Хмельницький національний університет
Факультет інформаційних технологій
Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА

Ніколайчук Мар'ян Сергійович

на здобуття ступеня вищої освіти Бакалавра

Система захисту конфіденційних даних від шкідливого програмного забезпечення

Галузь знань 12 – Інформаційні технології
Спеціальність 125 – Кібербезпека
Освітня програма Кібербезпека

Шифр КРБКБ. 2102160.21.02.22 ПЗ

Виконав студент 4 курсу група КБ-21-2 Мар'ян НІКОЛАЙЧУК
Керівник канд. техн. наук, доцент Володимир ДЖУЛІЙ
Нормоконтролер старший викладач Сергій МОСТОВИЙ

До захисту допускаю:
Завідувач кафедри кібербезпеки Юрій КЛЬОЦ

2 06 2025 р.

Хмельницький 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Кібербезпеки

Рівень вищої освіти Бакалавр

Галузь знань 12 – Інформаційні технології

Спеціальність 125 – Кібербезпека

Освітня програма Кібербезпека

ЗАТВЕРДЖУЮ

Завідувач кафедри кібербезпеки

Юрій КЛЬОЦ

15 лютого 2025 р.

З А В Д А Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Ніколайчук Мар'ян Сергійович

1 Тема роботи Система захисту конфіденційних даних від шкідливого програмного забезпечення

Керівник роботи Володимир Джулій

Затверджено наказом ректора університету від 7 лютого 2025 № 23

2 Строк подання студентом кваліфікаційної роботи на кафедру 06.06.2025

3 Вихідні дані до роботи Розробити ефективну систему захисту конфіденційних даних від шкідливого програмного забезпечення. Дослідити предметну область, що стосується загроз інформаційній безпеці, пов'язаних із діяльністю шкідливих програм.

4 Зміст пояснювальної записки (перелік питань, які потрібно розробити)
Аналіз предметної області та постановка задачі,Проєктування системи захисту конфіденційних даних,Розробка програмного забезпечення,Тестування та оцінка ефективності

5 Перелік графічного матеріалу (із зазначенням обов'язкових креслень) «Алгоритм роботи системи», «Структура програмного забезпечення системи захисту конфіденційних даних», «Схема процесу проведення оглядового дослідження»

6 Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7 Дата видачі завдання 16 лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
Вибір і затвердження теми кваліфікаційної роботи	Лютий	
Аналіз загроз та вразливостей конфіденційних даних	Лютий	
Дослідження існуючих методів захисту від шкідливого ПЗ	Лютий	
Постановка задачі та визначення вимог до системи захисту	Березень	
Визначення загальних принципів побудови системи захисту	Березень	
Розробка архітектури системи захисту	Квітень	
Реалізація механізмів захисту	Квітень	
Тестування системи захисту на наявність вразливостей	Травень	
Оформлення пояснювальної записки згідно вимог	Травень	
Оформлення графічної частини	Червень	
Захист КР	Червень	

Студент Мар'ян НІКОЛАЙЧУК

Керівник кваліфікаційної роботи Володимир ДЖУЛІЙ

АНОТАЦІЯ

Тема кваліфікаційної роботи: Система захисту конфіденційних даних від шкідливого програмного забезпечення.

Автор роботи: Ніколайчук Мар'ян Сергійович.

Керівник роботи: Джулій Володимир Миколайович.

Пояснювальна записка:72 с., 3 додатки,30 рисунків,0 таблиці,45 джерел.

Графічна частина: плакати, презентаційних слайдів.

СИСТЕМА ЗАХИСТУ КОНФІДЕНЦІЙНИХ ДАНИХ, ШКІДЛИВЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, МОДЕЛЬ ЗАГРОЗ, ПОЛІТИКА БЕЗПЕКИ, АНТИВІРУСНИЙ ЗАХИСТ, ТЕХНІЧНЕ ЗАВДАННЯ, ПЛАН ЗАХОДІВ, МОНІТОРИНГ БЕЗПЕКИ.

Кваліфікаційна робота бакалавра присвячена розробці системи захисту конфіденційних даних від шкідливого програмного забезпечення. У роботі проаналізовано сучасні загрози інформаційній безпеці, зокрема види та методи атак шкідливого програмного забезпечення. Визначено основні вразливості інформаційних систем та розроблено комплекс заходів для їх усунення.

У результаті розроблено та оформлено супровідну документацію до системи захисту план заходів із протидії шкідливому ПЗ, технічне завдання, технороботчий проєкт, політику безпеки, модель загроз.Виконано підготовку до впровадження розробленої системи захисту конфіденційних даних у дію.

28.05.2025 _____

ABSTRACT

Subject of qualification work: Confidential data protection system against malicious software.

Author: Nikolaychuk Maryan Serhiyovych.

Head of work: Dzhuly Volodymyr Mykolayovych.

Explanatory note:72 p., 3 appendices,30 figures,0 tables, 45 sources

Graphic part: posters, presentation slides.

CONFIDENTIAL DATA PROTECTION SYSTEM, MALWARE, THREAT MODEL, SECURITY POLICY, ANTIVIRUS PROTECTION, TECHNICAL TERMS OF USE, ACTION PLAN, SECURITY MONITORING.

The bachelor's qualification work is devoted to the development of a confidential data protection system against malicious software. The work analyzes modern threats to information security, in particular the types and methods of malicious software attacks. The main vulnerabilities of information systems were identified and a set of measures to eliminate them was developed.

As a result, the accompanying documentation for the protection system was developed and drawn up: a plan of measures to counter malicious software, a technical task, a technical design, a security policy, and a threat model. Preparations were made for the implementation of the developed confidential data protection system into operation.

28.05.2025 _____

ЗМІСТ

Вступ 8
1 Аналіз системи захисту конфіденційних даних від шкідливого програмного забезпечення 10
1.1 Аналіз загроз для конфіденційних даних підприємства 10
1.2 Аналіз сучасного шкідливого програмного забезпечення для конфіденційних даних підприємства 15
1.3 Оцінка ризиків для систем захисту конфіденційних даних 22
1.4 Постановка задач 27
2 Проєктування системи захисту конфіденційних даних від шкідливого програмного забезпечення 28
2.1 Принципи та механізми захисту конфіденційних даних від шкідливого програмного забезпечення 28
2.2 Алгоритми та засоби забезпечення захисту конфіденційних даних підприємства 34
2.3 Проєктування архітектури системи захисту конфіденційних даних від шкідливого програмного забезпечення 39
2.4 Висновки 43
3 Розробка системи захисту конфіденційних даних від шкідливого програмного забезпечення 44
3.1 Проєктування алгоритму роботи системи захисту конфіденційних даних від шкідливого програмного забезпечення 44
3.2 Реалізація системи захисту конфіденційних даних від шкідливого програмного забезпечення 52
3.3 Налаштування системи захисту конфіденційних даних від шкідливого програмного забезпечення 61

3.4 Висновки 65
Висновки 67
Перелік джерел посилання 68
Додаток А 73
Додаток Б 80
Додаток В 83

Вступ

Світ переживає революційні зміни, викликані бурхливим розвитком комп’ютерних та інформаційних технологій, які проникають у всі сфери людської діяльності. Завдяки цьому інформація стала найважливішим ресурсом, від якого залежить ефективність, стабільність і безперервність сучасних організацій.

Переваги цифрової трансформації очевидні – швидкість обробки даних, глобальна доступність інформації, інтеграція різноманітних сервісів та інструментів. Однак разом зі зростанням обсягів і ускладнення інформаційних потоків з'являються нові загрози, які можуть мати руйнівний вплив як на окремі підприємства, так і на державні структури.

Кіберзлочинці та хакерські групи використовують все більш витончені методи атак, які підривають основи інформаційної безпеки та ставлять під сумнів традиційні підходи до захисту даних.Актуальність проблеми захисту конфіденційних даних особливо очевидна в світі, де інформація є стратегічним ресурсом, а її втрата або компрометація може призвести до серйозних економічних і репутаційних втрат. Розвиток цифрових технологій у бізнесі, уряді, медицині, освіті та інших галузях породжує безпрецедентну залежність від інформаційних систем.

Водночас зростає кількість і різноманітність загроз, пов’язаних зі шкідливим програмним забезпеченням – вірусами, троянами, програмами-вимагачами та шпигунськими програмами. Ці загрози не тільки порушують роботу систем, але також можуть спричинити витік конфіденційних даних, змінити їх вміст або навіть зробити їх повністю недоступними. Таким чином, питання розробки надійних і комплексних систем захисту конфіденційної інформації набуває особливого значення в умовах сучасної кібервійни. За останні роки методи і засоби шкідливого програмного забезпечення отримали значний розвиток.

Зловмисники активно використовують можливості штучного інтелекту, машинного навчання та інші сучасні технології для автоматизації та підвищення ефективності своїх атак. Це призводить до того, що традиційні методи захисту, такі як антивірусне програмне забезпечення або базові брандмауери, вже не завжди здатні вчасно виявити та нейтралізувати новітні загрози. Тому існує потреба у створенні комплексних систем захисту, здатних інтегрувати різні інструменти та технології, адаптовані до сучасних умов кібербезпеки. Серед основних завдань, які стоять перед дослідниками в галузі інформаційної безпеки, є розробка методів раннього виявлення загроз, попередження атак і швидкого реагування на інциденти.

Сучасні системи захисту повинні враховувати численні фактори: як технічні особливості програмного забезпечення, так і людський фактор, який часто є найслабшою ланкою в системі безпеки. Крім того, важливим аспектом є забезпечення сумісності та інтеграції різних засобів захисту, що дозволить створити єдину екосистему, здатну ефективно протистояти складним загрозам. З цією метою особлива увага приділяється використанню сучасних технологій шифрування, аутентифікації користувачів, а також реалізації адаптивних алгоритмів аналізу поведінки системи для виявлення аномалій у роботі інформаційних мереж.

Метою роботи є розробка ефективної системи захисту конфіденційних даних від шкідливого програмного забезпечення на основі інтеграції сучасних технологій захисту та методів аналізу кіберзагроз. Дослідження спрямовані на розробку комплексного підходу, який включатиме як програмні засоби, так і апаратні та організаційні заходи для підвищення стійкості інформаційних систем до зовнішніх атак.

Особлива увага приділяється аналізу наявних уразливостей, розробці нових алгоритмів виявлення шкідливих компонентів у системах, а також методів оперативного реагування на кіберінциденти.

За темою дипломної роботи опубліковано 1 теза доповідей.

1 АНАЛІЗ СИСТЕМИ ЗАХИСТУ КОНФІДЕНЦІЙНИХ ДАНИХ ВІД ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Аналіз загроз для конфіденційних даних підприємства

Експерти з інформаційної безпеки ESET прогнозують, що у 2025 році кіберзагрози стануть ще більш витонченими та небезпечними, особливо коли мова йде про конфіденційні дані підприємств.

