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 протидії зловмисним програмам з використанням методів машинного навчання”

Виконав здобувач вищої освіти
ІІ курсу, групи КН-24М
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Фоменко С.О.
«____» _____ 2025 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Минайленко Р.М.
«____» _____ 2025 р.
Рецензент _____
_____

м. Кропивницький

АНОТАЦІЯ

Фоменко С.О. Дослідження та програмна реалізація системи протидії зловмисним програмам з використанням методів машинного навчання. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2025.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи протидії зловмисним програмам з використанням методів машинного навчання.

Метою розробки є дослідження та програмна реалізація системи протидії зловмисним програмам з використанням методів машинного навчання.

Об'єктом дослідження є процес протидії зловмисним програмам з використанням методів машинного навчання.

Предметом дослідження є методи протидії зловмисним програмам з використанням методів машинного навчання.

Методи дослідження базуються на методах машинного навчання, методах захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи протидії зловмисним програмам з використанням методів машинного навчання.

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'ютерні науки, протидія зловмисним програмам, машинне навчання

ABSTRACT

Fomenko S.O. Research and software implementation of a system for combating malicious programs using machine learning methods. 122 Computer Science. 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 a system for combating malicious programs using machine learning methods.

The purpose of the development is the research and software implementation of a system for combating malicious programs using machine learning methods.

The object of the research is the process of combating malicious programs using machine learning methods.

The subject of the research is methods for combating malicious programs using machine learning methods.

The research methods are based on machine learning methods, information protection methods, mathematical statistics methods, software development methods.

The result of the work is a software implementation of a system for combating malicious programs using machine learning methods.

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 user-friendly 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.

The program was developed in the Python environment.

Keywords: computer science, countermeasures against malicious programs, machine learning

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

АС	–	антивірусна система
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. Антивірусний бізнес зараз на підйомі. Тільки в 2025 році, за даними компанії Gartner, по усьому світі було продано програм безпеки на суму в 22 мільярда доларів. Однак скільки б засобів не витрачали підприємства й частки користувачі, стовідсоткового захисту не існує.

Особливо «улюбленим» у розроблювачів антивірусів підвидом шкідливого ПЗ є програми-здирники. Усього за третій квартал минулого року було заблоковано 821 865 подібних атак на користувачів продуктів.

Жахає при цьому те, наскільки зловмисники успішні у своїй справі: за інформацією цієї компанії, через відсутність мер протидії кожна третя жертва перераховує хакерам викуп – однак 20 відсотків таких користувачів ніколи не одержать від здирників код для дешифрування своїх даних. На кожному потерпілому злочинці заробляють у середньому близько 17 000 грн. – досить вигідний бізнес.

Щодня з'являються 300 000 нових видів шкідливого ПЗ – безперервна гра в кішки-мишки для антивірусів. Стандартними базами сигнатур з небезпекою впоратися практично неможливо. Тому сучасні захисні рішення роблять ставку на аналіз поводження. У цьому випадку захисне ПЗ в реальному часі перевіряє, як поводиться передбачуваний вірус на комп'ютері. При підозрілих зверненнях сканер блокує програму й повідомляє про це користувача.

Проблема: зловмисники розробляють методи для обходу евристики, наприклад, укриваючи своїх шантажистів у серйозних продуктах. Покласти кінець такому принципу дії здатні тільки зовсім нові техніки.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи протидії зловмисним програмам з використанням методів машинного навчання.

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

– Огляд існуючих систем протидії зловмисним програмам з використанням методів машинного навчання.

– Дослідження системи протидії зловмисним програмам з використанням методів машинного навчання.

– Програмна реалізація системи протидії зловмисним програмам з використанням методів машинного навчання.

Об'єктом дослідження є процес протидії зловмисним програмам з використанням методів машинного навчання.

Предметом дослідження є методи протидії зловмисним програмам з використанням методів машинного навчання.

Методи дослідження базуються на методах машинного навчання, методах 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 протидії зловмисним програмам з використанням методів машинного навчання.

– Розроблено вітчизняний продукт протидії зловмисним програмам з використанням методів машинного навчання, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі протидії зловмисним програмам з використанням методів машинного навчання.

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 протидії зловмисним програмам з використанням методів машинного навчання, 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

Антивірус – це перша програма, яку повинні встановити на новий комп'ютер. Навіть надійний захист можна одержати абсолютно безкоштовно.

Багато користувачів справедливо помітять, що їм досить убудованого в систему антивірусу «Захисника Windows». Однак, як показують тестування незалежних лабораторій, багато сторонніх рішень набагато краще справляються із завданнями захисту, і, крім того, пропонують більше широкі функціональні можливості.

Користувачі, які зовсім зневажають антивірусним захистом, є ідеальними цілями для вірусів, шкідливих атак, експлойтів і інших погроз.

Перші антивірусні алгоритми будувалися на основі порівняння з еталоном. Мова йде про програми, у яких вірус визначається класичним ядром по деякій масці. Зміст алгоритму полягає у використанні статистичних методів. Маска повинна бути, з одного боку, маленькою, щоб обсяг файлу був прийнятних розмірів, а з іншого боку – досить великою, щоб уникнути помилкових спрацьовувань (коли «свій» сприймається як «чужий», і навпаки).

Перші антивірусні програми, побудовані по цьому принципі (так звані сканери-поліфаги), знали деяку кількість вірусів і вміли їх лікувати. Створювалися ці програми в такий спосіб: розроблювач, одержавши код вірусу (код вірусу спочатку був статичний), становив по цьому коді унікальну маску (послідовність 10-15 байт) і вносив її в базу даних антивірусної програми. Антивірусна програма сканувала файли й, якщо знаходила дану послідовність байтів, робила висновок про те, що файл інфікований. Дана послідовність (сигнатура) вибиралася таким чином, щоб вона була унікальною й не зустрічалася у звичайному наборі даних.

Описані підходи використовувалися більшістю антивірусних програм аж до середини 90-х років, коли з'явилися перші поліморфні віруси, які змінювали своє тіло по непередбаченим заздалегідь алгоритмах. Тоді сигнатурний метод був доповнений так званим емулятором процесора, що дозволяє знаходити що шифруються й поліморфні віруси, що не мають у явному виді постійної сигнатури.

Якщо звичайно умовний ланцюжок складається із трьох основних елементів: ЦПУ – ОС – Програма, то при емуляції процесора в такий ланцюжок додається емулятор. Емулятор як би відтворює роботу програми в деякому віртуальному просторі й реконструює її оригінальний уміст. Емулятор завжди здатний перервати виконання програми, контролює її дії, не даючи нічого зіпсувати, і викликає антивірусне скануюче ядро.

Другий механізм, що з'явився в середині 90-х років і який використовувався для всіх антивірусів, – це евристичний аналіз. Справа в тому, що апарат емуляції процесора, що дозволяє одержати вижимку дій, чинених аналізованою програмою, не завжди дає можливість здійснювати пошук по цих діях, але дозволяє зробити деякий аналіз і висунути гіпотезу типу «вірус або не вірус?».

У цьому випадку ухвалення рішення ґрунтується на статистичних підходах. А відповідна програма називається евристичним аналізатором.

Для того щоб розмножуватися, вірус повинен робити які-небудь конкретні дії: копіювання на згадку, запис у сектори й т.д. Евристичний аналізатор (він є частиною антивірусного ядра) містить список таких дій, переглядає виконуваний код програми, визначає, що вона робить, і на основі цього приймає рішення, є дана програма чи вірусом ні.

При цьому відсоток пропуску вірусу, навіть невідомого антивірусній програмі, дуже малий. Дана технологія зараз широко використовується у всіх антивірусних програмах.

1.2 Область застосування

Областю застосування є антивірусні програми. Антивірусні програми розвивалися паралельно з еволюцією вірусів. У міру того як з'являлися нові технології створення вірусів, ускладнювався й математичний апарат, що використовувався в розробці антивірусів.

Класифікація антивірусних програм

Класифікуються антивірусні програми на чисті антивіруси й антивіруси подвійного призначення.

Чисті антивіруси відрізняються наявністю антивірусного ядра, що виконує функцію сканування по зразках. Принциповим у цьому випадку є те, що можливо лікування, якщо відомо вірус. Чисті антивіруси, у свою чергу, по типі доступу до файлів підрозділяються на дві категорії:

– здійснюючий контроль за доступом (on access);

– на вимогу користувача (on demand).

Звичайно on access-продукти називають моніторами, а on demand-продукти – сканерами.

On demand-продукт працює за наступною схемою: користувач хоче що-небудь перевірити й видає запит (demand), після чого здійснюється перевірка. On access-продукт – це резидентна програма, що відслідковує доступ і в момент доступу здійснює перевірку.

Крім того, антивірусні програми, так само як і віруси, можна розділити залежно від платформи, усередині якої даний антивірус працює. У цьому змісті поряд з Windows або Linux до платформ можуть бути віднесені Microsoft Server, Microsoft Office.

Програми подвійного призначення – це програми, використовувані як в антивірусах, так і в ПЗ, що антивірусом не є. Наприклад, CRC-checker – ревізор змін на основі контрольних сум – може використовуватися не тільки для лову вірусів. Різновидом програм подвійного призначення є поведінкові блокатори, які

аналізують поводження інших програм і при виявленні підозрілих дій блокують їх. Від класичного антивірусу з антивірусним ядром, що розпізнає й лікує від вірусів, які аналізувалися в лабораторії і яким був прописаний алгоритм лікування, поведінкові блокатори відрізняються тим, що лікувати від вірусів вони не вміють, оскільки нічого про їх не знають. Дана властивість блокаторів дозволяє їм працювати з будь-якими вірусами, у тому числі й з невідомими. Це сьогодні здобуває особливу актуальність, оскільки розповсюджувачі вірусів і антивірусів використовують ті самі канали передачі даних, тобто Інтернет. При цьому антивірусній компанії завжди потрібно час на те, щоб одержати сам вірус, проаналізувати його й написати відповідні лікувальні модулі. Програми із групи подвійного призначення саме й дозволяють блокувати поширення вірусу до того моменту, поки компанія не напише лікувальний модуль.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи протидії зловмисним програмам з використанням методів машинного навчання, 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

Bitdefender Antivirus Free Edition

Загальна інформація:

– Підтримувані ОС: Windows, Mac, Android.

– Функції: захист від фішингу, поведінковий аналіз, автоматичне сканування.

Рисунок 2.1 – Інтерфейс користувача Bitdefender

Переваги:

– висока швидкість сканування;

– відмінне виявлення вірусів.

Недоліки:

– обмежені можливості контролю;

– відсутні перевірки за розкладом.

Безкоштовний антивірус Bitdefender побудований на антивірусному модулі, що використовується в платних продуктах компанії. Bitdefender Antivirus Free Edition є дуже легким, простим і відрізняється ефективними механізмами сканування, які працюють швидше, ніж у конкурентів. Висока ефективність виявлення шкідливого ПЗ підтверджується тестами антивірусних лабораторій. Однак, тести AV-Test, проведені в грудні 2025 року, виявили кілька недоліків при виявленні погроз нульового дня.

Антивірус працює в автоматичному режимі, і втручання користувача практично не потрібно. Bitdefender надійно виконує базові функції захисту.

Avast Free Antivirus

Загальна інформація:

– Підтримувані ОС: Windows, Mac, Android.

– Функції: виявлення вірусів, ігровий режим, менеджер паролів, антивірусний сканер.

Переваги:

– не сповільнює продуктивність ПК;

– відмінний захист від погроз.

Недоліки:

– дратівні налаштування конфіденційності;

– містить посилання на платні компоненти.

Об'єднання антивірусів Avast і AVG у єдиний продукт не відбулося дотепер, хоча фактично вони стали належати одній компанії ще в середині 2016 року. Avast офіційно оголосив, що компанія буде розвивати ці продукти окремо,

але ймовірність виходу єдиного антивірусного ПЗ зберігається. Завдяки угоді з AVG користувальницька база Avast значно розширилася й нараховує більше 400 мільйонів активних користувачів.

Новітня версія Avast Free Antivirus одержала автоматичний ігровий режим, що дозволяє блокувати повідомлення й знижувати навантаження на системні ресурси під час запуску ігор. Крім того, був серйозно перероблений інтерфейс продукту. Убудований менеджер паролів є корисним доповненням до основних захисних функцій.

Продукт демонструє високу ефективність у випробуваннях AV-Test – причому як при виявленні невідомих вірусів, так при ідентифікації новітніх погроз нульового дня. Видимо, позначається розширена користувальницька база. З іншого боку, Avast сповільнює запуск програм і іноді може дратувати достатком спливаючих вікон.

Sophos Home

Загальна інформація:

– Підтримувані ОС: Windows, Mac.

– Функції: виявлення вірусів, захист від фішингу, батьківський контроль, захист до 10 ПК.

Переваги:

– простий і ненав'язливий;

– хмарний контроль пристроїв, що захищаються.

Недоліки:

– відсутні перевірки за розкладом;

– обмежені можливості налаштування для досвідчених користувачів.

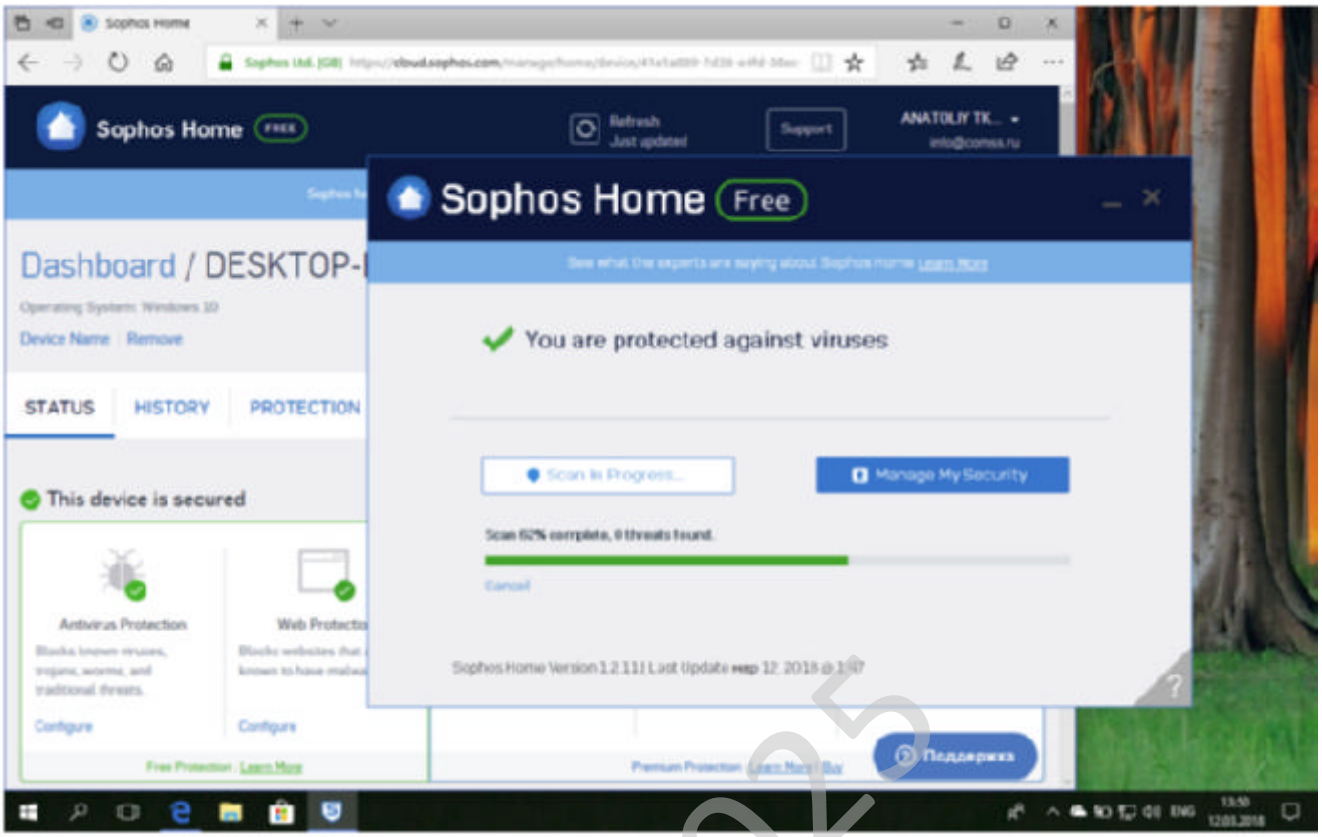

Рисунок 2.2 – Інтерфейс користувача Sophos Home

Sophos Home позиціонується розроблювачем як “захист бізнес-класу”. Продукт пропонує більше можливостей, чим більшість безкоштовних антивірусів і здається більше підходящим для родин. Ви одержуєте стандартний захист від вірусів і інших шкідливих програм, а також компоненти веб-захисту, наприклад, захист від фішингу й фільтрацію контенту. У сполученні із централізованим керуванням до 10 комп'ютерів ви одержуєте відмінний інструмент для блокування небажаного контенту на пристроях ваших дітей.

