналітиків), бухгалтерів-економістів визначається за потребою в залежності від функціональних обов'язків. Після визначення чисельності персоналу складається штатний розклад.

Складемо штатний розклад виконавців.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	11448	34344
Продакт-менеджер	0,25	8000	6000
Інженер-програміст	3	11500	103500
Інженер-електронщик	0,2	8000	4800
Інженер-системотехнік	0,25	8000	6000
Адміністратор мережі	0,5	8000	12000
Системний програміст	0,25	8000	6000
Дизайнер WEB	0,25	8000	6000
Інженер-верстальник	0,25	8000	6000
Бухгалтер-економіст	0,5	9000	13500
Всього за період розробки	$R_{сп}$ = 6,45	-	$Ф_{роб}$ = 198144

Розрахуємо середньоденну зарплату одного виконавця:

$$З_{сд} = \frac{Ф_{роб}}{R_{сп} F_{рq}}, \quad (7.8)$$

де: $Ф_{роб}$ – загальна сума зарплати за плановий період, грн.

$$З_{сд} = \frac{198144}{6,45 \cdot 60} = 512 \; грн.$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

Балансова вартість будівель визначається з урахуванням кількості робочих місць виконавців, питомої площі на одне робоче місце, та вартості одного квадратного метра виробничої площі:

$$Б_{уд} = R^1_{сп} S_у Ц_{пл}, \quad (7.9)$$

де: $R^1_{сп}$ – кількість робочих місць виконавців, шт. Приймаємо 8 робочих місць;

$S_у$ – питома площа на одне робоче місце, $м^2$;

$Ц_{пл}$ – вартість одного квадратного метра площі, грн.

Згідно даних інтернет ресурсу DOM.RIA (https://dom.ria.com) ціна одного квадратного метра площі, вік якої не перевищує 30 років, по місту складає 500…1600 $у.о./м^2$. Враховуючи, що курс складає 1 у.о. = 38 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./м^2. На кожне робоче місце у середньому потрібно 8 $м^2$. З урахуванням цього:

$$Б_{уд} = 8 \cdot 8 \cdot 20000 = 1280000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 128000 грн.

Балансова вартість інвентарю розраховується за нормою 3500 грн. на одне робоче місце. Тобто:

$$I_{нв} = R^1_{сп} \cdot Ц_м, \qquad (7.10)$$

де: $Ц_м$ – ціна меблів для одного робочого місця, грн.

$$I_{нв} = 8 \cdot 3500 = 28000 \text{ грн.}$$

Балансова вартість обчислювальної техніки визначається по оптовим цінам постачальника з врахуванням витрат на транспортування.

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались по прайсу фірми Компбест за 06.11.22 – джерело https://compbest.com.ua/.

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп’ютер		10947
Системний блок		7347
Процесор	Intel Core i5-4590 (4 ядра по 3.3-3.7GHz), 6 MB cache	-
Системна плата	MSI H81M-P33 6 x USB 2.0, 2 x USB 3.0, VGA, DVI, 2 x PS/2, LAN (RJ-45), 5 x Audio, FireWire	-
Відеокарта	nVidia GeForce GTX 660, 2 GB GDDR5, 192-bit	-
Жорсткий диск	240 GB SSD	-
Оперативна пам’ять	Samsung DDR3-1333 8Gb PC3-10600R ECC Registered (M393B1K70CH0-CH9Q5)	-
DVD-привод	DVDRW Pioneer DVR-TD10RS SATA Slim Black Bulk (DVR-TD10RS)	-
Корпус	ATX Middle Tower FOXCONN Pro, 3GTLA-489, PSU 350W(FSP Brand: ATX-350PNR, 12cm), black, (front bezel – black+light silver; body material – 0.6mm), 80mm fan (rear), 2xUSB2.0/AUDIO/MIC, Air Duct, Tool-less chassis design,Thermally Advantaged Chassis	-

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Кулер	–	–
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4-E int. 3.5", 1*USB3.0+AUDIO+1394, multi: A Type Cards, black	220
інше	Клавіатура, мишка	Подару-нок
Монітор	22" TFT, ASUS VW223D (5ms, 300/3000: 1 170/160, D-SUB, Wide)	3600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USE cable	5500
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробовування.	Загальна вартість, грн.
Персональні комп'ютери	15	10947	16420,5	180625,5
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Сканери	-	-	-	0
Копіюв. апарат	1	5965	596,5	6561,5
Всього	–	–	–	199177

Витрати на транспорт, монтаж та випробування можуть бути прийняті в межах до 10% від оптової ціни.

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	1280000	-	-
2. Передавальні пристрої	128000	-	-
Всього по групі	1408000	5	70400
Група 4			
3. Обчислювальна техніка	199177	-	-
Всього по групі	199177	50	99588,5
Група 5, 6			
4. Вимірювальні пристрої	5190	25	1297,5
5. Транспортні засоби	0	20	0,0
6. Господарський інвентар	28000	25	7000
Всього по групі	33190	-	8297,5
7. Нематеріальні активи	120000	10	12000
Разом	$К_р$ = 1760367		$А_р$ = 190286

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

$$З_о = \frac{З_{сд} \cdot Т_{пз}}{N_е}, \quad (7.11)$$

де: $N_е$ – кількість екземплярів програм, шт.

$$З_о = 512 \cdot 164/280 = 300 \text{ грн.}$$

Визначимо додаткову зарплату (оплата відпусток, виконання державних та суспільних обов'язків) на рівні 10%:

$$З_д = З_о \cdot Н_q \cdot 0{,}01, \quad (7.12)$$

де: $Н_q$ – норматив додаткової зарплати, %.

$$З_д = 300 \cdot 10 \cdot 0{,}01 = 30 \text{ грн.}$$

Відрахування на соціальні потреби за нормативом $Н_с = 22\%$ від суми основної та додаткової зарплати:

$$С_{оц} = 0{,}01 \cdot Н_с (З_о + З_д), \quad (7.13)$$

де: $Н_с$ – відрахування на соціальні потреби, %.

$$С_{оц} = 0{,}01 \cdot 22(300+30) = 73 \text{ грн.}$$

Визначимо загальногосподарські витрати (електроенергію, ремонт і утримання приміщень і т.д) за нормативом $Н_г = 15\%$ від основної зарплати:

$$Г_{осп} = З_о \cdot Н_г \cdot 0{,}01, \quad (7.14)$$

де: $Н_г$ – загальногосподарські витрати, %.

$$Г_{осп} = 300 \cdot 15 \cdot 0{,}01 = 45 \text{ грн.}$$

Визначимо витрати на матеріали для розробки програмної продукції за нормами споживання та діючими цінами за одиницю виміру:

$$З_М = (З_{М1} + З_{М2} + З_{М3})/N_е, \quad (7.15)$$

де: $З_{М1}$ – вартість паперу, грн.; $З_{М2}$ – вартість запам'ятовуючих пристроїв, грн.; $З_{М3}$ – вартість фарби, картріджей, тонеру, грн.; $N_е$ – кількість екземплярів програм, шт.

Згідно прийнятих норм на підприємстві $n_{вит}$ приймаємо 0,5 пачки паперу на період розробки. Тоді, враховуючи, що вартість пачки паперу складає $Ц_п$=210 грн., визначаємо вартість паперу за період розробки:

$$З_{М1} = Ц_п \cdot N_м. \quad (7.16)$$

$$З_{М1} = 210 \cdot 0,5 = 105 \text{ грн.}$$

Згідно прийнятих норм по комплектації до вартості запам'ятовуючих пристроїв входить вартість CD/DVD дисків. Їх кількість дорівнює кількості коробочних версій запропонованого продукту (приймаємо 150):

$$З_{М2} = \sum Ц_д \,, \quad (7.17)$$

де: $Ц_д$ – вартість дисків CD/DVD: CDR box – 22,4 грн./шт., DVD-R box – 35 грн./шт.

$$З_{М2} = 150 \cdot 22,4 = 3360 \text{ грн.}$$

Згідно норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

$$З_{М3} = \sum Ц_з., \quad (7.18)$$

де: $Ц_з$ – вартість розхідних матеріалів друкуючих пристроїв: відновлення та заправка картриджу для Canon i-SENSYS LBP6030W – 574 грн.; картридж для Epson Stylus Photo P50 – 558 грн.; відновлення картриджу для MF217W – 570 грн.

$$З_{М3} = 574+558+570 = 1702 \text{ грн.}$$

$$З_М = (105 + 3360 + 1702)/280 = 18 \text{ грн.}$$

Визначимо витрати на освоєння нових мов програмування або операційних систем за нормативом ($Н_п$ = 15%) від основної зарплати виконавців:

$$О_п = З_о \cdot Н_п \cdot 0,01, \quad (7.19)$$

де: $Н_п$ – норматив витрат на освоєння нових мов програмування, %.

$$О_п = 300 \cdot 15 \cdot 0,01 = 45 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_е$ = 280 прим.):

$$A_м = \frac{A_р \cdot N_{міс}}{N_е \cdot 12}, \quad (7.20)$$

де: $A_р$ – загальна річна сума амортизаційних відрахувань, грн.

$$A_м = 190286 \cdot 3/(280 \cdot 12) = 170 \text{ грн.}$$

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції:

$$C_п = З_о + З_д + С_{оц} + Г_{осп} + З_м + О_п + А_м. \quad (7.21)$$

$$C_п = 300+30+73+45+18+45+170 = 681 \text{ грн.}$$

Величини ціна підприємства, податок на додану вартість, відпускна ціна програмної продукції визначаються за формулами, приведеними в таблиці 7.9

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн
1. Основна зарплата виконавців	$З_о$	300
2. Додаткова зарплата виконавців	$З_д$	30
3. Відрахування на соціальні потреби	$С_{оц}$	73
4. Загальногосподарські витрати	$Г_{осп}$	45
5. Витрати на матеріали	$З_М$	18
6. Освоєння нових операційних систем, мов програмування	$О_п$	45
7. Амортизація основних фондів	$А_м$	170
8. Повна собівартість програмного забезпечення	$С_п$	681
9. Плановий прибуток	$П_р$	341
10.Ціна підприємства $Ц_п = С_п + П_р$	$Ц_п$	1022
11.Податок на додану вартість $ПДВ = 0.01 \cdot Н_{дв} \cdot Ц_п$	$ПДВ$	204,4
12.Відпускна ціна програмної продукції $Ц = Ц_п + ПДВ$	$Ц$	1226,4

Визначимо плановий прибуток за рівнем рентабельності (P_n) програмної продукції, яка залежить від складності програми та ступеня новизни задачі.

Для даного програмного забезпечення рівень рентабельності складає 50%.

$$П_р = 0{,}01 \cdot P_n \cdot C_n, \qquad (7.22)$$

де: P_n – рівень рентабельності, %.

$$П_р = 0{,}01 \cdot 50 \cdot 681 = 341 \text{ грн.}$$

7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції

Об'єм капітальних вкладень у споживача програмної продукції визначаємо на основі балансової вартості основних фондів, яка враховує ціну, транспортно-заготівельні витрати, вартість будівель, монтажних та пусконалагоджувальних робіт, а також витрати на випробування у виробничих умовах. Результати розрахунків зводимо у таблицю 7.10.