Сучасні зловмисники використовують штучний інтелект для створення реалістичних фейкових повідомлень, що створює додаткові труднощі для виявлення спроб соціальної інженерії. Завдяки потужним мовним моделям зловмисники можуть створювати переконливі фішингові електронні листи, маніпулювати думкою співробітників і навіть створювати фальшиві аудіо- та відеоматеріали, які використовують технологію deepfake, щоб видавати себе за керівників або громадських діячів.

У цьому контексті використання штучного інтелекту перетворюється з простого інструменту атаки на складну систему, здатну проводити багатоетапну компрометацію даних, включаючи автоматизацію шкідливих електронних листів і організацію масштабних дезінформаційних кампаній у соціальних мережах [1].

Мобільні пристрої, які стали невід'ємною частиною корпоративної інфраструктури, також перебувають під постійним ризиком атак. Зловмисники активно користуються можливістю створювати фейкові мобільні додатки, які імітують офіційні сервіси в обхід традиційних заходів безпеки. Такі технології, як прогресивні веб-додатки та WebAPK, дозволяють розробляти шкідливі програми, які виглядають легітимно, але здатні збирати конфіденційну інформацію.

Крім того, використання Flutter SDK для розробки мобільних додатків відкриває нові можливості для приховування шкідливого коду, що ускладнює його виявлення навіть для сучасних систем безпеки. Незважаючи на високий рівень безпеки, iOS-пристрої також можуть бути атаковані через фішингові сайти, перехоплені вкладення електронної пошти та рекламні кампанії в соціальних

мережах, що додає додатковий рівень невизначеності до захисту корпоративних даних.

В умовах глобальних геополітичних конфліктів та економічних загроз кібершпигунство набуває все більшого поширення, ставлячи під загрозу не лише корпоративні, а й державні інтереси. Зловмисники використовують VPN для анонімізації своїх дій, що дозволяє їм здійснювати атаки на державні установи та підприємства без можливості їх швидкого виявлення.

Атаки на об'єкти критичної інфраструктури, включаючи телекомунікаційні компанії, енергетичні компанії та транспортні системи, можуть мати руйнівні наслідки як для національної безпеки, так і для стабільної роботи бізнесу. Особливо небезпечними є спроби отримати стратегічно важливу інформацію, яка може бути використана для маніпулювання внутрішньою політикою або економічного шантажу. Зростаюча активність кібершпигунських кампаній, спрямованих на отримання конфіденційних даних, вимагає від бізнесу швидкого реагування та впровадження комплексних заходів безпеки.

Варто також зазначити, що програми-вимагачі постійно еволюціонують, адаптуючись до сучасних методів кіберзахисту.

Новітні версії шкідливого програмного забезпечення розробляються для обходу систем моніторингу та реагування, таких як EDR та SIEM.

Такі кіберзлочинні групи, як RansomHub, активно конкурують з традиційними організаціями, постійно вдосконалюючи свої технології, що змушує фахівців з кібербезпеки шукати нові методи протидії.

Через здатність програм-вимагачів блокувати інструменти моніторингу бізнес стикається з додатковими труднощами у виявленні та нейтралізації загроз, що вимагає постійного оновлення систем безпеки та швидкого реагування на нові виклики.

Через зростаючу складність та багатогранність кіберзагроз бізнесу необхідно впроваджувати комплексний підхід до захисту конфіденційних даних, що включає як технічні, так і організаційні заходи.Системи моніторингу та

реагування, багатофакторна автентифікація та регулярне оновлення програмного забезпечення дозволяють значно знизити ризики несанкціонованого доступу до корпоративних ресурсів. На рисинку 1.1 зображено схему роботи MFA.

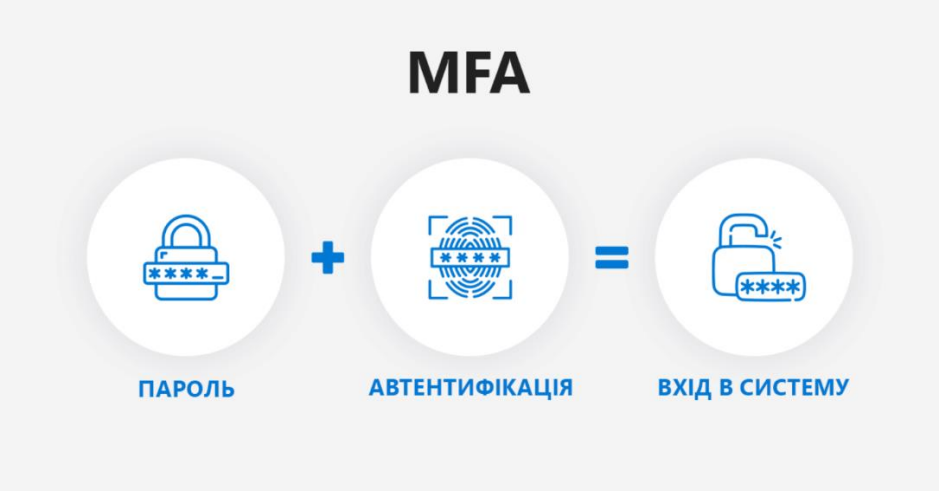


Рисунок 1.1 – Схема роботи MFA

Водночас важливо розробити внутрішні політики безпеки, які регламентують поводження з конфіденційною інформацією, і регулярно навчати співробітників. Аудит безпеки допомагає виявити слабкі місця системи та оперативно вжити заходів для їх усунення. Особлива увага приділяється інтеграції хмарних сервісів, що відкриває нові вектори атак, але також надає можливості для зручного зберігання та обробки даних. Забезпечення безпечного доступу до хмарних платформ, шифрування даних під час передачі та зберігання, контроль доступу до корпоративних ресурсів стають ключовими елементами сучасної стратегії кібербезпеки [2].

Бізнес також повинен враховувати зовнішні фактори, оскільки атаки можуть відбуватися не тільки безпосередньо, але й через постачальників або партнерів, які мають доступ до внутрішніх систем компанії.

Неправильна організація роботи персоналу, відсутність чітких внутрішніх правил використання корпоративних ресурсів, недостатня кіберграмотність можуть стати вирішальними у випадку спроби компрометації даних.

Компрометація конфіденційної інформації може призвести до значних фінансових втрат, шкоди репутації компанії, а також юридичних наслідків у разі порушення стандартів зберігання персональних даних.

Тому компаніям варто активно інвестувати у впровадження сучасних технологій, що дозволяють швидко реагувати на загрози, а також у розвиток внутрішньої культури кібербезпеки.

Враховуючи сучасний ландшафт кіберзагроз, експерти прогнозують подальший розвиток технологій безпеки, заснованих на машинному навчанні та аналізі великих даних.

Інтеграція штучного інтелекту в системи безпеки дозволить виявляти аномалії в поведінці користувачів і швидко реагувати на потенційні атаки. Співпраця між бізнесом, державними органами та міжнародними організаціями сприятиме обміну інформацією про кіберінциденти та виробленню спільних стандартів інформаційної безпеки.

Водночас розвиток нормативно-правової бази та впровадження етичних стандартів у сфері кібербезпеки стане запорукою зниження ризиків та підвищення довіри до цифрових технологій. Ефективна протидія сучасним загрозам вимагає комплексного підходу, що поєднує технічні заходи з організаційно-правовими аспектами для забезпечення надійного захисту конфіденційних даних компанії у світі, де кіберзагрози постійно розвиваються[3].

Таким чином, сучасний ландшафт кібербезпеки стає все більш складним і вразливим, що вимагає надійної стратегії безпеки. Підприємства повинні зосередитися на покращенні якості інформації для поширення інформації та навчання своїх працівників.

Хороша стратегія безпеки, яка включає регулярні оновлення програмного забезпечення, використання багатофакторної автентифікації та створення політик безпеки, зменшує ризик несанкціонованого доступу.

Інтеграція хмарних служб, контроль доступу та швидке реагування на кіберзагрози стають дедалі важливішими в сучасному середовищі безпеки.

Водночас зростаюча складність кібершпигунства та поширення програм-вимагачів означає, що підприємства повинні адаптуватися до нових загроз, включаючи технології та можливості кібербезпеки.

Поєднання сучасних технологій у поєднанні з інтенсивним навчанням персоналу може захистити конфіденційні дані та забезпечити ефективність організації в умовах загрози.

Сучасні EDR-платформи та пісочниці, які аналізують поведінкові сигнатури шкідливих двійкових файлів, не встигають за все більш витонченими «сплячими» шкідливими програмами, які розпізнають віртуалізовані середовища та активуються лише на живій системі, тому гібридні стратегії з емпіричними профілями та крос-платформними сигнатурами стають все більш важливими.

Щоб мінімізувати латеральне переміщення загроз всередині корпоративної мережі, організації впроваджують концепцію Zero Trust і сегментацію мережі: перевірка кожного запиту незалежно від його походження дозволяє блокувати шкідливий код на рівні окремих зон, зменшуючи потенційний масштаб атаки.

Водночас активно розвиваються платформи SOAR, які агрегують дані з SIEM, EDR та Threat Intelligence Feeds, автоматизуючи типові сценарії та значно скорочуючи час реагування на інциденти, хоча й потребують регулярного оновлення для уникнення хибних спрацьовувань.

Людський фактор залишається ключовим елементом: впроваджуються адаптивні навчальні платформи, які аналізують роботу співробітників під час фішингових симуляцій і пропонують персоналізовані уроки безпеки, а регулярні настільні навчання допомагають відточувати процедури оповіщення та реагування. Обсяги телеметрії та логів зростають в геометричній прогресії, тому

для виявлення аномалій активно використовується машинне навчання та аналіз великих даних: зміщення часових патернів активності, незвичні обсяги передачі даних або доступ з нових геолокацій негайно генерують оповіщення про ризики, які надсилаються на платформи GRC для подальшого розслідування.

Регуляторна складова все більше впливає на стратегії безпеки: дотримання GDPR, ISO/IEC 27001 та NIST SP 800-53 вимагає не лише технічних заходів, але й інтеграції перевірок у конвеєри CI/CD, таких як автоматичне сканування коду на наявність персональних даних перед розгортанням. Недотримання цих стандартів може призвести до штрафів у розмірі до 4% річного обороту та підірвати довіру клієнтів. Зростає роль кіберстрахування при прийнятті рішення про поліс аналізується не лише вартість потенційних збитків від витоку даних та простою бізнес-процесів, але й рівень внутрішньої готовності та історія інцидентів, що впливає на вартість страхових премій.

1.2 Аналіз сучасного шкідливого програмного забезпечення для конфіденційних даних підприємства

У контексті цифрової трансформації підприємства стикаються зі значними викликами у сфері інформаційної безпеки. Шкідливе програмне забезпечення є одним із основних інструментів кіберзлочинців, спрямованих на компрометацію, викрадення або знищення конфіденційних даних. У цьому розділі розглядаються основні типи шкідливих програм, механізми атак і реальні випадки кіберінцидентів, які мали значні наслідки для бізнесу.