AV-Test не проводила оцінку Sophos Home, але тести AV-Comparatives повідомляють про гідний рівень виявлення й гарних антивірусних можливостей. Деяким користувачам повідомлення можуть здатися нав'язливими. З іншого боку, антивірус не буде пропонувати вам платні рішення для споживачів, тому що компанія насамперед орієнтована на бізнес сегмент.

Avira Free Antivirus

Загальна інформація:

– Підтримувані ОС: Windows, Mac, Android, iOS.

– Функції: виявлення вірусів, захист від фішингу, убудований VPN, інструменти оптимізації системи.

Переваги:

– невеликий вплив на продуктивність системи;

– високий рівень виявлення.

Недоліки:

– установник просуває інші продукти Avira;

– велика кількість спливаючих вікон під час роботи.

Avira Antivirus для Windows набирає високі бали в тестах AV-Test. В останньому раунді випробувань рішення змогло виявити 99,7% погроз тестової колекції й не зробило помітного впливу на продуктивність системи.

Безкоштовний антивірус **Avira Free Antivirus** має дуже простий, дружній і сучасний інтерфейс і відрізняється мінімальним числом помилкових спрацьовувань. Як опція ви можете запускати з безкоштовним антивірусом інші компоненти безпеки – захист від фішингу, захист від шифрувальників або VPN-Підключення з обмеженням в 500 мегабайта трафіку на місяць.

З іншого боку, безкоштовний антивірус Avira дуже нав'язливий і показує велика кількість спливаючих повідомлень і рекламних модулів. Звичайно, це прийнятно для безкоштовного продукту, але здається, що Avira переходить всі межі.

AVG AntiVirus Free

Загальна інформація:

– Підтримувані ОС: Windows, Mac, Android.

– Функції: виявлення вірусів, захист від фішингу, інструменти оптимізації системи, антивірусний сканер.

Переваги:

– гнучкі можливості налаштування;

– відмінна репутація захисту від шкідливих програм.

Недоліки:

– повільне сканування;

– не найефективніший захист від фішингу.

AVG AntiVirus Free пропонує високоякісний антивірусний захист абсолютно безкоштовно й при цьому дбайливо ставиться до ресурсів вашої системи.

У випробуваннях AV-Test показники Avast і AVG все-таки розрізняються не на користь останнього, незважаючи на той факт, що обидва продукти мають однакову користувальницьку базу. Крім того, тестування показало, що захист від фішингу має потребу в поліпшенні. Це говорить про те, що після придбання AVG, антивірус став одержувати менше уваги з боку розроблювачів. Швидше за все, у найближчому майбутньому нам варто очікувати єдиний сполучений антивірус.

Panda Free Antivirus

Загальна інформація:

– Підтримувані ОС: Windows.

– Функції: хмарне сканування, ігровий режим, “вакцинація” USB-Пристроїв, завантажувальний диск відновлення системи.

Переваги:

– гарний рівень виявлення шкідливого ПЗ;

– інструмент вакцинації USB-флешок.

Недоліки:

– помітний вплив на продуктивність.

Як показує тестування AV-Test, Panda Free Antivirus значно впливає на типові повсякденні завдання – установка програм, копіювання файлів,

завантаження додатків. Піки навантаження процесора виникають у випадковому порядку, а також під час відновлення Panda.

З позитивної сторони, Panda Free Antivirus поставляється з корисними додатковими функціями. Ігровий режим дозволить поліпшити продуктивність, коли вам потрібні ресурси процесора для гри, а функція вакцинації USB-Пристроїв дозволить запобігти зараженню комп'ютера вірусами, що використовують техніку автозапуску. Завантажувальний диск відновлення буде відмінним інструментом для очищення зараженої системи, а монітор процесів допоможе виявити шкідливий процес.

360 Total Security

Загальна інформація:

– Підтримувані ОС: Windows.

– Функції: кілька антивірусних сканерів, сканер шкідливого ПЗ, захист від фішингу, мобільний додаток.

Переваги:

– багатий набір функцій;

– використовуються антивірусні движки Bitdefender і Avira.

Недоліки:

– більшість функцій недостатньо ефективні;

– багато реклами.

360 Total Security має не саму бездоганну репутацію в тестах незалежних лабораторій, але незважаючи на це продукт безкоштовно пропонує комплексний антивірусний захист із використанням відразу трьох антивірусних движків: Bitdefender, Avira і власної розробки Qihoo. Крім того, антивірус оснащений корисними додатковими функція, наприклад захистом від фішингу, а мобільний додаток Qihoo дозволить контролювати відразу кілька систем.

360 Total Security намагається не дратувати користувача достатком спливаючих вікон, але є серйозні побоювання із приводу політики конфіденційності продукту. Зверніть увагу, що збір даних у рамках програми

поліпшення якості включений за замовчуванням. Компанія не соромиться збирати вашу особисту інформацію, а що з нею буде відбуватися далі – більша загадка.

Інтерфейс антивірусу є небагато заплутаним і незручним, а багато компонентів у його составі виглядають зайвими. У тестах продукт часто видавав помилкові спрацьовування. У кожному разі, якщо вам потрібний ефективний безкоштовний продукт – це гарний варіант.

ZoneAlarm Free Antivirus+

Загальна інформація:

– Підтримувані ОС: Windows.

– Функції: антивірусний сканер, персональний фаєрвол, захист особистих даних, хмарне резервне копіювання.

Переваги:

– добре зарекомендував себе в антивірусних тестах;

– убудований фаєрвол і захист особистих даних.

Недоліки:

– хитра спроба встановити ПЗ від Yahoo;

– дуже повільна установка.

Фаєрвол ZoneAlarm має дуже гарну репутацію, що запрацьовувалася роками. Якщо ви не довіряєте роутеру або убудованому брандмауерові Windows, то ZoneAlarm буде відмінним рішенням.

ZoneAlarm Free Antivirus+ пропонує ліцензований антивірусний движок “Лабораторії Касперського”, тобто є підконтрольною ФСБ і дуже рідко попадає в програми тестування лабораторій. Продукт гранично простий у використанні й фактично є єдиним безкоштовним варіантом пакетного рішення антивірусу й фаєрвола. ZoneAlarm завжди буде виводити оповіщення при блокуванні або виявленні погрози, що може не сподобається деяким користувачам.

Adaware Antivirus Free

Загальна інформація:

– Підтримувані ОС: Windows.

– Функції: антивірусний сканер, поведінковий аналіз, захист завантажень.

Переваги:

– ігровий режим;

– можна встановлювати разом з іншими програмами безпеки.

Недоліки:

– не найкращий рівень виявлення шкідливого ПЗ;

– відсутня захист від шкідливих і шахрайських сайтів.

Ad-Aware Free Antivirus+ використовує ліцензований движок Bitdefender, тому логічно очікувати від рішення високої ефективності. На ділі не всі так райдужно.

Оновлений Adaware Antivirus Free 12 втратився компонента веб-компаньйон, що був в 11 версії й відповідав за блокування шкідливих і шахрайських сайтів. Тепер антивірус захищає тільки від шкідливих завантажень. Інтерфейс перевантажений рекламними блоками, що утрудняє навігацію й пошук потрібних функцій.

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 протидії зловмисним програмам з використанням методів машинного навчання.

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

Опишемо які технології повинна реалізовувати сучасна система протидії зловмисним програмам з використанням методів машинного навчання.

Система протидії зловмисним програмам з використанням методів машинного навчання є платформою безпеки, призначення якої – виявлення й відбиття різних погроз, включаючи погрози нульового дня, як на робочих станціях, так і в мережі за допомогою міжмережних екранів (Firepower) або ж засобами захисту контенту. Вона дозволяє безпосередньо виявляти шкідливі програми, проводити історичний аналіз файлових подій на робочій станції або ж у мережі, взаємодіяти з «пісочницею». Функціонал ретроспективного аналізу допомагає виявити шкідливе ПЗ, що змогло обійти традиційні механізми безпеки. Наприклад, система протидії зловмисним програмам з використанням методів машинного навчання ідеально підійшла для виявлення й блокування на крапці входу недавніх WannaCry і Petya.

Безперервна аналітика

Використання хмарної аналітики (Talos) більших обсягів даних для визначення, чи є файл безпечним або шкідливим, що підвищує точність виявлення схованих атак на вході в систему й для закриття шляхів проникнення, а також мінімізації уразливостей.

Швидке виявлення погроз

За допомогою технології безагентового виявлення (аналіз і тестування поводження підозрілих файлів) здійснюється реєстрація всієї файлової активності й, як наслідок, швидке виявлення вірусів.

Ретроспективний аналіз

Автоматичний аналіз історії дій файлів (звідки проникнув, де виконувався, що робив) не тільки знижує до мінімуму ризик глобального поширення зловмисного ПЗ, але й дозволяє виявити точну локалізацію атаки.

Захист від атак

Забезпечення повсюдного захисту від погроз із використанням ретроспективної безпеки до традиційного виявлення, що значно підвищує ефективність, продуктивність мережі й охват потенційно шкідливих файлів.

Контроль ланцюжка атаки

Новий рівень аналітики в режимі реального часу для виявлення погроз, що зіставляє шаблони шкідливого поводження на окремому кінцевому пристрої й на всіх кінцевих пристроях мережі.

Розширена аналітика

Автоматизоване виявлення підозрілого поводження, що забезпечують розподілене подання всіх областей, які найбільш піддані ризику атаки.

Поведінковий аналіз

Цілеспрямований пошук порушень на підставі реальних подій.

Відстеження подій

Можливість спрощеної й швидкої розбивки ланцюжка атак для своєчасного виявлення основних причин зараження (окремі додатки, файли, документи, віруси й т.д.)

Моніторинг мережі

Звіти не обмежені збором і підрахунком подій. Звітність у системі протидії зловмисним програмам з використанням методів машинного навчання для прикінцевих пристроїв включає панелі, що дають можливість діяти, моніторингу й визначення тенденцій, що показує доречність із погляду бізнесу й вплив з погляду ризику.

Інтеграція з іншими рішеннями

Система протидії зловмисним програмам з використанням методів машинного навчання виконує передачу й кореляцію інформації про погрози по всій архітектурі, включаючи як екосистему система протидії зловмисним програмам з використанням методів машинного навчання, так і інші платформи безпеки, таких як Email і Web Security.

Джерелами найбільшої в галузі бази даних, сформованої на основі моніторингу погроз і аналізу більших даних, є системи колективної інформаційної безпеки, група по інформаційній безпеці й дослідженням і канали системи протидії зловмисним програмам з використанням методів машинного навчання.

Система протидії зловмисним програмам з використанням методів машинного навчання працює в такий спосіб:

– До початку атаки система протидії зловмисним програмам з використанням методів машинного навчання використовує глобальні аналітичні дані по погрозах, які надходять із відділу колективної інформаційної безпеки компанії, групи по інформаційній безпеці й дослідженням, а також по каналах дані системи протидії зловмисним програмам з використанням методів машинного навчання. Це допомагає підсилити захист від відомих і невідомих погроз.

– Під час атаки система протидії зловмисним програмам з використанням методів машинного навчання використовує аналітичні дані, відомі сигнатури файлів і динамічний аналіз на основі технології система протидії зловмисним програмам з використанням методів машинного навчання, щоб виявити й блокувати файли, що порушують політику безпеки, експлойти й шкідливе ПЗ, що намагається проникнути в мережу.

– Після завершення атаки або після первісної перевірки файлу система безупинно контролює й аналізує всю активність і трафік файлу, незалежно від його статусу, відслідковуючи будь-які ознаки шкідливого поводження. Якщо

файл, що одержав раніше статус «невідомий» або «безпечний», діє підозріло, система протидії зловмисним програмам з використанням методів машинного навчання фіксує це й відправляє повідомлення в службу інформаційної безпеки із вказівкою потенційної погрози. система протидії зловмисним програмам з використанням методів машинного навчання забезпечує своєчасне одержання даних про те, звідки з'явилася шкідлива програма, які системи вона торкнулася й що робить у цей момент. Система надає інструменти, які дозволяють швидко реагувати на проникнення й усунути його за допомогою декількох клацань миші. Такі інструменти дозволяють службам інформаційної безпеки вчасно одержувати дані, допомагаючи швидко виявити атаку, оцінити масштаб вторгнення й знешкодити шкідливий код до того, як він завдасть шкоди

До загроз безпеки інформації у операційній системі відносяться наступні:

– Віруси – комп'ютерна програма, яка має здатність до прихованого саморозмноження. Одночасно зі створенням власних копій віруси можуть завдавати шкоди: знищувати, пошкоджувати, викрадати дані, знижувати або й зовсім унеможливлювати подальшу працездатність операційної системи комп'ютера. Розрізняють файлові, завантажувальні та макро-віруси. Можливі також комбінації цих типів. Нині відомі десятки тисяч комп'ютерних вірусів, які поширюються через мережу Інтернет по всьому світу.

– Троянські програми – проводять шкідливі дії замість оголошених легальних функцій або наряду з ними. Вони не спроможні на самовідтворення і передаються тільки при копіюванні користувачем.

– Хробаки – віруси, що поширюються автоматично в комп'ютерній мережі за знайденою адресою в адресній книзі.

– Макровіруси – це програми, написані на так званих макромови, вбудованих в деякі системи обробки даних (текстові та графічні редактори, електронні таблиці і т. д.). Для свого розмноження такі віруси використовують можливості макромови, вони переносяться від одного зараженого файлу до іншого. Найбільшого поширення набули макровіруси для Microsoft Word, Excel .

