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
“____” _____ 2025 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи мережевого антивірусного комплексу дата-центру”

Виконав здобувач вищої освіти
ІІ курсу, групи КІ-24М
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Грищенко М.О.
«____» _____ 2025 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Буравченко К.О.
«____» _____ 2025 р.
Рецензент _____
_____

м. Кропивницький

АНОТАЦІЯ

Грищенко М.О. Дослідження та програмна реалізація системи мережевого антивірусного комплексу дата-центру. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2025.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи мережевого антивірусного комплексу дата-центру.

Метою розробки є дослідження та програмна реалізація системи мережевого антивірусного комплексу дата-центру.

Об'єктом дослідження є процес мережевого антивірусного комплексу дата-центру.

Предметом дослідження є методи мережевого антивірусного комплексу дата-центру.

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

Результат роботи – програмна реалізація системи мережевого антивірусного комплексу дата-центру.

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 з ОС Windows 10/11.

Програму розроблено в середовищі Python.

Ключові слова: комп'ютерна інженерія, антивірусний комплекс, дата-центру

ABSTRACT

Hryshchenko M.O. Research and software implementation of the network antivirus complex data center system. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2025.

In this final qualification work for the second (master's) level of higher education, software has been developed, which is intended for the network antivirus complex data center system.

The purpose of the development is the research and software implementation of the network antivirus complex data center system.

The object of the research is the process of the network antivirus complex data center.

The subject of the research is the methods of the network antivirus complex data center.

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

The result of the work is the software implementation of the network antivirus complex data center system.

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 the software are provided.

The program can be used on a PC with Windows 10/11 OS.

The program was developed in the Python environment.

Keywords: computer engineering, antivirus complex, data center

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

КМ	–	комп’ютерна мережа
КСАЗ	–	комплексна система антивірусного захисту
МЕ	–	міжмережний екран
ПЗ	–	програмне забезпечення
ПК	–	персональний комп’ютер
ACL	–	Access Control List
FTP	–	File Transfer Protocol
http	–	HyperText Transfer Protocol
POP3	–	Post Office Protocol Version 3
SMTP	–	Simple Mail Transfer Protocol
VLAN	–	Virtual Local Area Network

ВСТУП

Актуальність теми. Масове використання комп'ютерів, а також швидкий розвиток комп'ютерних мереж (зокрема мережі Інтернет) сприяли появі й поширенню шкідливих програм – комп'ютерних вірусів. Віруси часом роблять роботу на комп'ютері неможливою. Вони, залежно від ситуації, можуть завдати значної шкоди як інформації, так і самому комп'ютеру.

Незважаючи на поширену думку, центри обробки даних не є «куленепробивними» від хакерських атак. Ви можете бути в більшій небезпеці, якщо володієте сервером, оскільки кіберзлочинці нападуть на вас з усією силою. На ринку є чимало антивірусних продуктів, але не всі вони підходять для центру обробки даних. Гарна новина полягає в тому, що з найкращим антивірусом ви можете перестати турбуватися про будь-які зовнішні загрози.

Іншими словами, центр обробки даних – це обладнання, яке зберігає інформацію (для бізнесу, урядів тощо). Крім того, той факт, що до цих даних можна отримати доступ з будь-якого куточка світу, перетворює центри обробки даних на корисний інструмент. Усі уповноважені сторони можуть отримати ту саму інформацію, фактично не відвідуючи місцезнаходження центру обробки даних. Однак, оскільки інформація вільно передається між різними комп’ютерами, це полегшує злочинцям її отримання.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи мережевого антивірусного комплексу дата-центру.

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

– Огляд існуючих систем мережевого антивірусного комплексу дата-центру.

– Дослідження системи мережевого антивірусного комплексу дата-центру.

– Програмна реалізація системи мережевого антивірусного комплексу дата-центру.

Об'єктом дослідження є процес мережевого антивірусного комплексу дата-центру.

Предметом дослідження є методи мережевого антивірусного комплексу дата-центру.

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

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

– Удосконалено метод мережевого антивірусного комплексу дата-центру.

– Розроблено вітчизняний продукт мережевого антивірусного комплексу дата-центру, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі мережевого антивірусного комплексу дата-центру.

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 LVII Науково-технічній конференції здобувачів вищої освіти LV науково-технічної конференції «Наука в ЦНТУ: основні досягнення та перспективи розвитку» (2025 р.), основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №15.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи мережевого антивірусного комплексу дата-центру, 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

Програма, усередині якої перебуває вірус, називається зараженою (інфікованою).

Для комп'ютерних вірусів характерні певні стадії існування:

– пасивна стадія – не вживає ніяких дій;

– стадія розмноження – вірус намагається створити якнайбільше своїх копій;

– активна стадія – вірус переходить до виконання руйнівних дій на локальному комп'ютері або в комп'ютерній мережі.

Проникнувши в комп'ютерну систему, вірус може обмежитися необразливими візуальними або звуковими ефектами, але може викликати втрату або перекручування даних, витік (крадіжку) особистої або конфіденційної інформації.

Ознаки присутності вірусу на комп'ютері:

– безпричинна з роботи комп'ютера;

– неможливість завантаження системних або прикладних програм, їхнє нестандартне функціонування;

– невмотивована зміна розмірів файлів, дати й часу їхнього створення, поява нових файлів (особливо – з незрозумілими іменами);

– зменшення обсягу доступної користувачеві оперативної пам'яті;

– мимовільні зміни файлової структури дисків;

– помітне збільшення кількості збоїв у роботі комп'ютера, у тому числі його мимовільні перезавантаження й ін.

Слід зазначити, що перераховані ознаки не обов'язково викликаються присутністю вірусу, вони можуть бути наслідком і іншими причинами.

Для захисту від вірусів можна використовувати загальні засоби захисту інформації, такі як дублювання інформації, створення резервних копій, розмежування доступу. Розмежування доступу дозволяє не тільки запобігти несанкціонованому використанню інформації, але й захистити дані від шкідливих дій вірусів, за рахунок обмеження доступу до файлів. Одним із самих зручних методів захисту від комп'ютерних вірусів є використання спеціалізованих програм.

1.2 Область застосування

Областю застосування програмного забезпечення, що розробляється, є дати-центри.

Цінну інформацію можна продати конкуруючим сторонам або використати для шантажу власників бізнесу. Шифрування, захист паролем, безпека протоколів та брандмауери є дуже ефективними. Однак жоден центр обробки даних не буде безпечним без належного антивіруса. Пошкоджені файли компанії можуть бути втрачені назавжди, а складне шкідливе програмне забезпечення здатне знищити великі фрагменти даних (якщо немає антивіруса для його видалення).

Коли йдеться про центри обробки даних, є три основні «точки», які потребують захисту. Це робочий стіл, сервер і шлюз. Давайте розглянемо кожну з них і подивимося, як антивірусне рішення може допомогти вам захистити критично важливу інформацію. Кожен пункт, який ми щойно згадали, має свої переваги та недоліки – ми також поговоримо про них.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи мережевого антивірусного комплексу дата-центру, 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

Norton 360

Переваги:

– Надійно висока продуктивність: Norton продовжує виділятися як у сторонньому, так і в власному тестуванні. Його виявлення шкідливого програмного забезпечення є однаковим у всіх операційних системах, і це один з небагатьох антивірусних програм, який забезпечує чудовий захист, не виснажуючи ресурси пристрою. Під час оцінювання мобільних пристроїв та настільних комп'ютерів сканування було швидким і не впливало на фонову активність. Продуктивність залишається стабільною, навіть з такими інструментами, як керування паролями та VPN, що працюють.

– Найкраще у своєму класі покриття: Norton включає один із найповніших пакетів безпеки на базовому рівні. Genie Scam Protection позначає текстові шахрайства, шахрайство в соціальних мережах та спроби фішингу в режимі реального часу. Norton Password Manager безкоштовний для всіх планів, а вищі рівні розблоковують моніторинг даркнету, батьківський контроль та Norton Secure VPN – все з однієї панелі керування. Ці функції вбудовані в кожен план, що робить Norton однією з небагатьох антивірусних платформ, де «все в одному» дійсно працює.

– Захист від шахрайства Genie: Цей інструмент на базі штучного інтелекту аналізує посилання в електронних листах, текстових повідомленнях та повідомленнях у соціальних мережах, щоб виявляти шахрайство, перш ніж ви з ним взаємодіятимете. Genie щодня навчається на тисячах нових загроз і надає

миттєві оцінки ризиків простою мовою. Неважливо, чи ви добре знаєтеся на технологіях, чи ні, адже він перетворює виявлення фішингу на щось, про що вам не доведеться думати двічі.

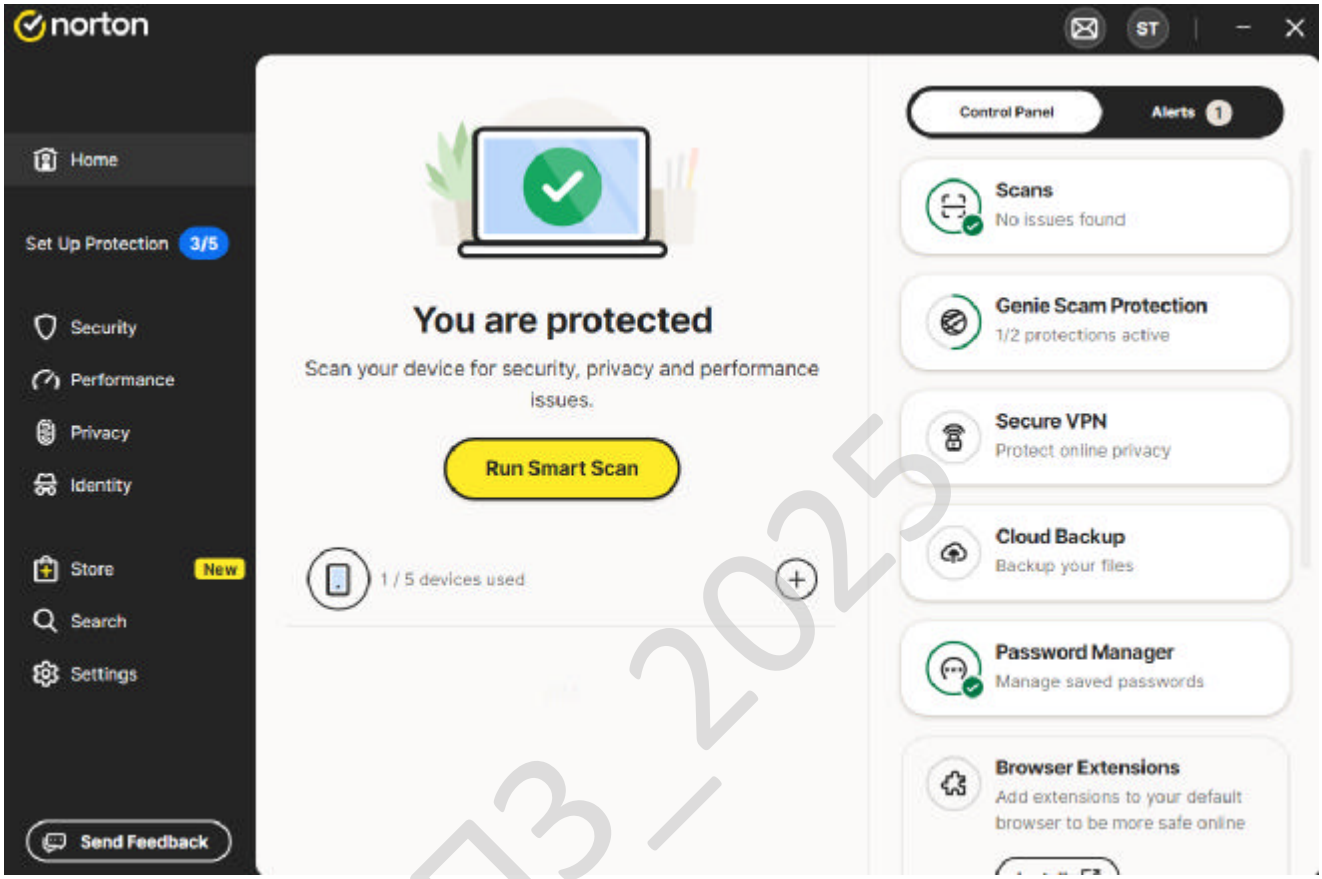


Рисунок 2.1 – Інтерфейс користувача

– Найкращий антивірус для сімей: Norton Family дозволяє батькам контролювати доступ до Інтернету, пошукові запити, час, проведений перед екраном, та використання програм. На відміну від інших інструментів батьківського контролю, він працює на всіх пристроях і забезпечує легку для розуміння звітність. Його інтеграція з ширшим пакетом Norton робить його особливо зручним для зайнятих сімей.

– Найкращий антивірус із захистом паролем: Norton Password Manager безпечно синхронізується між браузерами та пристроями, пропонуючи автозаповнення, аналіз сховища та навіть генератор паролів. Це чудове

доповнення до антивірусного захисту для тих, хто ще не використовує окремий менеджер паролів, і на відміну від конкурентів, це не платний додаток.

Недоліки:

– Найкращий план має обмеження щодо кількості пристроїв: хоча Norton підтримує кілька пристроїв, справжнє необмежене покриття недоступне навіть у найдорожчому плані: Norton 360 з LifeLock Select Plus. McAfee пропонує необмежену кількість пристроїв на всіх рівнях, що дає йому явну перевагу для великих домогосподарств або користувачів, які керують кількома системами. Підхід Norton тут здається надмірно обмежувальним, особливо враховуючи вартість оновлення.

– Мобільні додатки мають обмеження: Norton Mobile Security непогано справляється з фішинговими сповіщеннями та сповіщеннями про конфіденційність додатків, але не зупиняє завантаження шкідливих файлів у режимі реального часу. Це означає, що користувачам Android, зокрема, потрібно бути особливо обережними зі сторонніми додатками та APK, оскільки Norton може позначити загрозу лише після того, як її вже завантажено.

Aura Antivirus

Переваги:

Надійний захист особистих даних: Aura вирізняється вбудованим антивірусом у ширший пакет захисту особистих даних. Окрім виявлення та видалення шкідливого програмного забезпечення, вона також забезпечує моніторинг кредитної історії, сповіщення про шахрайство, сканування на випадок порушення безпеки облікових записів та інструменти для автоматичних запитів на видалення брокерів даних. Хоча Aura не пропонує брандмауера чи захисту веб-камери, вона робить акцент на запобіганні довгостроковому шахрайству з особистими даними за допомогою моніторингу кредитної історії.

Повний пакет безпеки: Окрім надійного захисту від крадіжки особистих даних, Aura Antivirus включає всі основні функції антивірусної безпеки, включаючи захист у режимі реального часу (у Windows), сканування на наявність

програм-вимагачів, вбудований VPN, безпечний перегляд, керування паролями та можливості захисту від відстеження через розширення браузера. Хоча жодна з цих функцій окремо не є найкращою у своєму класі, вони вражаюче добре інтегровані. Сканування антивірусом швидке, а інтерфейс простий у навігації.

Простота використання: Завдяки простій панелі інструментів, сповіщенням електронною поштою та інтуїтивно зрозумілим опціям спільного доступу для всієї родини, Aura є одним із найпростіших інструментів, який можна встановити та забути. Він навіть надсилає щотижневі зведення виявлених загроз або змін у налаштуваннях захисту, що робить його ідеальним для користувачів, які віддають перевагу підходу «налаштував і забув». Захист у режимі реального часу активно поміщає загрози в карантин під час тестування, а щотижневі зведення інформують користувачів, не вимагаючи постійного нагляду.

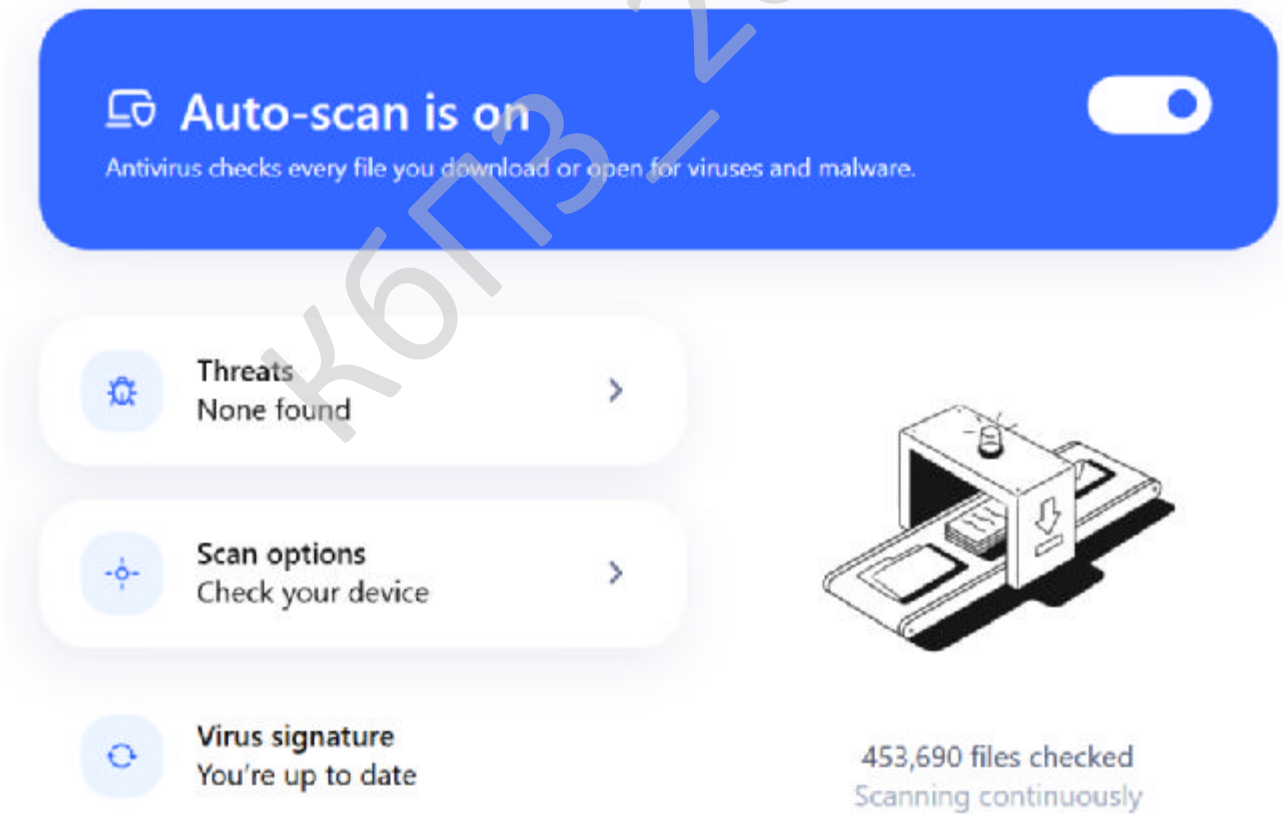


Рисунок 2.2 – Інтерфейс користувача Aura

– Найкраще для моніторингу кредитної історії: інтеграція Aura з бюро кредитних історій та системами оповіщення є однією з найкращих у своєму класі. Ви можете контролювати фінансові рахунки, отримувати сповіщення про шахрайство, а також блокувати або розблоковувати свою кредитну історію з одного й того ж інтерфейсу, що робить її унікальним гібридом антивіруса та захисту від крадіжки особистих даних.