Шкідливе програмне забезпечення поділяється на кілька основних категорій, кожна з яких має свої особливості та методи атаки. Комп’ютерні віруси це саморозмножувальні програми, які заражають інші файли та поширюються по системі. Вони можуть знищити або пошкодити дані. Троянські коні маскуються під корисне програмне забезпечення, але надають зловмисникам прихований доступ до системи.

Шпигунське програмне забезпечення мовчки збирає інформацію про користувача, наприклад паролі, фінансові дані або історію веб-перегляду.

Програмне забезпечення для банківського шахрайства викрадає інформацію про банківську картку та онлайн-банківський рахунок. Програми-вимагачі шифрують файли користувача та вимагають викуп за їх розшифровку. Бекдори створюють приховані логіни, які дозволяють зловмисникам отримати доступ до зараженого пристрою. Руткіти змінюють системні процеси та приховують наявність шкідливих програм у системі.

Сучасне зловмисне програмне забезпечення використовує різні методи атак, зокрема: фішингові атаки з використанням електронної пошти з вкладеними файлами або посиланнями на шкідливі веб-сайти. Соціальна інженерія, що маніпулює користувачами для отримання доступу до конфіденційної інформації. Експлойти з використанням уразливостей у програмному забезпеченні для несанкціонованого доступу.

Заражені веб-сайти під час відвідування таких сайтів шкідливий код автоматично завантажується на пристрій користувача. Зараження через знімні носії з використанням інфікованих USB-накопичувачів або інших фізичних носіїв інформації. Мережеві атаки, що проникають в систему через слабкі місця в мережевих протоколах або незахищені порти. Зловмисне програмне забезпечення неодноразово спричиняло серйозні наслідки для глобальних компаній і організацій.

Для захисту конфіденційної корпоративної інформації необхідно застосовувати комплексні заходи кібербезпеки. Антивірусне та антишкідливе програмне забезпечення регулярно оновлюйте та використовуйте найновіші рішення для виявлення шкідливих програм.

Захист мережі налаштуйте брандмауер і систему виявлення вторгнень IDS/IPS. Систематичні оновлення програмного забезпечення – встановлюйте останні оновлення для усунення вразливостей.

Навчання персоналу підвищення обізнаності співробітників щодо фішингових атак, соціальної інженерії та правил безпечного використання інформаційних систем.

Резервне копіювання регулярно створюйте резервні копії даних, щоб відновити інформацію після атаки програм-вимагачів. Найменші права доступу застосуйте принцип найменших привілеїв, щоб обмежити потенційні поверхні для атак.Моніторинг і аудит безпеки безперервний аналіз трафіку, виявлення відхилень і перевірка безпеки вашої IT-інфраструктури. На рисинку 1.2 зображено схему процесу проведення оглядового дослідження.

Рисунок 1.2 – Схема процесу проведення оглядового дослідження

Аналіз сучасних шкідливих програм набуває особливої актуальності в умовах постійного розвитку кіберзагроз. Сучасні мережеві атаки більше не обмежуються використанням базових експлойтів, а включають складні методи, які використовують слабкі місця в мережевих протоколах, незахищені порти та вразливості програмного забезпечення. Такі атаки включають як зловмисні дії, спрямовані на отримання доступу до корпоративних мереж, так і масштабні операції, спрямовані на компрометацію конфіденційних даних.

Одним із найвідоміших прикладів є атака програм-вимагачів WannaCry, яка стала справжнім каталізатором змін у підходах до кібербезпеки на глобальному рівні. WannaCry, використовуючи критичну вразливість в операційній системі Windows протокол SMB, швидко поширювався корпоративними мережами, не вимагаючи взаємодії з користувачем. Наслідки атаки були масштабними - під

загрозою блокування даних опинилися сотні тисяч комп'ютерів у різних країнах, що підтвердило необхідність постійного моніторингу систем і швидкого впровадження оновлень безпеки[4]. Ще один яскравий приклад атака Petya, яка завдала мільярдних збитків великим компаніям і державним установам. Особливістю Petya є те, що він використовував складні механізми шифрування даних і мав ознаки кіберзброї, націленої на критичну інфраструктуру.

Ця атака підкреслила важливість застосування принципу «найменших привілеїв» і створення резервних копій, які дозволяють швидко відновити дані в разі зламу[5]. Ботнет Emotet є прикладом того, як зловмисне програмне забезпечення може розвинутися від простого трояна до потужної модульної платформи, здатної доставляти додаткові шкідливі компоненти. Спочатку Emotet зосереджувався на крадіжці банківських облікових даних, але згодом Emotet перетворився на створення додаткових загроз, зокрема програми-вимагачі та шпигунські програми.

Його здатність адаптуватися до різних середовищ і інтегрувати нові модулі робить його особливо небезпечним для корпоративних систем.На рисинку 1.3 наведено візуальне представлення архітектури ботнету Emotet.

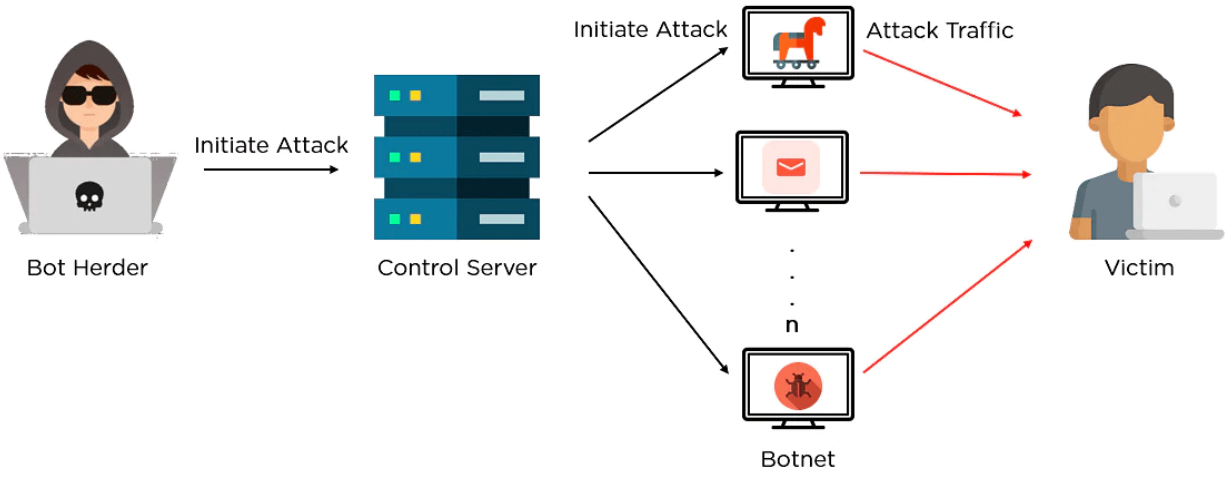


Рисунок 1.3 – Візуальне представлення архітектури ботнету Emotet

Не менш важливою є атака SolarWinds, яка продемонструвала складність сучасних ланцюжків поставок програмного забезпечення. Зловмисники використовували бекдор, вбудований в оновлення програмного забезпечення SolarWinds, щоб отримати доступ до урядових і корпоративних мереж у Сполучених Штатах. Ця атака є яскравим прикладом того, як компрометація одного постачальника може мати серйозні наслідки для великої кількості організацій[6].

Уразливість Log4Shell, виявлена в бібліотеці Log4j, стала новим викликом для експертів з безпеки. Він дозволяв дистанційне виконання коду без аутентифікації, що відкривало можливості для масового використання цього експлойту з несанкціонованим доступом до системи. Атаки, спрямовані на Log4Shell, показали, що вкрай важливо регулярно оновлювати залежності програмного забезпечення, а також впроваджувати механізми моніторингу аномалій у роботі програми.

Щоб продемонструвати цей процес, ви можете використовувати схему, що показує процес виявлення та усунення вразливості, яку можна розмістити на корпоративних ресурсах з аналізом загроз.

Усі ці приклади свідчать про те, що сучасні шкідливі програми постійно розвиваються, пристосовуючись до нових умов і обходячи традиційні методи захисту. Для ефективної протидії подібним загрозам необхідний комплексний підхід, що включає низку заходів.Перш за все, критичним моментом є оновлення системного програмного забезпечення. Компанія повинна впроваджувати політику регулярних оновлень операційних систем, програм і бібліотек, що використовуються у виробничих середовищах.

Використання системи керування виправленнями дозволяє швидко реагувати на виявлені вразливості та запобігати їх використанню зловмисниками.Щоб забезпечити захист від нових загроз, необхідно постійно оновлювати антивірусні та шкідливі програми.

Сучасні системи захисту використовують алгоритми машинного навчання, які можуть виявляти аномальну поведінку зловмисників навіть у разі невідомих загроз.

Щоб підвищити рівень захисту від захисту, інтегруйте такі рішення, як EDR виявлення кінцевої точки та реагування і SIEM інформаційна система безпеки та керування подіями, які забезпечують моніторинг подій у реальному часі та швидке реагування на інциденти.

Не менш важливим є питання підготовки кадрів. Людський фактор часто є найслабшою ланкою в системах безпеки, тому регулярне навчання, моделювання фішингових атак і семінари з кібербезпеки зменшать ризик успішної соціальної інженерії.

Рекомендується використовувати інтерактивні платформи та сервіси для підвищення обізнаності співробітників, що дозволяє вчасно виявляти спроби шахрайства.

Ще одним важливим елементом захисту корпоративної інформації є принцип найменших привілеїв. Обмеження доступу до критичних ресурсів мінімізує поверхню атаки. Використання інструментів управління ідентифікацією та доступом допомагає посилити контроль над тим, хто має доступ до яких даних.

Щоб візуалізувати цей підхід, можна створити діаграму розподілу прав доступу, яка показує, як привілеї розподіляються між користувачами в організації.

Резервне копіювання даних є останнім, але не менш важливим заходом для забезпечення безперервності бізнес-процесів.

Регулярне резервне копіювання дозволяє швидко відновити дані в разі атаки програм-вимагачів або іншої кіберзлочинної діяльності.

Важливо не тільки зберігати резервні копії, а й перевіряти їх на цілісність і готовність до відновлення. На цьому етапі може бути корисною інфографіка, яка демонструє процес резервного копіювання та відновлення даних.

У сучасних умовах особливе місце займає аналіз поведінкових характеристик шкідливого програмного забезпечення.

Статичний аналіз дозволяє виявити основні характеристики коду, але саме динамічне дослідження в ізольованому середовищі показує реальні шкідливі дії програмного забезпечення в мережі підприємства. Такий підхід сприяє точнішому визначенню векторів атак, що, в свою чергу, дозволяє оптимізувати протидію.

Тому компанія пропонує запровадити вимоги безпеки для всіх постачальників програмного забезпечення та послуг, провести аудит їхніх систем безпеки та використовувати інструменти перевірки програмного забезпечення перед встановленням у робочому середовищі.

Комплексний підхід до захисту конфіденційних даних передбачає інтеграцію кількох рівнів безпеки: від апаратного та мережевого рівнів до рівня програмного забезпечення та людського фактору.