Таблиця 7.10 – Розрахунок об'єму капітальних вкладень у споживача програмної продукції

Найменування капітальних вкладень	Сума за варіантами, грн.	
	Базовий	Новий
Вартість програмної продукції	–	1226
Всього капітальних витрат	–	1226

7.7 Визначення експлуатаційних витрат

Експлуатаційні витрати у споживача програмної продукції визначаємо при умові роботи підсистеми на протязі року. Результати зводимо до таблиці 7.11.

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на обслуговування системи	$З_р$	20290	5032
2. Витрати на електроенергію	$З_{ел}$	1488	1030
3. Витрати на амортизацію	$З_{ам}$	0	307
Всього витрат за рік	$І$	21778	6369

Витрати на профілактичні роботи:

$$З_р = Т_р \cdot З_г \cdot (1 + 0{,}01 \cdot Н_q) \cdot (1 + 0{,}01 \cdot Н_с), \qquad (7.23)$$

де: $Т_р$ – кількість годин обслуговування кожного комп'ютера за рік, *год.*;

$З_г$ – заробітна плата обслуговуючого персоналу, *грн / год*.

Після купівлі нового програмного забезпечення витрати на обслуговування системи зменшились з 20290 грн до 5032 грн на рік.

Витрати по амортизації визначаються на основі норм амортизаційних відрахувань, вартості програмної продукції і основних фондів. Для розрахунку складаємо таблицю 7.12.

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	25	–	1226	–	306,5
Всього відрахувань	-	–	1226	–	306,5

Витрати на електроенергію визначаються з урахуванням споживаємої потужності ($П_{ел}$) в кіловатах, часу експлуатації технічних засобів ($T_р$) в годинах та ціни однієї кіловат-години ($Ц_{ел}$):

$$З_{ел} = П_{ел} \cdot T_р \cdot Ц_{ел}. \quad (7.24)$$

$$З_{ел\ баз} = 0{,}545 \cdot 1300 \cdot 2{,}1 = 1487{,}85 \text{ грн.}$$

$$З_{ел\ нов} = 0{,}545 \cdot 900 \cdot 2{,}1 = 1030{,}05 \text{ грн.}$$

7.8 Визначення економічної ефективності програмної продукції

Економічна ефективність програмного забезпечення визначається для виготовлювача і споживача за такими показниками.

Величина економічного ефекту при виготовленні програмної продукції, розраховуємо за формулою:

$$E_в = (Ц_п - C_п) \cdot N_е - \sum_{i=1}^{m} E_{p_m} \cdot K_{p_m}, \quad (7.25)$$

де: $K_р$ – балансова вартість основних фондів розробника, грн.; $E_р$ – розрахунковий коефіцієнт капіталовкладень.

$$Ев = (1022 - 681) \cdot 280 - (0{,}05 \cdot 1408000 + 0{,}5 \cdot 199177 +$$
$$+0{,}25 \cdot 33190 + 0{,}1 \cdot 28000) \cdot 3/12 = 50208 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

$$T_в = \frac{K_р}{(Ц_п - C_п) \cdot N_е}, \quad (7.26)$$

де: $K_р$ – балансова вартість основних фондів розробника.

$$T_в = \frac{1760367}{(1022 - 681) \cdot 280 \cdot 12 / 3} = 4{,}5 \text{ роки}$$

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

$$E_{сп} = (I_б - I_н) - E_н(K_н - K_б), \quad (7.27)$$

де: $I_б$, $I_н$ – величина експлуатаційних витрат за базовим и новим варіантом відповідно;

$K_б$, $K_н$ – об'єм капітальних вкладень за варіантами, що порівнюються.

$$E_{сп} = (21778\text{-}6369)\text{-}0{,}25 \cdot 1226 = 15003 \text{ грн.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	280
2. Повна собівартість розробленої програми	Грн.	681
3. Ціна розробленої програми	Грн.	1022
4. Плановий прибуток від реалізації розробленої програми	Грн.	341
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1760367
7. Загальний прибуток від реалізації програмної продукції	Грн.	95480
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	50208
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Роки	4,5
10.Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	1226
11.Величина економічного ефекту у користувача програмної продукції	Грн.	15003
12.Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,1

Визначимо період окупності додаткових капітальних вкладень у споживача програмної продукції за рахунок зниження експлуатаційних витрат:

$$T_{сп} = \frac{K_н - K_б}{I_б - I_н}, \tag{7.28}$$

$$T_{сп} = \frac{1226}{21778 - 6369} = 0{,}1 \; року .$$

7.9 Висновки

Розроблена програма економічно вигідна. За рахунок впровадження програмного забезпечення досягається скорочення часу обробки інформації, підвищується культура праці, підвищення якості приймаючих управлінських рішень.

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Електронно-обчислювальна машина (ЕОМ) відіграє важливу роль у житті сучасної людини. Кожного дня мільйони людей використовують ЕОМ для пошуку необхідної інформації, спілкуванні у соціальних мережах, перегляду новин, роботи тощо. Багато людей користуються ЕОМ у професійних цілях, оскільки завдяки ЕОМ з'явилося багато нових професій.

Аналізуючи умови працівників it-сфери, на перший погляд, може здатися, що працівники сфери інформаційних технологій не схильні до ризиків на виробництві, та якщо більш глибоко розглянути умови і специфіку праці фахівців сфері it-індустрії, можна виявити ряд факторів які будуть мати негативний сплив на стан охорони праці, та на самого it-фахівця зокрема. Сюди можна віднести як невідповідність освітлення, так і високий рівень шуму, що негативно позначатимуться як на емоційному так і на фізичному стані фахівця, призводитимуть до зниження ефективності праці та виробничих травм. Також, важливим моментом охорони праці it-фахівця є врахування його психологічних можливостей (швидкість реакції, особливості пам'яті та уваги, емоційний стан, тощо). Для того, щоб забезпечити ефективну роботу it-фахівця, потрібно враховувати та максимально компенсувати такі негативні фактори як: надмірне нервово-емоційне навантаження, довготривалі статичні перевантаження, обмежена рухова активність. Всі ці чинники призводить до різноманітних відхилень у стані здоров'я, зокрема до перевтоми, зниження фізичної та розумової працездатності, неврозів, захворювань серцево-судинної системи тощо. Метою даного розділу є огляд конкретних умов праці спеціаліста у сфері it-індустрії. Завданнями для даного розділу є: аналіз умов праці на робочому місці фахівця it-індустрії, розробка конкретних рекомендацій щодо покращення умов

праці фахівців it-індустрії, огляд пожежної безпеки на it-підприємстві та розрахунок системи загального штучного освітлення виробничого приміщення де працюють ІТ-фахівці.

8.2 Аналіз умов праці на робочому місці ІТ-фахівця

На робочому місці ІТ-фахівця (або програміста) виникають небезпечні та шкідливі для безпечної життєдіяльності фактори:

– підвищений рівень шуму;

– несприятливі мікрокліматичні умови;

– недостатній рівень освітленості;

– шкідливі речовини;

– підвищений рівень електромагнітних випромінювань радіочастот;

– висока напруга електричної мережі;

– статична електрика та інші.

Робота програміста супроводжується також підвищеним ступенем напруженості трудового процесу. При систематичному впливі виробничих факторів, які не відповідають нормативним показникам, зростає рівень професійно зумовленої захворюваності працюючих та можуть виникнути професійні захворювання органів зору, руху, нервової системи. Таким чином, вивчення умов праці на робочому місці програміста є необхідною умовою запобігання негативних наслідків впливу небезпечних та шкідливих факторів. Робоче місце, добре пристосоване до трудової діяльності інженера, правильно і доцільно організоване, щодо простору, форми, розміру забезпечує йому зручне положення при роботі і високу продуктивність праці при найменшому фізичному і психічному напруженні.

Нормування параметрів проводиться в залежності від періоду року та категорії важкості виконуваних робіт. Для постійних робочих місць, якими є робочі місця ІТ-фахівців, встановлені оптимальні параметри мікроклімату, а за

неможливості їх дотримання використовують допустимі параметри. Робота ІТ-фахівця за важкістю відноситься до Іа (роботи, що виконуються сидячи і не потребують фізичного напруження) та Іб (роботи, що виконуються сидячи, стоячи або пов'язані з ходінням та супроводжуються деяким фізичним напруженням) категорій. В таблиці 8.1. наведені оптимальні параметри мікроклімату в приміщеннях.

Таблиця 8.1 – Параметри мікроклімату для приміщень з ПК

Період року	Параметр мікроклімату	Величина
Холодний	Температура повітря в приміщенні; відносна вологість; швидкість руху повітря	22...24°С; 40... 60%; до 0,1 м/с
Теплий	Температура повітря в приміщенні; відносна вологість; швидкість руху повітря	23...25 °С 40...60% 0,1...0,2 м/с

Виміряні за допомогою приладів температура та вологість у приміщеннях праці ІТ-фахівців повинні відповідати зазначеним у таблиці для теплого періоду року. Слід зазначити, що для нормалізації параметрів мікроклімату слід використовувати у приміщеннях кондиціювання повітря, або забезпечити подачу свіжого повітря системами вентиляції. Норми подачі свіжого повітря наведені у таблиці 8.2.

Таблиця 8.2 – Норми подачі свіжого повітря в приміщення

Характеристика приміщення	Об'ємна витрата свіжого повітря, що подається в приміщення, $м^3$ на одну людину в годину
Об'єм до 20 $м^3$ на людину 20... 40 $м^3$ на людину Більше 40 $м^3$на людину	Не менше 30 Не менше 20 Може біти використана природна вентиляція

Створення сприятливих умов праці і правильне естетичне оформлення робочих місць на виробництві має велике значення як для полегшення праці, так і для підвищення його привабливості, позитивно впливає на продуктивність праці. Забарвлення приміщень і меблів повинні сприяти створенню сприятливих умов для зорового сприйняття, гарного настрою. У службових приміщеннях, у яких виконується одноманітна розумова робота, що вимагає значної нервової напруги і великого зосередження, забарвлення повинно бути спокійних тонів – малонасичені відтінки холодного зеленого або блакитного кольорів.