Недоліки:

– Відсутність стороннього тестування: Aura не проходила тестування в AV-TEST, AV-Comparatives або SE Labs, що ускладнює порівняння її продуктивності. Вона пройшла наш внутрішній тест EICAR, який перевіряє наявність шкідливого програмного забезпечення та фішингових можливостей, але відсутність прозорості від сторонніх розробників може бути червоним прапорцем для деяких покупців.

– Хибнопозитивні результати під час тестування: Сертифіковані сторонні лабораторії безпеки тестують антивірусне програмне забезпечення на наявність хибнопозитивних результатів або позначених файлів чи програм, які насправді не є шкідливими програмами. Під час нашого практичного тестування ми помітили, що Aura помилково позначає наше завантаження Steam як троян. Хоча ми цінуємо захист у режимі реального часу, нам не подобається, що Aura може бути надмірно ретельною.

TotalAV

Переваги:

– Видатний захист у режимі реального часу: TotalAV послідовно блокує загрози, перш ніж вони досягнуть вашого пристрою, завдяки аналізу хмарних даних у режимі реального часу та виявленню на основі сигнатур. Він добре працює на різних операційних системах та ефективно виявляє спроби фішингу, шкідливе програмне забезпечення для криптоджекінгу та загрози на основі браузера під час тестування.

– Total Adblock: Total Adblock – це не просто блокувальник реклами, він

запобігає завантаженню шкідливих скриптів, трекерів та спливаючих вікон. Він зменшує час завантаження браузера та зменшує ризик клікбейтів, що особливо корисно на повільніших або старіших пристроях. Як і Surfshark CleanWeb, це один із найбільш адаптивних блокувальників реклами для блокування відеореклами на YouTube.

– Розширення Total Browser: Розроблене з урахуванням безпеки, це розширення фільтрує шкідливі веб-сайти та забезпечує безпечне HTTPS-з'єднання, де це можливо. WebShield, доступний для Chrome, додає додатковий рівень захисту, блокуючи підозрілі завантаження та надаючи попередження в режимі реального часу про URL-адреси з високим рівнем ризику.

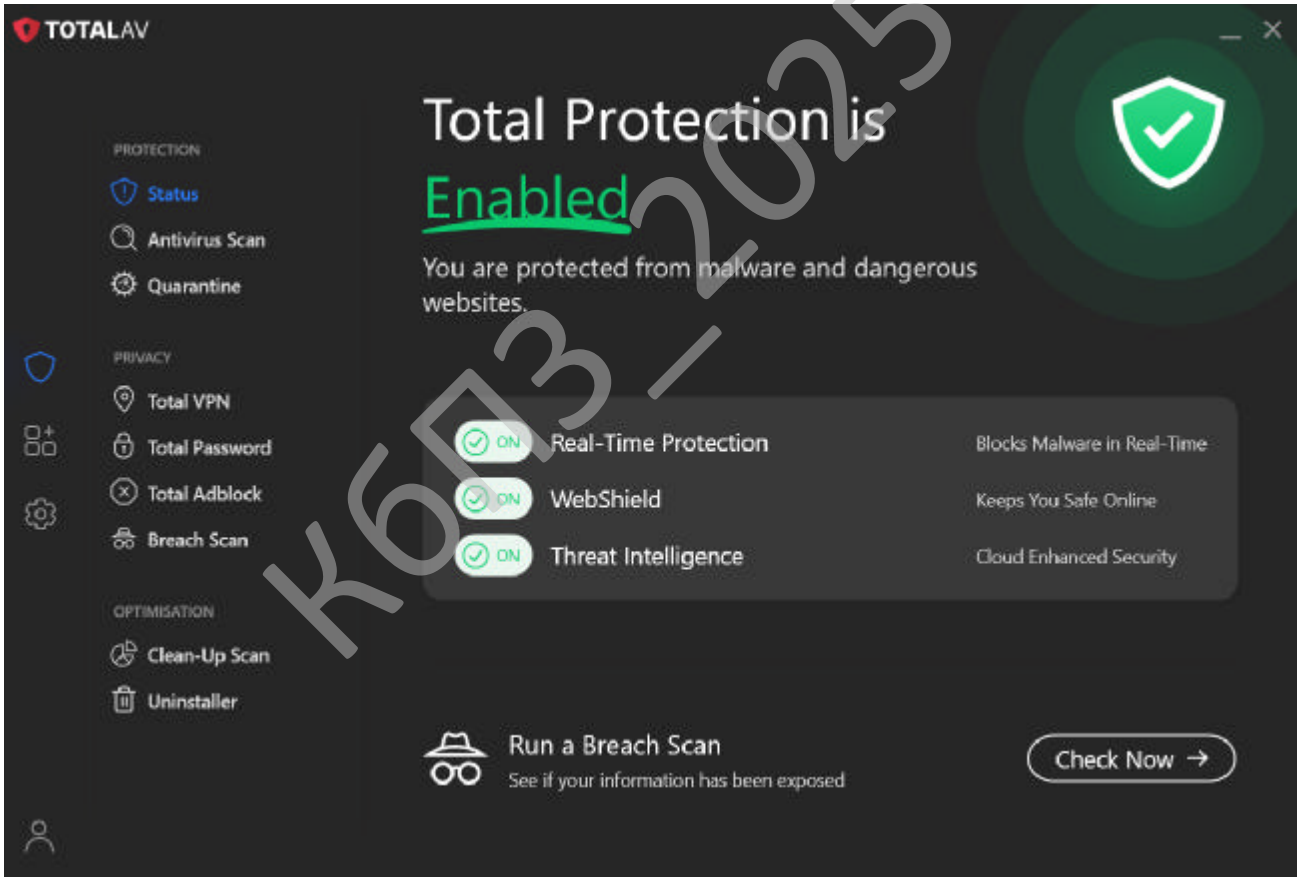


Рисунок 2.3 – Панель керування захистом TotalAV

– Найкращий пакет блокувальників реклами: пакет безпеки TotalAV постачається з Total Adblock, що пропонує більший контроль над

конфіденційністю в Інтернеті, ніж більшість антивірусних конкурентів. Інтеграція блокувальника реклами є безперебійною та не потребує жодної технічної конфігурації; просто встановіть його та безпечно переглядайте веб-сторінки.

– Найкращі інструменти для оптимізації продуктивності: Завдяки таким функціям, як видалення непотрібних файлів, менеджер автозавантаження та видалення програм, TotalAV чудово підходить для підтримки швидкості системи. Ці інструменти корисні одразу після встановлення та пропонують реальну цінність, що перевищує просто захист від вірусів.

Недоліки:

– Бракує спеціального брандмауера: Хоча TotalAV охоплює більшість основ, він не містить повноцінного інструменту брандмауера. Це обмежує його можливості моніторингу або блокування вхідних з'єднань на мережевому рівні. Це не є вирішальним фактором, але варто врахувати, якщо ви особливо стурбовані загрозами на мережевому рівні.

McAfee

Переваги:

– Необмежений захист пристроїв: McAfee – єдиний великий постачальник антивірусного програмного забезпечення, який пропонує необмежений захист пристроїв у всіх преміум-планах. Це кардинально змінює правила гри для домогосподарств з кількома телефонами, планшетами, ноутбуками та настільними комп'ютерами. Це також зменшує плутанину, пов'язану з вибором підписок на конкретні пристрої; один план покриває все.

– Легкий захист Windows: Незважаючи на широкий функціонал, McAfee не навантажує системні ресурси, навіть завдяки вбудованому брандмауеру. Він отримав найвищі оцінки за продуктивність у середовищах Windows, працюючи безперебійно в режимі реального часу. Незалежно від того, чи скануєте ви великі файли, чи встановлюєте оновлення, робота відбувається безперебійно.

– McAfee WebAdvisor: Це розширення для браузера працює в Chrome,

Edge та Firefox, блокуючи спроби фішингу та небезпечні веб-сайти. Воно також надає рейтинги безпеки в результатах пошуку та сповіщення про потенційно підозрілі завантаження. Інтеграція є безперебійною, надаючи сповіщення в режимі реального часу, не порушуючи ваш робочий процес.

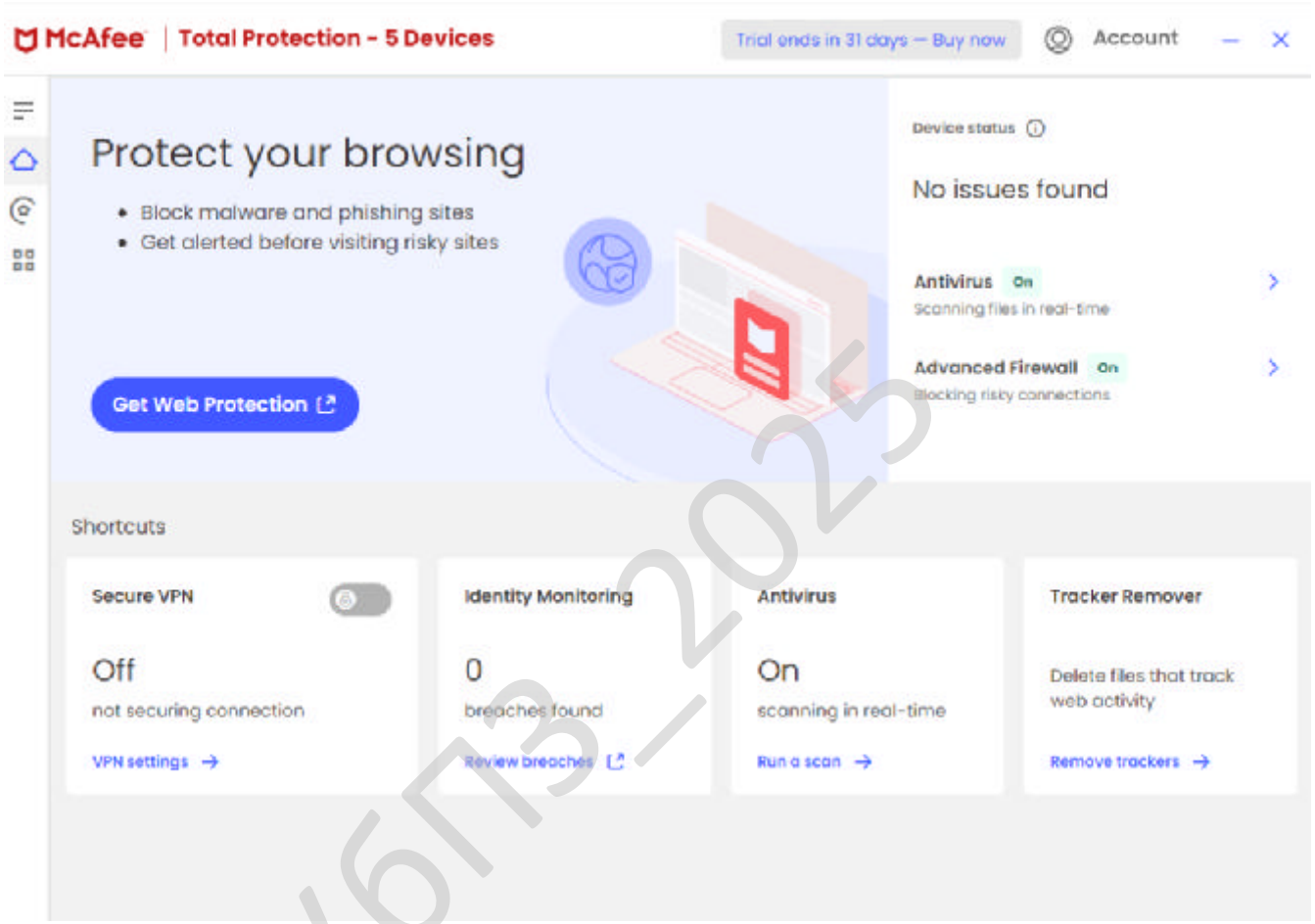


Рисунок 2.4 – Інтерфейс користувача McAfee

– Найкращий захист від програм-вимагачів: Усі плани McAfee включають можливості відкату програм-вимагачів та покриття до 25 000 доларів США на випадок атак програм-вимагачів. У поєднанні з аналізом хмарних даних у режимі реального часу та локальним моніторингом файлів, це один із найбезпечніших інструментів для бізнес-користувачів або тих, хто часто завантажує файли.

Недоліки:

– Не включає інструменти оптимізації: McAfee продає окремий додаток

PC Optimizer, який не входить до складу антивірусного пакету. Це означає, що функції очищення системи, такі як видалення непотрібних файлів та оновлення драйверів, не включені до базового пакету, що розчаровує порівняно з іншими продуктами.

Антивірус Surfshark

Переваги:

– Доступність: Передплати Surfshark One пропонують комплексний пакет безпеки з простою моделлю ціноутворення, яка конкурує з більшістю автономних антивірусних інструментів. Ви точно знаєте, за що платите і що отримуєте. Ще краще те, що у них немає прихованих додаткових витрат на поновлення або заплутаних обмежень на кількість пристроїв. Це привабливий бюджетний вибір для тих, хто хоче охопити всі свої пристрої однією підпискою.

– Пакет безпеки в комплекті: З пакетом Surfshark One ви отримуєте антивірусний захист, VPN, блокування реклами, сповіщення про порушення та навіть інструменти для створення псевдонімів електронної пошти. Хоча Surfshark Antivirus є відносно новим на ринку, Surfshark об'єднує інструменти, які зазвичай розподілені по кількох сервісах, таких як Surfshark CleanWeb та Incogni. Це особливо корисно для користувачів, стурбованих витоком даних та їх відстеженням.

– Тестування Windows сторонніми розробниками: Хоча деякі нові антивірусні інструменти проходять незалежне тестування, Surfshark пройшов оцінювання AV-TEST і отримав найвищі оцінки за захист від шкідливих програм та зручність використання. Така прозорість вселяє довіру, особливо для продукту, який починався як VPN.

– Найкращий пакет VPN: Surfshark Antivirus постачається з одним із найкраще оцінених VPN на ринку: Surfshark VPN. Це поєднання антивіруса та VPN забезпечує захист зашифрованого трафіку, блокування трекерів та безпечний перегляд веб-сторінок в одному пакеті, що робить його особливо корисним для користувачів публічних Wi-Fi та віддалених працівників.

– Найкраще для ігор: Surfshark – чудовий вибір для геймерів, оскільки він не потребує багато програмного забезпечення та займає мало місця на системі. Захист у режимі реального часу не впливає на продуктивність, а вбудований CleanWeb блокує шкідливу рекламу та трекери, які можуть уповільнювати ігровий трафік.

Недоліки:

– Немає окремого антивіруса: Surfshark не продає свій антивірусний інструмент окремо. Щоб отримати антивірусний захист, ви повинні підписатися на повний пакет Surfshark One, який включає VPN та інші інструменти захисту ідентифікаційних даних і веб-сайтів. Це цінний пакет безпеки, але користувачі, які шукають базовий антивірусний інструмент, можуть вважати додаткові функції зайвими.

Bitdefender

Переваги:

– Комплексний захист від шахрайства: Bitdefender серйозно ставиться до захисту від фішингу, пропонуючи кілька рівнів захисту, які виходять далеко за рамки простих фільтрів електронної пошти. Він сканує веб-трафік у режимі реального часу, аналізує поведінку програм на наявність шкідливих намірів та використовує штучний інтелект для виявлення шахрайських шаблонів у повідомленнях та на веб-сайтах. Система проактивно блокує ризиковані з'єднання до того, як ви клацнете, а не після.

– Scam Copilot: Ця функція діє як особистий помічник для виявлення шахрайства. Коли надходять підозрілі текстові повідомлення або текстові повідомлення, Scam Copilot аналізує небезпечний контент і пояснює, як залишатися в безпеці. Це особливо корисно для менш технічно підкованих користувачів, пропонуючи поради без залякування.

– Надійно висока продуктивність: Bitdefender – це не лише безпека, він ще й швидкий. Він тихо працює у фоновому режимі та обробляє все: від повного сканування диска до оцінки вразливостей з мінімальним впливом на

продуктивність системи. Його система Advanced Threat Defense відстежує поведінкові аномалії, позначаючи загрози перед їх виконанням, що додає другий рівень захисту в режимі реального часу.

– Найкращий антивірус для Windows: Панель інструментів Bitdefender добре інтегрована з екосистемою безпеки Windows. Такі функції, як захист веб-камери, безпечний онлайн-банкінг та захист від експлойтів нульового дня, роблять його найкращим варіантом для користувачів Windows, які шукають більш комплексний захист, ніж базовий антивірус.

Недоліки:

– Не розроблено для macOS: вся потужність Bitdefender зарезервована для Windows. Хоча технічно він працює на macOS, багато його найкорисніших функцій, таких як Scam Copilot та налаштування брандмауера, або відсутні, або обмежені. Це робить його кращим вибором для користувачів ПК, ніж для домогосподарств Apple.

MacKeeper

Переваги:

– Рідний антивірус для macOS: Більшість антивірусного програмного забезпечення розроблено для ПК, тому антивірусна програма, яка пропонує хороший рідкісний захист для macOS, є рідкістю. На відміну від інструментів оптимізації продуктивності, таких як CleanMyMac, MacKeeper може похвалитися повним пакетом онлайн-захисту для користувачів Mac. Преміум-покриття включає VPN, моніторинг витоків даних, інструменти оптимізації, технологію блокування реклами та захист від шкідливих програм у режимі реального часу для Mac.

– Відмінна продуктивність macOS: MacKeeper досяг високих результатів тестування зручності використання, продуктивності та захисту під час стороннього тестування з використанням macOS Monterey. Фактично, MacKeeper перевершує середні показники по галузі в кількох тестах продуктивності та не виявив жодного хибнопозитивних результатів. MacKeeper щорічно проходить

AV-TEST з березня 2021 року, демонструючи стабільно високу продуктивність.

– Інструменти оптимізації Mac: MacKeeper містить різноманітні інструменти оптимізації пристроїв, зокрема Smart Uninstaller, Duplicates Finder, Safe Cleanup, Adware Cleaner, Login Items та StopAd. Від видалення непотрібних програм до блокування реклами, яка уповільнює завантаження сторінок, інструменти MacKeeper працюють разом для покращення загальної продуктивності пристрою.

Недоліки:

– Відсутність захисту від фішингу: MacKeeper пропонує хороший пакет безпеки, але йому бракує вбудованого захисту від фішингу. Оскільки він не має спеціальної технології захисту від фішингу, MacKeeper не зміг виявити фішинговий файл під час нашого внутрішнього тестування. З огляду на те, що так багато зловмисників намагаються обманом викрасти вашу особисту інформацію, ми вважаємо, що захист від фішингу є обов'язковим.

Переваги:

– Простота використання: Сила Avast полягає в його доступності. Кожна функція чітко позначена, інтерфейс користувача простий, і практично немає кривої навчання. Для користувачів, які вперше знайомляться з антивірусом або не знайомі з надмірно технічними інструментами, це чудова відправна точка. Від встановлення до щоденного сканування Avast підтримує інтуїтивно зрозумілий користувацький досвід. Крім того, його інтерфейс на macOS відображає чіткий макет версії для Windows, що дозволяє легко перемикати функції, перевіряти історію сканування та налаштовувати параметри.