Поєднання технічних засобів, політик безпеки і лише постійного навчання персоналу дозволяє створити багаторівневу систему захисту, здатну ефективно протидіяти сучасним кіберзагрозам.

Підсумовуючи, можна зазначити, що сучасне шкідливе програмне забезпечення характеризується високою динамікою розвитку та адаптованістю до ринкових умов.

Атаки, які використовують уразливості в мережевих протоколах, незахищені порти, а також складні багатомодульні рішення на образах Emotet або SolarWinds, змушують організації переглянути свої підходи до безпеки.

Основою ефективної протидії цим загрозам є впровадження сучасних технологій безпеки, таких як системи EDR і SIEM, а також регулярні аудити, моніторинг і навчання персоналу.

1.3 Оцінка ризиків для систем захисту конфіденційних даних

Система захисту конфіденційних даних від шкідливих програм базується на багаторівневому підході, що поєднує в собі як превентивні заходи, так і механізми виявлення та реагування на інциденти.

Першою лінією захисту є впровадження сучасних антивірусних рішень корпоративного рівня, які безперервно сканують файли та процеси в режимі реального часу, аналізуючи їх на наявність відомих сигнатур шкідливих програм.

Однак покладатися лише на аналіз сигнатур недостатньо, оскільки зловмисники постійно модифікують своє програмне забезпечення, щоб обійти традиційні засоби виявлення.

Тому необхідно доповнювати антивірусні продукти поведінковим аналізом, який відстежує аномалії в системі: нехарактерні спроби запису в критичні ключі реєстру, підозрілу активність мережевих з'єднань, запуск невідомих процесів з правами адміністратора. Поєднання сигнатурного та поведінкового аналізу значно підвищує ймовірність виявлення нових і модифікованих загроз[7],[8].

Другим важливим компонентом є система Endpoint Detection and Response, яка записує повну хронологію кожного процесу на хостах підприємства. На рисинку 1.4 наведено представлення Endpoint Detection and Response.

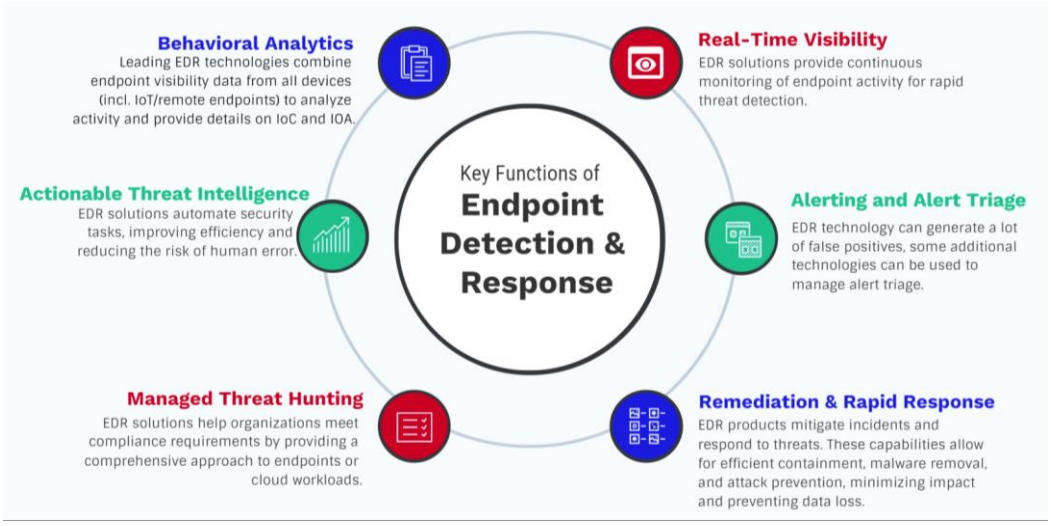


Рисунок 1.4 – Представлення Endpoint Detection and Response

Цей інструмент надає адміністраторам детальну інформацію про походження інциденту, його поширення мережею та повну картину шкідливого коду. Сучасні EDR-рішення мають можливість автоматично ізолювати уражений хост, запобігаючи подальшому поширенню загрози[9].

Бажано налаштувати політики таким чином, щоб у разі виявлення підозрілої активності сервер або робоча станція потрапляли на карантин з обмеженим доступом до корпоративних ресурсів до повного розслідування інциденту.

На рисинку 1.5 наведено схему взаємодії компонентів EDR з SIEM та антивірусом.

Рисунок 1.5 – Схема взаємодії компонентів EDR з SIEM та антивірусом

Не менш важливою є організація сегментації мережі на віртуальні зони з різними рівнями довіри та контролю. Критично важливі сервіси повинні бути розміщені в ізольованих підмережах, доступ до яких суворо обмежений за допомогою брандмауерів і проксі-серверів з глибокою перевіркою пакетів [10].

Поділ внутрішньої корпоративної мережі на сегменти допомагає запобігти латеральному переміщенню шкідливого програмного забезпечення в разі успішного проникнення.

Також слід застосовувати принцип нульової довіри, коли всі користувачі та пристрої, навіть ті, що знаходяться всередині периметра, проходять сувору автентифікацію та авторизацію перед доступом до ресурсів[11].

На рисинку 1.6 наведено блок-схему архітектури сегментованої мережі з демілітаризованими зонами DMZ, внутрішніми VLAN і зонами контролю.

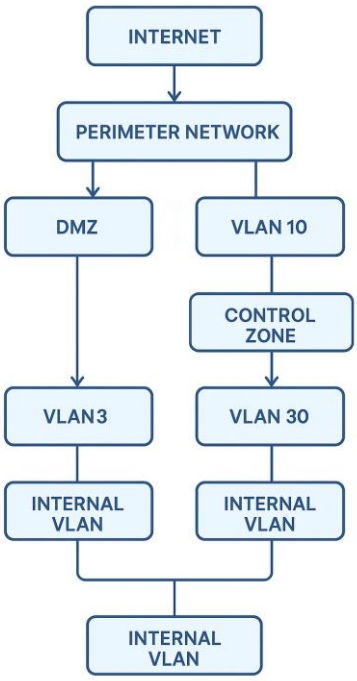


Рисунок 1.6 – Блок-схема архітектури сегментованої мережі

Основою безпеки є рольовий контроль доступу RBAC і принцип найменших привілеїв, коли жоден користувач або процес не має більше прав, ніж необхідно для виконання своїх завдань.

Надзвичайно ефективним є впровадження багатофакторної автентифікації MFA для всіх критично важливих систем, включаючи VPN-з'єднання та приватні хмарні сервіси. MFA знижує ризик успішного використання викрадених

облікових даних, адже навіть якщо пароль буде скомпрометований, зловмисник не зможе пройти другий рівень перевірки. Регулярний аудит журналів доступу та дотримання політик безпеки допомагають вчасно виявити спроби витоку або модифікації привілейованих даних.

Важливим компонентом є система резервного копіювання та відновлення даних, яка забезпечує збереження конфіденційної інформації, навіть якщо файли зашифровані шкідливим програмним забезпеченням . Копії даних слід зберігати на окремих дисках поза основною мережею, а цілісність архівів слід перевіряти. Успішна стратегія включає поєднання точок відновлення і архівування журналів транзакцій, що дозволяє відновити систему до стану до інциденту з мінімальними втратами[12].

Для раннього виявлення нових загроз корисно впроваджувати системи honeypot — пастки, які імітують вразливі сервери або робочі станції, призначені виключно для відлову зловмисної активності. Аналіз трафіку та поведінки шкідливих програм у такому середовищі дозволяє своєчасно оновлювати правила захисту в основних системах. Інтеграція результатів аналізу honeypot зі службами аналізу загроз сприяє оперативному поповненню баз знань про нові загрози. На рисинку 1.7 наведено принцип роботи системи honeypot.

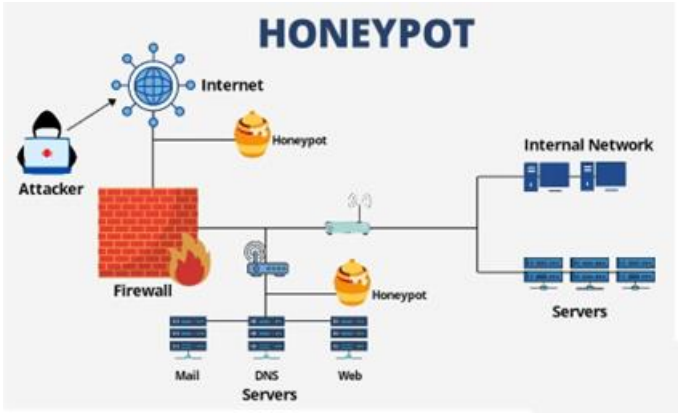


Рисунок 1.7– Принцип роботи системи honeypot

У складній системі захисту неможливо обійтися без чіткого Плану реагування на інциденти, який включає ролі та обов’язки групи безпеки, алгоритми зв’язку з керівництвом і зовнішніми партнерами, а також процедури супроводу постраждалих систем в ізольоване середовище.

Регулярні тренування та відпрацювання сценаріїв «що-якщо» забезпечують готовність особового складу швидко та злагоджено діяти в критичній ситуації. Після завершення відповіді обов’язково проводиться ретроспектива для виявлення слабких місць у процесі та коригування політики захисту.

Завершальним етапом є постійний моніторинг ефективності впроваджених заходів. Для цього використовуються ключові показники ефективності, такі як середній час до виявлення, середній час до відповіді, кількість зупинених атак на етапі запобігання тощо. Аналіз KPI дозволяє гнучко переглядати пріоритети інвестицій у безпеку та планувати оновлення технологічних рішень[13],[14].

Завдяки поєднанню описаних компонентів — антивірусної та EDR-систем, SIEM із SOAR, сегментованої інфраструктури, суворих політик контролю доступу, резервного копіювання, honeypot traps і чітких процедур реагування — створюється надійний щит від шкідливого програмного забезпечення, який забезпечує захист конфіденційних даних компанії та мінімізує можливі фінансові та репутаційні втрати. На рисинку 1.7 наведено принцип роботи SOAR з SIEM.

Рисунок 1.8– Принцип роботи SOAR з SIEM

1.4 Постановка задач

За результатами проведеного дослідження сучасних загроз інформаційній безпеці, а також аналізу наявних технологій захисту, постає задача розробки ефективної системи захисту конфіденційних даних від шкідливого програмного забезпечення. З огляду на зростання кількості кібератак, спрямованих на викрадення персональної та службової інформації, актуальність створення такої системи значно зросла.

Об'єктом роботи визначено розробку системи, яка забезпечуватиме виявлення, блокування та запобігання несанкціонованому доступу до конфіденційної інформації з боку шкідливого програмного забезпечення. Йдеться про дані користувачів, що зберігаються та обробляються на пристроях або в інформаційних системах організацій, які можуть стати мішенню для атак.