При розробці оптимальних умов праці програміста необхідно враховувати освітленість. Раціональне освітлення робочого місця є одним з найважливіших факторів, що впливають на ефективність трудової діяльності людини, що попереджають травматизм і професійні захворювання. Правильно організоване освітлення створює сприятливі умови праці, підвищує працездатність і продуктивність праці. Освітлення на робочому місці програміста повинно бути таким, щоб працівник міг без напруги зору виконувати свою роботу. Стомлюваність органів зору залежить від ряду причин: недостатність освітленості; надмірна освітленість; неправильний напрям світла. Недостатність освітлення приводить до напруги зору, ослабляє увагу, приводить до настання передчасної стомленості. Надмірно яскраве освітлення викликає засліплення, роздратування і різь в очах. Неправильний напрямок світла на робочому місці

може створювати різкі тіні, відблиски, дезорієнтувати працюючого. Всі ці причини можуть призвести до нещасного випадку або профзахворювань. [2]

8.3 Пропозиції щодо підвищення працездатності ІТ-фахівців

Поява та впровадження нових інформаційно-комунікаційних технологій зумовлює необхідність подальшого вдосконалення охорони праці фахівців it-індустрії. Все це потребує розробки нових нормативно-правових актів з регламентації праці та відпочинку фахівців it-індустрії і стандартів підприємств, центрів комп'ютерної техніки, центрів інформаційних технологій, сучасних комп'ютерних класів. Для підвищення розумової працездатності то зорової роботи повинна здійснюватися ергономічна оптимізація в рамках системи «оператор-термінал», яка сприятиме результативній фізичній та інтелектуальний працездатності і відновленню психосоматичного здоров'я фахівців it-індустрії. Всі наведені заходи щодо вдосконалення охорони праці фахівців it-індустрії повинні контролюватися службою охорони праці та комісією з охорони праці підприємства. Особливе значення у соціальному захисті цієї категорії працівників належить прийняття комплексного договору, який може забезпечити фахівців додатковими пільгами та компенсаціями.

Для більшого розуміння, пропозиції щодо підвищення працездатності it-фахівців, розіб'ємо на декілька категорій:

1 Середовище і розпорядок праці. Для мінімізації негативних ефектів, що пов'язані з перевтомленням it-фахівців, потрібно чітко прописати і реалізувати графік періодів праці-відпочинку, щоб фахівець міг можливість переключити увагу, дати можливість відпочити очам, мозку, елементарно, встати розім'яти ноги. Також потрібно зробити максимально комфортними умови мікроклімату у офісному приміщенні, де працюють it-фахівці. Мається на увазі встановлення і експлуатація, коли виникає необхідність, кондиціонерів, опалення, та системи вентиляції, задля попередження перегрівання, переохолодження it-фахівців, і

подальшої неможливості ними виконувати свої функції. Також, за можливості, нами пропонується введення практики віддаленої праці it-фахівцями, якщо роботодавець не може забезпечити оптимальні і безпечні умови в офісному приміщенні, або якщо фахівця вони не влаштовують із певних причин.

2 Фізичні і психоемоційні чинники. Першим і найважливішим чинником, що впливає на працездатність it-фахівців є робоче місце, і саме тому, роботодавець має забезпечити максимальний його комфорт і безпеку. Гарантією цих факторів може слугувати сертифікація меблів, що використовуються на підприємстві it-галузі. Тому нами пропонується закупівля тільки меблів , які пошли сертифікацію на відповідність. Під психоемоційними чинниками ми розуміємо гарне самопочуття фахівців, позитивний настій, гарний психологічний клімат у колективі, тощо. Задля того, щоб психоемоційні чинники мали максимально позитивний ефект, керівництву слід поводити заходи, які сприятимуть укріпленню і покращенню міжособистісних стосунків у колективі, таких як психологічні тренінги, тімбілдінг, спортивні змагання і естафети. Також, сюди можна віднести розробку і впровадження системи мотивації працівників, як фінансової, так моральної і адміністративної.

8.4 Розрахунок системи загального штучного освітлення виробничого приміщення де працюють ІТ-фахівці

Приміщення з ПК повинні мати природне і штучне освітлення, яке відповідало б вимогам ДБН В.2.5-28-2006 «Природне і штучне освітлення» [1], ДСАНПІН 3.3.2.007-98 «Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [2]. Приміщення для роботи із ПЕОМ повинні мати природне й штучне освітлення. Віконні прорізи повинні бути орієнтовані на північ або на північний схід, забезпечувати коефіцієнт природної освітленості (К.П.О.) не менш 1,5% і мати жалюзі або штори. Віконні прорізи повинні мати регульовані пристрої для відкривання, а

також жалюзі, завіски, зовнішні козирки тощо. Приміщення із ПЕОМ повинні бути обладнані системою загального рівномірного освітлення. У виробничих і адміністративно-суспільних 130 приміщеннях, де переважно ведеться робота з документами, допускається комбінована система штучного освітлення. Штучне освітлення має здійснюватися системою загального рівномірного освітлення, яка включає суцільні або такі, що перериваються лінії світильників, розташованих збоку робочих місць (переважно ліворуч), паралельно лінії зору користувачів ПК. Світильники повинні мати розсіювачі світла. У світильниках місцевого освітлення можна використовувати лампи накалювання. При розміщенні ПК по периметру приміщення лінії світильників штучного освітлення повинні розміщуватися локально над робочими місцями. Система освітлення робочого місця користувача ПК має відповідати наступним вимогам (рис. 8.1).

Рисунок 8.1 – Вимоги до системи освітлення робочого місця користувача ПК

Освітленість на робочому столі користувача в зоні розташування документів має бути в межах 300-500 лк. Якщо цей рівень освітленості неможливо забезпечити системою загального освітлення то допускається застосування світильників місцевого освітлення, але при цьому не повинно бути відблисків на поверхні екрану (яскравість відблисків не повинна перевищувати 40 кд/м^2) та перевищення його освітленості більше ніж 300 лк. Яскравість світильників загального освітлення, а також яскравість стелі при застосуванні системи відбитого освітлення не повинна перевищувати 200 кд/м^2. Величина коефіцієнта пульсації освітленості не повинна перевищувати 5%. Що стосується розподілу яскравості в полі зору працюючих за дисплеями ПК, то відношення значень яскравості робочих поверхонь не повинно перевищувати 3:1

Проведемо розрахунок штучного освітлення за методом коефіцієнта використання світлового потоку для приміщення ширина якого складає 6 м, довжина – 7 м, висота – 2,9 м. У зазначеному приміщенні працює 7 людей.

Для того, щоб визначити потрібну кількість світильників, які повинні забезпечити нормований рівень освітленості, визначимо світловий потік, що падає на робочу поверхню за формулою [1]:

$$F=ESKZ/n,$$

де:

F – світловий потік, що розраховується, Лм;

E – нормована мінімальна освітленість, Лк; *E = 300 Лк;*

S – площа освітлюваного приміщення (у нашому випадку *S*=6 х 6,8 = 40,8 *м*^{2});

K – коефіцієнт запасу, що враховує зменшення світлового потоку лампи в результаті забруднення світильників в процесі експлуатації (його значення залежить від типу приміщення і характеру робіт, що проводяться в ньому, в нашому випадку *K = 1,5);*

Z – відношення середньої освітленості до мінімальної (зазвичай приймається рівним 1.1... 1.2, в нашому випадку *Z = 1,1*);

n – коефіцієнт використання світлового потоку, (відношення світлового потоку, що падає на розрахункову поверхню, до сумарного потоку всіх ламп і обчислюється в долях одиниці [7]; залежить від характеристик світильника, розмірів приміщення, забарвлення стін і стелі, що характеризуються коефіцієнтами відбиття від стін **($\rho_{стін.}$)** і стелі **($\rho_{стелі}$),** значення коефіцієнтів дорівнюють $\rho_{стін}$ **= 50% і** $\rho_{стелі}$ **= 50%.**

Обчислимо індекс приміщення за формулою:

$$i=S/(h(A+B)),$$

де:

S – площа приміщення, *S* = 40,8 $м^2$;

h – розрахункова висота підвісу, *h = 3* м *(співпадає з висотою стелі, т.я. лампи освітлення закріплюються на стелі);*

A – ширина приміщення, *A* = 6 м;

B – довжина приміщення, *B = 6,8* м.

Підставимо всі значення у формулу та визначимо індекса приміщення:

$$i=1,1.$$

Знаючи індекс приміщення, за знаходимо *n = 0,46 (з табличних даних коефіцієнтів використання світлового потоку (n) світильників з відповідним типом лампам) [7].* Підставимо всі значення у формулу, визначемо світловий потік: *F=43904 Лм.*

Для розрахунку будемо використовувати *світлодіодні стельові панелі Lezard 6400K 72 Вт.*, світловий потік яких $F_л$ *= 7200* Лм.

Число ламп визначається по формулі:

$$N=F/F_л$$

де:

F – світловий потік,

$F_л$ – світловий потік однієї лампи.

Підставимо всі значення у формулу та визначимо індекса приміщення:

$$N= 43904 / 7200=6,09 шт.$$

Приймаємо необхідну кількість *світлодіодних світильників* 7 шт.

8.5 Висновки до розділу

Дотримання всіх необхідних умов праці не лише сприяє збереженню здоров'я працівників, а також підвищує ефективність виробництва в цілому.

З цих міркувань було здійснено аналіз умов праці на робочому місці ІТ-фахівця, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи. Виконано розрахунок штучного освітлення, як одного з ключових факторів впливу на працездатність та здоров'я програміста. Розроблено заходи з умов поліпшення охорони праці.

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи CAN-мережі на основі технології CSDN.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів CAN-мережі на основі технології CSDN.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем CAN-мережі на основі технології CSDN.

– Досліджена система CAN-мережі на основі технології CSDN.

– На основі отриманих результатів досліджень створена програмна реалізація системи CAN-мережі на основі технології CSDN.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання CAN-мережі на основі технології CSDN.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створені програмного забезпечення було використано об’єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Delphi 10.4 Sydney. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм Md5.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

Розроблена програма має реальний економічний ефект від її впровадження у виробництво у сумі 15003 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,1 роки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Олійник А.О. Дослідження та програмна реалізація системи CAN-мережі на основі технології CSDN // Збірник праць молодих науковців ЦНТУ. – Вип. 13. – Кропивницький: ЦНТУ, 2022.

2. Смирнов А.А. Анализ и сравнительное исследование перспективных направлений развития цифровых телекоммуникационных систем и сетей / А.А.Смирнов, В.В.Босько, Е.В.Мелешко // Системи обробки інформації. – Х.: ХУ ПС, 2008. – Вип.7(74). – С.120-123.

3. Смирнов А.А. Усовершенствование метода управления очередями в многопротокольных узлах телекоммуникационной сети / А.А.Смирнов, Е.В.Мелешко // Збірник тез та доповідей другої всеукраїнської науково-практичної конференції «Системний аналіз. Інформатика. Управління». Запоріжжя. Тези доповідей. Запоріжжя: КПУ, 2011.

4. Смирнов С.А. Метод безопасной маршрутизации метаданных в облачные антивирусные системы / А.К. Дидык, С.А. Смирнов // Информационные технологии в управлении, образовании, науке и промышленности: монография / Под редакцией профессора В.С. Пономаренко. – Х.: Видавець Рожко С.Г., 2016. – 566 с.