Макровіруси одержують управління при відкритті або закриття зараженого файлу, перехоплюють стандартні файлові функції і потім заражають файли, до яких яким-небудь чином йде звертання. Більшість макровірусів є резидентними вірусами: вони активні не тільки в момент відкриття або закриття файлу, але до тих пір, поки активний сам текстовий або табличний редактор (а деякі можуть залишатися в оперативній пам'яті до виключення ПК). Легкість створення макровірусів вражає уяву, все знаходиться буквально під рукою: достатньо запустити Word, Вибрати меню Сервіс – Макрос – Редактор Visual Basic – І запуститься програмне середовище VBA При роботі з документом MS Word виконує різні дії: відкриває документ, зберігає, друкує, закриває і т. д. При цьому Word шукає і виконує відповідні вбудовані макроси: при збереженні файлу по команді Файл – Зберегти викликається макрос FileSave, При збереженні по команді Файл – Зберегти як ... – FileSaveAs, При друці по команді Друк ... – FilePrint і т. д. Існують також кілька макросів, автоматично викликаються при виникненні відповідних ситуацій. Наприклад, при відкритті документа Word перевіряє його на наявність макросу AutoOpen . Якщо такий макрос присутній, то Word виконує його. При закритті документа Word виконує макрос AutoClose, При запуску Word викликається макрос AutoExec, При завершенні роботи – AutoExit, При створенні нового документа – AutoNew (Схожі механізми, але з іншими іменами макросів, використовуються і в Excel). Макровіруси, що вражають файли Word, Як правило, користуються одним з чотирьох прийомів: у вірусі присутній автомакрос; у вірусі перевизначений один зі стандартних системних макросів (асоційований з яким-небудь пунктом меню); макрос вірусу автоматично викликається при натисканні на яку-небудь клавішу або комбінацію клавіш; вірус починає розмножуватися тільки в тому випадку, якщо користувач запускає його на виконання. Основний механізм зараження такою: коли ми відкриваємо заражений документ Word, Макровірус копіює свій код в область глобальних макросів документа. А при виході з Word глобальні макроси (включаючи макроси вірусу) автоматично записуються в dot-файл глобальних

макросів (шаблон Normal.dot). Потім вірус перевизначають стандартні макроси (наприклад, FileOpen, FileSave, FileSaveAs, FilePrint) І з їх допомогою перехоплює команди роботи з файлами. Коли Ви робите цих команд заражається файл, до якого йде звернення.Характерними ознаками присутності макровірусів є: неможливість збереження зараженого документа Word в інший формат (по команді Зберегти як ...); неможливість запису документа в інший каталог або на інший диск командою Зберегти як ...; неможливість збереження внесених змін в документ (команда Зберегти); недоступність вкладки « Рівень безпеки »(Меню Сервіс – Макрос – Безпека ...); тому що багато вірусів написані з помилками (або некоректно працюють в різних версіях пакета Microsoft Office), то можлива поява відповідних системних повідомлень з кодом помилки; інші «дивацтва» в поведінці документів Word ; найчастіше макровіруси можна виявити візуально. Справа в тому, що більшість вірусописьменників відрізняються марнославством: у властивостях файлу Word (Вікно Властивості Викликається по кліку правої кнопки миші – вибрати з контекстного меню Властивості) На вкладці Підсумок заповнюють поля вводу (Назва, Тема, Автор, Категорія, Ключові слова та Коментар). Цю інформацію (як правило, в макровірусів вона пишеться сумішшю латинки з кирилицею і включає – серед іншого – деякі безглузді слова, типу Муніципалізмо Тощо) можна побачити при наведенні покажчика миші на значок файлу Word – Вона з'являється в підказці і внизу зліва у папці, у віконці Детально (Якщо включено Використання типових завдань для папок).

– Руткіти – програма або набір програм для приховування слідів присутності зловмисника або шкідливої програми в системі. Термін Rootkit історично прийшов зі світу UNIX, і під цим терміном розуміється набір утиліт або спеціальний модуль ядра, які зломщик встановлює на зламаній їм комп'ютерній системі відразу після отримання прав суперкористувача. Цей набір, як правило, включає в себе різноманітні утиліти для «замітання слідів» вторгнення в систему, робить непомітними сніфери, сканери, кілоггери, троянські програми, які заміщають основні утиліти UNIX (у разі неядерного руткіта).

Rootkit дозволяє зломщикові закріпитися у зламаної системі і приховати сліди своєї діяльності шляхом захования файлів, процесів, а також самої присутності руткіта в системі.

– Спам – масова розсилка комерційної, політичної та іншої реклами (інформації) або іншого виду повідомлень особам, які не висловлювали бажання їх отримувати. Легальність масової розсилки деяких видів повідомлень, для яких не потрібна згода одержувачів, може бути закріплена в законодавстві країни. Наприклад, це може стосуватися повідомлень про насуваються стихійних лихах, масової мобілізації громадян тощо У загальноприйнятому значенні термін «спам» у українській мові вперше став вживатися стосовно до розсилки електронних листів. Чи не запитані повідомлення в системах миттєвого обміну повідомленнями (наприклад, ICQ) носять назву SPIM.

– Фішинг – вид інтернет-шахрайства, метою якого є отримання доступу до конфіденційних даних користувачів – логінів і паролів. Це досягається шляхом проведення масових розсилок електронних листів від імені популярних брендів, а також особистих повідомлень всередині різних сервісів, наприклад, від імені банків (Приват-банк,…), сервісів (Gmail.com,…) або всередині соціальних мереж (Facebook,…). У листі часто міститься пряме посилання на сайт, зовні не відрізняється від справжнього, або на сайт з перенаправленням. Після того, як користувач потрапляє на підроблену сторінку, шахраї намагаються різними психологічними прийомами спонукати користувача ввести на підробленій сторінці свої логін і пароль, які він використовує для доступу до певного сайту, що дозволяє шахраям отримати доступ до облікових записів і банківських рахунків. Фішинг – один з різновидів соціальної інженерії, заснована на незнанні користувачами основ мережевої безпеки: зокрема, багато хто не знають простого факту: сервіси не розсилають листів з проханнями повідомити свої облікові дані, пароль і інше. Для захисту від фішингу виробники основних інтернет-браузерів домовилися про застосування однакових способів інформування користувачів про те, що вони відкрили підозрілий сайт, який може належати шахраям. Нові

версії браузерів вже мають таку можливість, яка відповідно іменується «антифішинг».

– Програми шпигуни – програмне забезпечення, яке встановлюється таємно на персональному комп'ютері для перехоплення або часткового контролю над взаємодії користувача з комп'ютером без згоди користувача. Програми-шпигуни – будь-яке програмне забезпечення, яке збирає інформацію через інтернет користувачів зв'язку без їх відома, як правило, в рекламних цілях. Як правило, в комплекті з безкоштовною або умовно-безкоштовних програм, які можна завантажити з Інтернету. Це шкідлива програма, яка призначена для крадіжки особистих даних, як номери кредитних карток, номери соціального страхування, паролі і т.д. Це програмне забезпечення здатне виконувати певні операції на вашому комп'ютері без вашої згоди, таких як показ реклами, збір вашої особистої інформації або зміна конфігурації комп'ютера. Spyware програми можуть збирати різні види особистої інформації, таку як Інтернет-серфінг звички, сайти, які були відвідані, але також може заважати користувачеві контролем над комп'ютером іншими способами, такими, як встановлення додаткового програмного забезпечення, перенаправлення діяльності веб-браузеру, доступ до веб-сайтів сліпий, що призведе до зростання кількості шкідливих вірусів, або перенаправлення доходів від реклами з третьою стороною.

– DoS-атака (від англ. Denial of Service, відмова в обслуговуванні) – атака на обчислювальну систему з метою вивести її з ладу, тобто створення таких умов, при яких легітимні (правомірні) користувачі системи не можуть отримати доступ до надаваних системою ресурсів, або цей доступ ускладнений. Відмова «ворожої» системи може бути як самоціллю (наприклад, зробити недоступним популярний сайт), так і одним із кроків до оволодіння системою (якщо під позаштатної ситуації ПЗ видає будь-яку критичну інформацію – наприклад, версію, частина програмного коду і т. д .).

Якщо атака виконується одночасно з великої кількості комп'ютерів, говорять про DDoS-атаці (від англ. Distributed Denial of Service, розподілена атака

типу «відмова в обслуговуванні»). У деяких випадках до DDoS-атаці призводить легітимне дію, наприклад, розміщення на популярному інтернет-ресурсі посилання на сайт, розміщений на не дуже продуктивному сервері (слешдот-ефект). Великий наплив користувачів, призводить до перевищення допустимого навантаження на сервер і відмови в обслуговуванні частини з них

– Ping-флуд – тип атаки на мережеве обладнання, що ставить своєю метою відмову в обслуговуванні. Ключовою особливістю (у порівнянні з іншими видами флуд-атак) є можливість здійснення атаки «побутовими засобами » (програмами і утилітами, що входять до складу домашніх/офісних версій операційних систем).

– Експлойт – це комп'ютерна програма, фрагмент програмного коду або послідовність команд, що використовують уразливості в програмному забезпеченні та застосовуються для проведення атаки на обчислювальну систему. Метою атаки може бути як захоплення контролю над системою (підвищення привілеїв), так і порушення її функціонування (DoS-атака).

– IP-спуфінг – Вид хакерської атаки, що полягає у використанні чужого IP-адреси з метою обману системи безпеки. Метод, який використовується в деяких атаках. Полягає в проставленні у поле зворотного (source) адреси IP-пакета неправильно вказану адресу. Застосовується з метою приховування справжнього адреси атакуючого, з метою викликати у відповідь пакет на потрібну адресу і з іншими цілями. Протокол транспортного (4) рівня TCP має вбудований механізм для запобігання спуфінга – так звані номери послідовності та підтвердження (sequence number, acknowledgement number). Протокол UDP не має такого механізму, отже, побудовані на його основі програми більш уразливі для спуфінга.

– Атака Man-in-the-Middle – термін в криптографії, що позначає ситуацію, коли атакуючий здатний читати і видозмінювати по своїй волі повідомлення, якими обмінюються кореспонденти, причому жоден з останніх не може здогадатися про його присутність у каналі. Метод компрометації каналу зв'язку, при якому зломщик, підключившись до каналу між контрагентами, здійснює

активне втручання в протокол передачі, видаляючи, спотворюючи інформацію або нав'язуючи хибну.

– Атака SQL-іньєкція – один з розповсюджених способів злому сайтів і програм, що працюють з базами даних, заснований на впровадженні в запит довільного SQL-коду. Впровадження SQL, в залежності від типу використовуваної СУБД та умов впровадження, може дати можливість атакуючому виконати довільний запит до бази даних (наприклад, прочитати зміст будь-яких таблиць, видалити, змінити або додати дані), отримати можливість читання і/або запису локальних файлів і виконання довільних команд на атакується сервері. Атака типу впровадження SQL може бути можлива через некоректну обробки вхідних даних, використовуваних в SQL-запити. Розробник прикладних програм, що працюють з базами даних, повинен знати про такі уразливості та вживати заходів протидії запровадженню SQL.

– Атака PHP-іньєкція – один із способів злому веб-сайтів, що працюють на PHP, що полягає у виконанні стороннього коду на серверній стороні.

– Міжсайтовий скриптінг – тип вразливості інтерактивних інформаційних систем в інтернеті. XSS виникає, коли в генеруються сервером сторінки з якоїсь причини потрапляють користувальницькі скрипти. Специфіка подібних атак полягає в тому, що замість безпосередньої атаки сервера вони використовують уразливий сервер в якості засобу атаки на клієнта. Для терміна використовують скорочення «XSS», щоб не було плутанини з каскадними таблицями стилів, що використовують скорочення «CSS». Зараз XSS складають близько 15% всіх виявлених вразливостей. Довгий час програмісти не приділяли їм належної уваги, вважаючи їх безпечними. Однак ця думка помилкова: на сторінці або в HTTP-Cookie можуть бути досить вразливі дані (наприклад, ідентифікатор сесії адміністратора). На популярному сайті скрипт може влаштувати DoS-атаку

– Рекламні системи (adware) – той вид реклами, на показ якої користувач не давав своєї згоди і яка показується користувачеві примусово.

3.2 Розробка структурної схеми

Машинне навчання складається із двох великих розділів: фази навчання й розпізнавання.

– Фаза навчання. На початку фахівці повинні пояснити комп'ютеру, що характеризує вірус. На наступних етапах комп'ютер буде стає усе розумніше до того ж почне вчитися.

– Розпізнавання. Потім натренований алгоритм на підставі поводження виконуваної програми здатний вирішити, чи йде мова про легітимний додаток або про схований у файлі вірусі.

У сучасні комерційні продукти безпеки інтегрований захист від програм-здирників. Вона в реальному часі постійно сканує систему на наявність дивних звертань до файлів. Одночасно із цим перевіряються показники системи, з'ясовується, чи не підвищується раптово навантаження на процесор і чи не зростає стрімко число звертань до дисків (перші індикатори атаки).

В офіційних заявах затверджується, що розроблювачі антивірусів знають свою справу й гарантують ефективний захист від нападів. Однак інсайдери пошепки розповідають, що й у цієї технології є слабкі місця: кібергангстери послідовно встановлюють на ізольовану систему всі розповсюджені антивірусні рішення, а потім спокійно розробляють ідеальний вірус, що не буде замічений сканером і зможе обдурити навіть найсучаснішу евристику.

Проти таких, розроблених з більшими зусиллями, варіантів практично будь-який антивірусний продукт неспроможний. Незважаючи на те що розроблювачі протягом декількох годин після поширення вірусу підготовляють захист, до перших потерпілих протиотрута попадає занадто пізно.

Винуватий у цьому людський фактор: фахівцям зі шкідливого ПЗ необхідно вручну розібрати новий вірус, проаналізувати його й підготувати опис, що потрапить на систему клієнта при відновленні. У випадку зі складними

«шкідниками» процес може зайняти кілька днів. На аналіз особливо вишуканого утвору злочинців можуть піти навіть місяці.

Деякі розроблювачі бачать майбутнє вірусного аналізу у відмові від людського фактора: у центрі будуть перебуває машини, що розпізнають шкідливе ПЗ й запускають відповідне відновлення.

Аналіз через машинне навчання

Усілякі компанії, від Symantec до Malwarebytes, сьогодні займаються розробкою методів, при яких роботу з аналізу вірусів візьмуть на себе IT-системи. У випадку з машинним навчанням дослідники «зкормлюють» суперкомп'ютеру кілька мільйонів файлів – як шкідливих, так і безпечних.

Програми з машинним навчанням. Malwarebytes починаючи із третьої версії робить ставку на автоматизоване навчання.

За допомогою так званих ознакових описів потім комп'ютеру пояснюється, як виглядає вірус, якими характеристиками він володіє і як звичайно поводиться на комп'ютері своєї жертви – абсолютно так само, як і при розвитку й вихованні дитини.

Таким чином, поступово машина знайде самостійність і буде усе надійніше розрізняти вірус і легітимну програму. І нехай частота помилок залишається ще досить високої, поступово вона буде знижуватися, а алгоритм удосконалюватися.

Компанія Symantec для машинного навчання звела навіть окремий обчислювальний центр у Великобританії

Однак такий розвиток вимагає часу й ресурсів. В Symantec є свій окремий обчислювальний центр для машинного навчання, де займаються оптимізацією методики.

У сучасних антивірусних продуктах цей алгоритм уже є елементом стратегії розпізнавання. Незважаючи на це на заднім тлі однаково присутні люди-дослідники, оскільки нова технологія ще страждає від дитячих хвороб.

Проблеми з панацеєю

Як би не була гарна ідея машинного навчання, до цих спостерігаються ключові недоліки, що затримують тріумфальний хід. У першу чергу, усе впирається в гроші. Лише великі компанії можуть дозволити собі інфраструктуру, необхідну для реалізації цього методу.

Крім того, незважаючи на перехід до машинного навчання, на місці завжди повинна бути присутнім команда експертів. Вона контролює алгоритм і в неоднозначних випадках сама приймає рішення, вірус це чи ні, вносячи відповідні коректування.

Фірми, що скорочують витрати на персонал і стовідсотково належні на алгоритм, ризикують допустити в навчальний процес помилкові результати. Таким чином, їхня безпека стрімко знижується.

Одного лише машинного навчання недостатньо

Проблема машинного навчання впирається насамперед у мову. Якщо представити нинішній сценарій погрози у вигляді англомовної країни, такий алгоритм був би успішний лише в боротьбі зі шкідливим ПЗ англійською мовою.

Однак оскільки ландшафт погрози постійно змінюється, і відповідно постійно з'являється нова мова, доводиться увесь час із чистого аркуша адаптувати алгоритм під ситуацію, що змінилася. А це складно. Приміром, сімейство Trojan-Ransom.Win32.Shade складається з 30 000 підтипів. Одним лише розпізнаванням такого сімейства алгоритм не здатний забезпечити захист від всіх його підвидів. Для цього знадобилися б сотні прикладів на кожний вірус – лише так алгоритм навчається автоматично.

Утиліта машинного навчання також не здатна захистити від спеціалізованих атак. Просто відсутня база даних. На одній методиці машинного навчання далеко не виїхати. Вона може бути винятково шестірнею в складній антивірусній машині. Але навіть із машинним навчанням і іншими засобами антивірусні системи катастрофічно відстають від розроблювачів шкідливого ПЗ.

Всі захисні продукти роблять помилки

Навіть якщо в системі використовуються «пісочниці», карантин, евристичні методи й машинне навчання, зловмисники умудряються неї обманювати. Як уже було сказано вище, досить протестувати антивірусне рішення на ізольованій системі до релізу «шкідника» і знайти в такий спосіб пробіли в розпізнаванні.