– Включає брандмауер для Windows та macOS: Хоча багато антивірусних програм не пропонують розширеного захисту мережі за допомогою брандмауера, Avast One включає повнофункціональний брандмауер, який допомагає контролювати вхідний та вихідний трафік. Він позначає підозрілі з'єднання *без* необхідності ручного налаштування. Це особливо цінно для користувачів Mac, яким потрібен захист рівня ПК без сторонніх інструментів.

– Висока продуктивність: Під час сторонніх та внутрішніх тестів Avast постійно доводив свою надійність. Він отримав високі оцінки за продуктивність, зручність використання та захист у різних операційних системах, що робить його ідеальним для будь-якого користувача.

Недоліки:

– Не пропонує повного пакету безпеки: у планах Avast відсутні кілька ключових функцій, які ви знайдете в інших місцях, таких як менеджер паролів, батьківський контроль та моніторинг особистості. Якщо ви шукаєте універсальне рішення, Avast може призвести до необхідності налаштовувати додаткові інструменти.

AVG

Переваги:

– Продуктивність: AVG постійно посідає перше місце в незалежних тестах продуктивності, пропонуючи один з найлегших антивірусів на ринку. Він не уповільнює завантаження та не витрачає фонові ресурси навіть під час повного сканування системи. Він особливо добре підходить для користувачів, які запускають ресурсоємні програми (такі як ігри або програмне забезпечення для редагування) і не можуть дозволити собі перерв.

– Легкий захист: AVG посідає одне місце позаду McAfee за оцінкою продуктивності AV-Comparatives, але в практичному використанні AVG перевершив його. Він працював чистіше та тихіше, сканування завершувалося швидше, піки ресурсів були ледь помітними, а інтерфейс не здавався роздутим. На відміну від інструментів, які переривають вас постійними підказками або затримками під час багатозадачності, AVG підтримував повний захист у режимі реального часу, не відволікаючи уваги. Цей баланс між високими результатами тестів та фактичною простотою використання – це те, що відрізняє його від інших під час щоденного використання.

– Швидке сканування: AVG пропонує різноманітні типи сканування, не лише швидке чи повне. Налаштування сканування та вибір папок додають

гнучкості. Під час тестування повне сканування працювало помітно швидше, ніж у аналогічних інструментів, водночас виявляючи приховані загрози.

Недоліки:

– Набридлива реклама оновлень: Навіть на платних планах AVG пропонує «оновлення» для функцій оптимізації, таких як видалення непотрібних файлів та оновлення драйверів, деякі з яких не входять до вашої поточної підписки. Це не реклама сторонніх розробників, але інтерфейс все одно здається захараженим для користувачів, які просто хочуть чисте програмне забезпечення без реклами.

ESET

Переваги:

– Розширений захист: ESET блокує стандартне шкідливе програмне забезпечення, але також спеціалізується на захисті від складних атак, таких як безфайлове шкідливе програмне забезпечення та експлойти нульового дня. Його механізм моніторингу поведінки створений для професіоналів, виявляючи нерегулярну активність до того, як вона стане порушенням.

– Розширені функції: ESET містить функції, які рідко зустрічаються у споживчому антивірусному програмному забезпеченні, такі як зашифроване банківське середовище, аналіз мережевого трафіку та захист розумного дому. Ці інструменти забезпечують безпеку на підключених пристроях і маршрутизаторах, що особливо корисно для користувачів із розширеними домашніми налаштуваннями.

– Безпека корпоративного рівня: Завдяки підтримці ESET Inspect (XDR), повного шифрування диска та багатофакторної автентифікації, ESET має унікальні можливості для обслуговування віддалених працівників або малого бізнесу. Мало які антивірусні засоби так добре заповнюють розрив між споживачем та підприємством, як ESET.

Недоліки:

– Страшно для початківців: інтерфейс не завжди інтуїтивно зрозумілий, а термінологія може бути заплутаною для користувачів, які не мають технічних

знань. Такі налаштування, як «оцінка правил HIPS» або «Виключення обробника виявлення», вимагають глибшого розуміння принципів кібербезпеки.

Вибір найкращого антивірусного програмного забезпечення може бути складним завданням. Роблячи свій вибір, враховуйте ціну, продуктивність та зручність використання.

– Ціноутворення: Антивірусне програмне забезпечення пропонується в широкому діапазоні цін, що може бути вражаючим під час порівняння. Визначаючи, який антивірусний план придбати, враховуйте такі фактори, як потрібен вам індивідуальний чи сімейний план, кількість пристроїв, які ви плануєте охопити, та чи шукаєте ви базове антивірусне рішення, чи комплексний пакет кібербезпеки. Багато антивірусних програм пропонують безкоштовні пробні версії або гарантії повернення грошей, що дозволяє користувачам випробувати продукт, перш ніж оформити підписку.

– Продуктивність: Хоча антивірусна програма може стверджувати, що блокує всі типи шкідливих програм, ви не можете бути впевнені в цьому, поки її не протестують. Антивірусні програми повинні пройти незалежне тестування в сертифікованих лабораторіях, щоб підтвердити, наскільки добре працює програмне забезпечення з неупередженої точки зору.

Такі надійні сайти, як AV-TEST, AV-Comparatives та SE Labs, тестуватимуть програмне забезпечення в реальних умовах та оцінюватимуть результати після тестування на предмет зручності використання, хибнопозитивних результатів та функціональності. Якщо відома антивірусна програма не пройшла стороннє тестування, це може викликати занепокоєння, і вам слід розглянути інші варіанти.

– Досвід користувача: Антивірусна програма зазвичай не є універсальною. Деякі мають комплексні продукти для Windows, але не мають багатьох функцій для macOS. Деякі мають інтерфейси користувача, які більше підходять для технічно підкованих користувачів, тоді як інші надають розширені рекомендації для початківців у сфері антивірусів. Ви також можете ознайомитися

з оглядами антивірусів, які часто містять примітки щодо досвіду користувача, такі як тривалість сканування, будь-які уповільнення під час сканування та простота використання програмного забезпечення.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Python – це об’єктно-орієнтована мова програмування високого рівня загального призначення з відкритим кодом. Це визначення може бути важким для новачків, тому розглянемо кожну характеристику окремо, щоб зрозуміти, що вона означає:

– Відкритий вихідний код: це безкоштовно та доступно для подальших покращень, таких як додавання корисних функцій або виправлення помилок.

– Об’єктно-орієнтована: заснована не на функціях, але в об’єктах з певними атрибутами й методами.

– Високий рівень: зручний для людини, а не для комп’ютера.

– Загальне призначення: можна використовувати для створення будь-яких програм.

Ця мова використовується в будь-якому програмному забезпеченні, про яке ви тільки можете подумати. Ви можете використовувати його для створення веб-сайтів, штучного інтелекту, серверів, програмного забезпечення для бізнесу та багато іншого. Також застосовується в науці про дані, аналізі даних, машинному навчанні, інженерії даних, веб-розробці, розробці програмного забезпечення та інших галузях.

Переваги та недоліки Python

Переваги:

– Її легко читати, вчити та писати. Це мова програмування високого рівня з англійським синтаксисом. Це полегшує читання та розуміння коду. Її дійсно легко зрозуміти і вивчити, тому багато людей рекомендують Python новачкам.

Вам потрібно менше рядків коду для виконання того ж завдання в порівнянні з іншими основними мовами, такими як C/C++ та Java.

– Підвищує продуктивність. Це дуже продуктивна мова. Завдяки її простоті розробники можуть зосередитися на розв'язанні проблеми. Їм не потрібно витрачати багато часу на розуміння синтаксису або поведінку мови програмування. Ви пишете менше коду та виконуєте більше завдань.

– Інтерпретована мова. Python мова, що інтерпретується, а це означає, що вона безпосередньо виконує код по рядку. Якщо сталася помилка, вона зупиняє подальше виконання та повідомляє про її виникнення. Вона показує лише одну помилку, навіть якщо у програмі їх кілька. Це спрощує налагодження.

– Динамічно типізована. Python не визначає тип змінної, доки ми не запустимо код. Вона автоматично надає тип даних, коли відбувається процес виконання. Фахівець може не турбуватися про оголошення змінних та типи даних.

– Безкоштовна та з відкритим вихідним кодом. Ця мова постачається під схваленою OSI ліцензією з відкритим вихідним кодом. Це робить його безкоштовним для використання та розповсюдження. Ви можете завантажити вихідний код, змінити його та навіть розповсюджувати свою версію. Це корисно для організацій, які хочуть використати свою версію для розробки.

– Підтримка великих бібліотек. Стандартна бібліотека Python є величезною, ви можете знайти майже всі функції, необхідні для вашого завдання. Таким чином ви не залежите від зовнішніх бібліотек.

– Портативність .У багатьох мовах, таких як C/C++, потрібно змінити свій код, щоб запустити програму на різних платформах. З Python все інакше. Ви тільки пишете один раз і запускаєте її будь-де.

Недоліки:

– Низька швидкість. Вище ми обговорювали, що це інтерпретована мова з динамічною типізацією. Порядкове виконання коду часто призводить до повільного виконання. Динамічна природа Python також є причиною її низької

швидкості, оскільки їй доводиться виконувати додаткову роботу при виконанні коду. Тому вона не підходить для цілей, де швидкість важливий аспект проєкту.

– Неефективна для пам’яті. Ця мова програмування використовує великий обсяг пам’яті, це може бути недоліком при створенні програм, коли віддають перевагу оптимізації пам’яті.

– Слабка у мобільних обчисленнях. Python зазвичай використовується у серверному програмуванні. Ми не бачимо – її на стороні клієнта або в мобільних програмах з таких причин: вона не заощаджує пам’ять і має повільну обчислювальну потужність у порівнянні з іншими мовами.

– Доступ до бази даних. Програмувати на цій мові легко, але коли ми взаємодіємо з базою даних, її не вистачає. Рівень доступу до бази даних у Python примітивний та недостатньо розвинений у порівнянні з іншими популярними технологіями.

– Помилки виконання. Це мова з динамічною типізацією, тому тип даних змінної може змінюватись у будь-який час. Змінна, що містить ціле число, у майбутньому може містити рядок, що може призвести до помилок виконання.

Застосування Python:

– Для аналізу даних. Дані стали цінним активом у будь-якій сучасній галузі, і більшість компаній зацікавлені у збиранні, обробці та аналізі релевантних даних, щоб витягти з них цінну інформацію для бізнесу. І тут Python виходить за межі будь-якої конкуренції. Python особливо цінна тим, що крім великої стандартної бібліотеки надає величезний набір додаткових модулів, розроблених спеціально для аналітичних цілей. Найвідоміші бібліотеки Python для аналізу даних – це pandas і NumPy . Ці інструменти дозволяють робити з вашими даними майже все, наприклад, очищати і аналізувати їх, вивчати статистику або візуалізувати приховані тенденції у ваших даних.

– Для візуалізації даних. Візуалізація даних – це окрема частина аналізу даних, яка допомагає нам подавати інформацію, необроблену чи очищену, у більш змістовній формі. Тут Python знову входить у гру, пропонуючи широкий

спектр інструментів візуалізації даних. Найпопулярніші з них – matplotlib і заснований на ній seaborn. Використовуючи їх, ми можемо створювати буквально всі види візуалізації: від найпростіших до складніших.

– Для машинного навчання. Машинне навчання (ML) є основою більшості завдань науки даних. Він є областю штучного інтелекту, пов'язаною з використанням алгоритмів, що дозволяють машинам вивчати закономірності та тенденції на основі історичних даних, щоб робити прогнози на основі невідомих даних. – Використовуючи методи ML, ми можемо створювати моделі, які можуть точно передбачити швидкість відтоку клієнтів компанії, оцінити ризик виникнення у людини певного захворювання, визначити оптимальне розташування автомобілів таксі й т.д. За допомогою Python ми можемо побудувати модель ML, використовуючи лише три рядки коду.

– Для розробки програмного забезпечення. Крім свого багатостороннього застосування в галузях науки про дані, Python використовується на кожному етапі розробки програмного забезпечення, включаючи контроль складання, автоматичну безперервну компіляцію, прототипування, відстеження помилок, тестування та обслуговування програмного забезпечення. За допомогою цієї мови можемо створювати аудіо- або відеопрограми на основі методів штучного інтелекту, машинного навчання, API (інтерфейсів прикладного програмування), GUI (графічних інтерфейсів) або будь-якого іншого типу програмного забезпечення.

– Для веброзробки. У той час як для створення візуальної частини вебсайту ми переважно будемо використовувати такі мови, як HTML, CSS та JavaScript, для його невидимої частини ми часто вибираємо Python. Серед масштабних вебсайтів та програм, створених за допомогою цієї мови, варто згадати Google, Facebook, Instagram, YouTube, Dropbox та Reddit.

– Для автоматизації задач/скриптингу. Це відмінний інструмент для написання програм для автоматизації різних завдань, що повторюються. Цей процес називається скриптингом. Зокрема, можна робити скрипти для роботи з

файлами та папками. Наприклад, можна створювати, перейменовувати, перетворювати, розділяти, об’єднувати або видаляти файли, перевіряти їх на наявність помилок. Ви також можете використовувати автоматизацію Python для пошуку та завантаження інформації з Інтернету, заповнення та надсилання онлайн-форм та надсилання регулярних повідомлень або електронних листів.

Яким фахівцям потрібно володіти Python:

– Фахівець з даних.

– Аналітик даних.

– Інженер даних.

– Інженер з машинного навчання.

– Журналіст даних.

– Архітектор даних.

– Повний стек веб-розробника.

– Backend-розробник.

– DevOps-інженер.

– Інженер-програміст.

Можемо зробити висновок, що Python ще довго буде популярною мовою, хоч і має низку недоліків. Цю мову використовують для створення вебсайтів, штучного інтелекту, серверів, програмного забезпечення для бізнесу, аналізу даних, машинного навчання, інженерії даних та для багатьох інших областей. Це перспективна і затребувана навичка, яка необхідна у всіх галузях.

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 мережевого антивірусного комплексу дата-центру.

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

Класифікація комп'ютерних вірусів

Вся наявна розмаїтість комп'ютерних вірусів можна класифікувати по різних ознаках, – наприклад по середовищу перебування, по руйнівних можливостях, по особливостях побудови й т.д.

За способом зараження:

– резидентні – віруси, що залишають в оперативній пам'яті свою резидентну (постійну) частина, що потім перехоплює звертання до програм, що заражаються, і впроваджується в них;

– нерезидентні – віруси, які не заражають оперативну пам'ять і проявляються лише при запуску інфікованої програми.

По цілісності:

– монолітні – віруси, програми яких являють собою єдиний блок;

– розподілені – віруси, програми яких розділені на частини, що містять інструкцію з відтворення вірусу(наприклад, одна якась частина заражає комп'ютер, а потім завантажує з Інтернету інші частини, що виконують шкідливу дію).

Інші шкідливі програми

Крім розглянутих вище комп'ютерних вірусів, існують і інші види шкідливих програм. Серед них, наприклад, можна назвати:

– мережні хробаки – шкідливі програми, які поширюються по комп'ютерних мережах, обчислюючи адреси мережних комп'ютерів і розсилаючи по цих адресах свої копії;

– троянські програми («троянські коні», квазивіруси) – шкідливі програми, які не здатні до самопоширення, а маскуються під якусь корисну або цікаву

програму, руйнують завантажувальний сектор і файлову систему або збирають і пересилають своєму творцеві інформацію, що не підлягає розголошенню (наприклад, ваші особисті паролі);

Антивірусні програми

Антивірусні програми призначені для захисту комп'ютерів від більшості вірусів, хробаків і «троянських коней», які можуть видаляти файли, одержувати доступ до особистих даних або використовувати заражену систему як засіб атаки на інші комп'ютери.

Антивірусні програми звичайно використовують два різних методи для виконання своїх завдань:

1) сканування (перегляд) файлів для пошуку вже відомих вірусів, для яких у вірусній базі (щовходить у комплект антивірусної програми спеціальної БД) є інформація про характерні фрагменти вірусного програмного коду (сигнатурах вірусів);

2) виявлення підозрілого поводження будь-якої програми, що схоже на поводження зараженої програми («евристичне сканування»).

Антивірусне програмне забезпечення складається з пакета програм, які виявляють, запобігають розмноженню й видаляють комп'ютерні віруси й інші шкідливі програми.

При виборі антивірусної програми необхідно враховувати наступні параметри, яким антивірус повинен відповідати:

1. Сталість і надійність роботи. Цей параметр є визначальним. При стабільній роботі антивірусної програми немає відчуття, що якісь заражені файли залишилися непоміченими.

2. Великий обсяг і постійне відновлення вірусної бази. Сюди ж ставиться вміння програми швидко пізнавати види вірусів працювати з файлами різних типів (архівами, документами), і здійснювати автоматичну перевірку всіх нових файлів у міру їхнього копіювання.

3. Швидкість роботи антивірусу й додаткові функції. До додаткових функцій можна віднести наявність евристичного сканування й можливість лікування заражених файлів (коли віруси з них віддаляються, а файли приводяться у вихідний стан, що був до їхнього зараження).

4. Підтримка різних програмою багатьох операційних систем – багатоплатформеність. При роботі в мережному варіанті немаловажним є також наявність в антивірусної програми серверних модулів, призначених для адміністрування, і наявність можливості роботи на різних серверах.

Захист робочого столу

У минулому захист робочих столів був єдиною «сферою», де використовувалися антивіруси. Сьогодні він все ще має вирішальне значення для захисту центрів обробки даних, але має великий недолік. Річ у тім, що для його роботи кожен кінцевий користувач повинен регулярно оновлювати своє програмне забезпечення. Крім того, важко відстежувати їх усі, що наражає центр на небезпеку.

Сучасні антивіруси для комп'ютерів мають численні автоматичні сповіщення. Люди часто забувають про них, відкладають їх або ігнорують, що, знову ж таки, робить всю систему вразливою. Ось як працює захист комп'ютера: антивірус сканує диски та оперативну пам'ять (RAM), шукаючи шкідливий код у базі даних.

Також, щойно система виявляє потенційно небезпечний файл/програму, користувач отримує сповіщення. Далі користувач може вибрати одну з наступних дій: видалення вірусу, спроба очищення файлів/програм або поміщення їх у карантин. Це все, що вам потрібно знати про антивірусний захист робочого столу. Він не ідеальний, але й донині центри обробки даних включають його як надійний рівень захисту.

Захист сервера

У цьому випадку антивірусне рішення працює на рівні поштового сервера. Це означає, що більшість шкідливих кодів будуть знищені ще до того, як вони

потраплять на комп’ютери користувачів. На жаль, навіть за наявності антивіруса, що захищає сервери, вірусам все ж вдається потрапити в мережу. Захист сервера так само важливий, як і захист робочого столу, але в більшості випадків він є «бонусом», і саме тому міжнародні компанії його використовують.

Якщо ви покладаєтеся лише на антивірус сервера, а шкідливе програмне забезпечення потрапляє на робочі столи, ніщо не завадить йому пошкодити/знищити конфіденційні бізнес-дані. Тому вам потрібно використовувати все це разом, включаючи наступний (і останній) захист – на рівні «шлюзу». Він працює подібно до захисту сервера, але має дещо інший підхід.