5. Смирнов С. А. Сравнительные исследования математических моделей технологии распространения компьютерных вирусов в информационно-телекоммуникационных сетях / Мохамад Абу Таам Гани, А. А. Смирнов, А. В. Коваленко, С. А. Смирнов // Системи обробки інформації: зб. наук. праць. – Х.: ХУПС, 2014. – Вип. 9(125). – 105-110.

6. Смирнов С. А. Математическая модель интеллектуального узла коммутации с обслуживанием информационных пакетов различного приоритета / Мохамад Абу Таам Гани, А. А. Смирнов, Н. С. Якименко, С. А. Смирнов //

Збірник наукових праць Харківського університету Повітряних Сил. – Харків: ХУПС, 2014. – Вип. 4 (41). – С. 48-52.

7. Смирнов С. А. Исследование показателей качества функционирования интеллектуальных узлов коммутации в телекоммуникационных системах и сетях / Мохамад Абу Таам Гани, А. А. Смирнов, Н. С. Якименко, С. А. Смирнов // Наука і техніка Повітряних Сил Збройних Сил України: наук. журн. –Х.: ХУПС, 2014. – № 4(17). – С. 90-95.

8. Смирнов С. А. Усовершенствованный алгоритм управления доступом к «облачным» телекоммуникационным ресурсам / Мохамад Абу Таам Гани, А. А. Смирнов, Н. С. Якименко, С. А. Смирнов // Системи обробки інформації: зб. наук. праць. – Х.: ХУПС, 2015. –Вип. 1(126). – С. 150-153.

9. Smirnov S.A. Method of controlling access to intellectual switching nodes of telecommunication networks and systems / A.A. Smirnov, Mohamad Abou Taam, S.A. Smirnov // International Journal of Computational Engineering Research (IJCER). – Volume 5, Issue 5. – India. Delhi. – 2015. – P. 1-7.

10. Смирнов С. А. Анализ и исследование методов управления сетевыми ресурсами для обеспечения антивирусной защиты даннях / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Системи озброєння і військова техніка: наук. журн. – Х.: ХУПС, 2015. – № 3(43). – С. 100-107.

11. Смирнов С. А. Исследование эффективности метода управления доступом к облачным антивирусным телекоммуникационным ресурсам / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Наука і техніка Повітряних Сил Збройних Сил України: наук. журн. –Х.: ХУПС, 2015. – № 3(20). – С. 134-141.

12. Смирнов С. А. Комплекс gert-моделей технологии облачной антивирусной защиты телекоммуникационной системы / А. А. Смирнов, А. К. Дидык, А. Н. Дреев, С. А. Смирнов // Безпека інформації: наук. – практ. журн. – К.: НАУ, 2015. – Т. 21, № 3. – С. 251-262.

13. Смирнов С. А. Метод безопасной маршрутизации метаданных в облачные антивирусные системы / А. А. Смирнов, А. К. Дидык, С. А. Смирнов // Системи озброєння і військова техніка: наук. журн. – Х.: ХУПС, 2016. – № 2 (46). – С. 146-149.

14. Смирнов С. А. Модели системы нейросетевых экспертов безопасной маршрутизации в облачных антивирусных системах / А. А. Смирнов, А. К. Дидык, А. Н. Дреев, С. А. Смирнов // Системи обробки інформації: зб. наук. праць. – Х.: ХУПС, 2016. – Вип. 3 (140). – С. 36-39.

15. Смирнов С. А. Метод безопасной маршрутизации на базовом множестве путей передачи метаданных в облачные антивирусные системы / В. Л. Бурячок, С. А. Смирнов // Системи управління, навігації та зв'язку. – Полтава, 2016. – Вип. 4(40). – С. 57-62.

16. Смирнов С. А. Способ контроля линий связи телекоммуникационной системы облачного антивируса / А. А. Смирнов, А. К. Дидык, А. Н. Дреев, С. А. Смирнов // Збірник наукових праць Харківського університету Повітряних Сил. – Харків: ХУПС, 2016. – № 2 (47). – С. 148-152.

17. Смирнов С. А. Дослідження та реалізація GERT-моделі технології розповсюдження комп'ютерних вірусів для захисту телекомунікаційних систем / В. Л. Бурячок, Мохамад Абу Таам Гани, С. А. Смирнов // Інформаційні технології та комп'ютерна інженерія: зб. тез доп. наук.-практ. конф., м. Кіровоград, 4 грудня 2014 р. – Кіровоград: КНТУ, 2014. – С. 168.

18. Смирнов С. А. Исследование математических моделей технологии распространения компьютерных виру сов / А. А. Смирнов, Мохамад Абу Таам Гани, С. А. Смирнов // Актуальні питання забезпечення кібернетичної безпеки та захисту інформації: зб. наук. праць міжнар. наук.-практ. конф., м. Київ, 25-28 лютого 2015 р. – К.: Європейський університет, 2015. – С. 90-91.

19. Смирнов С. А. Метод управления доступом к «облачным» ресурсам для защиты телекоммуникационных систем / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Всеукраїнська науково-практична конференція

«Інформаційна безпека держави, суспільства та особистості», м. Кіровоград, 16 квітня 2015 р.: зб. тез доп. – Кіровоград: КНТУ, 2015. – С. 50-52.

20. Смирнов С. А. Разработка метода управления доступом в интеллектуальных узлах коммутации / А. А. Смирнов, Мохамад Абу Таам Гани, С. А. Смирнов // Проблеми і перспективи розвитку ІТ-індустрії: зб. тез VII міжнар. наук.-практ. конф., м. Харків, 17-18 квітня 2015 р. – Х.: ХНЕУ, 2015. – С. 14.

21. Смирнов С.А. Реализация метода управления доступом в интеллектуальных узлах коммутации / А.А. Смирнов, Мохамад Абу Таам Гани, С.А. Смирнов // Збірник тез XVII міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування». м. Кіровоград. 17-18 квітня 2015 р. – Кіровоград: КНТУ. – 2015. – С. 91-92.

22. Смирнов С. А. технология передачи сигнатур в облачные антивирусные системы для обеспечения защищенности телекоммуникационных сетей / А. А. Смирнов, С. А. Смирнов // Збірник тез V міжнародної науково-технічної конференції «ITSEC», Київ, 19-22 травня 2015 р. – К.: НАУ 2015. – С. 12-13.

23. Смирнов С. А. Реализация математической модели интеллектуального узла коммутации для обеспечения защищенности телекоммуникационной сети / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Інформаційна та економічна безпека (INFECO-2015): зб. тез II Міжнар. наук.-практ. Інтернет-конф., м. Харків, 21-22 травня 2015 р. – Х.: ХІБС УБС НБУ, 2015. – С. 20-24.

24. Смирнов С. А. Разработка математической модели технологии распространения компьютерных вирусов в информационно-телекоммуникационных сетях / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Сборник тезисов XI международной конференции «Стратегия качества в промышленности и образовании», г. Варна, Болгария, 01-06 июня 2015 г. – Варна: ТУВ, 2015. – С. 488-491.

25. Смирнов С. А. Метод управления доступом к облачным телекоммуни-кационным ресурсам для обеспечения защиты даннях / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Комп'ютерні технології та інформаційна безпека: зб. тез доп. міжнар. наук.-практ. конф., м. Кіровоград, 2-3 липня 2015 р. – Кіровоград: КНТУ, 2015. – С. 4-5.

26. Смирнов С. А. Имитационная модель системы управления доступом к облачным антивирусным телекоммуникационным ресурсам / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Збірник тез першої всеукраїнської науково-практичної конференції «Перспективні напрями захисту інформації» (м. Затока, 7-9 вересня 2015 р.). – Одеса: ОНАЗ, 2015. – С. 90-94.

27. Смирнов С. А. Разработка комплекса gert-моделей технологии облачной антивирусной защиты телекоммуникационной системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Інформаційні технології та взаємодії» (IT & I): зб. тез II міжнар. наук.-практ. конф., м. Київ, 3-5 листопада 2015 р. – К.: КНУ ім. Тараса Шевченка, 2015. – С. 65-67.

28. Смирнов С. А. Разработка моделей телекоммуникационной системы формирования и обработки метаданных в облачных антивирусных системах / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Информационные и телекоммуникационные технологии: образование, наука, практика: сб. тезисов II междунар. научно-практ. конф., г. Алматы, Казахстан, 3-4 декабря 2015 г. – Алматы: КазНИТУ им. К.И. Сатпаева, 2015. – С. 309-313.

29. Смирнов С. А. gert-модели технологии облачной антивирусной защиты / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Безпека українського суспільства в концепції вступу в постіндустріальне суспільство ЄС: зб. тез Круглого столу, м. Київ, 16 грудня 2015 р. – К.: Європейський університет, 2015. – С.41-43.

30. Смирнов С. А. Алгоритмы формирования множества маршрутов передачи метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Актуальні питання забезпечення

кібернетичної безпеки та захисту інформації: зб. наук. праць II Міжнар. наук.-практ. конф., м. Київ, 24-27 лютого 2016 р. – К.: Європейський університет, 2016. – С. 140-142.

31. Смирнов С. А. Разработка и реализация метода безопасной маршрутизации метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Securitea informationala 2015-2016: Conferenta internationala (editia a XII-a), Chisinau, Moldova, 3 martie 2016. – Chisinau: ADSEM, 2016. – С. 90-96.

32. Смирнов С. А. Алгоритм формирования базового множества маршрутов передачи метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Інформатика та системні науки (ICH-2016): зб. тез VII всеукр. наук.-практ. конф., м. Полтава, 10-12 березня 2016 р. – Полтава: ПУЕТ, 2016. – С. 261-263.

33. Смирнов С. А. Система обработки и формирования начального состояния маршрутизации метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Проблеми кібербезпеки інформаційно-телекомунікаційних систем: зб. тез наук.-практ. конф., м. Київ, 10-11 березня 2016 р. – К.: КНУ ім. Тараса Шевченка, 2016. – С. 81-82.

34. Смирнов С. А. Алгоритм безопасной маршрутизации на базовом множестве путей передачи метаданных в программный сервер облачной антивирусной системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Інформаційна безпека та комп'ютерні технології (IS&CT): зб. тез міжнар. наук.-практ. конф., м. Кіровоград, 24-25 березня 2016 р. – Кіровоград: КНТУ, 2016. – С. 73.

35. Смирнов С. А. Исследование способа контроля линий связи телекоммуникационной системы для облачных антивирусов / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Збірник тез першої міжнародної науково-практичної конференції «Проблеми науково-технічного та правового

забезпечення кібербезпеки у сучасному світі» (ПНПЗК-2016), м. Харків, 30 березня – 1 квітня 2016 р. – Х.: НТУ «ХПІ», 2016. – С. 14.

36. Смирнов С. А. Разработка способа контроля линий связи телекоммуникационной системы для облачных антивирусов / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Матеріали XVIII міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування» (м. Кіровоград, 15-16 квітня 2016 р.). –Кіровоград: КНТУ, 2016. – С. 182-186.