Сучасні антивіруси повинні розпізнавати й зупиняти програми-здирники завчасно

Іншу небезпеку представляють самі антивіруси. Хакери, користуючись уразливостями, за допомогою перепрограмування можуть одержати за ними контроль так само, як над звичайним ПЗ. Так, у червні 2016 року постраждала компанія Symantec. Фахівці виявили сім критичних лазівок практично у всіх антивірусних продуктах цього розроблювача.

Приклавши мінімальні зусилля, зловмисники інфікують не тільки комп'ютери, але й цілі корпоративні мережі. Для використання уразливостей вони повинні примусити свою жертву до виклику модифікованої веб-сторінки. Оскільки антивірусним продуктам на комп'ютері видаються максимально розширені права, подібні уразливості стають фатальними. Гра в кішки-мишки між атакуючими й що обороняються, таким чином, триває. Однак машинне навчання в кожному разі ускладнить життя розроблювачам вірусів.

Напрямки розпізнавання

У сучасних антивірусних системах для розпізнавання використовується цілий ряд методів. Нові файли проходять через різні стадії за пару митей.

– **Аналіз поводження.** При виникненні підозр сканер запускає файл передбачуваного вірусу в пісочниці – захищеної області. Там рудиментарний алгоритм перевіряє, чи не заражена програма.

– **Швидка оцінка.** У майбутньому після впровадження машинного навчання нинішня модель буде пропонувати більше швидкий і надійний захист, а також відрізнятися підвищеною продуктивністю. Однак і в цьому випадку остаточне рішення, чи довіряти файлу або перемістити його в карантин, приймає користувач за комп'ютером.

Структурна схема розробленої, у результаті виконання магістерської роботи, системи зображена на рисунку 3.1.

З нього ми бачимо, що система складається з наступних структурних блоків:

– Модулю машинного навчання системи протидії злвмисни программам (антивірусної системи (АС)).

– Джерела погроз.

– Антивірусне програмне забезпечення на персональному комп’ютері.

– Загрози безпеці.

Розглянемо ці структурні блоки більш детально.

До джерел загроз відносяться наступні:

– Загрози з інтернету.

– Загрози при роботі зі змінними носіями інформації.

До антивірусного програмного забезпечення на персональному комп’ютері, відносяться наступні підсистеми:

– Підсистема контролю цілісності.

– Підсистема контролю процесів.

– Підсистема самозахисту антивірусу.

– Підсистема виявлення атак.

– Підсистема аналізу захищеності.

– Підсистема антивірусного захисту.

– Підсистема захисту від спаму.

– Підсистема мережного екранування.

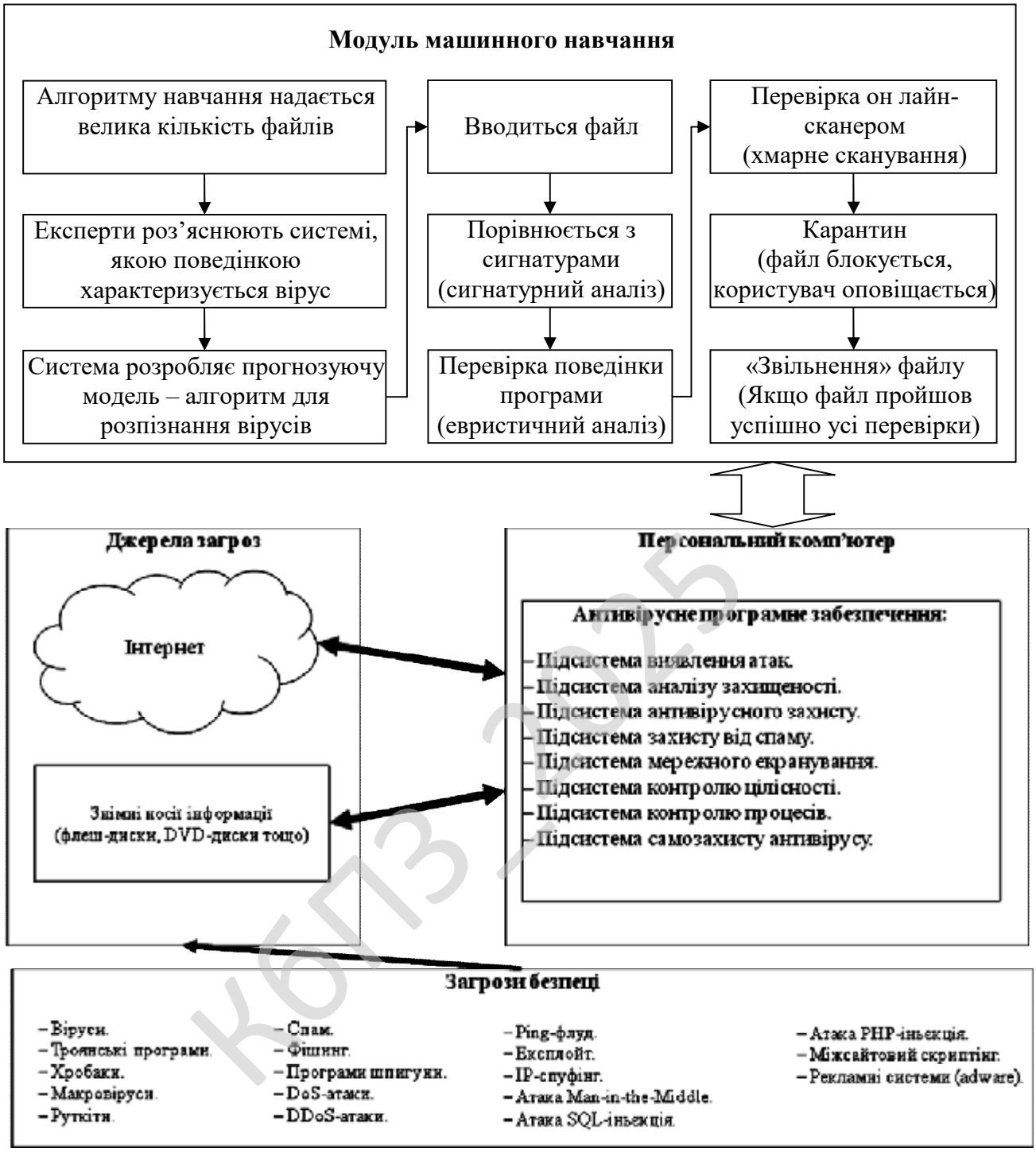

Рисунок 3.1 – Структурна схема системи

Розглянемо їх більш детально.

Підсистема виявлення атак призначена для виявлення несанкціонованої вірусної активності за допомогою аналізу пакетів даних, циркулюючих в антивірусній системі, а також подій, що реєструються на серверах і робочих

станціях користувачів. Підсистема доповнює функції міжмережевих і персональних екранів за рахунок можливості детальнішого контекстного аналізу вмісту передаваних пакетів даних. Дана підсистема включає наступні компоненти: мережеві і хостові сенсори, призначені для збору необхідної інформації про функціонування АС. Мережеві сенсори реалізуються у вигляді окремих програмно-апаратних блоків і призначені для збору інформації про всі пакети даних, що передаються в рамках того мережевого сегменту, де встановлений сенсор. Даний тип сенсорів встановлюється у всіх ключових сегментах АС, де розташовані вузли системи, що захищаються. Хостові сенсори встановлюються на робочі станції і сервери АС і збирають інформацію про всі події, що відбуваються на цих вузлах системи. Хостові сенсори можуть збирати інформацію не тільки про пакети даних, але і інших операціях, які виконуються додатками, запущеними на вузлі АС; модуль виявлення атак, що виконує обробку даних, зібраних сенсорами, з метою виявлення інформаційних атак порушника. Даний модуль підсистеми повинен реалізовувати сигнатурні і поведінкові методи аналізу інформації; модуль реагування на виявлені атаки. Модуль повинен передбачати можливість як пасивного, так і активного реагування. Пасивне реагування припускає сповіщення адміністратора про виявлену атаку, тоді як активне – блокування спроби реалізації вірусної атаки; модуль зберігання даних, в якому міститься вся конфігураційна інформація, а також результати роботи підсистеми..

Підсистема аналізу захищеності забезпечувати можливість виявлення технологічних і експлуатаційних уразливостей АС за допомогою проведення мережевого сканування. Як об’єкти сканування можуть виступати робочі станції користувачів, сервери, а також комунікаційне устаткування. Для проведення сканування можуть використовуватися як пасивні, так і активні методи збору інформації. За наслідками роботи підсистема повинна генерувати детальний звіт, що включає інформацію про виявлені уразливостях, а також рекомендації по їх усуненню. Спільно з підсистемою виявлення уразливостей в антивірусній системі

може використовуватися система управління модулями оновлень загальносистемного і прикладного ПЗ, встановленого в антивірусній системі. Сумісне використання цих систем дозволить автоматизувати процес усунення виявлених уразливостей шляхом установки необхідних оновлень на вузли АС (service pack, hotfix, patch і ін.)..

Підсистема антивірусного захисту призначена для виконання наступних функцій: видаленої установки і деінсталяції антивірусних засобів на серверах і робочих станціях користувачів; видаленого управління параметрами роботи підсистем захисту, що входять до складу комплексної системи антивірусного захисту; централізованого збору і аналізу інформації, що поступає від інших підсистем. Дана функція дозволяє автоматизувати процес обробки даних, що поступають, а також підвищити оперативність ухвалення рішень по реагуванню на виявлені інциденти, пов′язані з порушенням антивірусної безпеки.

Підсистема захисту від спаму. Антиспам дозволяє відслідковувати, фільтрувати та видаляти спам, що пересилається в вашу поштову скриньку. Система не видаляє повідомлення, які здадуться їй спамом. Вона лише робить відповідну позначку на них і доставляє їх, як завжди. Саме тому ці повідомлення також включаються до вашого трафіка. Рівень перевірки спаму – Spam check level – визначає, наскільки жорсткою буде фільтрація спаму. Антиспамові фільтри аналізують кожне електронне повідомлення, яке проходить крізь поштовий шлюз і оцінюють його за шкалою від 1 до 14. Чим більший номер, тим більша ймовірність того, що повідомлення буде віднесено до розряду спаму: Дуже жорстка фільтрація: гарантує, що до вашої поштової скриньки спам практично не буде надходити. Тим не менше, ви ризикуєте втратити і потрібні повідомлення (не пропускає повідомлення, що оцінені вище 2). Жорстка: практично всі спамові повідомлення будуть видалені, є ймовірність видалення потрібних повідомлень (не пропускає повідомлення, що оцінені вище 4). Нормальна: може заблокувати деякі розсилки (не пропускає повідомлення, що оцінені вище 7). Нежорстка: пропускає "другосортну" пошту (не пропускає

повідомлення, що оцінені вище 10). М'яка: пропускає майже всю пошту (не пропускає повідомлення, що оцінені вище 14). Рівень оцінки за замовчуванням зазвичай дорівнює 5, проте адміністратор хостингової системи може його змінити. Обробка спаму: Mark as spam: – присвоєння позначки "спам" – до теми електронного повідомлення буде додане слово СПАМ, і потім це повідомлення буде переслане клієнту як вкладення із зазначенням деталей. Remove: видалення спамового повідомлення. Коли клієнт запустить команду надіслати/отримати, повідомлення зі спамом до нього не надійде. Move To: дозволяє вказати поштову скриньку, куди буде відправлено спам. Клієнти не отримають спам у свою поштову скриньку, проте зможуть переглянути його у вказаній скриньці.

Підсистема мережного екранування призначена для захисту робочих станцій користувачів від можливих мережевих вірусних атак за допомогою фільтрації потенційно небезпечних пакетів даних. Підсистема повинна забезпечувати можливість фільтрації на канальному, мережевому, транспортному і прикладному рівнях стека TCP/IP. Як правило, дана підсистема реалізується на основі міжмережевих і персональних мережевих екранів. При цьому міжмережевий екран встановлюється в точці підключення АС до мережі Інтернет, а персональні екрани розміщуються на робочих станціях користувачів.

Підсистема контролю цілісності. Для жорсткішого контролю за спробами проникнення в систему використовується багаторівневий захист. Одним з таких додаткових рівнів є система контролю цілісності програмного забезпечення, яка контролює все програмне забезпечення на брандмауері і присилає звіти про всі віддалені файли, що щойно з'явилися і змінилися. Таким чином, при щонайменшій зміні конфігурації шлюзу відповідальний за його роботу отримає докладний звіт про те, що і коли було змінено.

Підсистема контролю процесів призначена для визначення які процеси відбуваються у системі, та чи є вони дозволеними операційною системою, чи ні.

Підсистема самозахисту антивірусу. Ефективна система самозахисту захищає його від зупинки або відключення навіть при проведенні цільових атак.

Антивірус здатний створювати диск аварійної регенерації системи. Будь-які посилання на шкідливі або фішингові сайти програмою блокуються автоматично. У разі поразок антивірус після ліквідації загроз самостійно здатний відновити систему, повернувши її в стан, який був до виникнення проблем з руйнівним кодом або вірусом. Програма виконує діагностику системи, виявляючи існуючі пробіли та визначаючи відповідність її оновлень за допомогою бази даних. Але програмою, на жаль, не передбачено надання інформації про те, яка версія програми, яка має прогалини, встановлювалася. Дану інформацію доведеться виявляти самостійно. Крім того, повна інформація про сам процес усунення вразливих місць недоступна.

3.3 Розробка функціональної схеми

Антивірус, що розроблений у результаті виконання проектування використовує новітні технології захисту, завдяки яким забезпечується безпека й стабільна робота комп'ютера.

Функціональна схема системи складається з наступних основних функціональних блоків:

– Контроль запущених процесів.

– Захист у режимі реального часу.

– Сканування системи.

– Ядро антивірусної програми.

– Бази даних вірусів.

– Блок машинного навчання

Розглянемо більш детальніше склад функціональних блоків.

Функціональний блок контролю запущених процесів складається з наступних функціональних підсистем:

– Контроль роботи запущених програм та процесів і обмеження їхнього доступу до важливих областей ОС.

– Список активних процесів.

– Блокування шкідливих та інфікованих процесів.

– Перевірка процесів зі списку автозавантаження.

– Видалення зі списку автозавантаження шкідливих процесів.

Функціональний блок захисту у режимі реального часу складається з наступних функціональних підсистем:

– Перевірка запущених файлів.

– Перевірка відкритих web-сторінок.

– Блокування посилань на заражені й фішингові веб-сайти.

– Перевірка отриманих поштових повідомлень.

– Захист від спаму й фішингу в поштових програмах.

– Самозахист антивірусу від спроб вимикання з боку шкідливого ПЗ.