Мета розробки системи захисту конфіденційних даних від шкідливого програмного забезпечення полягає в:

– забезпеченні високого рівня безпеки даних шляхом своєчасного виявлення шкідливого ПЗ та його нейтралізації; запобіганні несанкціонованому доступу до конфіденційної інформації шляхом застосування механізмів моніторингу, шифрування та ізоляції даних;

– використанні сучасних підходів до аналізу поведінки програм, штучного інтелекту та машинного навчання для підвищення ефективності виявлення нових та раніше невідомих загроз;

– формуванні безпечного середовища для роботи користувачів без зниження продуктивності та з урахуванням зручності в експлуатації.

2 ПРОЄКТУВАННЯ СИСТЕМИ ЗАХИСТУ КОНФІДЕНЦІЙНИХ ДАНИХ ВІД ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Принципи та механізми захисту конфіденційних даних від шкідливого програмного забезпечення

Сучасна інформаційна інфраструктура підприємств працює в умовах постійного ризику, пов'язаного із загрозами, що виникають з боку шкідливого програмного забезпечення. У зв'язку з цим особливо актуальним стає питання захисту конфіденційних даних, після чого ці дані можуть стати головною ціллю кібератаки. Для забезпечення належного рівня захисту розроблені як загальні підходи, так і конкретні принципи та механізми протидії загрозам, пов'язаним з діяльністю ЗП[15].

Частковий підхід з використанням окремих засобів протидії конкретним загрозам. До таких засобів належать антивірусні програми, засоби шифрування, спеціалізовані програми виявлення вторгнень, моніторингу та реєстрації подій. Його головною перевагою є висока ефективність у боротьбі з певною формою загрози, наприклад, певним типом вірусу або шпигунського програмного забезпечення.

Однак цей підхід має суттєві обмеження: фрагментація, локальність дій та порушення комплексного охоплення всієї інфраструктури. Він не гарантує цілісного захисту, особливо у випадках комбінованих атак або шкідливих дій внутрішніх користувачів.

Системний підхід вважається більш ефективним з точки зору довгострокового захисту інформаційних ресурсів. Він базується на комплексній інтеграції технічних, програмних та організаційних засобів у рамках єдиної політики безпеки підприємства. Такий підхід дозволяє створити безпечне середовище обробки інформації, здатне адаптуватися до змін загроз, враховуючи не лише зовнішні атаки, а й внутрішні ризики[16].

Однак реалізація системного підходу вимагає висококваліфікованого персоналу, значних фінансових витрат та правильного налаштування засобів захисту, хоча навіть незначна помилка може призвести до критичних наслідків.

На рисинку 2.1 зображено схему багаторівневого захисту, що демонструє взаємодію між різними шарами безпеки: апаратним, операційною системою, мережевим рівнем і прикладними рішеннями. Ця схема допоможе читачеві краще зрозуміти, як інтегровані заходи на різних рівнях забезпечують комплексний захист від атак[17].

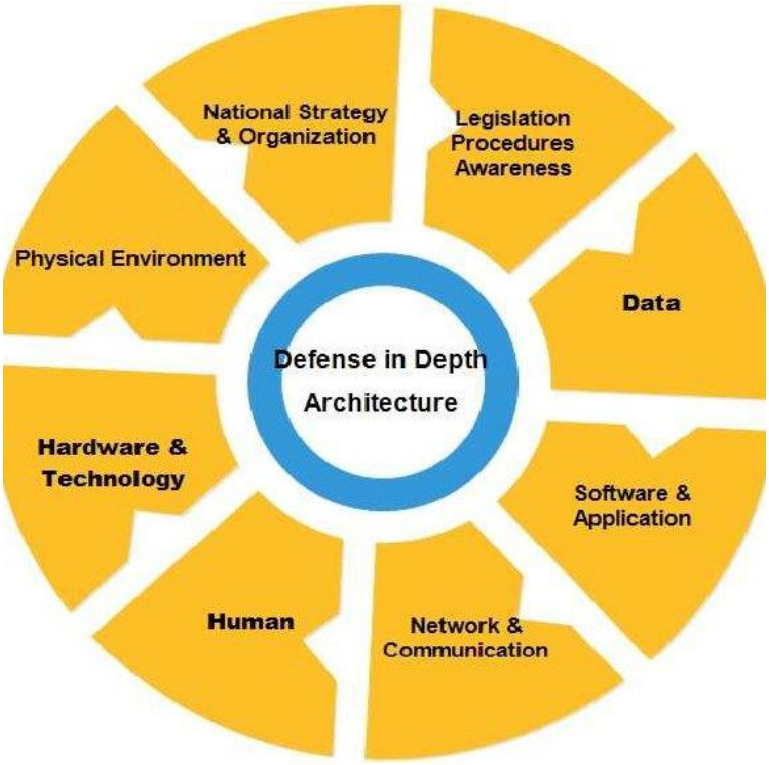


Рисунок 2.1 – Схема багаторівневого захисту

У рамках системного підходу ключову роль відіграє дотримання принципів інформаційної безпеки, що закладає ідейно-методологічну основу для побудови ефективної системи захисту від шкідливого програмного забезпечення.

Всі заходи повинні відповідати чинному законодавству щодо обробки та зберігання інформації, що є критично важливим при роботі з персональними або державними даними.

Водночас слід враховувати принцип невизначеності, оскільки поведінка зловмисника часто непередбачувана, тому система безпеки повинна бути достатньо гнучкою та адаптивною, здатною реагувати на нові загрози незалежно від їх джерела.

Оскільки жодна система не може гарантувати абсолютний рівень захисту, реалізується принцип неможливості створення ідеальної системи, що змушує шукати компроміс між рівнем безпеки та ресурсами для її досягнення[18].

На практиці це виражається в постійному аналізі, моніторингу вразливостей та своєчасному впровадженні змін до системи для мінімізації ризиків і втрат. Час реагування на інцидент має вирішальне значення – чим швидше виявлено та локалізовано загрозу, тим меншими будуть наслідки, що відповідає принципу безпечного часу.

Оскільки шляхи проникнення в інформаційну систему можуть бути дуже різноманітними, застосовується принцип «захисту всіх від усіх усі елементи інфраструктури повинні мати необхідний рівень захисту, а не лише окремі критичні вузли.

При цьому кожен співробітник несе особисту відповідальність за дотримання вимог безпеки в межах своїх повноважень, а доступ до інформації суворо обмежений даними, необхідними для виконання своїх обов'язків[19],[20].

Для скоординованої роботи технічного, адміністративного та інформаційного персоналу критично важливими є принципи взаємодії та співпраці, що формують канали довіри та координації дій.

Нарешті, принцип комплексності та індивідуальності передбачає, що система захисту повинна складатися з низки взаємопов'язаних заходів, кожен з яких враховує специфіку об'єкта, тип оброблюваної інформації та умови експлуатації.

Щоб ці принципи не залишалися суто теоретичними, вони втілюються в конкретних механізмах захисту від шкідливого програмного забезпечення.

Контроль доступу на рівні точок ініціації, проміжних вузлів та кінцевих точок забезпечує авторизацію користувачів та програм відповідно до їхніх прав.

Цифровий підпис з використанням асиметричного шифрування гарантує автентичність та цілісність даних, підтверджуючи джерело інформації та захищаючи її від підміни.

Щоб ускладнити аналіз мережевого трафіку, зловмисники використовують механізми додавання трафіку, які маскують розмір і частоту передачі даних. Для контролю змін в інформаційних блоках і потоках використовуються хеш-функції, MAC-алгоритми та динамічне шифрування зі змінними ключами для забезпечення цілісності даних.

Механізми керування маршрутизацією передбачають побудову безпечних каналів передачі, уникаючи небезпечних маршрутів, які можуть перебувати під контролем зловмисника.

Нарешті, механізми арбітражу через довірену третю сторону дозволяють перевірити відповідність переданих і отриманих даних заявленим властивостям, запобігаючи їх фальсифікації.

У сукупності ці принципи та механізми утворюють багаторівневу адаптивну систему захисту, здатну протидіяти сучасним загрозам шкідливого програмного забезпечення[21],[22].

Окрім вищезазначених механізмів, сучасна практика захисту інформаційних ресурсів активно використовує інтегровані рішення, такі як системи виявлення вторгнень, засоби моніторингу безпеки, а також хмарні платформи кіберзахисту, які аналізують поведінку користувачів та виявляють аномальні дії в режимі реального часу.

Таким чином, ефективні заходи протидії шкідливому програмному забезпеченню вимагають впровадження комплексного підходу, що поєднує принципи інформаційної безпеки з реальними механізмами їх технічної реалізації.

Саме ця синергія дозволяє знизити ризики витоку, знищення або несанкціонованого використання конфіденційних даних в умовах постійно зростаючих кіберзагроз[23].

З огляду на тенденції кіберзагроз, варто розглянути деякі конкретні типи шкідливого програмного забезпечення та відповідні методи протидії їм. Одним із найнебезпечніших типів ЗП є програмне забезпечення-вимагач, яке шифрує файли користувачів з метою отримання викупу.

Боротьба з ним передбачає не лише наявність надійних резервних копій, але й впровадження інструментів поведінкового аналізу, які можуть виявляти підозрілу активність ще до завершення шифрування даних. Наприклад, спроби масового відкриття та перезапису файлів або зміни розширень великої кількості документів можуть бути індикаторами атаки.

Ще одним поширеним типом загрози є шпигунське програмне забезпечення програмне забезпечення, призначене для збору конфіденційної інформації без відома користувача. Для його виявлення ефективні системи моніторингу поведінки та пісочниці, які дозволяють виявити, як програма поводиться в ізольованому середовищі, перш ніж дозволити їй повний доступ до системи.

Руткіти, які приховують свою присутність у системі та відкривають задні двері для зловмисників, також становлять значну загрозу для підприємств. Ефективна боротьба з ними вимагає глибокого аналізу на рівні ядра операційної системи, а також використання контролю цілісності апаратного забезпечення системних файлів та модуля довіреної платформи для захисту від несанкціонованих змін.

Ще однією серйозною проблемою є використання ботнетів, які об'єднують заражені пристрої в мережу під контролем зловмисника. Такі мережі можуть використовуватися для DDoS-атак або поширення інших шкідливих програм.

Протидія ботнетам вимагає постійного аналізу вхідного/вихідного трафіку, виявлення аномалій, характерних для діяльності управління та управління, та використання механізмів фільтрації.

У зв'язку з цим зростає значення інтелектуальних систем кіберзахисту на основі штучного інтелекту та машинного навчання. Вони здатні самостійно виявляти нові види шкідливої діяльності, навчатися на попередніх інцидентах та забезпечувати адаптивний захист. Такі системи ефективні у виявленні атак нульового дня, що використовують ще невідомі вразливості.

Ще одним перспективним напрямком є впровадження архітектури нульової довіри архітектури з нульовою довірою, яка передбачає відмову від припущення про безпеку внутрішнього середовища. У такій моделі кожна дія, запит чи транзакція потребує перевірки незалежно від джерела, що значно ускладнює дії зловмисника навіть у разі успішного проникнення в мережу.