Захист шлюзу

Ідея полягає в тому, щоб зупинити шкідливе програмне забезпечення в найвіддаленішій точці, задовго до того, як воно отримає шанс досягти мережі. Поки віруси/пошкоджені повідомлення тримаються на відстані, вони не зможуть завдати жодної шкоди центру обробки даних. Шлюзи існують не так давно, але вони вже довели свою ефективність як захист на рівні серверів і робочих столів.

Інтеграція антивірусного програмного забезпечення в мережеве обладнання є серцем і душею підходу шлюзу. На рівні шлюзу антивірус сканує всі вхідні повідомлення, шукаючи будь-які потенційні загрози. Крім того, коли він виявляє заражені повідомлення, він або зупиняє їх, поміщає в карантин, або видаляє їх. На думку експертів, це прогресивне рішення є дуже перспективним і, найімовірніше, стане новою нормою в найближчі роки.

Завдяки інтегрованим продуктам, вбудованим безпосередньо в мережі, захист центрів обробки даних від зовнішніх загроз стане набагато комфортнішим. Вам слід знати, що уряди різних країн також використовують підхід «центр обробки даних/сервер», і це одна з причин, чому безпека кінцевих точок стає важливою частиною світу, в якому ми зараз живемо.

3.2 Розробка структурної схеми

Антивірусний комплекс дата-центру – це не тільки антивірус і антишпигун. Він визначає й нейтралізує різні види небажаного ПЗ.

Основні переваги:

– Невимогливість до ресурсів – відмінно працює на настільних ПК, ідеальний для роботи на ноутбуках.

– Можливість установки на заражену машину й лікування ураженої системи.

– Коректна перевірка «на лету» вхідної й вихідної пошти по протоколах SMTP, POP3, IMAP, NNTP.

– Захист від шпигунських, рекламних програм, хакерських утиліт і програм-дозвонщиків.

– Висока частота відновлень вірусних баз – до декількох разів у годину! Відновлення виробляється з локальних серверів мережі.

Масове поширення комп'ютерних вірусів, а також активне обговорення в пресі планів інформаційної війни із залученням хакерів для придушення ворожих систем керування й передачі даних привели до того, що питання про створення засобів протидії й захисти здобуває нову якість. На думку ряду закордонних експертів, держава, що програла в інформаційній війні, буде відкинуто у своєму розвитку на багато десятиліть.

Сьогодні вже ясно, що традиційні методи побудови систем захисту інформації не принесуть бажаного результату. Треба шукати принципово нові підходи до рішення цієї проблеми. Справжня стаття покликана дати «інформацію до міркування» для розроблювачів антивірусних систем, щоб вони змогли глянути на свою предметну область із іншої сторони, а саме – з боку Природи, що створила, напевно, саму зроблену систему захисту – імунну систему організму.

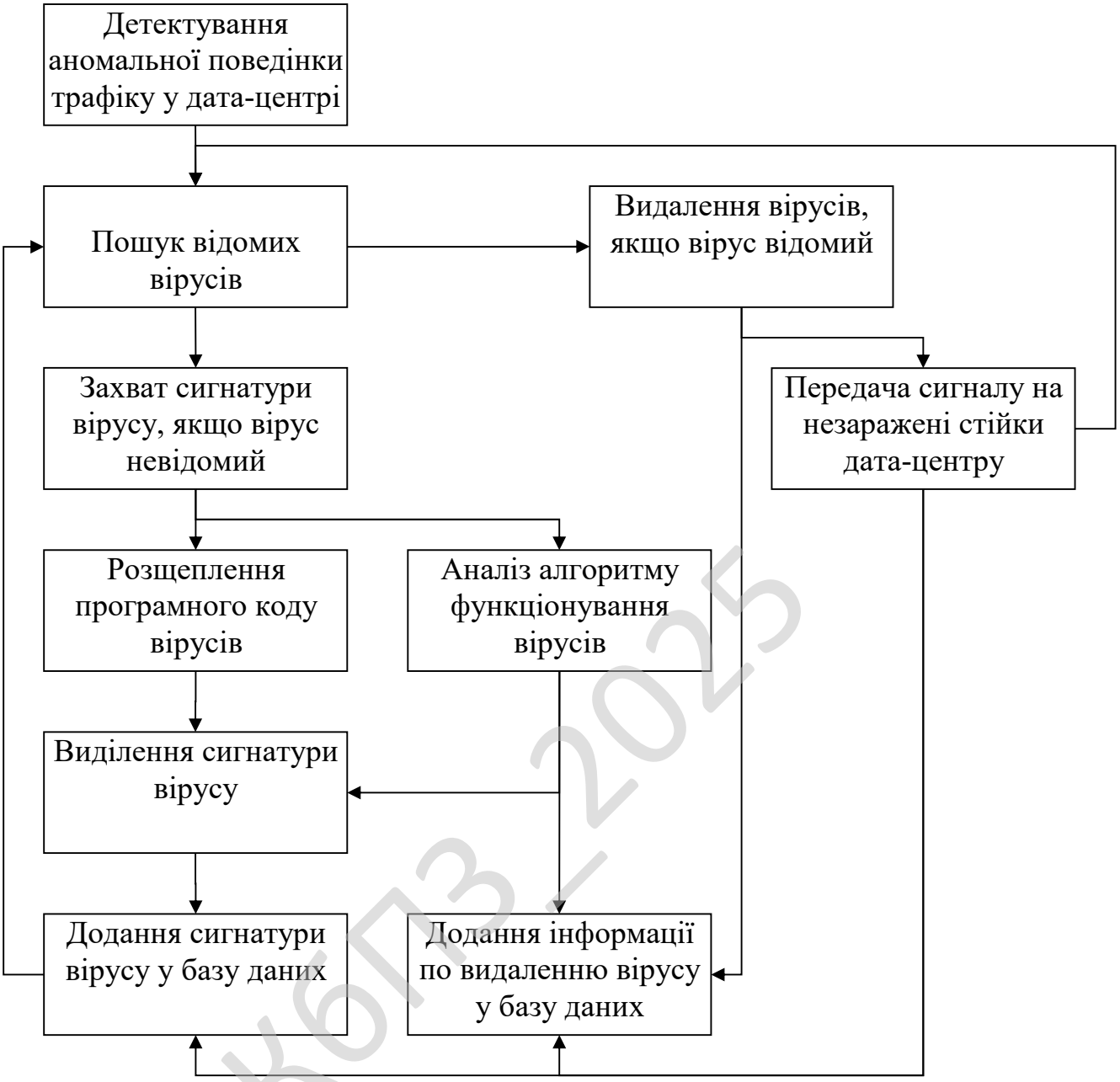


Рисунок 3.1 – Структурна схема системи

Уже розроблений ряд алгоритмів, що дозволяють писати віруси, які принципово не можна виявити жодним з існуючих способів. Багато хто відзначають, що самоідентифікуємий довільним образом код одержати просто неможливо. У кожному разі, є тверді рамки, які дозволяють ту саму операцію реалізувати обмеженим числом способів. Способи ці відомі заздалегідь, що, у

принципі, дозволяє перелічити всі ключові фрагменти вірусів, а значить – безпомилково їх розпізнати.

Однак якщо припустити, що архітектура процесора може бути довільною, або навіть динамічно синтезованої в процесі виконання, то досить написати емулятор відповідного процесора – деяку віртуальну машину, що буде виконувати код вірусу, побудований на певних принципах.

Вірус, написаний на віртуальній машині, вимагає дуже багато часу для аналізу традиційними методами. Виходить, потрібні засоби автоматичної боротьби з такого роду деструктивними програмами. Питання лише в тім, на яких принципах повинна базуватися така антивірусна система? Відповідь виявляється дивно простою: на принципах імунної системи людини. Дійсно, у нашім організмі функціонує чудова система, здатна боротися з мільярдами хвороботворних антигенів.

Антивірусна технологія антивірусного комплексу дата-центру побудована на основі моделі імунної системи людини.

Очевидно, що створити систему захисту інформації комп'ютерної мережі по прямій подобі імунної системи людини практично неможливо, так у цьому й немає необхідності. Однак той факт, що імунна система досягла досконалості в боротьбі із хвороботворними й чужорідними антигенами, говорить про те, що багато принципів, що сформували імунну систему, досить ефективні й можуть бути використані з тим допущенням, що працювати вони будуть не з біохімічними антигенами, а з антигенами програмними, тобто інформаційними.

Поряд із цим останні досягнення в області створення багатоагентних інтелектуальних систем дозволяють сподіватися, що найближчим часом штучна імунна система буде створена і її ефективність не опуститься нижче ефективності її природного прототипу.

3.3 Розробка функціональної схеми

Антивірус, що розроблений у результаті виконання дипломного проектування – захист вашого дата-центру від шкідливих програм, що включає базові функції забезпечення безпеки вашого дата-центру. Антивірус використовує новітні технології захисту, завдяки якому забезпечується безпека й стабільна роботу дата-центру.

Основні функції антивірусного комплексу дата-центру, що розроблений:

– Проактивний захист нового покоління від невідомих погроз.

– Віртуальна клавіатура для безпечного введення логінів, паролів і номерів кредитних карт на веб-сторінках.

– Перевірка операційної системи й установлених програм на наявність уразливостей.

– Налаштування операційної системи й інтернет-браузера для безпечної роботи в мережі Інтернет.

– Відновлення працездатності системи після вірусної атаки.

– Видалення тимчасових файлів інтернет-браузера.

– Захист у режимі реального часу.

– Базовий захист при роботі в мережі Інтернет і з електронною поштою.

– Мінімальне завантаження антивірусного комплексу дата-центру.

– Інтуїтивно зрозумілий інтерфейс.

– Для повноцінного захисту антивірусного комплексу дата-центру крім антивірусу рекомендується використовувати міжмережний екран.

– Перевірка файлів, веб-сторінок, поштових і ICQ-повідомлень.

– Блокування посилань на заражені веб-сайти й сайти, що перехоплюють інформацію.

– Проактивний захист від невідомих погроз, заснована на аналізі поводження програм.

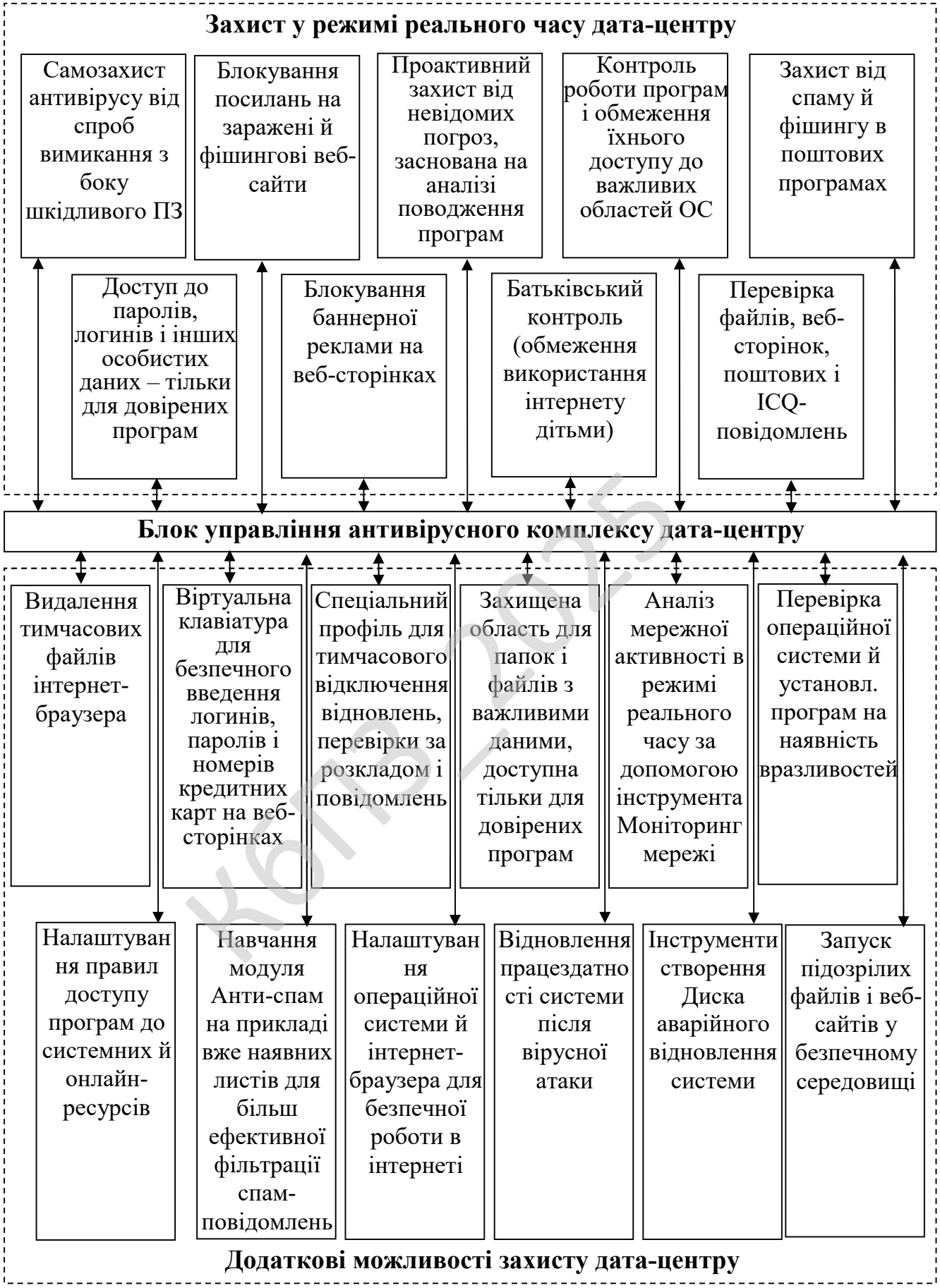


Рисунок 3.2 – Функціональна схема системи

– Самозахист антивірусу, що розроблений попереджає погрозу вимикання з боку шкідливого ПЗ.

– Система миттєвого виявлення погроз, що моментально блокує нові шкідливі коди.

– Реалізовано модуль «Перевірка посилань», що попереджає про заражені або небезпечних веб-сайти.

На рисунку 3.2 зображена функціональна схема системи. Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.3. При детальному її розгляді можна побачити як саме проходе взаємодія у розробленій системі.

Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування). Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи.

Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

Діаграми потоків даних містять чотири типи елементів:

– Процеси які являють собою трансформацію даних в рамках описуваної системи.

– Сховища даних (репозиторії).

– Зовнішні по відношенню до системи сутності.

– Потоки даних між елементами трьох попередніх типів.

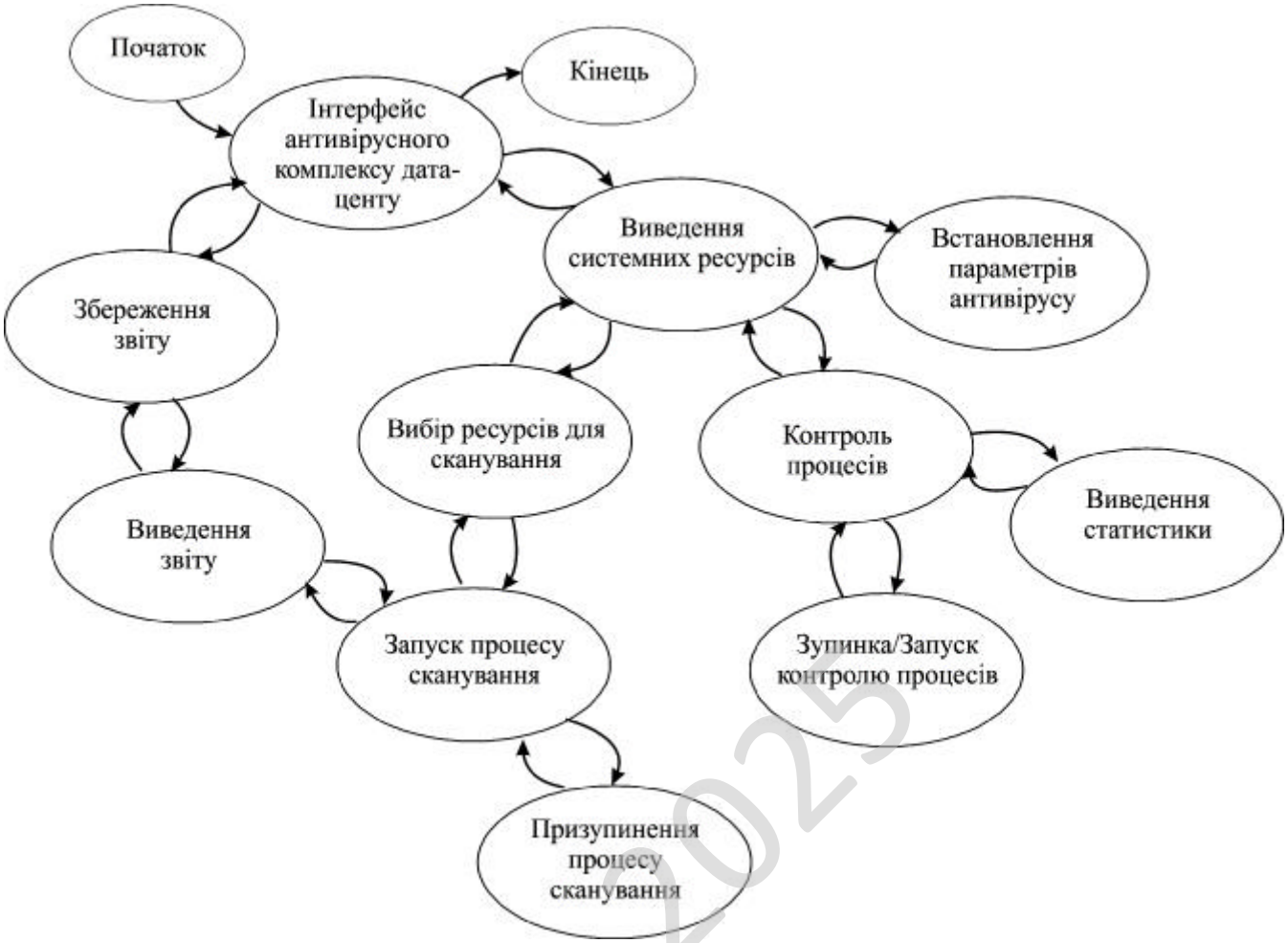


Рисунок 3.3 – Діаграма взаємодії процесів

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 ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Блок-схеми є основою ПЗ. Тому від точності і детальності проробки блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об’єктно-орієнтована що вимагає оптимізації, також те, що при розробці програми слід надати особливу увагу модулю антивірусного комплексу дата-центу.

Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів.

Функціональні блоки можуть виконуватися в укрупненому і розгорненому вигляді. У першому випадку на схемі зображають найважливіші блоки системи і зв'язки між ними.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підпрограми та останньої стадії – перевірки поточного стану та поверненням на початок схеми чи з завершенням роботи розробленого ПЗ.