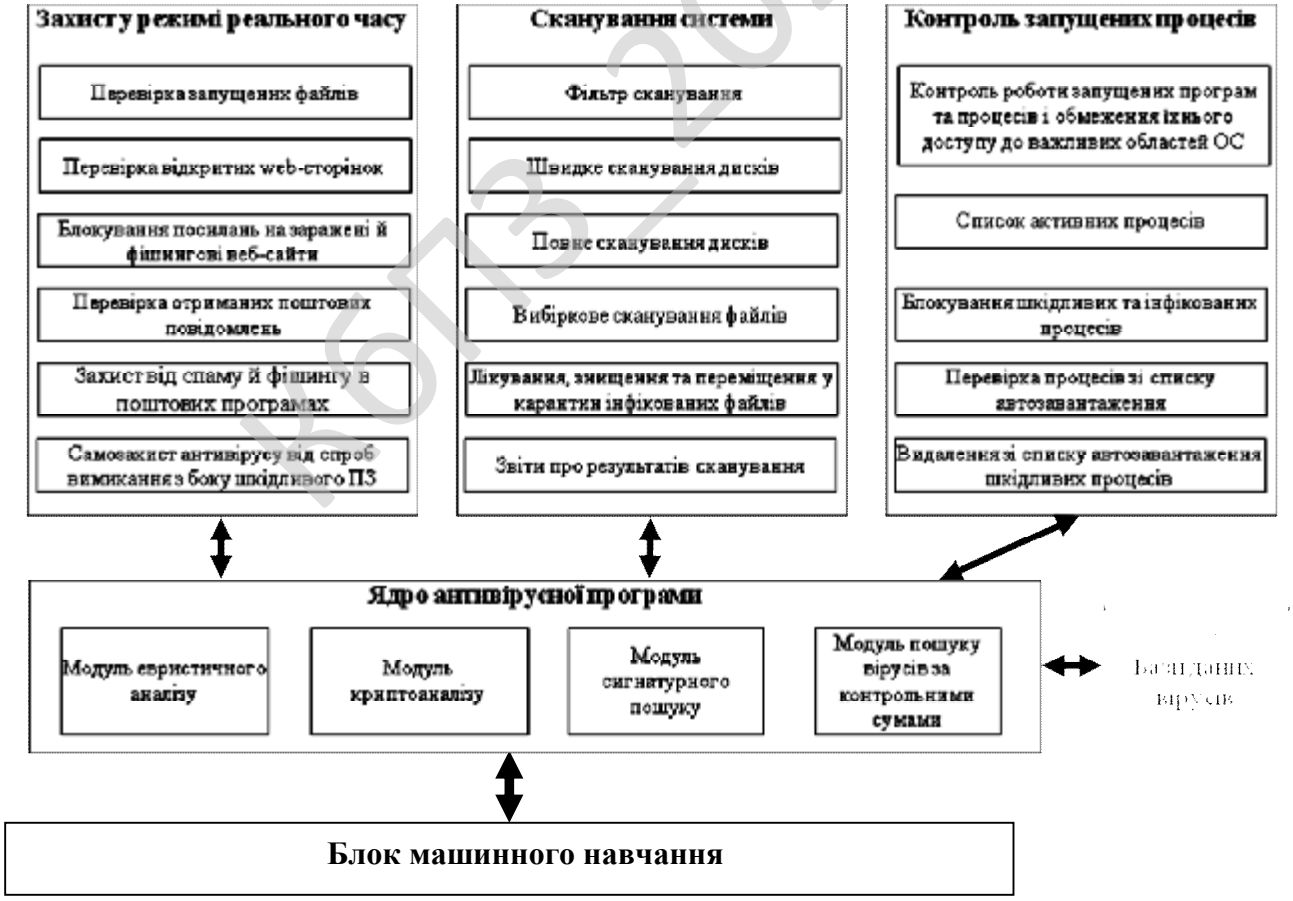

Рисунок 3.2 – Функціональна схема системи

Функціональний блок сканування системи складається з наступних функціональних підсистем:

– Фільтр сканування.

– Швидке сканування дисків.

– Повне сканування дисків.

– Вибіркове сканування файлів.

– Лікування, знищення та переміщення у карантин інфікованих файлів.

– Звіти про результатів сканування.

Функціональний блок ядра антивірусної програми складається з наступних функціональних підсистем:

– Модуль евристичного аналізу.

– Модуль криптоаналізу.

– Модуль сигнатурного пошуку.

– Модуль пошуку вірусів за контрольними сумами.

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

Незважаючи на те що я працював над ПЗ один в реалізації програми я використовував підходи пришвидшення розробки на основі методологій Agile.

Гнучка́ розробка програмного забезпечення (Agile software development, agile-методи) – клас методологій розробки програмного забезпечення, що базується на ітеративній розробці, в якій вимоги та розв'язки еволюціонують через співпрацю між самоорганізовуваними багатофункціональними командами.

Гнучка розробка – найкращий засіб для підвищення продуктивності розробників програмного забезпечення.

Більшість гнучких методологій націлені на мінімізацію ризиків, шляхом зведення розробки до серії коротких циклів, що мають назву ітерацій, які зазвичай тривають один-два тижні. Кожна ітерація сама по собі виглядає як програмний проект в мініатюрі, і включає всі завдання, необхідні для видачі мінімального приросту за функціональністю: планування, аналіз вимог, проектування, кодування, тестування і документування. Хоча окрема ітерація, як правило, недостатня для випуску нової версії продукту, мається на увазі те, що гнучкий програмний проект готовий до випуску наприкінці кожної ітерації. Після закінчення кожної ітерації, команда виконує переоцінку пріоритетів розробки.

Agile акцентує увагу на безпосередньому спілкуванні «віч-на-віч». Більшість agile команд розташовані в одному офісі, його іноді називають bullpen. Як мінімум вона включає і «замовників» (замовники, які визначають продукт, також це можуть бути менеджери продукту, бізнес аналітики або клієнти). Офіс

може також включати тестувальників, дизайнерів інтерфейсу, технічних авторів і менеджерів.

Основною метрикою agile методів є робочий продукт. Віддаючи перевагу безпосередньому спілкуванню, agile-методи зменшують обсяг письмової документації в порівнянні з іншими методами. Це привело до критики цих методів як недисциплінованих.

Agile – родина процесів розробки, а не єдиний підхід в розробці програмного забезпечення, і визначається Agile Manifesto. Agile не включає практик, а визначає цінності та принципи, якими керуються успішні команди.

Agile Manifesto розроблений і прийнятий 17 розробниками 11-13 лютого 2001 року на лижному курорті The Lodge at Snowbird в горах Юти. Маніфест підписали представники наступних методологій Extreme programming, Scrum, DSDM, Adaptive software development, Crystal Clear, Feature driven development, Pragmatic Programming. Agile Manifesto містить 4 основні ідеї та 12 принципів. Примітно, що Agile Manifesto не містить практичних порад.

Основні ідеї:

– Особистості та їхні взаємодії важливіші, ніж процеси та інструменти.

– Робоче програмне забезпечення важливіше, ніж повна документація.

– Співпраця із замовником важливіша, ніж контрактні зобов'язання.

– Реакція на зміни важливіша, ніж дотримання плану.

Принципи, які роз'яснює Agile Manifesto:

– задоволення клієнта за рахунок ранньої та безперебійної поставки коштовного програмного забезпечення;

– вітання змін вимог навіть наприкінці розробки (це може підвищити конкурентоспроможність отриманого продукту);

– часта поставка робочого програмного забезпечення (кожен місяць або тиждень або ще частіше);

– тісне, щоденне спілкування замовника з розробниками впродовж всього проекту;

– проектом займаються мотивовані особистості, які забезпечені потрібними умовами роботи, підтримкою і довірою;

– рекомендований метод передачі інформації – особиста розмова (віч-на-віч);

– робоче програмне забезпечення – найкращий вимірювач прогресу;

– спонсори, розробники та користувачі повинні мати можливість підтримувати постійний темп на невизначений термін;

– постійну увагу поліпшенню технічної майстерності та зручному дизайну;

– простота – мистецтво не робити зайвої роботи;

– найкращі технічні вимоги, дизайн та архітектура виходять у самоорганізованої команди;

– постійна адаптація до мінливих обставин.

Маніфест та Принципи гнучкої розробки містять високорівневі ідеї щодо того, як потрібно вибудовувати процес розробки програмного забезпечення, щоб успішно завершувати проекти й створювати команди, в яких приємно та цікаво працювати.

Документи визначають, що потрібно для цього зробити, але не говорять, як це зробити. По-іншому й не могло бути, оскільки Маніфест та Принципи народилися внаслідок консенсусу представників різних (хоча й споріднених) напрямів, які могли знайти спільну основу лише на рівні базових цінностей та принципів.

Критика

Багато керівників проектів, що працюють у традиційних методологіях на кшталт «водоспаду», критикують agile-методи.

Один з повторюваних пунктів критики: при agile-підході часто нехтують створенням «дорожньої карти» розвитку продукту, так само як і управлінням вимогами, в процесі якого і формується така «карта». Гнучкий підхід до управління вимогами не має на увазі далекосяжних планів (по суті, управління вимогами просто не існує в даній методології), а має на увазі можливість

замовника раптом і несподівано наприкінці кожної ітерації виставляти нові вимоги, що часто суперечать архітектурі вже створеного і поставленого продукту. Таке іноді призводить до катастрофічних «авралів» з масовим рефакторингом і переробками практично на кожній черговій ітерації.

Крім того вважається, що робота в agile мотивує розробників вирішувати всі прибулі завдання найпростішим і найшвидшим можливим способом, при цьому часто не звертаючи уваги на коректність коду з точки зору вимог базової платформи (підхід «працює, та й добре», при цьому не враховується, що може перестати працювати при найменшій зміні або ж породити важкі до відтворення дефекти після реального розгортання у клієнта). Це призводить до зниження якості продукту і накопиченню дефектів.

Методології

Існують методології, які дотримуються цінностей і принципів заявлених в Agile Manifesto, деякі з них:

1. Agile Modeling – набір понять, принципів і прийомів (практик), що дозволяють швидко і просто виконувати моделювання і документування в проектах розробки програмного забезпечення. Не включає в себе детальну інструкцію з проектування, не містить описів, як будувати діаграми на UML.

Основна мета – ефективне моделювання і документування; але не охоплює програмування та тестування, не включає питання управління проектом, розгортання і супроводу системи. Однак включає в себе перевірку моделі кодом.

2. Agile Unified Process (AUP) спрощена версія IBM Rational Unified Process (RUP), розроблена Скоттом Амблером, яка описує просте і зрозуміле наближення (модель) для створення програмного забезпечення для бізнес-додатків.

3 Agile Data Method – група ітеративних методів розробки програмного забезпечення, в яких вимоги та рішення досягаються в рамках співпраці різних крос-функціональних команд.

4. DSDM заснований на концепції швидкої розробки додатків (Rapid Application Development, RAD). Являє собою ітеративний і інкрементний підхід, який надає особливого значення тривалій участі в процесі користувача/споживача.

5. Essential Unified Process (EssUP).

6. Екстремальне програмування (Extreme programming, XP).

7. Feature driven development (FDD) – функціонально-орієнтована розробка. Використовуване в FDD поняття функції або властивості (feature) Системи досить близько до поняття прецеденту використання, використовуваному в RUP, істотна відмінність – це додаткове обмеження: «кожна функція повинна допускати реалізацію не більше, ніж за два тижні». Тобто якщо сценарій використання досить малий, його можна вважати функцією. Якщо ж великий, то його треба розбити на декілька відносно незалежних функцій.

8. Getting Real – ітераційний підхід без функціональних специфікацій, що використовується для веб-додатків. У даному методі спершу розробляється інтерфейс програми, а потім її функціональна частина.

9. OpenUP – це ітераційно-інкрементний метод розробки програмного забезпечення. Позиціюється, як легкий і гнучкий варіант RUP. OpenUP ділить життєвий цикл проекту на чотири фази: початкова фаза, фази уточнення, конструювання та передачі. Життєвий цикл проекту забезпечує надання зацікавленим особам та членам колективу точок ознайомлення і прийняття рішень впродовж усього проекту. Це дозволяє ефективно контролювати ситуацію і вчасно приймати рішення про задовільність результатів. План проекту визначає життєвий цикл, а кінцевим результатом є остаточний додаток.

10. Scrum встановлює правила керування процесом розробки та дозволяє використовувати вже існуючі практики кодування, коректуючи вимоги або вносячи тактичні зміни. Використання цієї методології дає можливість виявляти і усувати відхилення від бажаного результату на більш ранніх етапах розробки програмного продукту.

11. Бережлива розробка програмного забезпечення (lean software development). Використовує підходи з концепції бережливого виробництва.

Під час роботи над магістерською роботою було створено блок-схеми. Перед їх розглядом необхідно провести роз’яснення який саме тип блок-схем використовується.

Блок-схема це представлення задачі для її аналізу або розв'язування за допомогою спеціальних символів (геометричних образів), які позначають такі елементи, як операції, потік, дані тощо. Блок вхідних та вихідних даних прийнято позначати паралелограмом, блок обчислень (обробки) даних – прямокутником, блок прийняття рішень – ромбом, еліпсом – початок та кінець алгоритму.

У інформаційних технологіях функціональна схема складається з функціональних блоків, які являють собою конструктивно відособлені частини (елементи або пристрої) автоматичних систем, які виконують певні функції. 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 схеми можуть виконуватися в укрупненому і розгорненому вигляді. У першому випадку на схемі зображають найважливіші блоки системи і зв'язки між ними.

У другому варіанті схема відображається більш детально, що полегшує її читання та ілюструє принцип роботи.

Основні елементи схем алгоритму це термінатор, процес, рішення, зумовлений процес (підпрограма), дані та з'єднувач.

Термінатор це елемент відображає вхід із зовнішнього середовища або вихід з неї (найчастіше застосування – початок і кінець програми). Всередині фігури записується відповідна дія.

Процес це виконання однієї або кількох операцій, обробка даних будь-якого виду (зміна значення даних, форми подання, розташування). Всередині фігури записують безпосередньо самі операції.

Рішення це показує рішення або функцію перемикального типу з одним входом і двома або більше альтернативними виходами, з яких тільки один може бути обраний після обчислення умов, визначених всередині цього елементу. Вхід в елемент позначається лінією, що входить зазвичай у верхню вершину елементу. Якщо виходів два чи три то зазвичай кожен вихід позначається лінією, що виходить з решти вершин (бічних і нижній). Якщо виходів більше трьох, то їх слід показувати однією лінією, що виходить з вершини (частіше нижній) елемента, яка потім розгалужується. Відповідні результати обчислень можуть записуватися поруч з лініями, що відображають ці шляхи.

Зумовлений процес (підпрограма) це символ відображає виконання процесу, що складається з однієї або кількох операцій, що визначені в іншому місці програми (у підпрограмі, модулі). Всередині символу записується назва процесу і передані в нього дані.

Дані це перетворення у форму, придатну для обробки (введення) або відображення результатів обробки (виведення). Цей символ не визначає носія даних (для вказівки типу носія даних використовуються специфічні символи).

З'єднувач це символ відображає вихід в частину схеми і вхід з іншої частини цієї схеми. Використовується для обриву лінії та продовження її в іншому місці (приклад: поділ блок-схеми, що не поміщається на листі). Відповідні сполучні символи повинні мати одне (при тому унікальне) позначення.

Блок-схеми є першоджерелами стратегії розвитку ПЗ. Тому від точності і детальної 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 програми високого рівня, також те, що при розробці програми слід надати особливу увагу модулю протидії

зловмисним програмам з використанням методів машинного навчання, модулю обробки помилок програми.