37. Смирнов С. А. Разработка и исследование способа контроля линий связи телекоммуникационных сетей для облачных антивирусных систем / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Проблеми і перспективи розвитку ІТ-індустрії: VIII міжнар. наук.-практ. конф., м. Харків, 28-29 квітня 2016 р.: зб. тез. – Х.: ХНЕУ, 2016. – С. 48.

38. Смирнов С. А. Модель системы нейросетевых экспертов безопасной маршрутизации для облачных антивирусных систем / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Інформаційна та економічна безпека (INFECO-2016): зб. тез ІІІ міжнар. наук.-практ. конф., м. Харків, 28-30 кві. 2016 р. – Х.: ХННІ ДВНЗ «УБС», 2016. – С. 178-182.

39. Смирнов С. А. Метод безопасной маршрутизации метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Сборник тезисов XII международной конференции «Стратегия качества в промышленности и образовании» (г. Варна, Болгария, 30 мая – 02 июня 2016 г.). – Варна: ТУВ, 2016. – С. 581-585.

40. Смирнов С. А. Оценка эффективности метода безопасной маршрутизации метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. С. Коваленко // РадіоЕлектроніка та ІнфоКомунікації: зб. тез першої наук. – техн. конф., м. Київ, 11-16 вересня 2016 р. – К.: НТУУ «КПІ», 2016. – С. 17.

41. Современные телекоммуникации. Технологии и экономика / [В.Л. Банкет, О.В. Бондаренко, П.П. Воробьенко и др.]; под ред. С.А. Довгого. – М.: Эко-Трендз, 2003. – 320 с.

42. Столлингс В. Современные компьютерные сети / Вильям Столлингс. –СПб.: Питер, 2003. – 778 с.

43. Таненбаум Э. Компьютерные сети / Эндрю Таненбаум; пер. с англ. А. Леонтьев. – СПб.: Питер, 2002. – 848 с.

44. Телекоммуникационные системы и сети: учебное пособие. В 3 томах / [В.В. Величко, Е.А. Субботин, В.П. Шувалов, А.Ф. Ярославцев]; под ред. В.П. Шувалова. – М.: Горячая линия-Телеком, 2005, т. 3 – 592 с.

45. Хайкин С. Нейронные сети: полный курс / С. Хайкин. – М.:Вильямс, 2006. – 1103 с.

46. Шелухин О.И. Фрактальные процессы в телекоммуникациях: моногр. / О.И. Шелухин, А.М. Тенякшев, А.В. Осин – М.: Радиотехника, 2003. – 480 с.

47. Elwalid, D. Mitra, I. Saniee, and I. Widjaja. Routing and Protection in GMPLS Networks: From Shortest Paths to Optimized Designs // Journal of lightwave technology. – 2003. – №21(11), P. 2828-28-38.

48. A.B. Bagula, M. Botha, and A.E Krzesinski. Online Traffic Engineering: The Least Interference Optimization Algorithm // IEEE Communications Society – 2004, P. 1232-1236.

49. Anees Shaikh, Jennifer Rexford, and Kang G. Shin. Evaluating the Impact of Stale Link State on Quality-of-Service Routing // IEEE/ACM Transactions on Networking. – 2001. – №9(2), P. 162-176.

50. Basabi Chakraborty. Simultaneous Search for Multiple Routes using Genetic Algorithm / IEEE lntemational Conference on Computational oltelligencs for Measurement System and Applications Boston. MA, USA, 14-16, July 2004, P. 77-80/

51. Barakat, E. Altman, and W. Dabbous. On TCP performance in a heterogeneous network: a survey // IEEE Communications Magazine. – 2000. – №38(1). – P. 40 – 46.

52. Carpenter G., A., Grossberg S. Pattern Recognition by Self-Organizing Neural Networks, Cambridge, MA, MIT Press, 1991.

53. Cloud security, Deep Dive series, August 2011 [Электронный ресурс]. – Режим доступа к ресурсу: http://www.slideshare.net/kimrenejensen/cloud-security-deep-dive-2011#14375029197881&fbinitialized

54. Chris Loeser, Andre Brinkmann, Ulrich Ruckert. Distributed Path Selection (DPS) a Traffic Engineering Protocol for IP-Networks / Proceedings of the 37th Hawaii International Conference on System Sciences – 2004, P. 1-8.

55. Dai Boong Lee and Hwangjun Song. Dynamic Class Selecting Mechanism for Guaranteed Service with Minimum Cost over Relative Differentiated-Services Networks / IEEE International Conference on Multimedia and Expo (ICME) – 2004, P. 237-240.

56. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПІН 3.3.2-007-98. – Режим доступу до ресурсу: https://zakon.rada.gov.ua/rada/show/v0007282-98

57. Закон України «Про охорону праці» від 14.10.1992 р. № 2694-XII. – Режим доступу до ресурсу: https://zakon.rada.gov.ua/laws/show/2694-12

58. Зеркалов Д. В. Охорона праці в Галузі: Загальні вимоги: навч. посіб. Київ: Основа. 2011. 551 с.

59. Наказ Міністерства соціальної політики України 14.02.2018 № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». – Режим доступу до ресурсу: https://zakon.rada.gov.ua/laws/show/z0508

60. Охорона праці. Ч. 1. Захисне заземлення: метод. вказ. до викон. розрахунків з викор. персон. ЕОМ ІВМ сумісного типу / Кіровоград. ін-т с.-г.

машинобуд.; [укл. О. В. Оришака, Є. К. Солових, В. О. Оришака]. – Кіровоград: КІСМ, 1997. – 20 с. Режим доступу до ресурсу: http://dspace.kntu.kr.ua/jspui/handle/123456789/4358

61. Постанова № 42 від 01.12.1999 Головного державного санітарного лікаря України «Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99. – Режим доступу до ресурсу: https://zakon.rada.gov.ua/rada/show/va042282-99

62. Сакулин В.П., Шептовицкий В.М. Безопасность труда при монтаже и эксплуатации электроустановок / В.П.Сакулин, В.М.Шептовицкий. – Л. : “Колос”, 1973. – 238 с.

63. Центр післядипломної освіти та підвищення кваліфікації. – Режим доступу до ресурсу: https://cpo.stu.cn.ua

64. Оришака, О. В. Основи охорони праці: навч. посіб. / О. В. Оришака, Г. П. Горбачова, К. М. Марченко; М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т. – Кропивницький : ЦНТУ, 2022. – 175 с. – Режим доступу до ресурсу: http://dspace.kntu.kr.ua/jspui/handle/123456789/12161 (дата звернення 19.09.22).

Додаток А
(обов'язковий)

Технічне завдання

Зміст

					ВКРМ-123.22.0018.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив		Олійник А.О.			Дослідження та програмна реалізація системи CAN-мережі на основі технології CSDN	Літ.	Аркуш 1 / Аркушів 6
Перевірив		Доренський О.П.				М	
Н. Контр.		Гермак В.С.				ЦНТУ КІ-21М-1,4	
Затв.		Смірнов О.А.					

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи CAN-мережі на основі технології CSDN.

2 Підстава для розробки

Підставою для розробки служить завдання на випускну кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 19-13 від 17.08.2022 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи CAN-мережі на основі технології CSDN.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

– вибір і обґрунтування методів реалізації проекту;

– розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

– техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;

– аналіз умов праці;

– розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

– програмну реалізацію системи CAN-мережі на основі технології CSDN;

– цілісність даних у процесі роботи та при зберіганні;

– простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

– температура повітря: 19-20 град. по Цельсію;

– відносна вологість повітря до 80%;

– атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 10/11.

5.8.1 Обладнання

Комп’ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14” 1Mb або сумісні з ним.

5.8.2 Мова програмування

Середовище Delphi 10.4 Sydney.

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати розрахунок витрат показників економічного ефекту з урахуванням цін на 3 вересня 2022 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинен бути розглянутий аналіз умов праці на робочому місці ІТ-фахівця.

9 Перелік документів, що розробляються

– Наукова новизна – 1 аркуш.
– Структурна схема системи – 1 аркуш.
– Функціональна схема системи – 1 аркуш.
– Діаграма процесів – 1 аркуш.
– Блок-схема алгоритму роботи програми – 2 аркуша.
– Показники економічної ефективності – 1 аркуш.
– Пояснювальна записка – 117 аркушів.

10 Етапи розробки

10.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.

10.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 10.12.2022 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 21.12.2022 р.

Додаток Б

(обов’язковий)

Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за

другим (магістерським) рівнем вищої освіти

_____ Доренський О.П.

Дослідження та програмна реалізація

системи CAN-мережі на основі технології CSDN

Лістинг програми

Код документу 12

Носій: CD/DVD-диск / USB-флеш-накопичувач

Загальна кількість аркушів: 21

Літера: РП

Кропивницький – 2022 року

UnitMain.pas – основна програма

```
unit UnitMain;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, CheckLst, DB, DBTables, Grids, DBGrids, ComCtrls, Pumpdata,
  ExtCtrls, jpeg;

type
  TFormMain = class(TForm)
    GroupBox1: TGroupBox;
    CheckListBox1: TCheckListBox;
    StatusBar1: TStatusBar;
    Panel1: TPanel;
    Image1: TImage;
    Query1: TQuery;
    DataSource1: TDataSource;
    GroupBox2: TGroupBox;
    Memo1: TMemo;
    GroupBox3: TGroupBox;
    Memo2: TMemo;
    Image5: TImage;
    Image6: TImage;
    Label1: TLabel;
    Label2: TLabel;
    Image2: TImage;
    Image7: TImage;
    Label3: TLabel;
    Image8: TImage;
    Image9: TImage;
    Label4: TLabel;
    Image3: TImage;
    Image4: TImage;
    Label5: TLabel;
    Image10: TImage;
    Label6: TLabel;
    Image11: TImage;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    procedure FormCreate(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Image3Click(Sender: TObject);
    procedure Image6Click(Sender: TObject);
    procedure Label1Click(Sender: TObject);
    procedure Image2Click(Sender: TObject);
    procedure Image7Click(Sender: TObject);
    procedure Label3Click(Sender: TObject);
    procedure Image9Click(Sender: TObject);
    procedure Image4Click(Sender: TObject);
    procedure Image11Click(Sender: TObject);
  private
    { Private declarations }

    procedure ShowHint(Sender: TObject);
  public
    { Public declarations }

  end;

var
  FormMain: TFormMain;
```

```
implementation

uses UnitAddKr, AddZagrz, UnitView, UnitInfo;

{$R *.dfm}

procedure TFormMain.FormCreate(Sender: TObject);
begin
//Вивід списку критеріїв
  Query1.SQL.Clear;
  Query1.SQL.Add('select * from krt order by Name asc');
  Query1.Open;
  Query1.First;
  while not Query1.Eof do begin
    CheckListBox1.Items.Add(Query1.Fields[1].AsString);
    Query1.Next;
  end;
end;

procedure TFormMain.Button2Click(Sender: TObject);
begin

end;

//Процедура для StatusBar
procedure TFormMain.ShowHint(Sender: TObject);
begin
  StatusBar1.SimpleText:=Application.Hint;
end;

//Процедура для StatusBar
procedure TFormMain.FormShow(Sender: TObject);
begin
  Application.OnHint:=ShowHint;
end;

procedure TFormMain.Button3Click(Sender: TObject);

begin

end;

//Натиснення кнопки "Додати критерії"
procedure TFormMain.Image3Click(Sender: TObject);
begin
  FormAddKr.ShowModal;
end;



procedure TFormMain.Image6Click(Sender: TObject);
type
  Zagrzes=record
    name:string;
    probability:integer;
    all_kr:integer;
  end;
var str, prob:string;
  probab,probab2,p1,p2:real;
  i,j,count_krt,kol,num_zagr:integer;
  a:array[0..128] of integer;
  zagr:array[0..128] of Zagrzes;
  temp: Zagrzes;
begin
  Image6.Visible:=False;

  count_krt:=0;
```

```
  kol:=0;
  num_zagr:=0;
  for i:=0 to 128 do a[i]:=0;
  for i:=0 to 128 do zagr[i].probability:=0;

  //Очищення Memo1, Memo2
  Memo1.Clear;
  Memo2.Clear;

  //Формування списку номерів вибраних критеріїв
  for i:=0 to CheckListBox1.Items.Count-1 do begin
    if CheckListBox1.Checked[i] then begin
      str:=CheckListBox1.Items[i];
      Query1.First;
      while not Query1.Eof do begin
        if Query1.Fields[1].AsString=str then begin
          a[count_krt]:=Query1.Fields[0].AsInteger;
          count_krt:=count_krt+1;
        end;
        Query1.Next;
      end;
    end;
  end;

  //Відправляємо запит в БД загроз безпеці мережі
  Query1.Active:=False;
  Query1.SQL.Clear;
  Query1.SQL.Add('select * from Zagrzesses order by Name asc');
  Query1.Open;

  Query1.First;
  while not Query1.Eof do begin
    zagr[num_zagr].all_kr:=0;
    for i:=2 to 11 do begin
      if Query1.Fields[i].AsInteger<>0 then
zagr[num_zagr].all_kr:=zagr[num_zagr].all_kr+1;
      for j:=0 to count_krt do begin
        if (Query1.Fields[i].AsInteger=a[j]) and (Query1.Fields[i].AsInteger<>0)
then begin
          //Memo1.Lines.Add('OK - '+IntToStr(a[j]));
          zagr[num_zagr].probability:=zagr[num_zagr].probability+1;
          zagr[num_zagr].name:=Query1.Fields[1].AsString;
        end;
      end;
    end;
    Query1.Next;
    num_zagr:=num_zagr+1;

  end;

  //Сортування
{   for j:=0 to 128 do begin
   for i:=0 to 128 do begin
    probab:=zagr[i].probability/zagr[i].all_kr;
    if zagr[i+1].probability<>0 then begin
      probab2:=zagr[i+1].probability/zagr[i+1].all_kr;
      if probab<probab2 then begin
        temp.name:=zagr[i+1].name;
        temp.probability:=zagr[i+1].probability;
        temp.all_kr:=zagr[i+1].all_kr;

        zagr[i+1].name:=zagr[i].name;
        zagr[i+1].probability:=zagr[i].probability;
        zagr[i+1].all_kr:=zagr[i].all_kr;

        zagr[i].name:=temp.name;
        zagr[i].probability:=temp.probability;
        zagr[i].all_kr:=temp.all_kr;
      end;
```

```
    end;
    end;
   end;
}
  //Вивід результату
  for i:=0 to 128 do begin
    if zagr[i].probability<>0 then begin
      probab:=(zagr[i].probability/zagr[i].all_kr)*100;
      Memo1.Lines.Add(zagr[i].name + '   (Вірогідність:
'+FloatToStrF(probab,ffNumber,3,1)+'%)');
    end;
  end;

  //Вивід детальної інформації
  for i:=0 to 128 do begin
    if zagr[i].probability<>0 then begin
      Query1.SQL.Clear;
      Query1.SQL.Add('select * from about where Name="'+zagr[i].name+'"');
      Query1.Open;
      Query1.First;
      Memo2.Lines.Add('ЗАГРОЗА: '+Query1.Fields[0].AsString);
      Memo2.Lines.Add('       '+Query1.Fields[1].AsString);
      Memo2.Lines.Add(' ');
      Memo2.Lines.Add('КРИТЕРІЇ ВИЯВЛЕННЯ ЗАГРОЗИ БЕЗПЕКИ ІНФОРМАЦІЇ У
МЕРЕЖІ:');
      Memo2.Lines.Add('       '+Query1.Fields[2].AsString);
      Memo2.Lines.Add(' ');
      Memo2.Lines.Add('МЕТОДИ ТА ЗАСОБИ ЗАПОБІГАННЯ ДАНІЙ ЗАГРОЗІ БЕЗПЕКИ
ІНФОРМАЦІЇ У МЕРЕЖІ:');
      Memo2.Lines.Add('       '+Query1.Fields[3].AsString);
      Memo2.Lines.Add(' ');
      Memo2.Lines.Add('-------------------------------------------------------------
-------------------------------------------------------------------------------
-----------------------------------------------');
      Memo2.Lines.Add(' ');
    end;
  end;

  //Перемотування Memo на початок
  Memo1.SelStart := 0;
  Memo1.Perform(EM_SCROLLCARET, 0, 0);
  Memo2.SelStart := 0;
  Memo2.Perform(EM_SCROLLCARET, 0, 0);

  Query1.SQL.Clear;
  Query1.SQL.Add('select * from krt order by Name asc');
  Query1.Open;
  Query1.First;

  Image6.Visible:=True;
end;

procedure TFormMain.Label1Click(Sender: TObject);
begin
  Image6Click(Sender);
end;


procedure TFormMain.Image2Click(Sender: TObject);
begin
  Image2.Visible:=False;
  FormAddKr.ShowModal;
end;

procedure TFormMain.Image7Click(Sender: TObject);
begin
  FormMain.Close;
end;
```

```
procedure TFormMain.Label3Click(Sender: TObject);
begin
  Label3Click(Sender);
end;

procedure TFormMain.Image9Click(Sender: TObject);
begin
  Image9.Visible:=False;
  FormAddZagrz.ShowModal;
end;

procedure TFormMain.Image4Click(Sender: TObject);
begin
  FormView.ShowModal;
end;

procedure TFormMain.Image11Click(Sender: TObject);
begin
  FormInfo.ShowModal;
end;

end.
```

AddZagrz.pas – додавання загроз безпеці інформації у мережі в базу знань

```
unit AddZagrz;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, CheckLst;

type
  TFormAddZagrz = class(TForm)
    GroupBox3: TGroupBox;
    GroupBox4: TGroupBox;
    GroupBox5: TGroupBox;
    GroupBox6: TGroupBox;
    GroupBox7: TGroupBox;
    ListBox1: TListBox;
    Button1: TButton;
    Button2: TButton;
    Button4: TButton;
    GroupBox2: TGroupBox;
    Edit1: TEdit;
    Label1: TLabel;
    Button3: TButton;
    GroupBox1: TGroupBox;
    CheckListBox1: TCheckListBox;
    Memo1: TMemo;
    Memo2: TMemo;
    Memo3: TMemo;
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure FormShow(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure ListBox1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FormAddZagrz: TFormAddZagrz;

implementation

uses UnitMain, UnitAddKr;

{$R *.dfm}

procedure TFormAddZagrz.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
  FormMain.Image9.Visible:=True;
end;

procedure TFormAddZagrz.FormShow(Sender: TObject);
begin
  ListBox1.Clear;
  CheckListBox1.Clear;
  Memo1.Clear;
  Memo2.Clear;
  Memo3.Clear;
  Edit1.Clear;
  Button4.Enabled:=False;
```

```
  with FormMain do begin
//Вивід списку загроз безпеці інформації у мережі
  Query1.SQL.Clear;
  Query1.SQL.Add('select * from Zagrzesses order by Name asc');
  Query1.Open;
  Query1.First;
  while not Query1.Eof do begin
    ListBox1.Items.Add(Query1.Fields[1].AsString);
    Query1.Next;
  end;
//Вивід списку критеріїв безпеки інформації у мережі
  Query1.Close;
  Query1.SQL.Clear;
  Query1.SQL.Add('select * from krt order by Name asc');
  Query1.Open;
  Query1.First;
  while not Query1.Eof do begin
    FormAddZagrz.CheckListBox1.Items.Add(Query1.Fields[1].AsString);
    Query1.Next;
  end;
  end;
end;

procedure TFormAddZagrz.Button2Click(Sender: TObject);
begin
  FormAddZagrz.Close;
end;

procedure TFormAddZagrz.Button3Click(Sender: TObject);
var
  str,str1,str2,str3: string;
  a:array[1..10] of integer;
  count_krt,i,num: integer;
begin
randomize;
for i:=1 to 10 do a[i]:=0;
//Перевірка чи заповнені поля
  str:='Не заповнені поля(е):';
  if(FormAddZagrz.Memo1.Lines[0]='') or (FormAddZagrz.Memo2.Lines[0]='') or
(FormAddZagrz.Memo3.Lines[0]='') then begin
    if FormAddZagrz.Memo1.Lines[0]='' then str:=str+' "Стисло про загрозу"';
    if FormAddZagrz.Memo2.Lines[0]='' then str:=str+' "Критерії виявлення
загрози безпеки інформації у мережі"';
    if FormAddZagrz.Memo3.Lines[0]='' then str:=str+' "Методи та засоби
запобігання загрозі безпеки інформації у мережі"';
    Application.MessageBox(PChar(str),'Увага!!!',MB_OK)
  end else begin
    //Вивід списку критеріїв безпеки інформації у мережі
    with FormMain do begin
      Query1.Close;
      Query1.SQL.Clear;
      Query1.SQL.Add('select * from krt order by Name asc');
      Query1.Open;
      Query1.First;
    end;

    count_krt:=1;
    for i:=0 to CheckListBox1.Items.Count-1 do begin
      if CheckListBox1.Checked[i] then begin
        str:=CheckListBox1.Items[i];
        FormMain.Query1.First;
        while not FormMain.Query1.Eof do begin
          if FormMain.Query1.Fields[1].AsString=str then begin
            a[count_krt]:=FormMain.Query1.Fields[0].AsInteger;
            count_krt:=count_krt+1;
          end;
          FormMain.Query1.Next;
        end;
      end;
```

```
    end;

    //якщо не введені критерії безпеки інформації у мережі
    str:='Критерії безпеки інформації у мережі не введені!';
    if count_krt=1 then Application.MessageBox(PChar(str),'Увага!!!',MB_OK)
    else begin

    num:=random(32767);
    FormMain.Query1.Close;
    FormMain.Query1.SQL.Clear;
    FormMain.Query1.SQL.Add('insert into Zagrzesses values
('+IntToStr(num)+',"'+FormAddZagrz.Edit1.Text+'", "'+IntToStr(a[1])+'",
"'+IntToStr(a[2])+'", "'+IntToStr(a[3])+'", "'+IntToStr(a[4])+'",
"'+IntToStr(a[5])+'", "'+IntToStr(a[6])+'", "'+IntToStr(a[7])+'",
"'+IntToStr(a[8])+'", "'+IntToStr(a[9])+'", "'+IntToStr(a[10])+'")');
    FormMain.Query1.ExecSQL;

    str1:='';
    str2:='';
    str3:='';
    for i:=0 to Memo1.Lines.Count do str1:=str1+Memo1.Lines[i];
    for i:=0 to Memo2.Lines.Count do str2:=str2+Memo2.Lines[i];
    for i:=0 to Memo3.Lines.Count do str3:=str3+Memo3.Lines[i];


    FormMain.Query1.Close;
    FormMain.Query1.SQL.Clear;
    FormMain.Query1.SQL.Add('insert into about values
("'+FormAddZagrz.Edit1.Text+'", "'+str1+'", "'+str2+'", "'+str3+'")');
    FormMain.Query1.ExecSQL;

    FormAddZagrz.ListBox1.Clear;
    FormMain.Query1.SQL.Clear;
    FormMain.Query1.SQL.Add('select * from Zagrzesses order by Name asc');
    FormMain.Query1.Open;
    FormMain.Query1.First;
    while not FormMain.Query1.Eof do begin
      ListBox1.Items.Add(FormMain.Query1.Fields[1].AsString);
      FormMain.Query1.Next;
    end;
    end;
  end;
end;

procedure TFormAddZagrz.Button4Click(Sender: TObject);
var i:integer;
begin
 with FormMain do begin
  for i:=0 to FormAddZagrz.ListBox1.Items.Count-1 do begin
    if FormAddZagrz.ListBox1.Selected[i] then begin

        Query1.Close;
        Query1.SQL.Clear;
        Query1.SQL.Add('delete from Zagrzesses where Name =
"'+FormAddZagrz.ListBox1.Items[i]+'"');
        Query1.ExecSQL;

        Query1.Close;
        Query1.SQL.Clear;
        Query1.SQL.Add('delete from about where Name =
"'+FormAddZagrz.ListBox1.Items[i]+'"');
        Query1.ExecSQL;

      FormAddZagrz.ListBox1.Clear;
      Query1.SQL.Clear;
      Query1.SQL.Add('select * from Zagrzesses order by Name asc');
      Query1.Open;
      Query1.First;
      while not Query1.Eof do begin
```

```
        FormAddZagrz.ListBox1.Items.Add(Query1.Fields[1].AsString);
        Query1.Next;
      end;
    break;
    end;
  end;
  end;
end;

procedure TFormAddZagrz.Button1Click(Sender: TObject);
begin
  FormAddKr.ShowModal;
   FormAddZagrz.CheckListBox1.Clear;
    FormMain.Query1.SQL.Clear;
    FormMain.Query1.SQL.Add('select * from krt order by Name asc');
    FormMain.Query1.Open;
    FormMain.Query1.First;
    while not FormMain.Query1.Eof do begin
      FormAddZagrz.CheckListBox1.Items.Add(FormMain.Query1.Fields[1].AsString);
      FormMain.Query1.Next;
    end;
end;

procedure TFormAddZagrz.ListBox1Click(Sender: TObject);
begin
  Button4.Enabled:=True;
end;

end.
```