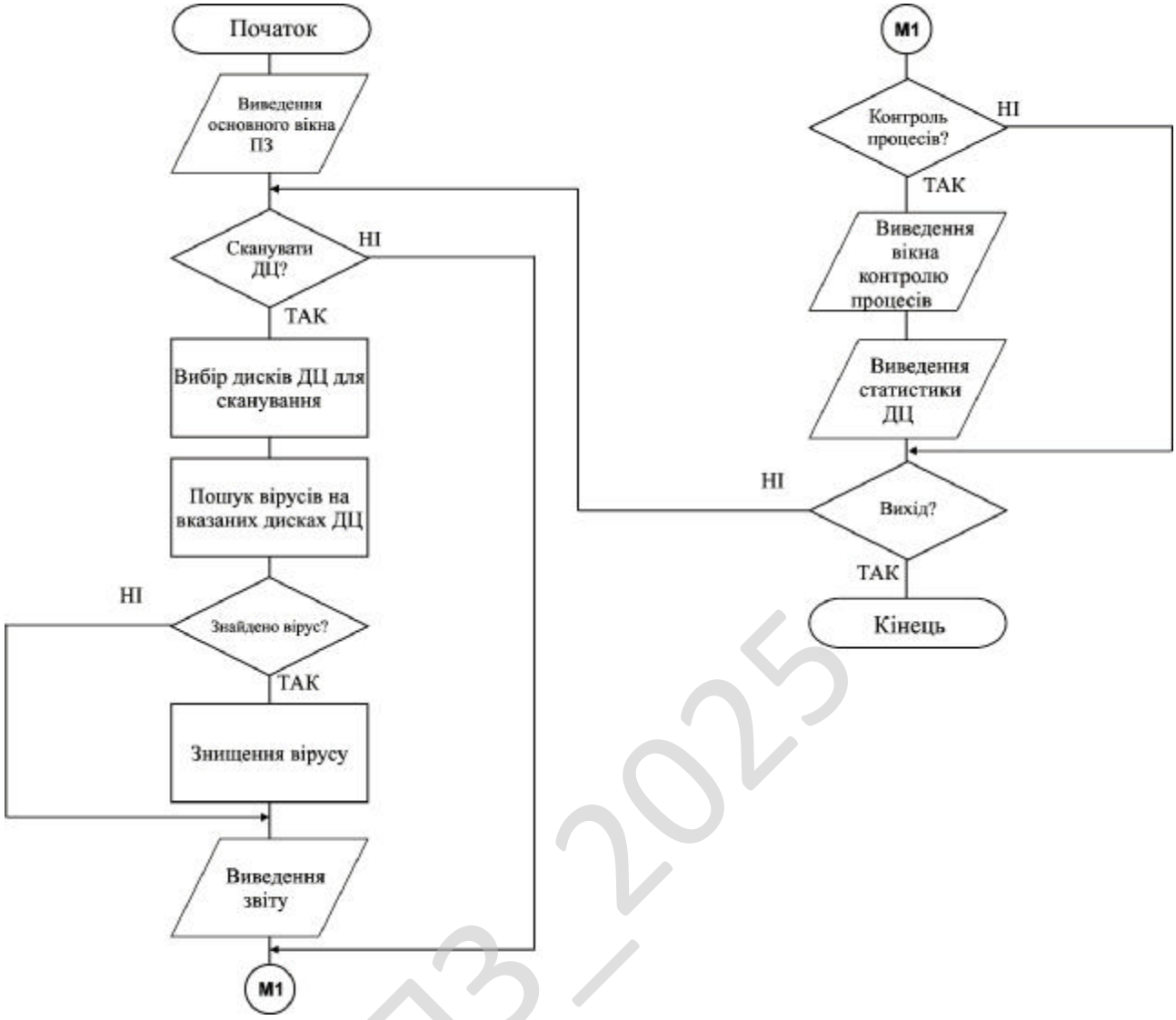


Рисунок 4.1 – Блок-схема основної програми

При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю.

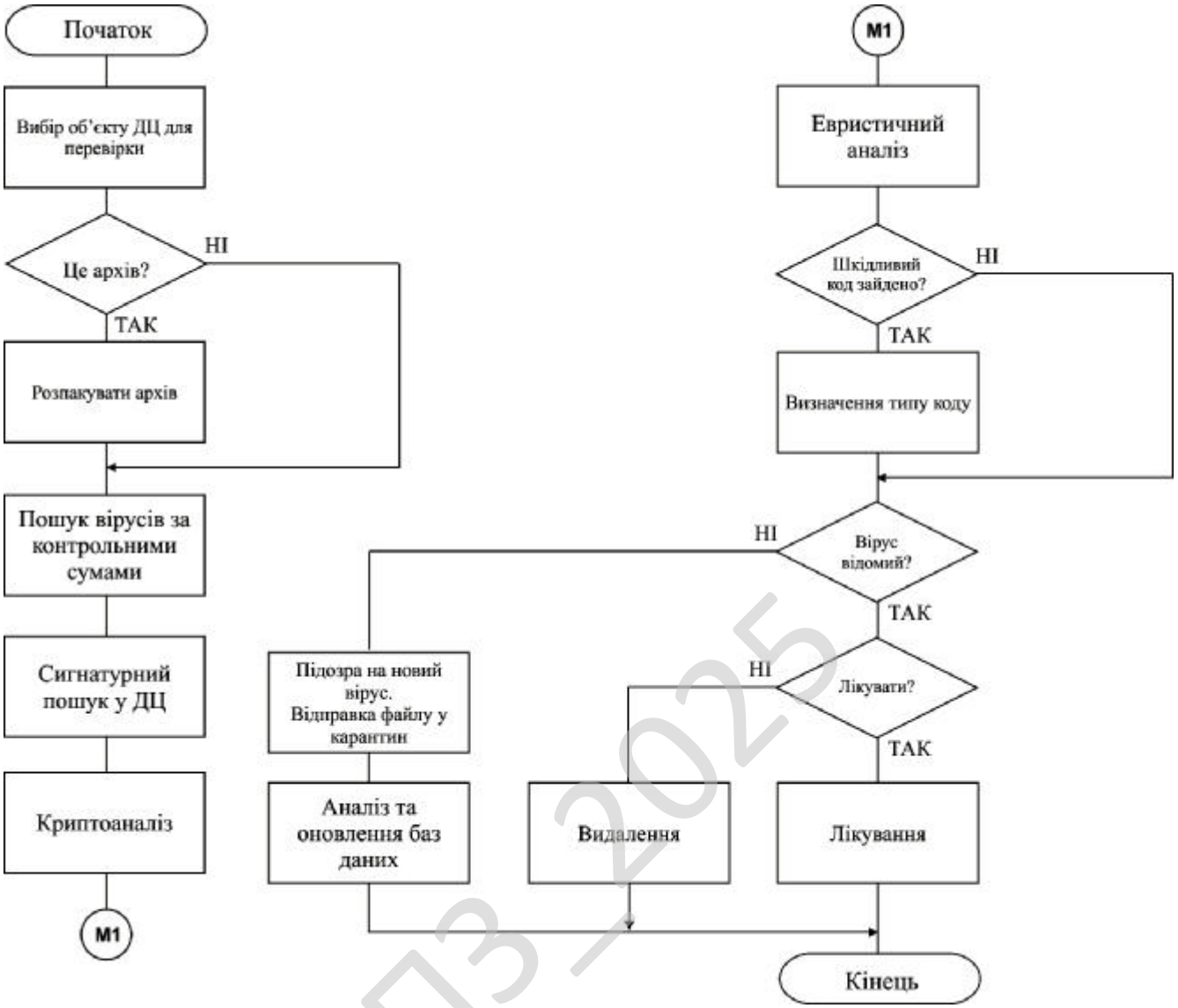


Рисунок 4.2 – Блок-схема роботи підпрограми

UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

Розглянемо використані технології та їх основні компоненти що підтверджують правильність використаних проектних рішень.

Подійно-орієнтована архітектура (Event-driven architecture, надалі EDA) – шаблон архітектури програмного забезпечення, який призначений для створення подій, їх виявлення, споживання і реагування на них.

Подія може бути визначена як значна зміна стану. Наприклад, коли споживач купує автомобіль, стан автомобіля змінюється з "на продаж" до "продано". Архітектура системи дилера автомобілів може трактувати цю зміну стану як подію, поява якої може стати відомою іншим програмам даної архітектури.

З формальної точки зору, те, що виробляється, публікується, поширюється, виявляється і споживається (як правило, асинхронно) є повідомленням, яке називають сповіщенням про подію (або нотифікацією), а не самою подією, яка є зміною стану, що викликає появу повідомлення.

Події не подорожують, вони просто відбуваються. Проте термін подія часто використовується метонімічно для позначення самого нотифікаційного повідомлення, що може призвести до певної плутанини.

Цей архітектурний шаблон може застосовуватися при проектуванні і реалізації ПЗ і систем, які передають події між слабкозв’язаними компонентами програмного забезпечення і сервісами (службами).

Подійно-орієнтована система як правило складається з емітерів подій (або агентів) і споживачів подій (або стоків).

Стоки несуть відповідальність за здійснення реагування на появу події. Реакція не завжди може бути повністю забезпечена самим стоком. Наприклад, стік, може бути відповідальним лише за фільтрацію, трансформацію і відправку події до іншого компонента або він може забезпечити повністю самостійну реакцію на таку подію. Перша категорія стоків може бути заснована на традиційних компонентах, таких як проміжне програмне забезпечення, орієнтоване на обробку повідомлень (message oriented middleware, MOM), в той час, як друга категорія стоків (самостійна реакція в режимі он-лайн) може вимагати більш придатної платформи (фреймворку) для виконання транзакцій.

Розробка ПЗ і систем в подійно-орієнтованій архітектурі дозволяє їм бути сконструйованими способом, який більш відповідає вимогам до їх створення, оскільки такі системи в більшій мірі пристосовуються до непередбачуваних і асинхронних середовищ.

Подійно-орієнтована архітектура (EDA) може доповнювати сервісно-орієнтовану архітектуру (SOA), оскільки сервіси (служби) можуть бути активовані тригерами, які ініціюються при настанні подій.

Ця парадигма особливо корисна, коли стік не забезпечує власного виконання будь-яких дій.

Подійно-орієнтована SOA (SOA-2) розвиває архітектури SOA і EDA для забезпечення більш глибокого і надійного рівня сервісів за рахунок використання раніше невідомих причинно-наслідкових зв'язків, щоб сформувати новий шаблон подій. Цей новий шаблон бізнес-аналітики дає поштовх до небаченого раніше зростання рівня автоматизації підприємства за рахунок привнесення додаткової цінної інформації в описану раніше модель діяльності.

Обчислювальна техніка та сенсорні пристрої (сенсори, датчики, контролери) можуть виявляти зміни стану об'єктів або умов і створювати події, які потім можуть бути оброблені сервісом (службою) або системою.

Подія може складатися з двох частин: заголовка події та тіла події. Заголовок події може включати в себе інформацію таку як, наприклад, назва події, часова мітка події і тип події. Тіло події – це частина, яка описує факт, що стався в дійсності. Тіло події не слід плутати з шаблоном або логікою, яка може бути застосована як реакція на саму події.

Архітектура, керована подіями, складається з чотирьох логічних рівнів (шарів). Вона починається з виявлення факту, його технічного подання у формі події і закінчується непустою множиною реакцій на цю подію.

Генератор подій

Першим логічним шаром є генератор подій, який виявляє факт і представляє цей факт подією. Оскільки фактом може бути практично все, що

може бути сприйнято, то ним може бути і генератор подій. Наприклад, генератором може бути клієнт електронної пошти, система електронної комерції або певний тип датчика.

Перетворення різних даних, отриманих від датчиків, в єдину стандартизовану форму даних, які можуть бути оцінені, є основною проблемою при розробці та реалізації цього шару. Однак, враховуючи, що подія є строго декларативною, можна легко застосовувати будь-які операції трансформації, тим самим усуваючи необхідність забезпечення високого рівня стандартизації.

Канал подій

Канал подій – це механізм, через який інформація від генератора подій передається до обробника подій (event engine) або стоку.

Це може бути з'єднання TCP/IP або вхідний файл будь-якого типу (простий текст, формат XML, e-mail тощо). В один і той же час може бути відкрито кілька каналів подій. Як правило, оскільки обробник подій повинен працювати в режимі, наближеному до реального часу, канали подій зчитуються асинхронно. Події зберігаються в черзі, очікуючи наступної обробки механізмом обробки подій.

Механізм обробки подій

Механізм обробки подій (event processing engine) є місцем, де подія ідентифікується і вибирається відповідна реакція на нього, яка потім виконується. Це також може призвести до породження ряду тверджень. Якщо подія, яка надійшла до механізму обробки подій, є наприклад такою «Запаси продукту ID досягли нижнього допустимого рівня», це може ініціювати, наприклад, такі реакції як «Замовити продукт ID» і «Сповістити персонал».

Наступна подійно-орієнтована дія (післядія)

Щодо того, як можуть проявлятися наслідки події, слід відмітити, що вони можуть проявитись багатьма різними способами і у різноманітних формах (наприклад, повідомлення електронної пошти, надіслане комусь, або ПЗ, що виводить деяке попередження на екран). Залежно від рівня автоматизації, який

забезпечується стоком (механізмом обробки подій), ці дії можуть виявитись зайвими.

Є три основні стилі обробки подій: простий, потоковий і складний. Часто ці три стилі використовуються спільно у розвинутій подійно-орієнтованій архітектурі.

Проста обробка подій

Проста обробка подій стосується подій, які безпосередньо належать до специфічних вимірних змін умов. У випадку простої обробки подій, мають справу з появою відомих подій, що ініціюють післядію (післядії). Проста обробка подій зазвичай використовується для управління потоком робіт в реальному часі, скорочуючи тим самим час затримки і вартість робіт.

Наприклад, прості події можуть створюватись (породжуватись) датчиком, що виявляє зміну тиску в шині або температуру навколишнього середовища.

Обробка потоку подій

При обробці потоку подій (event stream processing, далі ESP) відбуваються як звичайні, так і відомі події. Звичайні події (заявки, передачі RFID) перевіряються на те, чи є вони відомими, і передаються інформаційним передплатникам. Обробка потоку подій зазвичай використовується для управління потоком інформації в реальному часі і на рівні підприємства, що дозволяє своєчасно приймати рішення.

Обробка складних подій

Обробка складних подій (Complex event processing (CEP)) дозволяє за шаблонами простих і звичайних подій проводити аналіз того, чи наступила складна подія. Обробка складних подій полягає в оцінюванні взаємного впливу подій і в наступному виконанні дій. При цьому, типи подій (відомих або звичайних) можуть перетинатись, а події можуть виникати протягом тривалого періоду часу.

Кореляція подій може бути причинною, тимчасовою або просторовою. CEP вимагає використання складних інтерпретаторів подій, визначення і підбору

шаблонів подій, а також відповідних кореляційних методів.

Обробка складних подій зазвичай використовується для виявлення і реагування на аномальну поведінку, загрози і можливості у бізнесі.

Антивірусний "движок" – це програмний модуль, що призначений для детектування шкідливого програмного забезпечення. "Движок" є основним компонентом будь-якої антивірусної програми, незалежно від її призначення.

Движок використовується як у персональних продуктах – персональний сканер або монітор, так і в серверних рішеннях – сканер для поштового або файлового сервера, мережного екрану або проксі-серверу.

Як правило, для детектування шкідливих програм, у більшості "движків" реалізовані наступні технології:

– Пошук за "сигнатурами" (унікальній послідовності байт).

– Пошук за контрольними сумами або CRC (контрольної суми з унікальної послідовності байт).

– Використання скороченої маски.

– Криптоаналіз.

– Статистичний аналіз.

– Евристичний аналіз.

– Емуляція.

Розглянемо кожний із цих методів докладніше.

Пошук за "сигнатурами".

Сигнатура – це унікальний "рядок" байт, що однозначно характеризує ту або іншу шкідливу програму. Сигнатурний пошук, у тій або іншій модифікації, використовується для виявлення вірусів та інших шкідливих програм, починаючи з найперших антивірусних програм і дотепер. Незаперечне достоїнство сигнатурного пошуку – швидкість роботи (при використанні спеціально розроблених алгоритмів) і можливості детектування декількох вірусів однією сигнатурою. Недолік – розмір сигнатури для впевненого детектування повинен бути досить великий, як мінімум 8-12 байт (звичайно для точного детектування

використовуються набагато більш довгі сигнатури, до 64 байт), отже, розмір антивірусної бази буде досить великим. Крім цього, останнім часом більшу поширеність одержали шкідливі програми, написані на мовах високого рівня (C++, Delphi, Visual Basic), а в таких програм є окремі частини коду, які практично не змінюються (так звана Run Time Library). Неправильно обрана сигнатура неминуче приведе до помилкового спрацьовування – детектування "чистого", не зараженого файлу як зараженого вірусом. Як рішення цієї проблеми пропонується використовувати або дуже великі сигнатури або використовувати детектування по деяких областях даних, наприклад, таблиці переміщень (relocation table) або текстові рядки, що не завжди добре.

Пошук за контрольними сумами (CRC).

Пошук за контрольними сумами (CRC – cyclic redundancy check), по суті, є модифікацією пошуку за сигнатурами. Метод був розроблений для запобігання основних недоліків сигнатурного пошуку – розміру бази й зменшення ймовірності помилкових спрацьовувань. Суть методу полягає в тому, що для пошуку шкідливого коду береться не тільки "опорний" рядок – сигнатура, а, точніше сказати, контрольна сума цього рядка, але й місце розташування сигнатури в тілі шкідливої програми. Місце розташування використовується для того, щоб не підраховувати контрольні суми для всього файлу. Таким чином, замість 10-12 байт сигнатури (мінімально) використовується 4 байти для зберігання контрольної суми й ще 4 байти – для місця розташування. Однак метод пошуку за контрольними сумами трохи повільніший, ніж пошук за сигнатурами.

Використання масок для виявлення шкідливого коду досить часто буває ускладнений наявністю шифрованого коду (так звані поліморфні віруси), оскільки при цьому або неможливо вибрати маску, або маска максимального розміру не задовольняє умові однозначної ідентифікації вірусу без помилкових спрацьовувань.

Неможливість вибору маски достатнього розміру у випадку поліморфного вірусу легко пояснюється. Шляхом шифрування свого тіла вірус домагається того,

що більша частина його коду в ураженому об'єкті є змінною, і, відповідно, не може бути обрана як маска.

Для детектування таких вірусів застосовуються наступні методи: використання скороченої маски, криптоаналіз і статистичний аналіз. Розглянемо ці методи докладніше.

Використання скороченої маски.

При поразці об'єктів вірус, що використовує шифрування, перетворить свій код у шифровану послідовність даних:

$$S = F(T),$$

де T – базовий код вірусу;

S – зашифровані коди вірусу;

F – функція шифрування вірусу, що довільно вибирається з деякої множини перетворень {F}.

Спосіб скороченої маски полягає в тому, що вибирається перетворення R зашифрованих кодів вірусу S, таке, що результат перетворення (тобто деяка послідовність даних S') не буде залежати від ключів перетворення F, тобто:

$$S = F(T),$$

$$S' = R(S) = R(F(T)) = R'(T).$$

При застосуванні перетворення R до всіляких варіантів шифрованого коду S результат S' буде постійним при постійному T. Таким чином, ідентифікація уражених об'єктів виконується шляхом вибору S' як скорочена маска й застосування до уражених об'єктів перетворення R.