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 – перевірка поточного стану з завершенням роботи розробленого ПЗ. При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

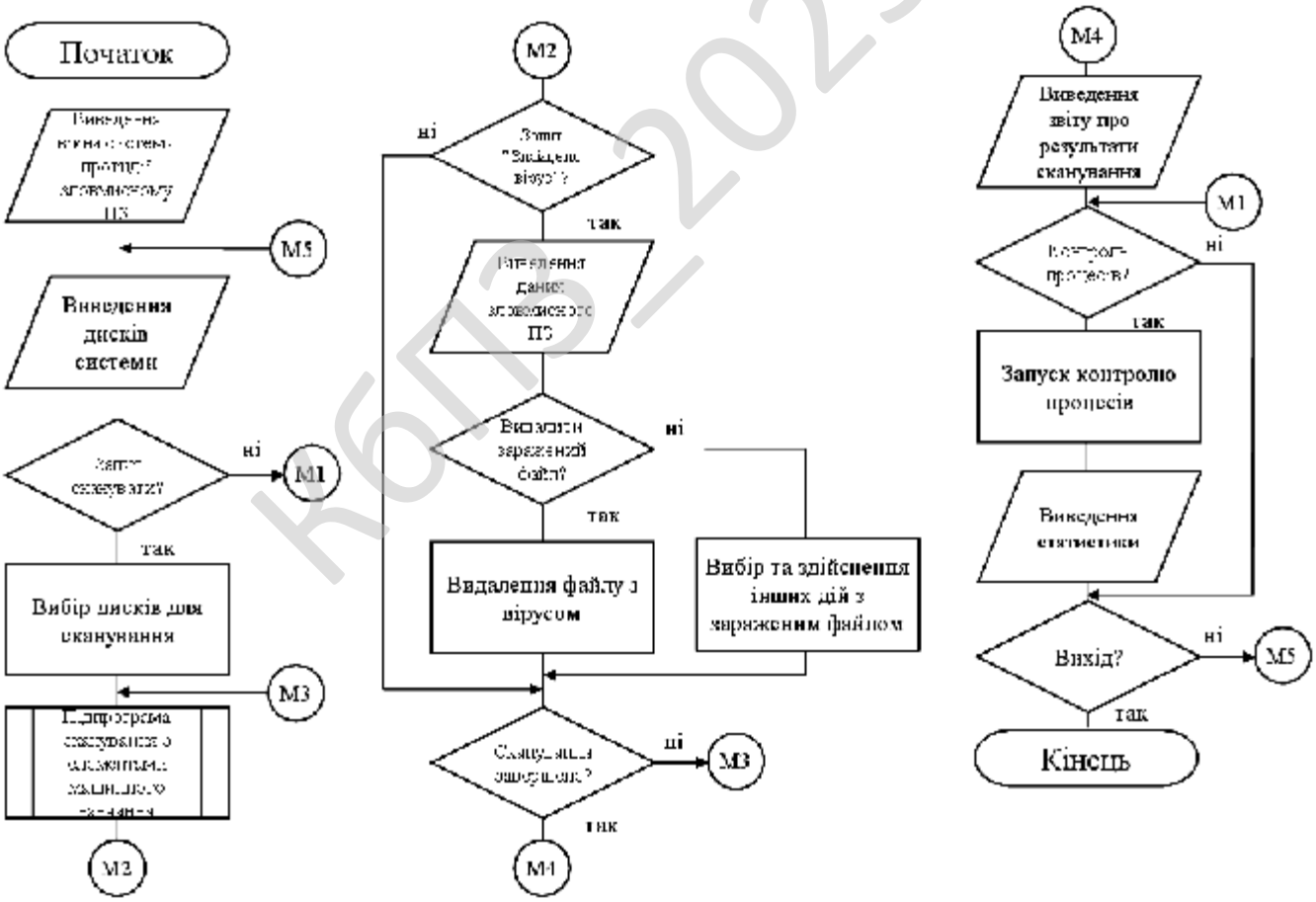

Рисунок 4.1 – Блок-схема основної програми

Jira – була використана комерційна система відслідковування помилок, призначена для організації взаємодії з користувачами, хоча в деяких випадках використовується і для управління проектами. Розроблено компанією Atlassian, є одним з двох її основних продуктів (поряд з вікі-системою Confluence). Має веб-інтерфейс.

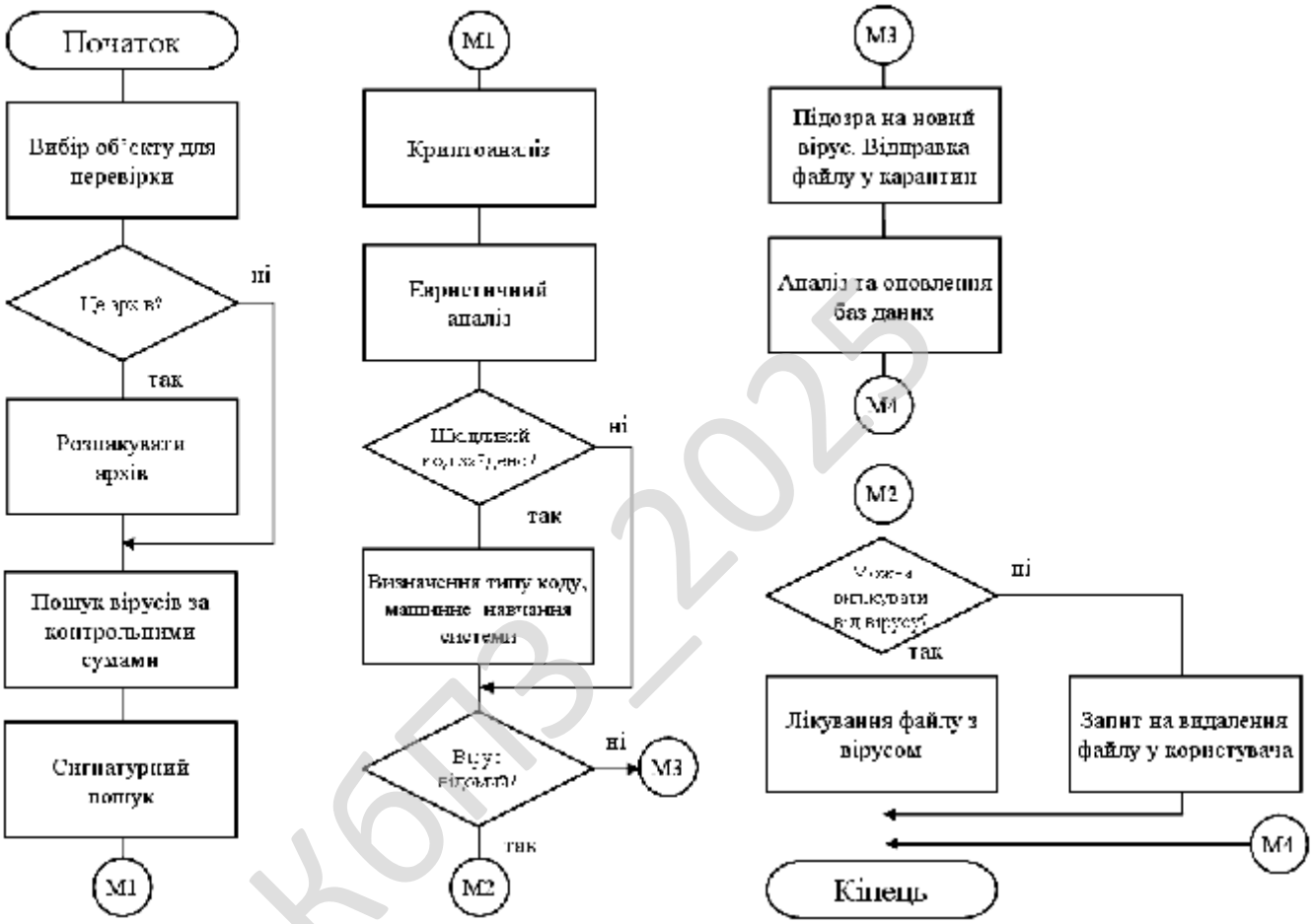

Рисунок 4.2 – Блок-схема роботи підпрограми

Назва системи отримано шляхом усічення слова «Gojira» – Японського імені монстра Годзилла, що, в свою чергу, є відсиланням до назви конкуруючого продукту – Bugzilla; створювалася в якості заміни Bugzilla і багато в чому повторює її архітектуру. Система дозволяє працювати з декількома проектами. Для кожного з проектів створює і веде схеми безпеки і схеми оповіщення.

До версії 3.13.5 (включно) розрізнялися редакції Enterprise, Professional і Standard, після – Залишилася тільки редакція Enterprise (для великих організацій).

Система заснована на Java EE і працює на кількох популярних системах управління базами даних і операційних системах.

Основний елемент обліку в системі – завдання (ticket або issue). Завдання містить назву проекту, тему, тип, пріоритет, компоненти і зміст. Завдання може бути розширена додатковими полями (також і нові призначені для користувача поля можуть бути визначені), додатками (наприклад – Фотографіями, скриншотами) або коментарями. Завдання може редагуватися або просто змінювати статус, наприклад, з «відкритий» в «закритий». Які переходи між станами можливі, визначається через настроюється потік операцій. Будь-які зміни в задачі записуються в журнал.

Jira має велику кількість можливостей конфігурації: для кожної програми може бути визначений окремий тип завдання з власним workflow, набором статусів, одним або декількома видами уявлення (screens). Крім того, за допомогою так званих «схем» можна визначити для кожного індивідуального Jira-проекту власні права доступу, поведінку і видимість полів і багато іншого.

Завдяки універсальному підходу можна пристосувати Jira для багатьох непрофільних завдань, наприклад, керування вимогами, керування ризиками, аж до реалізації невеликої системи бронювання, автоматизації процесу рекрутингу.

Для інтеграції з зовнішніми системами підтримує інтерфейси SOAP, XML-RPC і REST. Поставляється із засобами інтеграції з такими системами управління версіями, як Subversion, CVS, Git, Clearcase, Team Foundation Server, Mercurial і Perforce.

Існують доповнення, що дозволяють вбудувати Jira в інтегровані середовища розробки, в тому числі Eclipse і IntelliJ IDEA. Перекладена багатьма мовами, включаючи російську, англійську, японську, німецьку, французьку, іспанську.

Для сторонніх розробників надаються кошти розробки розширень системи – плагінів. Розробники розширень можуть викладати плагіни для продажу на спеціальний розділ сайту Atlassian.

Є комерційним продуктом, який може бути ліцензований для роботи на локальному сервері або доступний в якості віддаленого додатки. Ціноутворення залежить від максимального числа користувачів, при цьому близько $50 за користувача для локального і $7 на місяць за користувача для віддаленого доступу є типовими цінами.

Для академічних і комерційних клієнтів доступний повний вихідний код під ліцензією розробника.

Для проектів з відкритим вихідним кодом Atlassian надає спеціальну безкоштовну ліцензію при дотриманні наступних правил:

– проект використовує ліцензії, схвалені Open Source Initiative;

– Вихідний код проекту доступний для скачування;

– у проекту є публічно доступна веб-сайт;

– програмне забезпечення від Atlassian є на веб-сайті проекту.

Розглянемо більш детально роботу антивірусного движка, на якому працює розроблене програмне забезпечення.

Антивірусний модулю (Anti-Virus Engine) – це програмний модуль, що призначений для детектування шкідливого програмного забезпечення. модулю є основним компонентом будь-якої антивірусної програми, незалежно від її призначення. Модуль 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.

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.

База даних антивірусного "движка"

База даних є невід'ємною частиною антивірусного "движка". Більш того, якщо вважати що добре спроектований модулю змінюється не так часто, то антивірусна база змінюється постійно, тому що саме в антивірусній базі перебувають сигнатури, контрольні суми й спеціальні програмні модулі для детектування шкідливих програм. Як відомо, нові віруси, мережні хробаки й інші шкідливі програми з'являються із завидною частотою, і тому дуже важливо, щоб відновлення антивірусної бази відбувалися якнайчастіше. Якщо п'ять років тому було досить щотижневих відновлень, то сьогодні просто необхідно одержувати хоча б щоденні відновлення антивірусної бази.

Також дуже важливо, що саме перебуває в антивірусній базі: чи тільки записи про віруси або ще й додаткові програмні процедури. У другому випадку набагато легше обновляти функціонал антивірусного "движка" шляхом звичайного відновлення баз.

Підтримка "складних", вкладених об'єктів

За останні кілька років антивірусні "движки" сильно змінилися. Якщо першим антивірусам для того, щоб вважатися першокласною програмою, було досить перевіряти системну пам'ять, файли, що виконуються, й завантажувальні сектори, то вже через кілька років у зв'язку з ростом популярності спеціальних утиліт упакування виконавчих модулів перед розроблювачами виникло завдання розпакувати упакований файл перед тим, як його сканувати.

Потім нова проблема – віруси навчилися заражати архівні файли (та й самі користувачі найчастіше пересилали заражені файли в архівах). Антивіруси були змушені навчитися обробляти й архівні файли. В 1995 році з'явився перший макровірус, що заражає документи Microsoft Word. Варто помітити, що формат документів, використовуваний Microsoft Word, закритий, і дуже складний. Ряд антивірусних компаній дотепер не вміють повноцінно обробляти такі файли.

Сьогодні, у зв'язку з величезною популярністю електронної пошти, антивірусні "движки" також обробляють і бази поштових повідомлень і самі

повідомлення.

Методи детектування

У типовому антивірусному "движку", що реалізований у кожній антивірусній програмі, використовуються всі необхідні технології для виявлення шкідливих програм: ефективний евристичний аналізатор, високопродуктивний емулятор і, що саме головне, грамотна й гнучка архітектура підсистеми детектування шкідливих програм, що дозволяє використовувати всі перераховані вище методи детектування.

Майже в кожному антивірусному "движку" базовим є метод детектування за контрольними сумами. Цей метод був обраний виходячи з вимоги мінімізації розміру антивірусних баз. Однак архітектура "движка" часто настільки гнучка, що дозволяє використовувати кожний з перерахованих вище методів детектирування, що й робиться для деяких особливо складних вірусів. Це дозволяє домогтися високого рівня детектування вірусів. Докладніше архітектура антивірусного "движка" представлена на схемі далі в тексті.

Практичне застосування способів виявлення поліморфних вірусів (криптоаналіз і статистичний аналіз, застосування скороченої маски й емуляція), зводиться до вибору найбільш оптимального по швидкодії й обсягу необхідної пам'яті методу. Код більшості вірусів, що самошифруються, досить легко відновлюється процедурою емуляції. Якщо використання емулятора не є оптимальним рішенням, то код вірусу відновлюється за допомогою підпрограми, що реалізує зворотне перетворення – криптоаналіз. Для детектування емуляції вірусів, що не піддаються емуляції і вірусів, для яких не представляється можливим побудувати зворотне перетворення, використовується спосіб побудови скорочених масок.

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм SHACAL-1, який заснований на перетвореннях алгоритму SHA-1. SHACAL-1 шифрує 160-бітний блок даних з використанням 512-бітного ключа шифрування. Допускається використання більше коротких ключів шифрування (не коротше 128 біт), які перед виконанням розширення ключа (процедура розширення ключа також успадкована від SHA-1 і буде описана нижче) повинні бути доповнені нульовими бітами для досягнення 512-бітного розміру.

В алгоритмі SHACAL-1 передбачено 80 раундів шифрування. Шифруєме повідомлення представляється у вигляді п'яти 32-бітних субблоків A, B, C, D і E, над якими в кожному раунді виконуються наступні дії:

$$A_{i+1} = K_i + (A_i <<< 5) + f_i(B_i, C_i, D_i) + E_i + M_i,$$

$$B_{i+1} = A_i,$$

$$C_{i+1} = B_i <<< 30,$$

$$D_{i+1} = C_i,$$

$$E_{i+1} = D_i,$$

де i – номер раунду (i = 0...79),

K_i – фрагмент розширеного ключа для і-го раунду,

f_i – функція для і-го раунду (див. нижче),

<<< – операція побітового циклічного зрушення вліво,

M_i – константи, що модифікують, певні в такий спосіб:

Раунди	Значення константи
0...19	5A827999
20...39	6ED9EBA1
40...59	8F1BBCDC
60...79	CA62C1D6

Використовувані в раундах функції f_i визначені так:

Раунди	Функція
0...19	$f(x,y,z)=(x\&y)\|(x'\&z)$
20...39,60…79	$f(x,y,z)=x \oplus y \oplus z$
40...59	$f(x,y,z)=(x \oplus y)\|(x \oplus z)\|(y \oplus z)$

У таблиці символами &, | і ⊕ позначені, відповідно, побітові логічні операції «і», «або» й «або, що виключає» (XOR); *x*' позначає побітовий комплемент до *x*.

Шифртекстом є конкатенація вмісту змінних A_{80}, B_{80}, C_{80}, D_{80} і E_{80}.

Процедура розширення ключа в алгоритмі SHACAL-1 також досить проста, вона виконується у два етапи:

Етап 1. 512-бітний вихідний ключ шифрування ділиться на 16 фрагментів по 32 біта $K_0 \ldots K_{15}$...

Етап 2. Інші фрагменти розширеного ключа $K_{16} \ldots K_{79}$ обчислюються з перших 16 фрагментів у такий спосіб:

$$K_i = (K_{i-3} \oplus K_{i-8} \oplus K_{i-14} \oplus K_{i-16}) <<< 1.$$

Раунди розшифрування виконуються у зворотній послідовності; кожний з них має на увазі виконання наступних операцій:

$$A_i = B_{i+1},$$

$$B_i = C_{i+1} <<< 2,$$

$$C_i = D_{i+1},$$

$$D_i = E_{i+1},$$

$$E_i = K'_i + (B_{i+1} <<< 5)' + f_i'(C_{i+1} <<< 2, D_{i+1}, E_{i+1}) + A_{i+1} + M'_i + 4.$$

Тут запис $f'(x)$ позначає побітовий комплемент результату виконання операції $f(x)$.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ системи протидії зловмисним програмам з використанням методів машинного навчання яке зображено на рисунку 5.1. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

– Навігаційне меню: Файл; Дії; Параметри; Довідка.