Механізми сегментації мережі також відіграють значну роль. Поділ інфраструктури на ізольовані зони обмежує поширення шкідливого програмного забезпечення, зменшуючи масштаб можливих наслідків. Така сегментація може бути реалізована як на фізичному рівні, так і за допомогою віртуалізації та мікросегментації, зокрема в хмарних середовищах.

Особливу увагу слід приділити людському фактору, який часто виступає слабкою ланкою в системі безпеки. Згідно зі статистикою, більшість кібератак починаються з фішингових повідомлень, які змушують користувача завантажувати шкідливе програмне забезпечення або надавати доступ до ресурсів. Тому важливим компонентом захисту є регулярне навчання персоналу, симуляції атак та фішингові тести.

Для зменшення впливу людського фактору також впроваджуються системи багатофакторної автентифікації, контролю привілеїв та обмеження на геолокацію та час доступу, що дозволяє звузити вікно можливостей для зловмисника.

Не менш важливим є впровадження політик резервного копіювання з регулярною перевіркою можливості відновлення. Це дозволяє зберегти критично важливі дані навіть у разі успішної атаки програм-вимагачів. Резервні копії повинні зберігатися в ізольованому середовищі та захищатися від модифікації.

Використання цифрових сертифікатів та впровадження PKI також ефективні для захисту електронного документообігу, перевірки підписів та шифрування каналів зв'язку, зокрема в корпоративній електронній пошті та хмарних рішеннях.

2.2 Алгоритми та засоби забезпечення захисту конфіденційних даних підприємства

У корпоративних інформаційних системах для захисту конфіденційних даних традиційно використовуються як симетричні, так і асиметричні криптографічні алгоритми.

Симетричні шифри наприклад, AES з довжиною ключа 256 біт або 3DES забезпечують високу швидкість обробки великих обсягів інформації, тоді як асиметричні схеми RSA з мінімальним розміром ключа 2048 біт, ECC з еквівалентною безпекою при значно меншій довжині ключа також виконують автентифікацію без необхідності попереднього узгодження спільного секрету[24].На рисинку 2.2 наведено принцип роботи RSA.

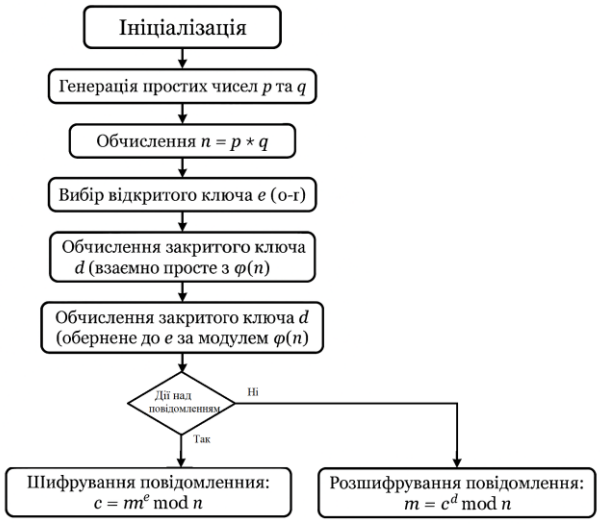


Рисунок 2.2 – Режим використання RSA

У реальних корпоративних реалізаціях часто потрібні гібридні протоколи, де низка симетричних ключів для шифрування сеансу обмінюється по асиметричному каналу, поєднуючи швидкість і високу стійкість до атак типу «людина посередині». Вибір конкретного алгоритму та тривалості включає результати аналізу моделі загроз, вимоги до продуктивності системи та нормативні стандарти зберігання даних, такі як GDPR або ISO 27001[25].

Центральним елементом захисту є централізоване рішення для управління ідентифікацією та доступом. Ці платформи підтримують моделі на основі ролей , атрибутів та заяв , що забезпечують гнучкість у визначенні політик доступу на основі відповідного контексту часу отримання даних, геолокації, змін стану пристрою.

У великих організаціях системи IAM інтегруються з Active Directory або LDAP для централізованого управління обліковими записами, делегування прав та детального ведення журналу подій входу[26].

Використання багатофакторної автентифікації MFA, що поєднує паролі, токени та біометрію, значно знижує ризик несанкціонованого доступу, навіть компрометації облікових даних. На рисинку 2.3 наведено принцип ідентифікації та доступу.

Рисунок 2.3 – Ідентифікація та доступ

Платформа захисту кінцевих точок EPP та система виявлення та реагування на кінцеві точки EDR все частіше впроваджуються для захисту кінцевих точок.

Початковий рівень складається з аналізу сигнатур відомих зразків шкідливого коду, тоді як розширена функціональність EDR використовує методи машинного навчання — кластеризацію, статистичний аналіз, нейронні мережі — для виявлення аномалій у поведінці процесів та активності мережі в режимі реального часу.

Раптове створення незвичайних процесів або підключень до незнайомих доменів автоматично запускає детальне розслідування, що дозволяє ізолювати заражений вузол до початку атаки[27].

На рівні мережевої інфраструктури брандмауери з відстеженням стану та глибока перевірка пакетів DPI аналізують потік даних, контролюють стан TCP/UDP-з'єднань та перевіряють вміст пакетів на наявність сигнатур загроз.

Завдяки складним правилам, які можна налаштувати за категоріями користувачів, служб або програм, адміністратори можуть створювати політики, які блокують не лише на основі відомих шаблонів, але й на основі поведінкових характеристик мережевого трафіку[28].

Це дозволяє ефективно захищатися від внутрішніх атак та виявляти тунелювання даних через нестандартні порти або протоколи. Для забезпечення цілісності та конфіденційності даних у стані спокою Data at Rest впроваджується повне шифрування диска, шифровані файлові системи LUKS, BitLocker та рішення на основі контейнеризації даних[29].

При цьому захист даних під час їх передачі забезпечується за допомогою протоколів Transport Layer Security, IPSec та SSH, які використовують сертифікати відкритого ключа для взаємної автентифікації клієнта та сервера. На рисинку 2.4 наведено взаємодію у TLS між сервером і клієнтом.

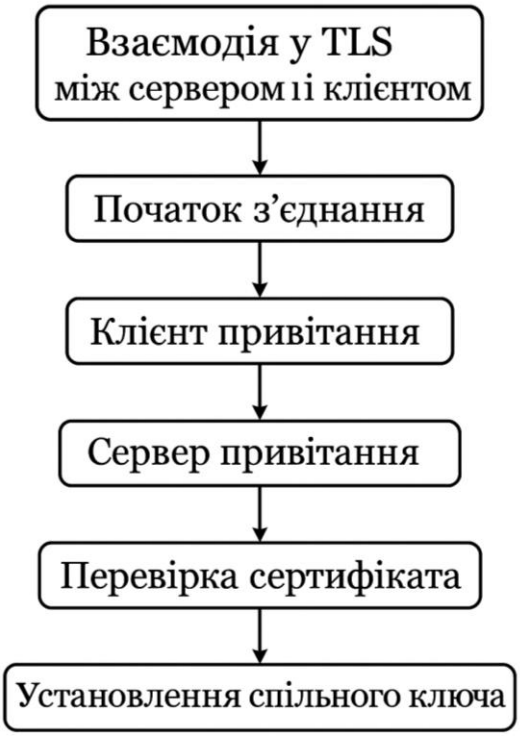


Рисунок 2.4 – Взаємодія у TLS між сервером і клієнтом.

Системи запобігання втраті даних аналізують вміст файлів та мережевий трафік за допомогою евристичних алгоритмів, регулярних виразів та моделей машинного навчання для виявлення спроб експорту конфіденційної інформації через електронну пошту, USB-пристрої або хмарні сервіси[30].

При спрацьовуванні сигнатур або шаблонів система DLP може блокувати операцію передачі, генерувати сповіщення для відповідальних осіб або ініціювати процес розгляду інциденту.

Для централізованого моніторингу безпеки впроваджено рішення Security Information and Event Management, які агрегують журнали з серверів, мережевих пристроїв, пристроїв кінцевих користувачів та програм.

Правила кореляції аналізують багатоканальні події, а алгоритми машинного навчання виявляють складні ланцюжки дій, типові для цільових атак. SIEM-системи здатні автоматично підвищувати пріоритет певних індикаторів компрометації та передавати інформацію на платформи для автоматизованого

реагування[31]. На рисинку 2.5 наведено алгоритм виявлення загроз системою моніторингу.

Рисунок 2.5 – Алгоритм виявлення загроз системою моніторингу

Інструменти Security Orchestration, Automation and Response SOAR доповнюють SIEM, виконуючи заздалегідь визначені сценарії реагування ізоляція заражених вузлів у мережі, блокування підозрілих IP-адрес на рівні брандмауера, створення інциденту в системі відстеження та автоматичне повідомлення відповідальних команд[32].Такий підхід скорочує час реагування на кібератаки та мінімізує людський фактор у початкових діях.

2.3 Проєктування архітектури системи захисту конфіденційних даних від шкідливого програмного забезпечення

Проектування систем захисту конфіденційних даних від шкідливого програмного забезпечення починається з глибокого аналізу інформаційних активів підприємства та оцінки ризиків, пов'язаних з кожним з них.

На цьому етапі вектора розробляється класифікація даних за рівнем конфіденційності від загальнодоступної інформації до критично важливих фінансових та особистих записів - а також існуючого потенціалу загроз та атак, властивих цим категоріям[33],[34].

Ключовим результатом такої оцінки є статистична карта загроз, яка дозволяє зрозуміти, які ресурси є пріоритетними для захисту та якими засобами найкраще їх прикрити.

На основі отриманих даних компанія формує базову концепцію глибокоешелонованого захисту , згідно з якою для інфраструктури кожного рівня створюється власний щит, взаємодія між якими гарантує резервування та стабільність у всіх системах[35].

Перший рівень захисту складається з фізичних засобів та мережевого зонування, які визначають поділ корпоративної мережі на окремі сегменти за ступенем довіри та критичності оброблюваних даних. Демілітаризована зона відокремлена від внутрішньої мережі, де зберігаються конфіденційні файли, та від зони архівування резервних копій.

Будуть підключені фізичні та віртуальні брандмауери, маршрутизатори з мережевою фільтрацією, а також периметральні брандмауери забезпечують основний бар'єр від несанкціонованого доступу[36].

У межах кожного сегмента висуваються власні вимоги до контролю: на периферії максимально обмежується вхідний трафік з Інтернету, у внутрішній мережі регулюється зв'язок між робочими станціями та серверами, а в зоні

резервного копіювання дозволяється передача даних на зовнішні накопичувачі або хмарні сховища без ініціації.

Така сегментація не тільки зменшує площу ураження у разі успішної атаки, але й значно ускладнює завдання для зловмисника, якому доводиться долати кілька незалежних бар'єрів[37],[38].

Другий рівень забезпечує комплексний контроль доступу на основі концепцій Role-Based Access Control та Attribute-Based Access Control, що дозволяють гнучко призначати та обмежувати права користувачів та сервісів. На рисинку 2.6 наведено алгоритм надання ролей користувачам RBAC.