Криптоаналіз

Цей спосіб полягає в наступному: за відомим базовим кодом вірусу й за відомим зашифрованим кодом (або за "підозріломим" кодом, схожим на зашифроване тіло вірусу) відновлюються ключі й алгоритм програми-розшифровувача. Потім цей алгоритм застосовується до зашифрованої ділянки, результатом чого є розшифроване тіло вірусу. При рішенні цього завдання доводиться мати справу із системою рівнянь.

Як правило, цей спосіб працює значно швидше й займає набагато менше пам'яті, ніж емуляція інструкцій вірусу. Однак рішення подібних систем часто є завданням високої складності.

Причому основна проблема – це математичний аналіз отриманого рівняння або отриманої системи рівнянь. Багато в чому завдання рішення систем рівнянь при відновленні зашифрованого тіла вірусу нагадує класичне криптографічне завдання відновлення зашифрованого тексту при невідомих ключах. Однак тут це завдання звучить трохи інакше: необхідно з'ясувати, чи є даний зашифрований код результатом застосування деякої відомої з точністю до ключів функції. Причому заздалегідь відомі багато даних для рішення цього завдання: ділянка зашифрованого коду, ділянка незашифрованого коду, можливі варіанти функції перетворення. Більш того, сам алгоритм цього перетворення й ключі також присутні в аналізованих кодах. Однак існує значне обмеження, що полягає в тому, що дане завдання повинне вирішуватися в конкретних границях оперативної пам'яті й процедура рішення не повинна займати багато часу.

Статистичний аналіз

Також використовується для детектування поліморфних вірусів. Під час своєї роботи сканер аналізує частоту використання команд процесора, будує таблицю команд, що зустрічаються, процесора, і на основі цієї інформації робить висновок про зараження файлу вірусом. Даний метод ефективний для пошуку деяких поліморфних вірусів, тому що ці віруси використовують обмежений набір команд у декрипторі, тоді як "чисті" файли використовують зовсім інші команди з іншою частотою. Наприклад, всі програми для MS-DOS часто використовують переривання 21h, однак у декрипторі поліморфних DOS-вірусів ця команда практично не зустрічається.

Основний недолік цього методу в тому, що є ряд складних поліморфних вірусів, які використовують майже всі команди процесора й від копії до копії набір використовуваних команд сильно змінюється, тобто за побудованою таблицею частот не представляється можливим виявити вірус.

Евристичний аналіз

Коли кількість вірусів перевищила кілька сотень, антивірусні експерти задумалися над ідеєю детектування шкідливих програм, про існування яких антивірусна програма ще не знає (немає відповідних сигнатур). У результаті були створені так звані евристичні аналізатори. Евристичним аналізатором називається набір підпрограм, які аналізують код файлів, що виконуються, макросів, скриптів, пам'яті або завантажувальних секторів для виявлення в ньому різних типів шкідливих комп'ютерних програм. Існують два принципи роботи аналізатора.

Статичний метод. Пошук загальних коротких сигнатур, які присутні в більшості вірусів (так звані "підозрілі" команди). Наприклад, велика кількість вірусів робить пошук вірусів по масці *.EXE, відкриває знайдений файл, робить запис у відкритий файл. Завдання евристик у цьому випадку – знайти сигнатури, що відбивають ці дії. Потім відбувається аналіз знайдених сигнатур, і, якщо знайдено деяку кількість необхідних і достатніх "підозрілих команд", то приймається рішення про те, що файл інфікований. Великий плюс цього методу – простота реалізації й хороша швидкість роботи, але при цьому рівень виявлення нових шкідливих програм досить низький.

Динамічний метод

Цей метод з'явився одночасно із впровадженням в антивірусні програми емуляції команд процесора (докладніше емулятор описаний нижче). Суть методу полягає в емуляції виконання програми й протоколюванні всіх "підозрілих" дій програми. На основі цього протоколу приймається рішення про можливе зараження програми вірусом. На відміну від статичного методу, динамічний метод більш вимогливий до ресурсів комп'ютера, однак і рівень виявлення в динамічному методі значно вище.

Емуляція

Технологія емуляції коду програм (або Sandboxing) стала відповіддю на появу великої кількості поліморфних вірусів. Ідея цього методу полягає в тому, щоб емулювати виконання програми (як зараженої вірусом, так і "чистої") у

спеціальному "оточенні", що називається також буфером емуляції або "пісочницею". Якщо в емулятор попадає заражений поліморфним вірусом файл, то після емуляції в буфері виявляється розшифроване тіло вірусу, готове до детектування стандартними методами (сигнатурний або CRC пошук).

Сучасні емулятори емулюють не тільки команди процесора, але й виклики операційної системи. Задача написання повноцінного емулятора є досить трудомісткою, не говорячи вже про те, що при використанні емулятора доводиться постійно контролювати дії кожної команди. Це необхідно для того, щоб випадково не виконати деструктивні компоненти алгоритму вірусу.

Слід особливо зазначити, що доводиться саме емулювати роботу інструкцій вірусу, а не трасувати їх, оскільки при трасуванні вірусу занадто велика ймовірність виклику деструктивних інструкцій або кодів, відповідальних за поширення вірусу.

Було використано підходи MSF. Microsoft Solutions Framework (MSF) – Методологія розробки програмного забезпечення, запропонована корпорацією Microsoft. MSF спирається на практичний досвід Microsoft і описує управління людьми і робочими процесами в процесі розробки рішення.

MSF є узгоджений набір концепцій, моделей і правил. У 1994 році, прагнучи досягти максимальної віддачі від IT -проектів, Microsoft випустила в світ пакет посібників з ефективного проектування, розробки, впровадження та супроводу рішень, побудованих на основі своїх технологій. Ці знання базуються на досвіді, отриманому Microsoft при роботі над великими проектами з розробки та супроводження програмного забезпечення, досвід консультантів Microsoft і кращому з того, що накопичила на даний момент IT-індустрія. Все це представлено у вигляді двох взаємопов'язаних і добре доповнюють один одного областей знань: Microsoft Solutions Framework (MSF) і Microsoft Operatio ns Framework (MOF).

Слід зазначити, що Microsoft розробила на базі загальних методів MSF методики для прикладного та спеціалізованого застосування. Причому, Microsoft

сертифікує експертів саме по прикладних знань в застосуванні MSF (наприклад, сертифікація MCTS 74-131 з експертизи в методиці управління проектами). Перед тим, як вивчати методи MSF, слід спочатку визначити, який прикладної варіант MSF мається на увазі.

Найбільш популярні прикладні варіанти MSF, розроблені Microsoft:

– методика впровадження рішень в області Управління проектами;

– методика управління IT-проектами на базі методологій MSF і Agile.

Важливість прикладних варіантів MSF підкреслює той факт, що в «чистому варіанті» саму методику MSF в своїх IT-проектах компанія Microsoft не використовує.

У проектах Microsoft Consulting Services використовується гібридна методологія MSF і Agile. Незважаючи на зовнішні істотні відмінності прикладних варіантів MSF, розроблених експертами Microsoft, загальна база методів MSF для них залишається загальна і відображає загальні методологічні підходи до ітеративному ведення проектів.

MOF покликаний забезпечити організації, що створюють критично важливі (mission-critical) IT рішення на базі продуктів і технологій Microsoft, технічним керівництвом по досягненню їх надійності (reliability), доступності (availability), зручності супроводу (supportability) і керованості (manageability). MOF зачіпає питання, пов'язані з організацією персоналу і процесів, технологіями і менеджментом в умовах складних (complex), розподілених (distributed) і різнорідних (heterogeneous) IT-середовищ. MOF заснований на кращих виробничих методиках, зібраних в IT Infrastructure Li brary (ITIL), складених Central Computer and Telecommunications Agency – Агентством уряду Великобританії.

Створення бізнес-рішення в рамках відведених часу і бюджету вимагає наявності випробуваної методологічної основи.

MSF пропонує перевірені методики для планування, проектування, розробки та впровадження успішних IT-рішень. Завдяки своїй гнучкості,

масштабованості і відсутності жорстких інструкцій MSF здатний задовольнити потреби організації або проектної групи будь-якого розміру.

Методологія MSF складається з принципів, моделей і дисциплін з управління персоналом, процесами, технологічними елементами і пов'язаними з усіма цими факторами питаннями, характерними для більшості проектів.

MSF складається з двох моделей і трьох дисциплін. Вони докладно описані в 5 whitepapers. Починати вивчення MSF краще з моделей, а потім перейти до дисциплін.

MSF містить:

1. Моделі: модель проектної групи; модель процесів.

2. Дисципліни: дисципліна управління проектами; дисципліна управління ризиками; дисципліна управління підготовкою.

Модель проектної групи MSF (MSF Team Model) описує підхід Майкрософт до організації працює над проектом персоналу і його діяльності з метою максимізації успішності проекту. Дана модель визначає рольові кластери, їх області компетенції та зони відповідальності, а також рекомендації членам проектної групи, що дозволяють їм успішно здійснити свою місію по втіленню проекту в життя.

Модель проектної групи MSF розроблялася протягом кількох років і виникла в результаті осмислення недоліків пірамідальною, ієрархічної структури традиційних проектних груп.

Відповідно до моделі MSF проектні групи будуються як невеликі багатопрофільні команди, члени яких розподіляють між собою відповідальність і доповнюють області компетенцій один одного. Це дає можливість чітко сфокусувати увагу на потребах проекту. Проектну групу об'єднує єдине бачення проекту, прагнення до втілення його в життя, високі вимоги до якості роботи і бажання самовдосконалюватися.

Нижче описуються основні принципи, ключові ідеї і випробувані методики MSF в застосуванні до моделі проектної групи.

MSF включає в себе ряд основних принципів. Ось ті з них, які мають відношення до успішної роботи команди:

1. Розподіл відповідальності при фіксації звітності.
2. Наділяє членів команди повноваженнями.
3. Концентруйтеся на бізнес-пріоритети.
4. Єдине бачення проекту.
5. Проявляйте гнучкість – будьте готові до змін.
6. Заохочуйте вільне спілкування.

Успішне використання моделі проектної групи MSF ґрунтується на ряді ключових концепцій (key concepts):

– Команда соратників.

– Сфокусованість на потреби замовника.

– Націленість на кінцевий результат.

– Установка на відсутність дефектів.

– Прагнення до самовдосконалення.

– Зацікавлені команди працюють ефективно.

MSF заснований на постулаті про шести якісних цілях, досягнення яких визначає успішність проекту. Ці цілі обумовлюють модель проектної групи. У той час як за успіх проекту відповідальна вся команда, кожен з її рольових кластерів, які визначаються моделлю, асоційований з однієї зі згаданих шести цілей і працює над її досягненням.

В проектну групу входять такі рольові кластери:

– Управління програмою.

– Управління продуктом.

– Розробка.

– Тестування.

– Управління релізом.

– Задоволення споживача.

Вони відповідальні за різні області компетенції (functional areas) і пов'язані з ними цілі і завдання. Іноді рольові кластери називаються просто ролями. Але в будь-якому випадку суть концепції залишається тією ж – Побудувати основу виробничих відносин і пов'язану з нею модель команди такими, щоб вони були пристосованими (масштабованими) для задоволення потреб будь-якого проекту.

Як вже було сказано вище, проектна група з MSF складається з шести рольових кластерів, кожен з яких відповідає за:

– Управління програмою (program manager) – розробку архітектури рішення, адміністративні служби.

– Розробку (developer) – розробку додатків та інфраструктури, технологічні консультації.

– Тестування (QAE) – планування, розробку тестів і звітність по тестах.

– Управління випуском (release manager) – інфраструктуру, супровід, бізнес-процеси, випуск готового продукту.

– Задоволення замовника (user experience) – навчання, ергономіку, графічний дизайн, технічну підтримку.

– Управління продуктом (product manager) – бізнес-пріоритети, маркетинг, представництво інтересів замовника.

Наявність шести рольових кластерів не означає, що кількість членів команди має бути кратним шести – одна людина може поєднувати кілька ролей і навпаки, рольової кластер може складатися з декількох осіб в залежності від розміру проекту, його складності і професійних навичок, необхідних для реалізації всіх областей компетенції кластера. Мінімальний колектив по MSF може складатися всього з трьох осіб. Модель не вимагає призначення окремого співробітника на кожен рольової кластер. Сенс полягає в тому, що в команді повинні бути представлені всі шість якісних цілей. Зазвичай, виділення як мінімум одну людину на кожен рольової кластер забезпечує повноцінне увагу до інтересів кожної з ролей, але це економічно виправдано не для всіх проектів. Найчастіше члени проектної групи можуть об'єднувати ролі.

У малих проектних групах об'єднання ролей є необхідним. При цьому повинні дотримуватися дві засади:

1. Роль команди розробників не може бути об'єднана ні з якою іншою роллю.

2. Уникнення поєднання ролей, що мають зумовлені конфлікти інтересів.

Як і в будь-який інший командної діяльності, відповідна комбінація ролей залежить від самих членів команди, їх досвіду і професійних навичок. На практиці поєднання ролей зустрічається нерідко. І якщо проектна група виробляє його обдумано і управляє пов'язаними з таким об'єднанням ризиками, що виникають проблеми будуть мінімальними.

MSF не надає конкретних рецептів управління проектами і не містить пояснень різноманітних методів роботи, які застосовують досвідчені менеджери. Принципи MSF формують такий підхід до управління проектами, при якому:

– Відповідальність за управління проектом розподілена між лідерами рольових кластерів всередині команди – кожен член проектної групи відповідає за загальний успіх проекту і якість створюваного продукту.

– Професійні менеджери виступають в якості консультантів і наставників команди, а не виконують функції контролю над нею – в ефективно працюючій команді кожен її член має необхідні повноваження для виконання своїх обов'язків і впевнений, що отримає від колег все необхідне.

Як випливає з вищесказаного, одна з характерних особливостей MSF – відсутність посади менеджера проекту.

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм SEED – у криптографії симетричний блоковий криптоалгоритм на основі Мережі Фейстеля, розроблений Корейським агентством інформаційної безпеки (Korean Information Security Agency, KISA) в 1998 році. В алгоритмі використовується 128-бітний блок і ключ довжиною 128

біт. Алгоритм одержав широке поширення й використовується фінансовими й банківськими структурами, виробничими підприємствами й бюджетними установами Південної Кореї, оскільки 40-бітний SSL не забезпечує на даний момент мінімально необхідного рівня безпеки. Агентством по теорії побудови комп'ютерних мереж, теорії захисту інформації специфіковане використання шифру SEED у протоколах TLS і S/MIME. У той же час, алгоритм SEED не реалізований у більшості сучасних браузерів і інтернет-додатків, що утрудняє його використання в даній сфері поза межами Південної Кореї.

SEED являє собою мережу Фейстеля з 16 раундами, 128-бітовими блоками й 128-бітовим ключем. Алгоритм використовує дві 8×8 таблиці підстановки, які, як такі з Safer, виведені з дискретного зведення в ступінь (у цьому випадку, x^{247} і x^{251} – плюс деякі «несумісні операції»). Це є деякою подібністю с MISTY1 у рекурсивності його структури: 128-бітовий повний шифр – мережа Фейстеля з F-функцією, що впливає на 64-бітові половини, у той час як сама F-функція – Мережа Фейстеля, складена з G-функції, що впливає на 32-розрядні половини. Однак рекурсія не простягнеться далі, тому що G-функція – не Мережа Фейстеля. В G-функції 32-розрядне слово розглядають як чотири 8-бітових байта, кожний з яких проходить через одну або іншу таблицю підстановки, потім поєднується в помірковано комплексному наборі булевих функцій таким чином, що кожний біт виводу залежить від 3 з 4 вхідних байтів.

SEED має складний ключовий розклад, генеруючи тридцять два 32-розрядних додаткових символу, використовуючи G-функції на серіях обертань вихідного неопрацьованого ключа, комбінованого зі спеціальними раундовими константами (як в TEA) від «Золотого співвідношення» (англ. Golden ratio).

Згідно з дослідженнями KISA, алгоритм SEED «надійно протистоїть відомим атакам».

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ антивірусного комплексу дата-центу яке зображено на рисунку 5.1а та 5.1б. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

– Меню розділів: Загальне; Контроль процесів; Додатково; Фільтр сканування.

– Розділу обрання директорій.

– Розділу обрання типу сканування.

– Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.

– Функціональних кнопок ПЗ.

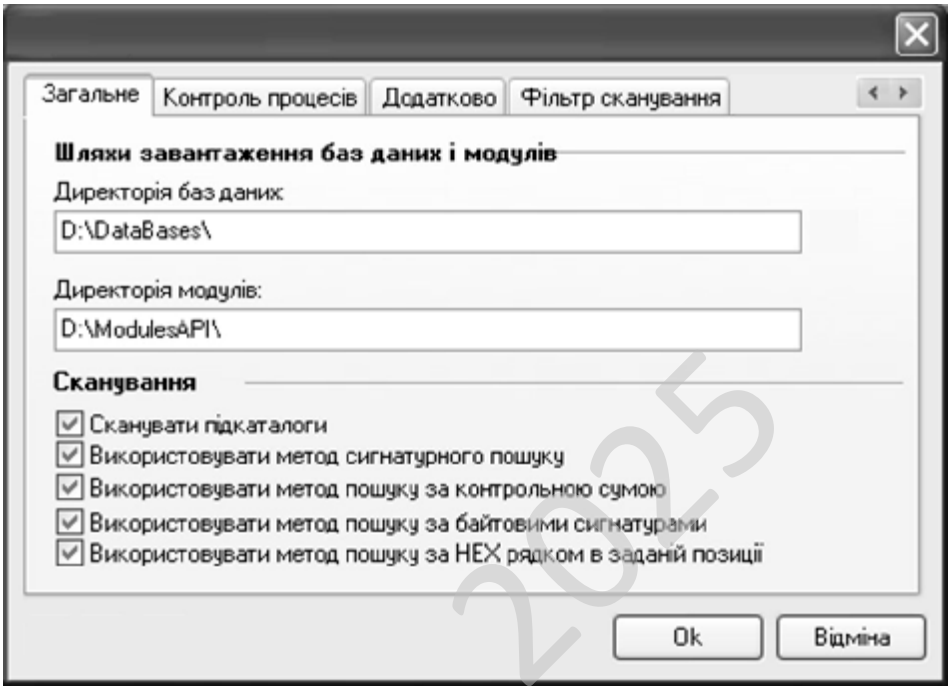
Розроблена програма має дуже простий і інтуїтивно зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий.