– Розділу обрання групи.

– Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.

– Функціональних кнопок ПЗ.

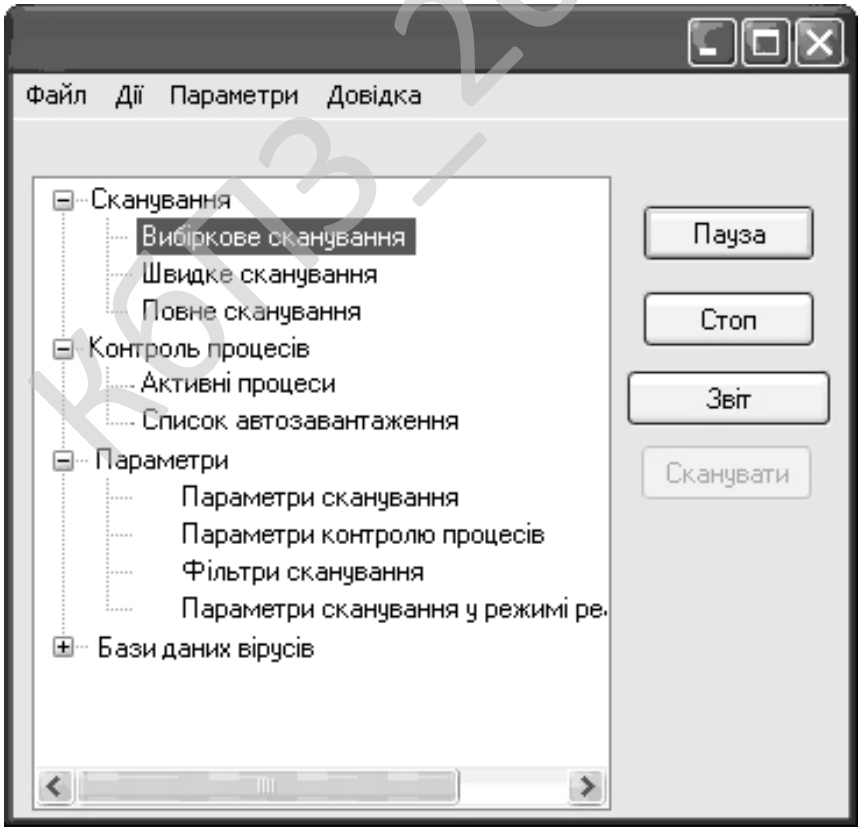

Рисунок 5.1 – Головне вікно ПЗ

Антивірус – це перша програма, яку повинні встановити на новий комп'ютер. Навіть надійний захист можна одержати абсолютно безкоштовно.

Багато користувачів справедливо помітять, що їм досить убудованого в систему антивірусу «Захисника Windows». Однак, як показують тестування незалежних лабораторій, багато сторонніх рішень набагато краще справляються із завданнями захисту, і, крім того, пропонують більше широкі функціональні можливості.

Користувачі, які зовсім зневажають антивірусним захистом, є ідеальними цілями для вірусів, шкідливих атак, експлойтів і інших погроз.

Перші антивірусні алгоритми будувалися на основі порівняння з еталоном. Мова йде про програми, у яких вірус визначається класичним ядром по деякій масці. Зміст алгоритму полягає у використанні статистичних методів. Маска повинна бути, з одного боку, маленькою, щоб обсяг файлу був прийнятних розмірів, а з іншого боку – досить великою, щоб уникнути помилкових спрацьовувань (коли «свій» сприймається як «чужий», і навпаки).

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.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

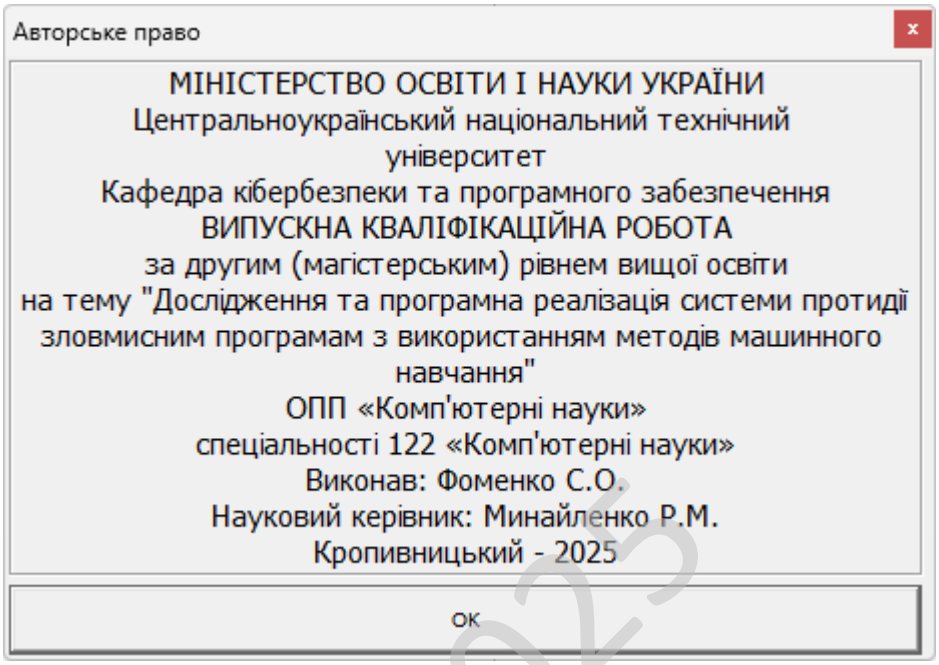

Рисунок 5.2 – Авторське право

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.

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 IT рішення для автоматизації операцій усередині саме цих процедур. Таким чином, розроблене IT 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 IT 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.

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 форматом білої скриньки засноване на аналізі керуючої структури програми. Програма вважається повністю перевіреною, якщо проведено вичерпне тестування маршрутів (шляхів) її графа управління.

У цьому випадку формуються тестові варіанти, в яких:

– Гарантується перевірка всіх незалежних маршрутів програми.

– Знаходяться гілки True, False для всіх логічних рішень.

– Виконуються всі цикли (у межах їхніх кордонів та діапазонів).

– Аналізується правильність внутрішніх структур даних.

Недоліки тестування "білої скриньки":

– Кількість незалежних маршрутів може бути дуже велика.

– Повне тестування маршрутів не гарантує відповідності програми вихідним вимогам до неї.

– У програмі можуть бути пропущені деякі маршрути.

– Не можна виявити помилки, поява яких залежить від даних.

Переваги тестування "білої скриньки" пов'язані з тим, що принцип «білої скриньки» дозволяє врахувати особливості програмних помилок:

– Кількість помилок мінімально в «центрі» і максимально на «периферії» програми.

– Попередні припущення про ймовірність потоку керування або даних у програмі часто бувають некоректними. У результаті типовим може стати маршрут, модель обчислень за яким опрацьована слабо.

– При записі алгоритму програмного забезпечення у вигляді тексту на мові програмування можливе внесення типових помилок трансляції (синтаксичних та семантичних).

– Деякі результати в програмі залежать не від вихідних даних, а від внутрішніх станів програми.

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 протидії зловмисним програмам з використанням методів машинного навчання.

Метою розробки є дослідження та програмна реалізація системи протидії зловмисним програмам з використанням методів машинного навчання.

Об'єктом дослідження є процес протидії зловмисним програмам з використанням методів машинного навчання.

Предметом дослідження є методи протидії зловмисним програмам з використанням методів машинного навчання.

Методи дослідження базуються на методах машинного навчання, методах 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 протидії зловмисним програмам з використанням методів машинного навчання.

– Розроблено вітчизняний продукт протидії зловмисним програмам з використанням методів машинного навчання, 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 цього дослідження можуть бути цікавими насамперед компаніям, що займаються кібербезпекою, оскільки вони постійно шукають нові способи підвищення ефективності захисту інформаційних систем. Використання машинного навчання дозволяє швидше реагувати на загрози, виявляти невідомі віруси та мінімізувати вплив людського фактора. Для таких компаній цей проєкт може стати прикладом інтеграції інноваційних технологій у вже наявні антивірусні рішення. Також розробка може зацікавити великі підприємства, банки, державні установи та освітні організації, які зберігають великі обсяги конфіденційних даних. Вони все частіше стикаються з кібератаками, які не виявляються традиційними засобами захисту. Тому система на базі машинного навчання є привабливою, адже здатна адаптуватися до нових типів загроз і самостійно “навчатися” в процесі роботи. Окрім цього, проєкт буде цікавим для ІТ-компаній, які розробляють інструменти моніторингу, аналітики або хмарних сервісів. Вони зможуть інтегрувати такі модулі у власні продукти, що підвищить їхню ринкову привабливість. Для академічного середовища та студентів ІТ-спеціальностей цей проєкт є цінним з точки зору досвіду практичного використання машинного навчання для вирішення реальних проблем кіберзахисту.

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Щоб оцінити привабливість проєкту, можна залучити експертів з галузі кібербезпеки, аналітиків ІТ-ринку та розробників програмного забезпечення. Вони можуть оцінити систему за такими критеріями, як технологічна

інноваційність, потенціал масштабування, простота інтеграції у корпоративну інфраструктуру, конкурентоспроможність і комерційна доцільність. Наприклад, кожен критерій оцінюється за десятибальною шкалою, а потім виводиться середній показник привабливості. Припустимо, експерти оцінили інноваційність у 9 балів, потенціал масштабування – у 8, конкурентоспроможність – у 9, а комерційний потенціал – у 10. Середній показник привабливості становить 9 балів, що свідчить про високу перспективність рішення. Така оцінка демонструє, що проєкт має всі шанси на успішну реалізацію та подальшу комерціалізацію. Крім кількісної оцінки, експерти можуть надати якісні коментарі – наприклад, рекомендації щодо покращення алгоритмів, підвищення швидкодії системи або вдосконалення інтерфейсу користувача. Це дозволяє не лише оцінити привабливість, а й зробити проєкт більш конкурентоспроможним. Такий підхід дає змогу збалансувати технічну та бізнесову частину дослідження.

7.3 Вибір методу оцінки вартості ПЗ

Для визначення вартості проєкту доцільно застосувати комбінований підхід, який поєднує метод оцінки витрат і метод очікуваних грошових потоків (Discounted Cash Flow). Перший дозволить визначити реальні витрати на розробку системи – заробітну плату програмістів, витрати на сервери, ліцензії, інфраструктуру та тестування. Другий метод дасть можливість оцінити потенційні прибутки від використання або продажу системи. Машинне навчання потребує значних початкових інвестицій, зокрема на збирання та підготовку навчальних даних. Але після розгортання система працює майже автономно, тому експлуатаційні витрати залишаються низькими. У довгостроковій перспективі саме аналіз дисконтованих потоків допоможе оцінити, як швидко проект зможе окупитися. Комбінований підхід забезпечує точність оцінки, адже враховує не лише поточні витрати, а й можливість масштабування та продажу ліцензій чи підписок. Таким чином, компанія може обґрунтовано прийняти рішення щодо інвестицій у проєкт, маючи чітке розуміння як фінансових, так і стратегічних вигід.

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

Вихідні умови (перед впровадженням / після впровадження): кількість інцидентів зараження на рік: до впровадження – 50; після – 10, середні прямі витрати на усунення одного інциденту (відновлення, ІТ-послуги, штрафи, втрата даних): до – 200 000 грн; після – 50 000 грн (за рахунок швидшого виявлення та реагування), витрати на ІТ-персонал, що займається розслідуванням/реакцією: до – 2 000 000 грн/рік; після – 1 200 000 грн/рік, втрати через простій бізнес-процесів (внаслідок інцидентів): до – 1 500 000 грн/рік; після – 500 000 грн/рік, ризик штрафів/репутаційних збитків (оцінка/рік): до – 500 000 грн; після – 100 000 грн, початкові інвестиції в проєкт (розробка/покупка рішення, інтеграція, навчання): 3 000 000 грн, річні витрати на підтримку та оновлення (хмарні сервіси, моделі, ліцензії): 600 000 грн/рік. Вхідні дані зафіксовано в таблиці 7.1.

Таблиця 7.1 – Вихідні дані для розрахунку

Показник	**До впровадження**	**Після впровадження**
Інциденти на рік	50	10
Середні витрати на інцидент, грн	200 000	50 000
Прямі збитки від інцидентів, грн/рік	10 000 000	500 000
Витрати ІТ-персоналу, грн/рік	2 000 000	1 200 000
Втрати від простою бізнес-процесів, грн/рік	1 500 000	500 000
Ризик штрафів/репутаційних збитків, грн/рік	500 000	100 000
Початкові інвестиції, грн	—	3 000 000
Річні витрати підтримки, грн	—	600 000

Розрахунок економічного ефекту демонструє наступне: економія на прямих збитках від інцидентів – 9 500 000 грн/рік, економія на IT-персоналі – 800 000 грн/рік, економія від скорочення простоїв – 1 000 000 грн/рік, зниження ризику штрафів/репутаційних втрат – 400 000 грн/рік, сумарна річна економія (передбачуваний ефект) – 11 700 000 грн/рік, чистий річний ефект з урахуванням витрат підтримки -11 100 000 грн/рік (чистий позитивний ефект), термін окупності ≈ 0,27 року (≈ 3,3 місяця), ROI (річний) ≈ 370%.

Нефінансові вигоди (важливі, але не завжди чисельно вимірювані): підвищення рівня кіберстійкості компанії: зниження шансу значних інцидентів і швидше реагування, покращення репутації та довіри клієнтів, що важливо для фінансових і сервісних організацій, автоматизація рутинних завдань з аналізу логів і сигналів – персонал може фокусуватись на проактивних заходах, краща відповідність нормативам і стандартам (ISO 27001, GDPR/локальні вимоги), спрощення аудиту, зростання внутрішньої культури безпеки: навчання персоналу, політики доступу, превентивні процеси.

Впровадження системи протидії зловмисним програмам із застосуванням машинного навчання може забезпечити значну фінансову та нефінансову віддачу: у наведеному прикладі очікувана чиста річна економія перевищує початкові інвестиції у кілька разів, а термін окупності – менше чотирьох місяців. Проте остаточна оцінка потребує індивідуального аналізу: збору історичних даних, моделювання з урахуванням специфіки бізнес-процесів і тестового пілоту перед масштабуванням.

7.5 Пропозиція алгоритму просування проєкту розробки ПЗ

Початковим етапом просування має стати створення пілотної версії системи, яку можна протестувати на обмеженій кількості підприємств або державних установ. Це дозволить отримати реальні відгуки користувачів, підтвердити ефективність роботи алгоритмів і зібрати кейси для подальшої

демонстрації потенційним клієнтам. Далі важливо запустити інформаційну кампанію, орієнтовану на ІТ-спільноту, за допомогою професійних конференцій, вебінарів і публікацій на спеціалізованих платформах. Демонстрація можливостей системи в реальному часі – наприклад, порівняння швидкості виявлення загроз з традиційними антивірусами – стане потужним маркетинговим інструментом.

Також можна залучити партнерів із галузі кібербезпеки, які вже мають клієнтську базу та канали збуту. Вони зможуть інтегрувати систему у свої рішення або поширювати її як додатковий сервіс. Такий підхід дає можливість охопити ширшу аудиторію без великих витрат на маркетинг.

7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Оптимізація каналів збуту може базуватися на використанні моделі Software as a Service (SaaS), коли клієнти сплачують підписку за користування системою без потреби встановлення локального програмного забезпечення. Це спрощує доступ до продукту та знижує бар'єр входу для нових користувачів. Додатково можна створити безкоштовний базовий пакет із обмеженими функціями для ознайомлення.