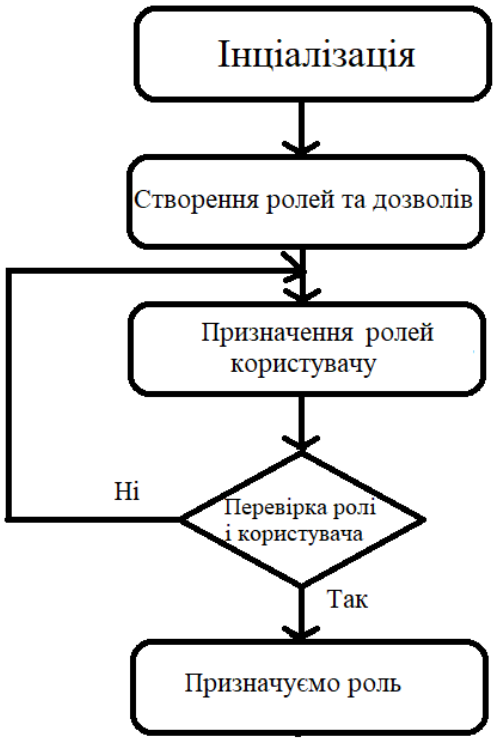


Рисунок 2.6 – Алгоритм надання ролей користувачам RBAC

Кожен співробітник отримує мінімальний набір привілеїв для виконання своєї роботи, а всі критичні операції супроводжуються багатофакторною автентифікацією, яка оплачується паролем, смарт-карткою або токеном, та біометричними факторами.

Журнали автентифікації та авторизації збираються та співвідносяться в режимі реального часу, що дозволяє однозначно виявляти спроби підбору облікових даних або використання скомпрометованих облікових записів.

Водночас для висококонфіденційних даних вводиться додатковий рівень перевірки – схвалення через незалежний канал, що забезпечує участь двох або більше уповноважених осіб[39].

Третій рівень захисту реалізує активне виявлення та блокування шкідливого програмного забезпечення. Захист кінцевих точок, встановлений на робочих станціях та серверах, відповідає традиційному скануванню сигнатур з модулями поведінкового аналізу та евристики.

Особлива увага приділяється виявленню атак Zero-Day та поліморфних вірусів: використання машинного навчання дозволяє аналізувати поведінкові процеси в режимі реального часу та фільтрувати підозрілі дії до завантаження коду в пам'ять.

Поряд з цим, на рівні мережевого шлюзу використовуються механізми IPS, які перевіряють кожен пакет на відповідність відомим сигнатурам, а у разі аномалії можуть не тільки сповіщати адміністраторів, але й автоматично розривати з'єднання з виявленим шкідливим вузлом.

Такі суворі правила реагування значно пришвидшують реагування на інциденти та гарантують, що атака не поширеться мережею за межі початкового сегмента. Четвертий рівень відбувається в централізованому моніторингу подій за допомогою системи SIEM[40].

Усі файли журналів від брандмауерів, IDS/IPS, антивірусних движків, контролю доступу, баз даних та серверів додатків належать до єдиної платформи, а спеціальні правила кореляції виявляють складні інциденти: багатовекторні атаки, поширення шкідливого програмного забезпечення на окремі сегменти та заражені ланцюги користувачів. SIEM-аналітика використовує набори індикаторів компрометації та поведінкових профілів, що дозволяють виявляти атаки на ранній стадії[41].

Для підвищення точності системи також використовуються механізми UEBA, які аналізують відхилення у звичайній поведінці користувачів та процесів.

У разі критичних сценаріїв SIEM автоматично запускає процедури реагування: ізолює вразливий сегмент, зупиняє перевірку сеансу, повідомляє команду реагування на інциденти через кілька каналів — електронну пошту, SMS, корпоративний месенджер.

П'ятий рівень захисту це заходи щодо забезпечення цілісності та конфіденційності даних: використання криптографічних протоколів SSL/TLS для всіх мережевих підключень, шифрування файлу та бази даних за допомогою сучасних алгоритмів AES-256, а також цифрові підписи документів та записів журналів. Окремі критичні таблиці в базі даних можуть зберігатися у вигляді зашифрованих стовпців , що унеможливлює витік інформації навіть у разі фізичної компрометації носія.

Ключі шифрування зберігаються в апаратному модулі безпеки (HSM), доступ до якого також суворо контролюється журналом. Завдяки такому підходу зловмисник втрачає можливість не лише красти, але й змінювати дані непомітно для системи моніторингу[42].

Шостий рівень охоплює архітектуру безпеки програмного забезпечення: розробка здійснюється за принципами Secure by Design, що включає статичний та динамічний аналіз коду SAST/DAST, регулярні пентести та аудит залежностей на наявність вразливостей .

Всі API доступні лише через захищені мікросервіси з автентифікацією на основі OAuth2/OpenID Connect та обмеженням швидкості, що запобігає DDoS-атакам. Використання сучасних фреймворків із вбудованим захистом від SQL-ін'єкцій, XSS та CSRF значно зменшує кількість точок входу для шкідливого коду[43].

Архітектура забезпечує засоби для аварійного відновлення та забезпечення безперервності бізнесу: стратегія резервного копіювання включає регулярні

знімки даних, реплікацію на географічно розподілені сайти, автоматизоване тестування відновлення DR drills.

Оновлення та виправлення програмного забезпечення розгортаються за допомогою систем управління конфігурацією Ansible, Puppet, Chef у тестовому середовищі перед впровадженням у виробництво, що запобігає появі критичних помилок[44],[45].

Відповідність побудованої архітектури міжнародним стандартам ISO/IEC 27001, NIST CSF та COBIT гарантує, що всі процеси задокументовані, а відповідальні особи мають чіткі інструкції та політики щодо підтримки та вдосконалення системи.

2.4 Висновки

У цьому розділі визначено основні вимоги до створення системи захисту конфіденційних даних від шкідливого програмного забезпечення. На основі аналізу сучасних загроз інформаційній безпеці було виявлено, що найбільшу небезпеку становлять цілеспрямовані атаки на персональні та службові бази даних користувачів, що зберігаються в інформаційних системах організацій.

Враховуючи динамічний розвиток шкідливого програмного забезпечення, традиційні засоби захисту необхідно доповнювати сучасними механізмами контролю та реагування. Завдання полягає в розробці програмно-апаратного комплексу, який забезпечує виявлення, блокування та запобігання несанкціонованому доступу до конфіденційної інформації.

Система повинна включати процеси моніторингу, перевірку цілісності файлів за допомогою хеш-функцій, ведення журналу подій, шифрування критичних даних, а також можливість оперативного повідомлення відповідальних осіб. Крім того, доцільно використовувати підходи, засновані на аналізі поведінки, а в перспективі методи штучного інтелекту.

Формування таких вимог до пошукової системи, як її архітектура: багаторівнева, модульна та легко масштабована з урахуванням майбутніх потреб організації. Особлива увага приділялася зручності використання системи для кінцевих користувачів без зниження продуктивності пристроїв.

3 РОЗРОБКА СИСТЕМИ ЗАХИСТУ КОНФІДЕНЦІЙНИХ ДАНИХ ВІД ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Проєктування алгоритму роботи системи захисту конфіденційних даних від шкідливого програмного забезпечення

Алгоритм системи захисту конфіденційних даних від шкідливого програмного забезпечення розроблено з урахуванням вимог безпеки та принципів сучасної кібербезпеки.

Система забезпечує багаторівневу перевірку запущених процесів, цілісності файлів, ведення журналу подій, шифрування важливих даних та повідомлення відповідальних осіб про виявлені загрози. Алгоритм базується на блок-схемі, яка реалізує послідовну перевірку на основі білих та чорних списків, ведення журналу, шифрування та моніторингу.

На початковому етапі система формує білий список – список довірених процесів або файлів, яким дозволено запускатися без додаткової перевірки. Одночасно формуються чорні списки, що містять відомі шкідливі або небажані об’єкти. Ці списки є основними правилами фільтрації на наступних етапах.

Далі система ініціалізує моніторинг процесів за допомогою спеціального модуля ProcessMonitor, який використовує функціональність System.Diagnostics.Process у середовищі .NET. Він отримує інформацію про активні процеси, їх імена, ідентифікатори та шляхи до виконуваних файлів.

Кожен щойно запущений процес перевіряється: якщо він включений до білого списку, йому дозволено продовжувати роботу, якщо він включений до чорного списку, він блокується та реєструється як потенційна загроза.

Усі події записуються в журнал подій системою LoggerService. Це дозволяє створювати прозору історію змін та інцидентів, яку згодом можна використовувати для аналізу або розслідування порушень безпеки.

Під час виправлення змін у процесах або доступі до файлів виконується хеш-перевірка за допомогою HashChecker, який використовує криптографічні алгоритми SHA-256 для виявлення змін або модифікацій даних. Якщо хеш-функція вказує на невідповідність, файл вважається зміненим або потенційно шкідливим.

Якщо виявлено підозрілу активність або проведено аналіз процесу, система визначає подальші дії.

Якщо ввімкнено офлайн-режим, система автоматично ініціює шифрування критичних файлів через компонент CryptoService, який реалізує алгоритм AES. Шифрування унеможливлює подальший несанкціонований доступ до конфіденційної інформації. Якщо офлайн-режим вимкнено, система генерує повідомлення адміністратору.

Це робиться за допомогою AlertService, який може надсилати повідомлення через Telegram за допомогою бібліотеки Telegram.Bot або електронною поштою за допомогою MailKit і MimeKit.На рисинку 3.1 наведено алгоритм роботи системи захисту конфіденційних даних від шкідливого програмного забезпечення.

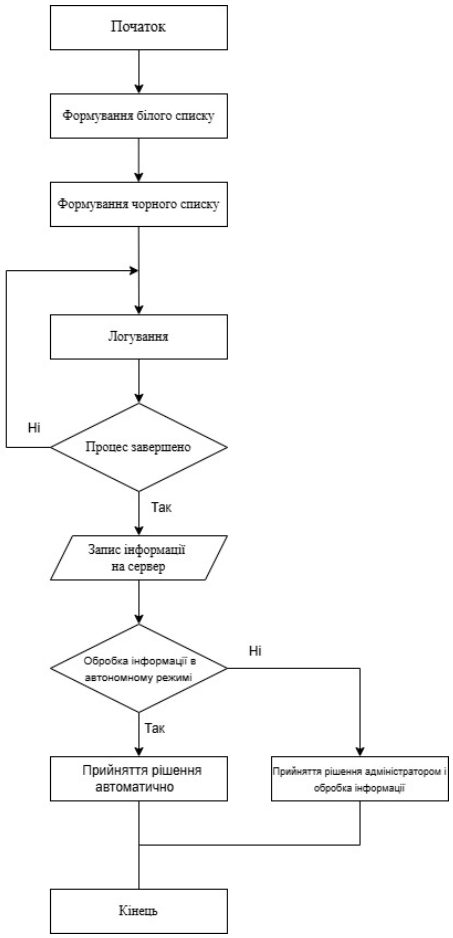


Рисунок 3.1 – Алгоритм роботи системи захисту конфіденційних даних від шкідливого програмного забезпечення

Система захисту складається з таких основних компонентів: білі та чорні списки використовуються для фільтрації дозволених та заборонених процесів. Білий список дозволяє запускати лише довірені програми, тоді як чорний список блокує відомі загрози. Порівняння створення за іменами процесів або хешами файлів.