Якщо програма не видала ніяких помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

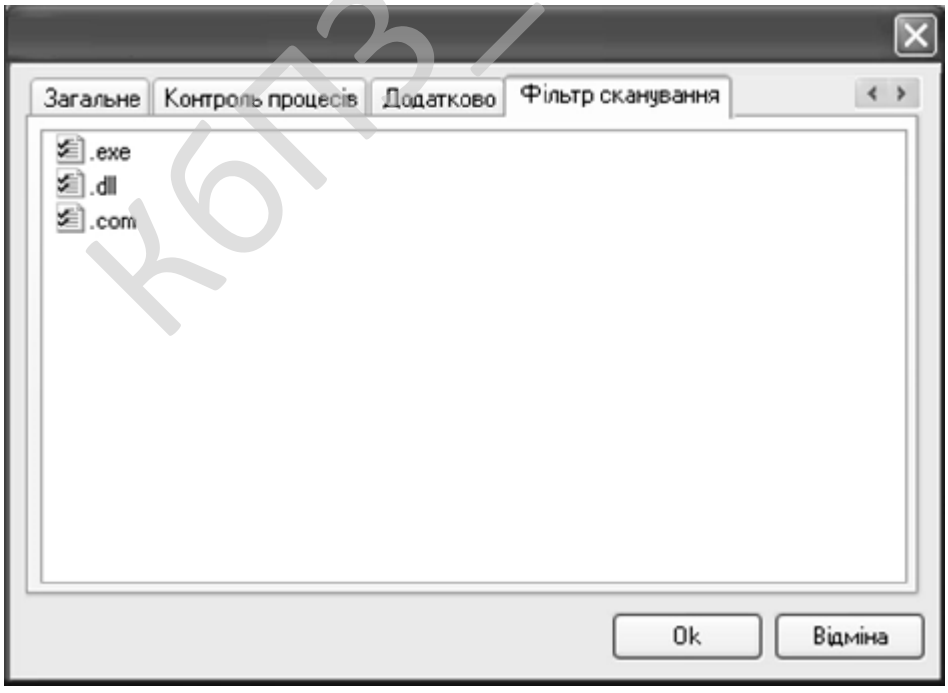
Розглянемо процес впровадження програмного забезпечення, це процес налаштування програмного забезпечення під певні умови використання, а також навчання користувачів роботі з програмним продуктом. Впровадження програмного забезпечення це усі дії, що роблять розроблену програмну систему готовою до використання. Даний процес є частинною життєвого циклу програмного забезпечення.

Загалом процес розгортання складається з кількох взаємопов'язаних дій із можливими переходами між ними. Ця активність може відбуватися як з боку виробника так і з боку споживача. Оскільки кожна програмна система є

унікальною, то усі процеси та процедури під час розгортання важко передбачити. Тому, "розгортання" можна трактувати як загальний процес відповідно до певних вимог та характеристик. Розгортання може здійснюватись програмістом і в процесі розробки програмного забезпечення.



а



б

Рисунок 5.1 – Головне вікно ПЗ

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

– Випуск.

– Встановлення та активація.

– Деактивація.

– Адаптація.

– Обновлення.

– Вмонтування.

– Відстежування версій.

– Видалення.

– Вилучення з обігу.

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

– Виділення критичних, з точки зору загального результату, процедур в діяльності організації. Коли набір таких процедур визначений, необхідно в першу чергу використовувати ІТ рішення для автоматизації операцій усередині саме цих процедур. Таким чином, розроблене ІТ рішення автоматично стає життєво важливим і затребуваним для організації, а також буде забезпечена публічність процесу впровадження;

– Розширення нормативної бази організації шляхом включення до неї регламентів, що описують порядок виконання процедур автоматизованих процесів. В іншому випадку є небезпека виникнення неузгодженості між автоматизованими процедурами та іншими процесами організації.

– Виконання робіт з загальної стандартизації існуючої діяльності організації, коли виділяються кращі практики виконання процедур і включаються в ІТ рішення за принципом найбільшої корисності для більшості учасників. Відсоток таких процедур щодо загального обсягу автоматизації може бути невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

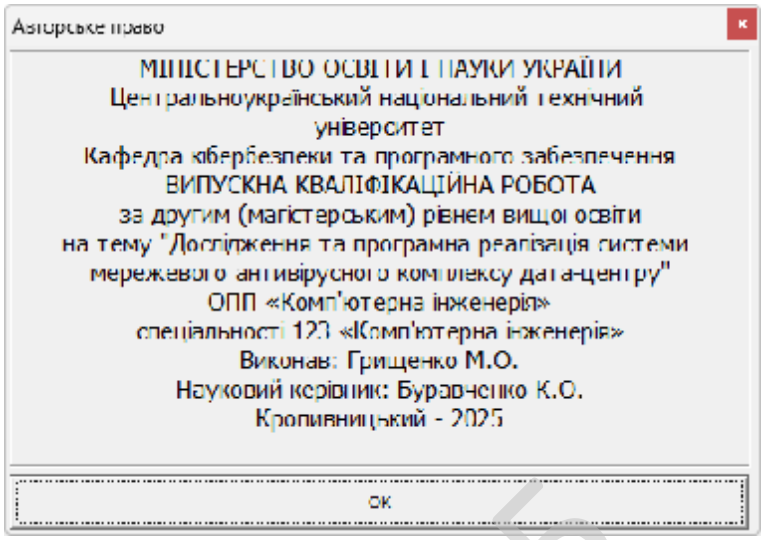


Рисунок 5.2 – Авторське право

Під час роботи над програмою було проведено тестування програмного забезпечення, тобто технічне дослідження, призначене для виявлення інформації про якість продукту відносно контексту, в якому воно має використовуватись.

Тестування включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою їх оцінки.

Проводилась оцінка:

– відповідності поставленим вимогам;

– правильна відповідь для усіх можливих вхідних даних;

– виконання функцій за прийнятний час;

– практичність;

– сумісність з ОС та стороннім ПЗ.

Оскільки число можливих тестів для програмних компонент практично нескінченне, тому стратегія тестування полягала в тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів.

Як результат ПЗ тестувалось стандартним виконанням програми з метою виявлення помилок або інших дефектів.

Проводилось тестування чорної скриньки. Основне місце програми тестів «чорної скриньки» – інтерфейс ПЗ. Відомі: функції програми. Досліджується: робота кожної функції на всій області визначення.

Ці тести демонструють:

– Як виконуються функції програми.

– Як приймаються вихідні дані.

– Як виробляються результати.

– Як зберігається цілісність зовнішньої інформації.

При тестуванні «чорної скриньки» розглядаються системні характеристики програм, ігнорується їхня внутрішня логічна структура. Вичерпне тестування, як правило, неможливе.

Наприклад, якщо в програмі 10 вхідних величин і кожна приймає по 10 значень, то кількість тестових варіантів становитиме 10^10. Тестування «чорної скриньки» не реагує на багато особливостей програмних помилок.

Тестування «чорної скриньки» (функціональне тестування) дозволяє отримати комбінації вхідних даних, які забезпечують повну перевірку всіх функціональних вимог до програми.

Програмний виріб тут розглядається як «чорна скринька», чию поведінку можна визначити тільки дослідженням його входів та відповідних виходів. При такому підході бажано мати:

– Набір, утворений такими вхідними даними, які призводять до аномалій у поведінці програми (назвемо його ITc).

– Набір, утворений такими вхідними даними, які демонструють дефекти програми (назвемо його OT).

Будь-який спосіб тестування «чорної скриньки» повинен:

– Виявити такі вхідні дані, які з високою ймовірністю належать набору ITc;

– Сформулювати такі очікувані результати, які з високою імовірністю є елементами набору ОТ.

Принцип «чорної скриньки» не альтернативний принципу «білої скриньки». Скоріше це доповнює підхід, який виявляє інший клас помилок.

Тестування «чорної скриньки» забезпечує пошук наступних категорій помилок:

– Некоректних чи відсутніх функцій;

– Помилок інтерфейсу;

– Помилок у зовнішніх структурах даних або в доступі до зовнішньої бази даних;

– Помилок характеристик (необхідна ємність пам'яті і т.д.);

– Помилок ініціалізації та завершення.

Обрано умови розповсюдження – Freeware.

Це власницьке програмне забезпечення, котре можна Безоплатно використовувати протягом необмеженого терміну без обмежень у функціональності, і поширюване без сирцевих кодів.

Автори такого програмного забезпечення, як правило, хочуть «дати щось спільноті», але хочуть також контролювати його подальшу розробку. Іноді, коли програмісти вирішують припинити розробку, вони передають сирцевий код іншим програмістам, або ж спільності як вільне програмне забезпечення.

Дуже часто плутають поняття «безплатне програмне забезпечення» та «вільне програмне забезпечення», хоча вони суттєво відрізняються.

Безплатне програмне забезпечення можна безоплатно встановлювати та використовувати (іноді з певними обмеженнями, як, наприклад, «безплатне для домашнього або некомерційного вжитку»), в той час як вільне програмне забезпечення можна продавати за будь-яку суму, але при тому, у користувача, котрий його отримує, повинні бути права на вивчення, модифікацію та поширення сирцевих кодів одержаної програми.

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи мережевого антивірусного комплексу дата-центру.

Метою розробки є дослідження та програмна реалізація системи мережевого антивірусного комплексу дата-центру.

Об'єктом дослідження є процес мережевого антивірусного комплексу дата-центру.

Предметом дослідження є методи мережевого антивірусного комплексу дата-центру.

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

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

– Удосконалено метод мережевого антивірусного комплексу дата-центру.

– Розроблено вітчизняний продукт мережевого антивірусного комплексу дата-центру, який має більш широкі можливості, на відміну від існуючих аналогів.

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та розробки системи мережевого антивірусного комплексу дата-центру можуть бути насамперед корисними для підприємств, які мають розгалужену ІТ-інфраструктуру й використовують сервери, мережеве обладнання та корпоративні сервіси для підтримки своєї діяльності. Для таких компаній стабільність і безперебійність роботи інформаційних систем є критично важливими, тому можливість своєчасного виявлення несправностей або перевантажень стає суттєвою конкурентною перевагою. Саме система моніторингу допомагає контролювати роботу мережевих пристроїв у режимі реального часу, виявляючи проблеми ще до того, як вони вплинуть на користувачів.

Особливий інтерес до таких систем можуть проявити ІТ-компанії, які займаються наданням послуг хостингу, розробкою програмного забезпечення або підтримкою клієнтів. Для них швидкість реагування на інциденти та якість технічного обслуговування є показниками репутації, а отже, від роботи системи моніторингу залежить рівень довіри клієнтів і лояльність користувачів. Такі підприємства часто працюють у середовищі, де навіть хвилинна затримка чи зупинка сервера призводить до фінансових збитків, тому автоматизація контролю за станом мережі – це не розкіш, а необхідність.

Крім комерційних компаній, результати дослідження будуть актуальними для державних структур, освітніх установ і організацій, які мають внутрішні мережі та зберігають великі обсяги інформації. У таких установах впровадження системи моніторингу підвищує ефективність роботи ІТ-відділів, зменшує ризик втрати даних і допомагає раціонально використовувати наявні ресурси.

Не менш важливим є значення цієї розробки для навчальних і наукових закладів. Вони можуть використовувати систему як навчальну платформу для підготовки фахівців у сфері інформаційних технологій. Студенти отримають можливість не лише спостерігати за реальною роботою системи моніторингу, а й аналізувати дані, моделювати різні ситуації та вчитися реагувати на інциденти. Таким чином, результати дослідження мають універсальний характер і можуть бути впроваджені як у бізнесі, так і в освіті.

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Для оцінки привабливості програмного продукту було проведено експертне опитування серед фахівців у галузі IT-інфраструктури, адміністраторів систем і представників компаній, що мають досвід використання схожих рішень. Експертам було запропоновано оцінити систему за основними критеріями – функціональні можливості, надійність, простота впровадження, масштабованість, вартість експлуатації та потенційна економічна ефективність.

Більшість експертів високо оцінили саме інтелектуальну частину системи – можливість автоматичного сповіщення про інциденти, генерацію аналітичних звітів і прогнозування потенційних відмов обладнання. Особливо було відзначено, що система працює стабільно навіть при великому навантаженні й може адаптуватися до різних типів мережевої інфраструктури, що робить її універсальною.

За результатами оцінки середній рівень привабливості продукту склав 8,7 бала з 10 можливих. Експерти зазначили, що така система може мати великий попит серед середніх і великих підприємств, особливо якщо її вартість залишатиметься конкурентною. Також було підкреслено, що простота інтерфейсу та можливість кастомізації під конкретного користувача є суттєвими перевагами, які підвищують комерційний потенціал рішення.

Таким чином, метод експертних оцінок показав, що система має високу ринкову привабливість, відповідає актуальним потребам бізнесу та може стати успішним продуктом за умови належного маркетингового просування та підтримки користувачів.

7.3 Вибір методу оцінки вартості ПЗ

Для оцінки вартості розробки системи мережевого антивірусного комплексу дата-центру доцільно використовувати витратний метод. Він передбачає визначення всіх фактичних витрат, які були понесені під час створення програмного продукту, включаючи оплату праці розробників, витрати на апаратне забезпечення, ліцензії, тестування та впровадження. Такий підхід дозволяє точно визначити базову собівартість проєкту, що є особливо важливим для невеликих команд і стартапів.

Однак, у випадку комерційного впровадження, доцільно поєднати цей підхід із дохідним методом. Дохідний метод дає змогу оцінити майбутні вигоди, які підприємство отримає після впровадження системи. Наприклад, скорочення простоїв серверів, підвищення ефективності роботи персоналу та зменшення витрат на ручну діагностику мережі є прямими джерелами економічної вигоди.

Такий комбінований підхід дозволяє не лише визначити початкову вартість розробки, а й обґрунтувати економічну доцільність проєкту. Він допомагає потенційним інвесторам побачити не просто витрати, а реальні фінансові перспективи, які відкриває впровадження системи.

У результаті використання комбінованої моделі оцінки можна отримати повну картину вартості та окупності проєкту, що стане основою для прийняття управлінських рішень щодо його реалізації чи масштабування.

7.4 Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості

Компанія має розгалужену IT-інфраструктуру, яка включає сервери, мережеве обладнання, робочі станції, системи зберігання даних і корпоративні сервіси. До впровадження системи моніторингу контроль за станом мережі здійснювався вручну: адміністратори виявляли проблеми лише після звернень користувачів або повного виходу сервісів із ладу. Це призводило до простоїв, затримок у роботі та фінансових втрат. Основна мета впровадження системи мережевого моніторингу – забезпечити цілодобове автоматичне відстеження стану обладнання, серверів і додатків, оперативне реагування на інциденти, зниження кількості простоїв і запобігання критичним збоям у роботі IT-інфраструктури. Вхідні дані зафіксовано в таблиці 7.1.

Таблиця 7.1 – Вихідні дані для розрахунку

Показник	До впровадження	Після впровадження	Економічний ефект
Кількість простоїв серверів на рік	20 випадків	5 випадків	–15
Середня тривалість простою одного сервера	4 години	1 година	–3 години
Середні втрати підприємства за 1 годину простою	25 000 грн	5 000 грн	–20 000 грн
Витрати на ручну діагностику й усунення збоїв	300 000 грн/рік	150 000 грн/рік	–150 000 грн
Вартість впровадження системи моніторингу	—	—	450 000 грн
Річні витрати на підтримку системи	—	—	100 000 грн

Розрахунок економічного ефекту демонструє наступне: зменшення збитків від простоїв – 1 975 000 грн/рік, економія на технічному обслуговуванні – 150 000 грн/рік, сукупний річний ефект – 2 125 000 грн/рік, чистий ефект – 2 025 000 грн/рік, термін окупності (Payback Period) – 0,22 року (~2,5 місяці), коефіцієнт ефективності (ROI) – 450%.

Додаткові (немонетарні) переваги: підвищення стабільності IT-інфраструктури завдяки ранньому виявленню збоїв, зменшення навантаження на IT-персонал через автоматизацію моніторингу, покращення SLA (Service Level Agreement) і задоволеності користувачів, прогнозування потенційних проблем через аналітику та звітність у реальному часі, зростання репутації підприємства, адже мінімізуються ризики затримок у наданні послуг або збою критичних бізнес-процесів.

Таким чином, моніторинг стає не лише технічним інструментом, а й важливою складовою операційної надійності та конкурентоспроможності підприємства.

7.5 Пропозиція алгоритму просування проєкту розробки ПЗ

Просування системи моніторингу має будуватися на поетапному підході, що включає як технічну демонстрацію, так і інформаційне просування. На першому етапі варто створити пілотний проєкт і запропонувати його впровадження у невеликій кількості підприємств для збору відгуків і реальних кейсів. Це дозволить перевірити ефективність системи в реальних умовах і створити довіру до продукту.

Далі важливо забезпечити інформаційну присутність продукту – через участь у галузевих конференціях, IT-форумах, онлайн-презентаціях і спеціалізованих публікаціях. Саме через публічну експертну комунікацію формується репутація розробника та усвідомлення цінності рішення на ринку.

Наступним етапом є розширення партнерських зв’язків. Доцільно співпрацювати з ІТ-компаніями, які займаються інтеграцією корпоративних систем, адже вони можуть пропонувати продукт своїм клієнтам як частину комплексного рішення. Водночас слід розробити гнучку цінову політику – наприклад, ліцензування за кількістю пристроїв або модель передплати, що зробить продукт доступнішим для малого та середнього бізнесу.

Просування має супроводжуватися технічною підтримкою користувачів, оновленнями та навчанням персоналу. Це створює позитивний досвід використання продукту та сприяє формуванню довгострокових відносин із клієнтами. У підсумку правильна стратегія просування допоможе не лише збільшити продажі, а й побудувати впізнаваний бренд на ринку ІТ-рішень.

7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Для оптимізації каналів збуту варто поєднати прямі продажі з цифровими платформами розповсюдження програмного забезпечення. Власний сайт компанії може стати не лише вітриною продукту, а й каналом комунікації з клієнтами, де вони зможуть отримати демо-версію, консультацію або підтримку. Це сприятиме зниженню витрат на маркетинг і збільшенню довіри.

Додатково ефективним буде впровадження партнерської програми для системних інтеграторів і реселерів, які вже мають доступ до корпоративних клієнтів. Така модель дозволяє розширити охоплення ринку без суттєвих додаткових інвестицій. Також можна запропонувати гібридну форму реалізації: ліцензування для великих компаній і модель SaaS (Software as a Service) для малого бізнесу. Це підвищить доступність системи та дозволить гнучко реагувати на потреби різних сегментів ринку.

Ключовим напрямом оптимізації збуту є створення якісного сервісу після продажу – технічна підтримка, регулярні оновлення, аналітичні звіти. Усе це забезпечує стабільність роботи клієнта й стимулює його до подальшої співпраці.

7.7 Визначення ключових факторів успіху конкретного проєкту

Основним фактором успіху є стабільність і надійність системи. Якщо система моніторингу працює без збоїв і забезпечує реальну користь, вона швидко здобуває довіру користувачів. Технологічна якість продукту, його здатність масштабуватися й інтегруватися з іншими ІТ-рішеннями відіграють ключову роль у його життєздатності.

Другим важливим чинником є професійна команда розробників і технічної підтримки. Клієнти цінують не лише продукт, а й можливість отримати швидку допомогу у випадку проблем або питань. Від рівня компетенції фахівців залежить не лише якість обслуговування, а й довгострокові відносини з партнерами.

Не менш значущим є гнучкість системи – можливість адаптувати її під специфіку кожного клієнта. Різні компанії мають різну інфраструктуру, тому універсальне, але налаштовуване рішення стає перевагою.

І, нарешті, успіх будь-якого ІТ-проєкту визначається здатністю постійно вдосконалюватися. Регулярні оновлення, впровадження нових технологій і зворотний зв’язок із користувачами формують довіру й підтримують актуальність продукту на ринку. Саме ці чинники разом створюють основу для стабільного розвитку та комерційного успіху системи моніторингу.