Для корпоративного сегмента доцільно передбачити персоналізовані рішення – наприклад, адаптацію моделі машинного навчання під конкретну інфраструктуру клієнта. Це підвищить лояльність і дозволить встановити довгострокові відносини. Розширення мережі партнерів серед системних інтеграторів допоможе швидше виходити на нові ринки.

Крім того, важливо розвивати зворотний зв'язок з користувачами через онлайн-підтримку, форуми або спільноти. Це дозволить швидко реагувати на зауваження, покращувати систему й підвищувати її конкурентоспроможність.

7.7 Визначення ключових факторів успіху конкретного проєкту

Успіх проєкту залежить насамперед від якості алгоритмів машинного навчання та повноти навчальних даних. Чим різноманітніші дані використовуються під час тренування моделей, тим точніше система зможе виявляти нові або змінені загрози. Не менш важливо забезпечити регулярне оновлення моделей, щоб підтримувати їх актуальність у мінливому середовищі кіберзагроз.

Другим фактором є зручність інтеграції системи у вже наявну інфраструктуру підприємств. Якщо рішення можна швидко розгорнути без потреби масштабних змін у мережевій архітектурі, воно буде привабливішим для клієнтів.

Також вагомим є рівень довіри користувачів до продукту. Компанія-розробник має забезпечити прозорість роботи алгоритмів і гарантувати безпеку даних клієнтів. Підтримка та якісний сервіс після продажу допоможуть утримати клієнтів і створити позитивну репутацію. Саме поєднання технологічної надійності, гнучкості та довіри є головним рецептом успіху подібного проєкту.

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Характерною ознакою сучасного науково-технічного прогресу практично у всіх сферах діяльності людини є широке застосування комп'ютерних технологій, заснованих на використанні електронно-обчислювальних машин (ЕОМ). Сьогодення, а тим більше, майбутнє, вже важко уявити без комп'ютерів та іншої електронної техніки. Адже саме завдяки їм стала можливою швидка обробка величезних обсягів інформації, виконання необхідних розрахунків та інших видів робіт, пов'язаних з обробкою текстових даних та ілюстраційних зображень, організація оперативного отримання та передачі інформації, збереження її значних обсягів електронним способом.

Стрімке впровадження комп'ютерів не тільки в сфері управління виробництвом, в банківській систему, бізнесі, системі освіти, але також на транспорті, сфері обслуговування призвело до того, що десятки мільйонів людей у всьому світі виявились втягнутими у взаємодію людини з комп'ютером. Природно виникає запитання: настільки безпечною є ця взаємодія для людини? Адже відома аксіома про те, що будь-яка взаємодія людини та засобів праці двостороння. Людина впливає на удосконалення засобів праці, а останні –на працюючу людину. Отже, навіть сучасні технології та техніка, до яких безперечно, залежать комп'ютерні технології та ЕОМ несуть у собі певні потенційні небезпеки. У зв'язку з цим набуває актуальності адекватна оцінка конкретних умов і характеру праці, яка сприяє обґрунтованому розробленню та впровадженню комплексу заходів і засобів, спрямованих на збереження здоров’я і працездатності людини в процесі праці за рахунок поліпшення параметрів виробничого середовища, зменшення важкості, напруженості трудового процесу та збереження здоров'я працівників на комп'ютеризованих робочих місцях.

Законодавством України чітко врегульовано норми та вимоги до використання комп’ютерної техніки на підприємстві, безпосередньо й охорона праці на підприємстві при роботі за комп’ютером., зокрема «Вимоги щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями», затверджені наказом Мінсоцполітики від 14.02.2018 № 207 [1], «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98. [2].

Загальні вимоги пожежної безпеки під час експлуатації комп’ютерної техніки визначають «Правила пожежної безпеки в Україні» (затверджені наказом МВС від 30.12.2014 № 1417) [3], комп’ютерних класів – пункт 3 розділу VIII «Правил пожежної безпеки для навчальних закладів та установ системи освіти України» (затверджені наказом МОН від 15.08.2016 № 974). [4] та інші державні стандарти, що регламентують експлуатування комп’ютерної техніки як радіоелектронної апаратури.

8.2 Шкідливі і небезпечні фактори при роботі з комп’ютером

Можна виділити наступні основні фактори, що впливають на стан здоров'я людей, які працюють за комп'ютером:

– сидяче положення на протязі тривалого періоду;

– вплив електромагнітного випромінювання монітора;

– втома очей, навантаження на зір;

– перевантаження суглобів кистей;

– стрес при втраті інформації або при виникненні критичних помилок.

У кожному з цих випадків ступінь ризику прямо пропорційний часу, що проводиться за комп’ютером і поблизу від нього. В сучасних умовах взаємодія людини з технікою значно ускладнилась, що вимагає комплексного підходу, який передбачає розгляд людини, технічних засобів праці та виробничого середовища, як взаємозв'язаних елементів єдиної системи. Все вищевказане в повній мірі

відноситься й до системи «людина–комп'ютер–середовище». Вагомий вплив на працездатність та здоров’я користувачів комп’ютерів здійснює виробниче середовище. Це середовище у виробничих приміщеннях (офісах), в основному, визначається мікрокліматом, освітленням, наявністю шкідливих речовин у повітрі, рівнем шуму та випромінювання.

Для того, щоб об’єктивно проаналізувати відповідність умов праці діючим нормативно-правовим актам та запропонувати заходи щодо зменшення негативного впливу комп’ютера на організм людини необхідно скласти санітарно-гігієнічну характеристику умов праці співробітника, який працює з програмним продуктом.

8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці користувача ПК

Розглянемо приміщення в якому працює користувач ПК з програмним продуктом.

Приміщення має одностороннє природне освітлення і загальне штучне освітлення. Стіни і стеля обклеєні світлими шпалерами, підлога вкрита темним ламінатом. У приміщенні відсутні сильні вібрації та шкідливі речовини. Склад повітря відповідає нормі. У кімнаті знаходиться ПК з 4-ядерним процесором і 23-дюймовим IPS-монітором, а також меблі.

Приміщення має довжину 4м, ширину 3,5м, висоту стелі 2,7м. Кількість робочих місць – одне. Площа – 14 м^2, об'єм – 37,8 м^3. Виходячи з цього, отримано дані, наведені в таблиці 8.1.

Таблиця 8.1–Фактичні та нормативні значення параметрів приміщення

Параметр	Норма *	Реальні параметри
Площа, S	не менше 6 м^2	14 м^2
Об’єм, V	не менше 20 м^3	37,8 м^3

*Згідно ДСанПіН 3.3.2.007-98 (Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин).

За даними, які наведено у табл. 8.1, можна зробити висновок, що отримані показники, площа та об'єм приміщення у розрахунку на одно робоче місце користувача ПК відповідає чинним нормам і вимогам.

Щодо мікроклімату, то згідно з ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень» [5] роботу з ПК можна віднести до категорії легка – 1а. Джерелами тепла в цьому приміщенні є люди, електроустаткування, освітлювальні прилади в темний час доби і система опалювання взимку. Оператором виділяється до 120 ккал теплової енергії за годину. Оптимальні та фактичні значення параметрів мікроклімату приведені в таблиці 8.2.

Таблиця 8.2 – Значення параметрів мікроклімату

Період року	Параметр	Оптимальний*	Фактичний
Теплий	Температура	23 – 25 ^{0} С	24^0 С
	Вологість	40 – 60%	50%
	Швидкість повітря	< 0,1м/с	
Холодний	Температура	22 – 24^0 С	23^0 С
	Вологість	40 – 60%	55%
	Швидкість повітря	< 0,1м/с	

*ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень»

За отриманими замірами параметрів мікроклімату можна зробити висновок, що всі показники задовольняють вимогам, зазначеним для робіт категорії праці легка 1а і є задовільними для здоров'я людини.

Щодо освітлення, то згідно з ДБН В.2.5-28:2006 «Природне і штучне освітлення» [6] ця робота відноситься до Va розряду зорових робіт. Передбачається використання природного, штучного і змішаного освітлення.

Природне освітлення здійснюється за допомогою вікна, площа якого складає S' = 1,8х1,5 = 2,7 м^2 і є бічним освітленням. У світильниках місцевого і загального освітлення використовуються світлодіодні лампи потужністю 20 Вт із світловим потоком однієї лампи 900 лм. Згідно замірів рівень освітлення в даному приміщенні і на робочому місці складає в межах 350-500 лк, що відповідає нормованому значенню.

Джерелом шуму в приміщенні є комп'ютер. Вентилятори (кулери) системного блоку, процесора, відеокарти і блоку живлення є сучасними і мають низький рівень шуму. Згідно з технічною документацією шум, зумовлений кулером в блоці живлення складає 25 дБ, кулером процесора – 30 дБ, загальний – 34 дБ. Враховуючи незначний рівень шуму від персонального комп'ютера і незначний рівень фонового шуму від іншого устаткування, можна стверджувати, що сумарний рівень шумового забруднення приміщення не перевищує максимально допустимий рівень коригованої звукової потужності і складає не більше 50 дБА, що відповідає рівню шуму для приміщень з комп’ютерною технікою згідно Державних санітарних правил і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98.

У приміщенні відсутні джерела інфрачервоного, ультрафіолетового і електромагнітного випромінювання, бо монітор ПК вироблений на основі рідкокристалічної матриці, підсвітка якої здійснюється неоновою лампою, які не мають сильного електромагнітного випромінювання і сертифіковані в Україні.

Блок живлення є екранованим і не випускає вищезазначених видів випромінювання.

8.4 Розробка заходів з умов поліпшення охорони праці

Перерахуємо проведені заходи щодо забезпечення умов праці на робочому місці користувача ПК. З точки зору забезпечення електробезпеки до цих заходів

можна віднести: устаткування розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв; періодична перевірка всіх приладів і пристроїв; щорічна здача іспитів з охорони праці. З точки зору забезпечення оптимальних умов мікроклімату, рівня звуку і освітленості до цих заходів можна віднести: організацію природної вентиляції, за допомогою дефлектора, для забезпечення необхідного повітрообміну в приміщенні вузла; організацію системи центрального опалювання, для підтримки оптимальної температури в холодний період року; організацію штучного загального освітлення, для забезпечення необхідних умов зорової роботи, що відповідають, оформлення паспорта на приміщення вузла, з занесенням в нього вимірювань освітленості і рівня звуку, проведених відділом охорони праці.

Крім рекомендацій щодо конкретного приміщення, де було проведено дослідження умов праці, існують загальні вимоги, які зарекомендовані відповідними нормативними документами.

Правильна організація робочих місць запобігає передчасній втомлюваності користувача і сприяє збереженню здоров'я. Організація робочого місця передбачає:

– правильне розміщення робочого місця у виробничому приміщенні;

– вибір ергономічного обґрунтованого робочого положення, виробничих меблів з урахуванням характеристик людини;.

– раціональне компонування обладнання на робочих місцях;

– урахування характеру й особливостей трудової діяльності. Стосовно робочих місць користувача ВДТ, то організація робочого має забезпечуватися відповідно до ДСанПіН 3.3.2-007-98. Для запобігання перевтомленню необхідно виконувати вправи для очей та дотримуватись розпорядку роботи та відпочинку. На робочому місці реалізовувався режим відпочинку: кожні дві години – перерва для виконання фізичних вправ для м‘язів очей.

8.5 Протипожежний захист

Пожежі в приміщеннях з оргтехнікою становлять особливу небезпеку, бо поєднані з великими матеріальними збитками. Пожежа може виникнути при взаємодії горючих речовин і джерел запалювання. Горючими речовинами є будівельні та опоряджувальні матеріали, пластмасові корпуси техніки, шнури тощо. Джерелами запалювання можуть бути електронні схеми комп'ютерів, принтерів, пристроїв електроживлення, де внаслідок різних порушень виникає перегрівання елементів, утворюються електричні іскри та дуги, здатні спричинити займання горючих матеріалів.

При обслуговуванні, ремонтних та профілактичних роботах використовуються різні легкозаймисті рідини, прокладаються тимчасові електропровідники, здійснюється паяння. Виникає додаткова пожежна небезпека, яка потребує відповідних заходів пожежного захисту.

До засобів гасіння пожежі, призначених для локалізації невеликих займань, належать вогнегасники, сухий пісок, азбестові ковдри. Приміщення, в якому встановлено комп'ютери і де немає необхідності влаштування систем автоматичного пожежогасіння, необхідно оснащувати переносними вуглекислотними вогнегасниками з розрахунку 2 шт. на кожні 20 $м^2$ в приміщеннях.

Звуковбирне облицювання стін, стель приміщень треба виконувати з негорючих та важко горючих матеріалів.

З метою виявлення початкової стадії займання необхідно використовувати пристрої систем автоматичного пожежогасіння там, де цього вимагають Правила пожежної безпеки.

З точки зору забезпечення пожежної безпеки до цих заходів можна віднести наявність схеми евакуації з приміщення у випадку пожежі, прикріплену на вхідні двері.

8.6 Розрахункова частина

В приміщенні (де відсутні джерела виділення шкідливих речовин) працює одна людина. Робота пов'язана з використанням ПЕОМ. Розміри приміщення: А = 4 м, В = 3.5 м, Н = 2.8 м, устаткування займає 15% об'єму. Визначимо найменшу необхідну кількість повітря для вентиляції.

Для приміщень, в яких відсутні виділення шкідливих речовин у повітрі, розрахунок вентиляції здійснюється залежно від кількості працюючих.

Необхідна кількість повітря ($м^3$/год.), яка забезпечує відповідність параметрів повітря робочої зони нормованим значенням, визначається за наступною формулою:

$$L = L' \cdot N,$$

де

L' – нормативна кількість повітря на одного працюючого, яка залежить від питомого об'єму приміщення, $м^3$/(год.-люд.);

N – кількість працюючих.

Питомий об'єм приміщення $V_п$, ($м^3$ /люд.), визначається за формулою:

$$V_п = V/N,$$

де V – об'єм приміщення, $м^3$.

Визначаємо вільний об'єм приміщення

$$V = A \cdot B \cdot H \cdot 0{,}85 = 4 \cdot 3{,}5 \cdot 2{,}8 \cdot 0{,}85 = 33{,}3 \text{ м}^3.$$

Питомий вільний об'єм складає

$$V' = V / N = 33{,}2 / 1 = 33{,}2 \text{ м}^3/\text{люд.} > 20 \text{ м}^3/\text{люд.}$$

Нормована кількість повітря на одну людину при V' > 20 $м^3$ /люд. становить 30 $м^3$/(год.·люд.).

Висновки до розділу

У даному розділі магістерської роботи проведено аналіз умов працівника робота якого пов'язана з комп'ютерною технікою. Проведено аналіз основних санітарно-гігієнічних показників в заданому приміщені , де працівник зайнятий

постійною роботою за комп’ютером.. Створені умови повинні забезпечувати комфортну роботу. На підставі вивченої літератури з даної проблеми, були зазначені оптимальні параметри мікроклімату, освітлення, допустимі рівні шуму та іонізуючого випромінювання при роботі з ПЕОМ, а також розраховано найменшу необхідну кількість повітря для вентиляції.

Дотримання умов, що визначають оптимальну організацію робочих місць працівників, дозволить зберегти гарну працездатність протягом усього робочого дня, підвищить як в кількісному, так і в якісному відносинах продуктивність їх праці.

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи протидії зловмисним програмам з використанням методів машинного навчання.

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 протидії зловмисним програмам з використанням методів машинного навчання.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем протидії зловмисним програмам з використанням методів машинного навчання.

– Досліджена система протидії зловмисним програмам з використанням методів машинного навчання.

– На основі отриманих результатів досліджень створена програмна реалізація системи протидії зловмисним програмам з використанням методів машинного навчання.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання протидії зловмисним програмам з використанням методів машинного навчання.

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 SHACAL-1.

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. Фоменко С.О. Дослідження та програмна реалізація системи протидії зловмисним програмам з використанням методів машинного навчання // 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.

52. Smirnov O., Kuznetsov A., Kiian A., Babenko V., Perevozova I.,Chepurko I. «New Approach to the Implementation of Post-Quantum Digital Signature Scheme». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT),* Ukraine, Kyiv, May 14-18. 2020. P. 166-171.