UnitAddKr.pas – додавання критеріїв безпеки інформації у мережі в базу знань

```
unit UnitAddKr;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, Grids, DBGrids, DBTables, StdCtrls;

type
  TFormAddKr = class(TForm)
    GroupBox1: TGroupBox;
    ListBox1: TListBox;
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    GroupBox2: TGroupBox;
    Edit1: TEdit;
    Label1: TLabel;
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure FormShow(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure ListBox1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FormAddKr: TFormAddKr;

implementation

uses UnitMain, AddZagrz;

{$R *.dfm}

procedure TFormAddKr.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
  FormMain.Image2.Visible:=True;
  FormMain.CheckListBox1.Clear;
    with FormMain do begin
    Query1.SQL.Clear;
    Query1.SQL.Add('select * from krt order by Name asc');
    Query1.Open;
    Query1.First;
    while not Query1.Eof do begin
      CheckListBox1.Items.Add(Query1.Fields[1].AsString);
      Query1.Next;
    end;
  end;
  Edit1.Clear;
end;

procedure TFormAddKr.FormShow(Sender: TObject);
var i:integer;
begin
  Button1.Enabled:=False;
  ListBox1.Clear;
//Вивід списку критеріїв безпеки інформації у мережі
  with FormMain do begin
    Query1.SQL.Clear;
    Query1.SQL.Add('select * from krt order by Name asc');
```

```
    Query1.Open;
    Query1.First;
    while not Query1.Eof do begin
      ListBox1.Items.Add(Query1.Fields[1].AsString);
      Query1.Next;
    end;
  end;
end;

procedure TFormAddKr.Button3Click(Sender: TObject);
var num:integer;
begin
Randomize;
if Edit1.Text='' then Application.MessageBox('Поле з ім`ям критерію
порожнє','Увага!!!',MB_OK)
else begin
  with FormMain do begin
      //Query1.Last;
      //num:=Query1.Fields[0].AsInteger+1;

    num:=random(32767);
    Query1.Close;
    Query1.SQL.Clear;
    Query1.SQL.Add('insert into krt values (' + IntToStr(num) +
','"'+Edit1.Text+'")');
    Query1.ExecSQL;

    ListBox1.Clear;
    Query1.SQL.Clear;
    Query1.SQL.Add('select * from krt order by Name asc');
    Query1.Open;
    Query1.First;
    while not Query1.Eof do begin
      ListBox1.Items.Add(Query1.Fields[1].AsString);
      Query1.Next;
    end;
  end;
end;
end;

procedure TFormAddKr.Button2Click(Sender: TObject);
begin
  FormAddKr.Close;
end;


procedure TFormAddKr.Button1Click(Sender: TObject);
var i:integer;
begin
  with FormMain do begin
  for i:=0 to FormAddKr.ListBox1.Items.Count-1 do begin
    if FormAddKr.ListBox1.Selected[i] then begin

      Query1.Close;
      Query1.SQL.Clear;
      Query1.SQL.Add('delete from krt where Name =
"'+FormAddKr.ListBox1.Items[i]+'"');
      Query1.ExecSQL;

      ListBox1.Clear;
      Query1.SQL.Clear;
      Query1.SQL.Add('select * from krt order by Name asc');
      Query1.Open;
      Query1.First;
      while not Query1.Eof do begin
        FormAddKr.ListBox1.Items.Add(Query1.Fields[1].AsString);
        Query1.Next;
      end;
      break;
```

```
    end;
  end;
  end;
end;

procedure TFormAddKr.ListBox1Click(Sender: TObject);
begin
  Button1.Enabled:=True;
end;

end.
```

UnitView.pas – перегляд бази знань

```
unit UnitView;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, CheckLst;

type
  TFormView = class(TForm)
    GroupBox1: TGroupBox;
    GroupBox2: TGroupBox;
    Memo1: TMemo;
    GroupBox3: TGroupBox;
    Memo2: TMemo;
    ListBox1: TListBox;
    procedure FormShow(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure ListBox1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FormView: TFormView;

implementation

uses UnitMain;

{$R *.dfm}

procedure TFormView.FormShow(Sender: TObject);
begin
  FormMain.Image4.Visible:=False;

   FormView.ListBox1.Clear;
//Вивід списку критеріїв безпеки інформації у мережі
  with FormMain do begin
    Query1.SQL.Clear;
    Query1.SQL.Add('select * from Zagrzesses order by Name asc');
    Query1.Open;
    Query1.First;
    while not Query1.Eof do begin
     FormView.ListBox1.Items.Add(Query1.Fields[1].AsString);
      Query1.Next;
    end;
  end;
end;

procedure TFormView.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  FormMain.Image4.Visible:=True;
end;

procedure TFormView.ListBox1Click(Sender: TObject);
var a:array[1..10] of integer;
i,j, count:integer;
name:string;
begin
  count:=1;
  //Визначаємо вибрану загрозу
  for i:=0 to FormView.ListBox1.Items.Count-1 do begin
```

```
    if FormView.ListBox1.Selected[i] then begin
      name:=FormView.ListBox1.Items[i];
      FormMain.Query1.Close;
      FormMain.Query1.SQL.Clear;
      FormMain.Query1.SQL.Add('select * from Zagrzesses where Name =
"'+FormView.ListBox1.Items[i]+'"');
      FormMain.Query1.Open;
      break;
    end;
  end;

  //Формування списку для виводу критеріїв безпеки інформації у мережі
  FormMain.Query1.First;
  for i:=2 to 10 do begin
    if FormMain.Query1.Fields[i].AsInteger<>0 then begin
      a[count]:=FormMain.Query1.Fields[i].AsInteger;
      count:=count+1;
    end else break;
  end;

  //Вивід критеріїв безпеки інформації у мережі
  FormView.Memo1.Clear;
  for i:=1 to count-1 do begin
    FormMain.Query1.Close;
    FormMain.Query1.SQL.Clear;
    FormMain.Query1.SQL.Add('select * from krt where Id="'+IntToStr(a[i])+'"');
    FormMain.Query1.Open;
    FormView.Memo1.Lines.Add(FormMain.Query1.Fields[1].AsString);
  end;

  //Вивід детальної інформації
    Memo2.Clear;
    FormMain.Query1.Close;
    FormMain.Query1.SQL.Clear;
    FormMain.Query1.SQL.Add('select * from about where Name = "'+name+'"');
    FormMain.Query1.Open;
    Memo2.Lines.Add('ЗАГРОЗА: '+FormMain.Query1.Fields[0].AsString);
      Memo2.Lines.Add('      '+FormMain.Query1.Fields[1].AsString);
      Memo2.Lines.Add(' ');
      Memo2.Lines.Add('КРИТЕРІЇ БЕЗПЕКИ ІНФОРМАЦІЇ У МЕРЕЖІ ВИЯВЛЕННЯ ЗАГРОЗИ
БЕЗПЕКИ ІНФОРМАЦІЇ У МЕРЕЖІ:');
      Memo2.Lines.Add('      '+FormMain.Query1.Fields[2].AsString);
      Memo2.Lines.Add(' ');
      Memo2.Lines.Add('МЕТОДИ ТА ЗАСОБИ ЗАПОБІГАННЯ ДАНІЙ ЗАГРОЗІ БЕЗПЕКИ
ІНФОРМАЦІЇ У МЕРЕЖІ:');
      Memo2.Lines.Add('       '+FormMain.Query1.Fields[3].AsString);

     with FormMain do begin
//Вивід списку загроз безпеці інформації у мережі
  Query1.SQL.Clear;
  Query1.SQL.Add('select * from Zagrzesses order by Name asc');
  Query1.Open;
  end;
end;

end.
```