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. Тому для розробника хмарних сервісів так важливо розробити зручний інтерфейс для зручного сприйняття інформації, та необхідний функціонал, який буде відповідати необхідним вимогам та навантаженням. Все це вимагає багато часу та великого навантаження з боку розробників. Тому так важливо слідкувати за умовами праці, в яких відбувається робочий процес. Оскільки захворювання можуть бути спричинені надмірним фізичним або розумовим навантаженням, через велику нервово-емоційну напругу, або через виробниче середовище. В даному розділі магістерської роботи проведемо аналіз основних чинників при роботі програміста.

Законом України "Про охорону праці" регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», яким затверджено нормативно-правовий акт з охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98.

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Електронно-обчислювальна машина (ЕОМ) та інше обладнання є джерелами небезпеки ураження електричним струмом. Так як робота програміста характеризується істотним зоровим навантаженням, то вимагає належного освітлення. У приміщенні, в якому працюють люди (у т.ч. програмісти) необхідно створити належний мікроклімат, параметри якого регламентуються, Державними санітарними правилами і нормами, зокрема ДСанПіН 3.3.2.007-98.

На роботу програміста впливають наступні фактори: невідповідний мікроклімат приміщення (температура, вологість), недостатня освітленість робочої зони, підвищений рівень шуму та електромагнітного випромінювання, порушення іонного складу повітря, неправильна ергономічна організація робочого місця, ризики, пов'язані із погіршенням зору, порушенням фізичного стану, стресом тощо.

Шкідливими факторами при роботі з персональним комп'ютером є неіонізуюче випромінювання промислової частоти, збільшене нервово- емоційне навантаження на оператора, збільшення навантаження на органи зору та дрібні стереостатичні рухи кінцівок. Ці фактори можуть викликати у працівника певні розлади здоров'я, зокрема підвищення артеріального тиску, кон'юктивіти, тендовагініти та інші захворювання.

Комп'ютер, як і будь-який електричний прилад, особливо при його неправильному підключенні, може бути джерелом ураження оператора електричним струмом. Саме тому всі працівники, які працюють з персональним комп'ютером, повинні мати першу (або другу) групу допуску з електробезпеки.

Через наявність зазначених факторів працівники, які працюють з персональними комп'ютерами, підлягають попередньому та періодичному медичному огляду згідно з пунктом 6.2.3 додатку 4 до наказу Міністерства охорони здоров'я України «Про затвердження Порядку проведення медичних оглядів працівників певних категорій» від 21 травня 2007 року № 246.

8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

Оптимальна температура в приміщенні для праці має становити 20-24°С, відносна вологість – 40-60 %, атмосферний тиск – 750 мм. рт. ст., запиленість не повинна перевищувати 10 мг/м^3, швидкість руху повітря – 0,1 м/с.

Через те, що обчислювальна техніка є джерелом тепловиділення, організація мікроклімату потребує додаткових зусиль: кондиціонування, провітрювання, використання систем опалення тощо. Об’єм приміщень повинен передбачатися з урахуванням як мінімум 20 м^3 /на особу [4].

Монітори комп’ютерів є джерелом випромінювання, яке може зашкодити здоров’ю людини. Для убезпечення роботи з комп’ютером відстань від монітора повинна становити не менше 50 см, бажано використовувати монітори зі зниженим рівнем, скорочувати час безперервної роботи за комп'ютером (робити п’ятнадцяти хвилинні перерви після кожних півтори години праці). Також в приміщенні необхідно встановлювати іонізатори повітря, використовувати нейтралізатори та зволожувачі.

Комп’ютери та периферійні пристрої є джерелами шуму, висока інтенсивність якого може призвести до проблем з органами слуху та негативно впливати на психологічний стан. Рівень шуму на робочому місці не повинен перевищувати 50 дБА [5]. Для зменшення рівня шуму можна використовувати звукопоглинальні пристрої, а стіни приміщень з комп’ютерами можуть бути покриті звукопоглинальними матеріалами. Поряд із шумом часто виникає вібрація. Для зменшення рівня вібрації в приміщенні на поверхні необхідно встановлювати віброізолятори.

Ергономічні показники робочого місця програміста мають бути наступними: висота робочої поверхні повинна складати 720 мм, розмір поверхні має становити 1600 х 1000 мм; під столом повинен бути простір з розмірами по глибині 650 мм; стіл повинен мати підставку для ніг, розташовану під кутом

15°до поверхні; відстань клавіатури від краю столу має бути не більше 300 мм; відстань між очима й екраном повинна складати 40 – 80 см; стілець повинен мати підйомно-поворотний механізм; висота сидіння має регулюватися в межах 400 – 500 мм, глибина – не менше 380 мм, а ширина – не менше 400 мм, висота опорної поверхні спинки має бути не менше 300 мм, ширина – не менше 380 мм. Кут нахилу спинки стільця до площини сидіння повинен змінюватися в межах 90 – 110° [6].

Проведений аналіз показує, що показники мікроклімату в приміщенні відповідають установленим нормам. Штучне опалення застосовується у холодний період року.

В літню пору застосовується кондиціонер.

Для боротьби з пилом робляться регулярні провітрювання та вологі прибирання приміщенні.

У приміщенні знаходяться наступні джерела шуму: принтер Prinics PicKit M1 Smartphone Photo Printer White, електродвигуни вентиляторів ЕОМ.

Робота програміста передбачає постійний візуальній контакт з моніторами комп'ютерів, та, як наслідок, значне навантаження на зір. Традиційно, це зорова робота високої або середньої точності. Для зорової роботи високої точності загальне освітлення (розподіл світла у всьому об'ємі приміщення) має становити 300 лк, комбіноване освітлення (поєднання загального і місцевого освітлення) – 750 лк. Штучне освітлення повинно бути рівномірним та використовуватися в світлий і темний час доби. Джерелами штучного освітлення можуть слугувати люмінесцентні лампи. Правильне освітлення передбачає уникнення відблисків на екранах.

З 2019 року діють Державні будівельні норми України "Природне і штучне освітлення" – ДБН В.2.5-28:2018 [4], у яких прописані вимоги до використання всіх освітлювальних приладів, у т.ч. світлодіодних.

Працю працівника, який постійно працює за комп'ютером, згідно ДБН В.2.5-28:2018 [4], можна віднести до роботи з малою точністю (найменший

розмір об'єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об'єкта розрізнення (символів на екрані дисплея), з темним тлом (під розряд зорової роботи В). Приміщення можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при поєднаному, спільному освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 Лк. [4], Крім того все поле зору повинно бути освітлено достатньо рівномірно – це основна гігієнічна вимога. Оскільки яскраве світло на ділянці периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

8.4 Розробка заходів з умов поліпшення охорони праці

Згідно аналізу умов праці в розглянутому приміщенні, ми одержали наступні результати:

– розмірі приміщення, у розрахунку на одному працюючого, відповідають нормативам;

– мікроклімат відповідає нормативному значенню;

– акустичні умови роботи не перевищують нормативних значень;

Таким чином можна припустити, що основною причиною можливого зниження працездатності програміста є психофізіологічний фактор, тому основна пропозиція буде така: дотримання позитивної психологічної атмосфери в колективі та регламентованого режиму праці та відпочинку, організація робочого місця з урахуванням ергономічних вимог.

Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який

повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов'язковою наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга).

Регулярна наочне знайомство персоналу із шляхами для евакуації людей із приміщення відповідно до плану евакуації, забезпечення розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв, які працюють при напрузі вище 36 В.

Так як при ураженні електричним струмом у людини може статися фібриляція шлуночків серця, в організації бажано мати дефібрилятор і підготовлений персонал для роботи з ним.

8.5 Розрахункова частина

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 4 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]:

$$F = E \cdot S \cdot K \cdot Z / n,$$

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 7 = 42 м^2);

K – коефіцієнт запасу, що враховує зменшення світлового потоку лампи в результаті забруднення світильників у 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, у нашому випадку $Z = 1,1$);

n – коефіцієнт використання світлового потоку, (відношення світлового потоку, що падає на розрахункову поверхню, до сумарного потоку від усіх ламп і обчислюється в долях одиниці [8]; 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:

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

де:

S – площа приміщення, $S = 42$ м2;

h – розрахункова висота підвісу, $h = 2,9$ м (співпадає з висотою стелі, оскільки лампи освітлення закріплюються на стелі);

A – ширина приміщення, $A = 6$ м;

B – довжина приміщення, $B = 7$ м.

Підставимо всі значення у формулу та визначимо індекс приміщення:

$$i=1,4.$$

Знаючи індекс приміщення, за знаходимо $n = 0,29$ (з табличних даних коефіцієнтів використання світлового потоку (n) світильників з відповідним типом лампам) [8]. Підставимо всі значення у формулу, визначимо світловий потік: F=71689 Лм.

Для розрахунку будемо використовувати світлодіодні стельові панелі Delux LED Panel 41 44 Вт, світловий потік яких $F_л = 3600$ Лм.

Число ламп визначається за формулою:

$$N=F/F_л$$

де:

F – світловий потік,

$F_л$ – світловий потік однієї лампи.

Підставимо всі значення у формулу та визначимо індекс приміщення:

$$N= 71689 / 3600 =19,9 \text{ шт.}$$

Приймаємо необхідну кількість світлодіодних світильників 20 шт.

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, 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 мережевого антивірусного комплексу дата-центру.

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 мережевого антивірусного комплексу дата-центру.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем мережевого антивірусного комплексу дата-центру.

– Досліджена система мережевого антивірусного комплексу дата-центру.

– На основі отриманих результатів досліджень створена програмна реалізація системи мережевого антивірусного комплексу дата-центру.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання мережевого антивірусного комплексу дата-центру.

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 Python. 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 SEED.

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.

Проведено маркетингове та економічне обґрунтування ІТ-проєкту, що дозволило визначити ключові фактори успіху даного проекту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Грищенко М.О. Дослідження та програмна реалізація системи мережевого антивірусного комплексу дата-центру // Збірник праць молодих науковців ЦНТУ. – Вип. 15. – Кропивницький: ЦНТУ, 2025.

2. Awais Rashid, Howard Chivers, George Danezis, Emil Lupu, Andrew Martin. CyBOK The Cyber Security Body of Knowledge. The National Cyber Security Centre. 2019. 854 p.

3. Loren Kohnfelder. Designing Secure Software. No Starch Press. 2022. 332 p.

4. Samir Kumar Rakshit. Ethical Hacker's Penetration Testing Guide. BPB Online. 2022. 509 p.

5. Corey J. Ball. Hacking APIs. No Starch Press. 2022. 353 p.

6. Kevin Beaver. Hacking for Dummies. John Wiley & Sons. 2022. 419 p.

7. Mark S. Merkow. Practical Security for Agile and DevOps. CRC Press. 2022. 236 p

8. Derek Fisher. Application Security Program Handbook. Manning Publications. 2021. 155 p.

9. Cameron Wyatt PH.D. Kali Linux Tutorial. Independently published. 2021. 60 p.

10. Alex Matrosov, Eugene Rodionov, Sergey Bratus. Rootkits and Bootkits. No Starch Press. 2019. 450 p.

11. Петрик В.М., Присяжнюк М.М., Аль-Файюмі Халед та ін. «Системи інформаційної зброї та технології інформаційної війни»: підручник / Петрик В.М., Присяжнюк М.М., Аль-Файюмі Халед, Жарков Я.М., Смірнов О.А, Буравченко К.О., Давидюк А.В., Кононович В.Г., Корчинский В.В., Кудирко В.М., Фесенко А.О.; за заг. ред. В.М. Петрика, М.М. Присяжнюка.– К.: Видавничий центр “Кафедра”, 2025.– 320 с.

12. Усік, П.С., Смірнова, Т.В., Буравченко, К.О., Смірнов, О.А., Улічев, О.С., Смірнов, С.А. «Дослідження технологій забезпечення кібербезпеки банківських систем з використанням штучного інтелекту». *Кібербезпека: освіта, наука, техніка.* 2025. Том 1 № 29. С.704–716, 2025

13. Kuznetsov, O., Frontoni, E., Kuznetsova, K., Arnesano, M., Smirnov, O. «A secure biometric authentication architecture for blockchain-driven cyber-physical systems». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 193–224.

14. Kuznetsov, O., Atzeni, G., Arnesano, M., Randieri, C., Smirnov, O. «Secure IoT-based smart wheelchair system: From implementation to security enhancement strategy». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 225–257.

15. Kuznetsov, O., Smirnov, O., Kuznetsova, T., Shaikhanova, A., Svatowsky, I. «Privacy-utility trade-offs in IoT networks: A comparative analysis of differential privacy mechanisms for sensor data aggregation». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 589–622.

16. Lakhno, V., Malyukov, V., Smirnov, O., Bebeshko, B., Chubaievskiy, V., Zhumadilova, M., Malyukova, I., Smirnov, S. «Multifactorial Model for Targeted Attacks Counteracting Within the Framework of a Multi-Step Quality Game with Fuzzy Information». *8th International Symposium on Intelligent Informatics, ISI 2023*, 2025. vol 389. pp 377-389. Springer, Singapore.

17. Kuznetsov O., Frontoni E., Kuznetsova Y., Smirnov O., Moskovchenko I. «Trust-Based Security Architecture for Edge Computing: A Simulation Study of Dynamic Trust Evolution and Attack Detection». *CEUR Workshop Proceedings,* 2024, 3909, pp. 227–241.

18. Kuznetsov, O., Frontoni, E., Kryvinska, N., Chevardin, V., Smirnov, O. «Wireless Network Encryption Stream Ciphers, Computational Modeling, and Security

Analysis». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 379–402.

19. Kuznetsov, O., Frontoni, E., Kryvinska, N., Smirnov, O., Imoize, G.L. «Computational Modeling of Enhanced Spread Spectrum Codes for Asynchronous Wireless Communication». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 403–447.

20. Ткаченко, О., Ільєнко, А., Улічев, О., Мелешко, Є., Смірнов, О. «Правові засади поширення інформаційних впливів в соціальних мережах». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»,* 2024. № 2(26), С. 170–188.

21. Смірнова Т.В., Конопліцька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка.* 2024. №4(24), С. 6-27.

22. Вінтенко, Б., Миронець, І., Смірнов, О., Кравчук, О., Козірова, Н., Савеленко, Г., Коваленко, А. «Дослідження вимог та аналіз кібербезпеки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Кібербезпека: освіта, наука, техніка.* 2024. №3(23), С. 111-131.

23. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем IP-телефонії». *Підводні технології,* 2024, № 13, с. 28-35.

24. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, 2023, 3628, pp. 106-115.

25. Akhalaia, G., Iavich, M., Iashvili, G., Prysiazhnyy, D., Smirnova, T. «Secure Encrypted Connection on Georgian Website». *CEUR Workshop Proceedings,* 2023, 3550, pp. 313-320.

26. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in

software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56

27. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchev, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings,* Volume 3530, 2023, pp. 256-265.

28. Kuznetsov, O., Kandiy, S., Frontoni, E., Smirnov, O. «Trade-offs in Post-Quantum Cryptography: A Comparative Assessment of BIKE, HQC, and Classic McEliece». *CEUR Workshop Proceedings,* Volume 3504, 2023, pp. 1-11.

29. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebeshko, B. «Selection of a Rational Composition of İnformation Protection Means Using a Genetic Algorithm». *In: Rajakumar, G., Du, KL., Vuppalapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies*, vol 131. 2023. Springer, Singapore. pp. 21-34.

30. Смірнова Т.В., Гнатюк С.О., Бердибаєв Р.Ш., Сидоренко В.М., Жигаревич О.К., «Система корелювання подій та управління інцидентами кібербезпеки на об'єктах критичної інфраструктури». *Кібербезпека: освіта, наука, техніка*, №*3*(19), 2023, С. 176-196.

31. Смірнов О.А. Козлов Я.О., Смірнова Т.В. «Дослідження застосування SIEM-систем для забезпечення кібербезпеки та захисту інформації». *II Міжнародна науково-практична Інтернет-конференція «Інновації та перспективні шляхи hозвитку інформаційних технологій (ІПШРІТ-2023)»* м.Черкаси 6 грудня 2023 року – Черкаси: ЧДТУ.– 2023. – С.251-252.

32. Козлов Я.О., Смірнова Т.В., Смірнов О.А. «Дослідження SIEM-систем для забезпечення кібербезпеки». *VII міжнародна науково-практична конференція "Інформаційна безпека та комп'ютерні технології" до 30-ти річчя кафедри кібербезпеки та програмного забезпечення,* м. Кропивницький. 1 листопада 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 26.

33. Козлов Я.О., Козірова Н.Л., Смірнов О.А. «Дослідження структури та принципу роботи SIEM-системи». *VII міжнародна науково-практична конференція "Інформаційна безпека та комп'ютерні технології" до 30-ти річчя кафедри кібербезпеки та програмного забезпечення,* м. Кропивницький. 1 листопада 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 59.

34. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А., Коваленко А.С. «Дослідження нормативних документів та галузевих стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 2(72), С. 170-178.

35. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» *CEUR Workshop Proceedings,* Volume 3187, 2022,

36. Smirnov O.A., Al-Oraiqat A.M., Ulichev O.S., Meleshko Ye.V., Al-Rawashdeh H.S., Polishchuk L.I. «Modeling strategies for information influence dissemination in social networks». *Journal of Ambient Intelligence and Humanized Computing* Volume 13, Issue 5. Springer, Cham. 2022, pp. 2463-2477.

37. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління,* № 2(70). 2022. С. 28-37.

38. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку,* 2022, № 3(69). С. 93-98.

39. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного

захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки»,* № 2 (307). С. 46-52. 2022.

40. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 1(67). С. 84-89.

41. Smirnov O., Kuznetsov A., Zhora V., Onikiychuk A., Pieshkova O. «Hiding Messages in Audio Files Using Direct Spread Spectrum». *11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2021,* Cracow, Poland, 22-25 September 2021. P. 414-418

42. Smirnov O., Kuznetsov A., Lokotkova I., Kuznetsova T., Florov S., Lebid O. «Using Orthogonal Signals to Hide Information in Images». *4 IEEE International Conference on Advanced Information and Communication Technologies (AICT) - 2021*, Lviv, Ukraine, September 21-25, 2021. P. 255-260.

43. Smirnov O., Kuznetsov A., Girzheva O., Kiian A., Nakisko O., Kuznetsova T. «Advanced Code-Based Electronic Digital Signature Scheme». *2020 IEEE International Conference on Problems of Infocommunications Science and Technology, PIC S and T 2020,* Kharkiv, 6 October 2020-9 October 2020, P. 358-362.

44. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». *CEUR Workshop Proceedings* Volume 2805, 2020, Pages *44-58.*

45. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». *International Journal of Computing*; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

46. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings*. Volume 2740, 2020, Pages 102-114.

47. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346.

48. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». *CEUR Workshop Proceedings* Volume 2654, 2020, Pages 122-131.

49. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». *CEUR Workshop Proceedings* Volume 2654, 2020, Pages 1-14.

50. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». *Lecture Notes in Networks and Systems*, vol 152. Springer, Cham. 2021, pp 66-84.

51. Smirnov O., Kuznetsov A., Onikiychuk A., Makushenko T., Anisimova O., Arischenko A. «Adaptive pseudo-random sequence generation for spread spectrum image steganography». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 161-165.