Pumpdata.pas – формування бази знань

```
unit PumpData;

interface

uses
  SysUtils, IniFiles, Classes, BaseTypes;

type

  EFieldNormError = class(Exception);
  EFieldKindError = class(Exception);

  TNeuroField = class;
  TNeuroFields = array of TNeuroField;

  TNeuroField = class(TObject)
  private
    FAlpha: double;
    FDataIn: TVectorFloat;
    FDispersion: double;
    FInd: boolean;
    FKind: byte;
    FName: string;
    FNormType: byte;
    FValueMax: double;
    FValueMid: double;
    FValueMin: double;
    function GetDataIn(Index: integer): double;
    function GetKindName: TNeuroFieldType;
    function GetNormTypeName: TNormalize;
    function GetDataInCount: integer;
    procedure SetDataIn(Index: integer; Value: double);
    procedure SetKind(Value: byte);
    procedure SetNormType(Value: byte);
    procedure SetDataInCount(Value: integer);
  public
    procedure FindMinMax;
    procedure CalcMid;
    procedure CalcDispersion;
    procedure Normalize;
    procedure DeNormalize;
    property Alpha: double read FAlpha write FAlpha;
    property DataIn[Index: integer]: double read GetDataIn write SetDataIn;
    property DataInCount: integer read GetDataInCount write SetDataInCount;
    property Dispersion: double read FDispersion write FDispersion;
    property Ind: boolean read FInd write FInd;
    property Kind: byte read FKind write SetKind;
    property KindName: TNeuroFieldType read GetKindName;
    property Name: string read FName write FName;
    property NormType: byte read FNormType write SetNormType;
    property NormTypeName: TNormalize read GetNormTypeName;
    property ValueMax: double read FValueMax write FValueMax;
    property ValueMin: double read FValueMin write FValueMin;
    property ValueMid: double read FValueMid write FValueMid;
  end;

  TNDataSource = class(TObject)
  private
    FName: TFileName;
    function IsHeaderChar(AValue: char): boolean;
  public
    function FieldCount(AHeader: string): integer;
    procedure ExtractHeaders(const AFields: TNeuroFields; AHeader: string);
    procedure ExtractValues(const AVector:TVectorFloat; AHeader: string);
    property Name: TFileName read FName write FName;
  end;
```

```
implementation

{ Клас TNeuroField }

function TNeuroField.GetDataIn(Index: integer): double;
begin
  Result := FDataIn[Index];
end;

function TNeuroField.GetDataInCount: integer;
begin
  Result := High(FDataIn) + 1;
end;

function TNeuroField.GetKindName: TNeuroFieldType;
begin
  case FKind of
    0 : Result := fdInput;
    1 : Result := fdOutput;
    2 : Result := fdNone;
  end;
end;

function TNeuroField.GetNormTypeName: TNormalize;
begin
  case FNormType of
    0 : if KindName = fdInput then
          Result := nrmLinear
        else if KindName = fdOutput then
          Result := nrmLinearOut;
    1 : Result := nrmSigmoid;
    2 : Result := nrmAuto;
    3 : Result := nrmNone;
  end;
end;

procedure TNeuroField.CalcMid;
var
  i: integer;
begin
  FValueMid := 0;
  for i := Low(FDataIn) to High(FDataIn) do
    FValueMid := FValueMid + FDataIn[i];
  FValueMid := FValueMid/(High(FDataIn) + 1);
end;

procedure TNeuroField.CalcDispersion;
var
  i: integer;
begin
  if High(FDataIn) > 1 then
  begin
    FDispersion := 0;
    for i := Low(FDataIn) to High(FDataIn) do
      FDispersion := FDispersion + sqr(FDataIn[i] - ValueMid);
    FDispersion := sqrt(FDispersion/High(FDataIn));
  end
  else
    FDispersion := 0;
end;

(*procedure TNeuroField.DeNormalize;
var
  i: integer;
  xTmp: double;
begin
  case NormTypeName of
    nrmLinear: for i := Low(FDataIn) to High(FDataIn) do
```

```
                FDataIn[i] := (FDataIn[i] + 1)*(FValueMax - FValueMin)/2 +
FValueMin;
    nrmLinearOut: for i := Low(FDataIn) to High(FDataIn) do
                FDataIn[i] := FDataIn[i]*(FValueMax - FValueMin) + FValueMin;
    nrmSigmoid: for i := Low(FDataIn) to High(FDataIn) do
                FDataIn[i] := - Ln(1/FDataIn[i] - 1)/Alpha;
  end;
end;*)

procedure TNeuroField.FindMinMax;
var
  i: integer;
begin
  FValueMax:= FDataIn[0];
  FValueMin:= FDataIn[0];
  for i := 1 to High(FDataIn) do
  begin
    if FValueMin > FDataIn[i] then
      FValueMin := FDataIn[i];
    if FValueMax < FDataIn[i] then
      FValueMax := FDataIn[i]
  end;
end;

procedure TNeuroField.Normalize;
var
  i: integer;
  xTmp: double;
begin
  case NormTypeName of
    nrmAuto:  begin
                CalcMid;
                CalcDispersion;
                for i := Low(FDataIn) to High(FDataIn) do
                begin
                  xTmp := (FDataIn[i] - FValueMid)/FDispersion;
                  FDataIn[i] := 1/(1 + exp(-xTmp));
                end;
              end;
    nrmLinear: for i := Low(FDataIn) to High(FDataIn) do
               FDataIn[i] := 2*(FDataIn[i] - FValueMin)/(FValueMax - FValueMin)-
1;
    nrmLinearOut: for i := Low(FDataIn) to High(FDataIn) do
              FDataIn[i] := (FDataIn[i] - FValueMin)/(FValueMax - FValueMin);
    nrmSigmoid: for i := Low(FDataIn) to High(FDataIn) do
              FDataIn[i] := 1/( 1 + exp(-Alpha * FDataIn[i]));
  end;
end;

procedure TNeuroField.SetNormType(Value: byte);
begin
  if (Value < 0) or (Value > 3) then
    raise EFieldNormError.CreateFmt(SFieldNorm, [Value])
  else
    FNormType := Value;
end;

procedure TNeuroField.SetKind(Value: byte);
begin
  if (Value < 0) or (Value > 2) then
    raise Exception.CreateFmt(SFieldKind, [Value])
  else
    FKind := Value;
end;

procedure TNeuroField.SetDataIn(Index: integer; Value: double);
begin
  FDataIn[Index] := Value;
end;
```

```
procedure TNeuroField.SetDataInCount(Value: integer);
begin
  SetLength(FDataIn, Value)
end;

// Клас TNDataSource

function TNDataSource.IsHeaderChar(AValue: char): boolean;
begin
  if (AValue in Letters) or (AValue in Capitals) or (AValue in DigitChars) then
    Result := True
  else
    Result := False;
end;

procedure TNDataSource.ExtractValues(const AVector:TVectorFloat; AHeader:
string);
var
  s: string;
  i, xCurPos: integer;
begin
  i := 0;
  AHeader := Trim(AHeader);
  xCurPos := Pos(SpaceChar, AHeader);
  try
    while xCurPos > 0 do
    begin
      s := Copy(AHeader, 1, xCurPos - 1);
      AVector[i] := StrToFloat(s);
      Inc(i);
      Delete(AHeader, 1, xCurPos - 1);
      AHeader := Trim(AHeader);
      xCurPos := Pos(SpaceChar, AHeader);
    end;
    s := AHeader;
    AVector[i] := StrToFloat(s);
  except
    on EConvertError do
      EConvertError.CreateFmt(SCannotBeNumber, [s])
  end;
end;

procedure TNDataSource.ExtractHeaders(const AFields: TNeuroFields; AHeader:
string);
var
  s: string;
  xFieldCount, j, xCurPos: integer;
begin
  // виділяє заголовки з файлу
  xFieldCount := 0;
  AHeader := Trim(AHeader);
  xCurPos := Pos(SpaceChar, AHeader);
  while xCurPos > 0 do
  begin
    s := Copy(AHeader, 1, xCurPos - 1);
    AFields[xFieldCount].FName := '';
    for j := 1 to Length(s) do
      if isHeaderChar(s[j]) then
        AFields[xFieldCount].FName := AFields[xFieldCount].FName + s[j];
    Inc(xFieldCount);
    Delete(AHeader, 1, xCurPos - 1);
    AHeader := Trim(AHeader);
    xCurPos := Pos(SpaceChar, AHeader);
  end;
  AFields[xFieldCount].FName := '';
  for j := 1 to Length(AHeader) do
    if isHeaderChar(AHeader[j]) then
      AFields[xFieldCount].FName := AFields[xFieldCount].FName + AHeader[j];
```

```
  // повертає кількість полів
end;

function TNDataSource.FieldCount(AHeader: string): integer;
var
  xFieldCount, xCurPos: integer;
begin
  // виділяє заголовки з файлу
  xFieldCount := 0;
  AHeader := Trim(AHeader);
  xCurPos := Pos(SpaceChar, AHeader);
  while xCurPos > 0 do
  begin
    Inc(xFieldCount);
    Delete(AHeader, 1, xCurPos - 1);
    AHeader := Trim(AHeader);
    xCurPos := Pos(SpaceChar, AHeader);
  end;
  // повертає кількість полів
  Result := xFieldCount + 1;
end;


end.
```

Файл about.pas - довідка про програму

```
unit about;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, ExtCtrls;

type
  Tfrm_about = class(TForm)
    Image1: TImage;
    Memo1: TMemo;
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frm_about: Tfrm_about;

implementation
{$R *.dfm}

procedure Tfrm_about.Button1Click(Sender: TObject);
begin
frm_about.Close;
end;

procedure Tfrm_about.FormCreate(Sender: TObject);
begin
Memo1.Clear;
Memo1.Lines.Add('МАГІСТЕРСЬКА РОБОТА');
Memo1.Lines.Add('');
Memo1.Lines.Add('на  тему:');
Memo1.Lines.Add('');
Memo1.Lines.Add('Дослідження та програмна реалізація системи CAN-мережі на
основі технології CSDN');
Memo1.Lines.Add('');
Memo1.Lines.Add('');
Memo1.Lines.Add('');
Memo1.Lines.Add('Керівник: Доренський О.П.');
Memo1.Lines.Add('');
Memo1.Lines.Add('Розробив: студент Олійник Артур Олегович ');
Memo1.Lines.Add('                  гр. КІ-21М-1,4');
Memo1.Lines.Add('');
Memo1.Lines.Add('');
Memo1.Lines.Add('м. Кропивницький 2022');
end;
end.
```