Вихідні дані моніторингу процесів ProcessMonitor проводять регулярну перевірку активних процесів у системі, порівнюють їх зі списками та ініціюють подальші дії, якщо виявлено підозрілі процеси.

Перевірка хешу забезпечує контроль цілісності файлів на основі SHA-256, що дозволяє виявляти модифіковані або шкідливі об'єкти, які могли б пройти звичайну перевірку за іменем.

Ведення журналу записує всі події у файли журналів або передає їх на віддалений сервер для подальшого аудиту. Це дозволяє відстежувати повний стан користувача та системи.

Шифрування реалізує захист критично важливих даних за допомогою алгоритму AES, забезпечуючи їх недоступність у разі загрози. AlertService надсилає сповіщення через Telegram або електронну пошту з детальним описом загрози, що дозволяє швидко реагувати на інциденти безпеки.

Розробка реалізована в середовищі Visual Studio з використанням платформи .NET. Кожен функціональний блок реалізовано як окремий клас у просторі імен SecuritySystem, що забезпечує модульність та гнучкість архітектури. Усі системні налаштування зберігаються у файлі appsettings.json.

Для їх обробки використовується бібліотека Microsoft.Extensions.Configuration, яка дозволяє легко завантажувати параметри з JSON-файлу.

Отримана система забезпечує багаторівневий та комплексний захист конфіденційних даних підприємства - від виявлення аномалій до автоматичного реагування.

Його використання дозволяє ефективно протидіяти загрозам шкідливого програмного забезпечення, зберігаючи при цьому цілісність, конфіденційність та доступність критично важливої інформації.

Розроблена система захисту конфіденційних даних включає три взаємопов'язані процеси: шифрування файлів, перевірку хеш-суми та реєстрацію подій.

Кожен із цих механізмів відповідає за окремий аспект безпеки: шифрування гарантує конфіденційність, перевірка хеш-суми забезпечує цілісність, а реєстрація надає можливість проводити аудит та відтворювати історію дій у разі виникнення інцидентів. Шифрування виконується за допомогою симетричного алгоритму AES з довжиною ключа 256 бітів у режимі CBC.

Першим кроком є вилучення бітового ключа довжиною до 32 байт з файлу конфігурації системи. Якщо в конфігурації вказано значення менше 256 бітів, воно розширюється з нульовим доповненням до необхідного розміру, забезпечуючи відповідність стандарту AES-256.

Далі для кожної операції шифрування генерується вектор ініціалізації (IV) розміром 16 байт. У тестовій реалізації IV складається з нульових байтів для відтворюваності та зручності налагодження, але в робочій версії рекомендується використовувати криптографічно стійкий генератор випадкових чисел.

Наступним кроком є ініціалізація об'єкта AES за допомогою стандартних бібліотек .NET, яким передаються ключ та IV; встановлюється режим CBC та схема доповнення PKCS7.

Після цього відкриваються два потоки: вхідний для зчитування вихідного файлу та вихідний для запису зашифрованого контенту. За допомогою методу CreateEncryptor() створюється CryptoStream, який зчитує дані з inFile, шифрує їх блочними операціями та записує їх у outFile.Після обробки всіх блоків виконується, потоки закриваються, і результатом є файл із зашифрованим контентом, доступ до якого неможливий без ключа.

Така послідовність дій забезпечує високий рівень конфіденційності даних навіть у разі фізичного видалення носія або несанкціонованого копіювання.На рисинку 3.2 наведено алгоритм шифрування файла.

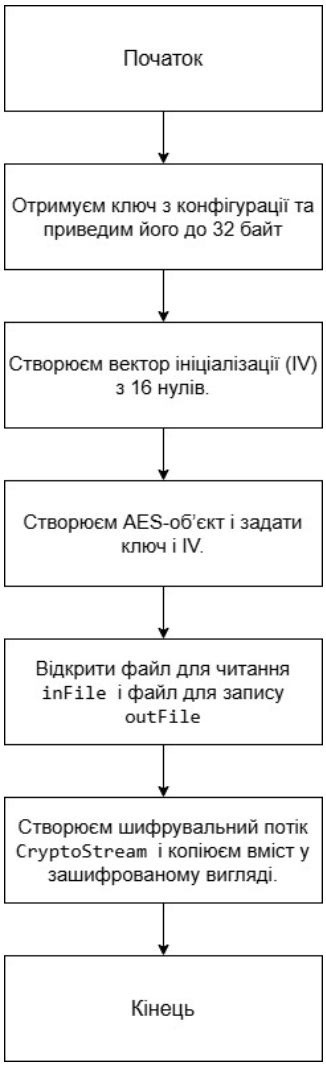


Рисунок 3.2 – Алгоритм шифрування файла

Цілісність файлів контролюється шляхом обчислення хеш-суми SHA-256 та порівняння її зі списком «дозволених» значень, що містяться у файлі конфігурації.Відстеження змін у режимі реального часу організовано через компонент, який контролює задану папку за вказаним розширенням та викликає обробник подій, коли файл створюється або змінюється.

Спочатку файл перевіряється на наявність у системі якщо об'єкт відсутній, алгоритм завершиться без подальших дій. Якщо файл існує, його вміст послідовно зчитується та обробляється класом SHA256Managed, який повертає 32-байтовий масив.Отриманий масив перетворюється на шістнадцятковий рядок та

порівнюється з кожним хеш-значенням зі списку, завантаженого з конфігурації JSON або XML.

Якщо хеш збігається, файл вважається цілим, і система ігнорує подію. Якщо хеші не збігаються, файл вважається підробленим — ця інформація передається до модуля реєстрації та сповіщень для подальшої реакції адміністратора або автоматизованої системи реагування.На рисинку 3.3 наведено алгоритм перевірки хеш-суми.

Рисунок 3.3 – Алгоритм перевірки хеш-суми

Ведення журналу є критично важливим для аудиту дій системи безпеки, оскільки воно фіксує успішні та невдалі операції шифрування, результати перевірки цілісності та внутрішні помилки.

Кожен виклик генерує рядок у форматі ISO з текстовим описом події. Перед записом рядка у файл вводиться критична секція за допомогою примітиву блокування, який запобігає одночасному доступу з різних потоків або процесів та запобігає пошкодженню файлу. Далі перевіряється умова запису якщо ведення журналу не потрібне, керування повертається без запису.

В іншому випадку файл журналу відкривається в режимі додавання та додається згенерований рядок, після чого потік закривається. Завдяки цьому механізму система зберігає структурований та хронологічний журнал подій, що дозволяє відновити хід операцій у будь-який час та виявити причину інциденту.На рисинку 3.4 наведено алгоритм запису лога у файл.

Рисунок 3.4 – Алгоритм запису лога у файл

Загалом, інтеграція описаних алгоритмів шифрування, перевірки цілісності та логування гарантує комплексний захист конфіденційних даних: шифрування для запобігання несанкціонованому доступу, хеш-контроль для оперативного виявлення підміни або пошкодження файлів та логування для забезпечення прозорості та відтворення історії подій.

3.2 Реалізація системи захисту конфіденційних даних від шкідливого програмного забезпечення

В основі системи захисту конфіденційних даних лежить головний модуль SecurityWorker, який координує роботу інших сервісів.Система складається з окремих компонентів, кожен з яких виконує свою роль :

– сервіс оповіщень AlertService генерує та надсилає сповіщення про загрози;

– криптосервіс CryptoService відповідає за шифрування та розшифровку файлів;

– модуль перевірки хешів HashChecker контролює цілісність даних за допомогою алгоритму SHA-256;

– сервіс ведення журналу LoggerService записує інформацію про події та загрози в журнал;

– компонент моніторингу процесів ProcessMonitor контролює запущені процеси на наявність підозрілої активності.

Головний модуль SecurityWorker реалізує фонову обробку даних, координує запуск перевірок та обробку виявлених загроз.

Конфігурацію системи зчитує бібліотека Microsoft.Extensions.Configuration, яка забезпечує гнучке зберігання параметрів. Кожен сервіс реалізовано окремим класом.

AlertService це сервіс сповіщення про загрози, який асинхронно надсилає повідомлення адміністратору за допомогою методу SendTextMessageAsync або подібного інтерфейсу.

Для цього використовується клієнт TelegramBotClient, який, наприклад, може використовувати виклик типу bot.SendMessage для надсилання повідомлень.

Це дозволяє швидко інформувати про підозрілі події, такі як зміни в захищеному файлі або запуск невідомого процесу реалізація наведена на рисинку 3.5 лістинг програмної реалізація відправки логів.

```
public async Task SendAsync(string subject, string body)
{
    //// Email
    var message = new MimeMessage();
    message.From.Add(MailboxAddress.Parse(_cfg["Email:From"]));
    message.To.Add(MailboxAddress.Parse(_cfg["Email:To"]));
    message.Subject = subject;
    message.Body = new TextPart("plain") { Text = body };

    using var smtp = new SmtpClient();
    await smtp.ConnectAsync(_cfg["Email:SmtpHost"], int.Parse(_cfg["Email:SmtpPort"]), true);
    await smtp.AuthenticateAsync(_cfg["Email:Username"], _cfg["Email:Password"]);
    await smtp.SendAsync(message);
    await smtp.DisconnectAsync(true);

    // Telegram
    var chatId = new ChatId(long.Parse(_cfg["Telegram:ChatId"]));
    await _bot.SendTextMessageAsync(chatId, $"{subject}: {body}");
}
```

Рисунок 3.5 – Лістинг програмної реалізація відправки логів

CryptoService це криптографічний сервіс для захисту файлів, який забезпечує симетричне шифрування метод EncryptFile та дешифрування метод DecryptFile конфіденційних даних. Реалізація побудована на стандартних криптографічних бібліотеках .NET, зокрема на класах Aes або AesCryptoServiceProvider, які реалізують алгоритм AES.

Після виявлення загрози метод EncryptFile шифрує файл за допомогою випадкового ключа та вектора ініціалізації. Цей же сервіс також може дешифрувати дані за допомогою збережених ключів реалізація наведена на рисинку 3.6 лістинг програмної реалізація шифрування і дешифрування.

```
public void EncryptFile(string inFile, string outFile)
{
    var key = Encoding.UTF8.GetBytes(_cfg["Crypto:Key"].PadRight(32).Substring(0, 32));
    var iv = new byte[16];
    using var aes = Aes.Create();
    aes.Key = key; aes.IV = iv;
    using var fsIn = File.OpenRead(inFile);
    using var fsOut = File.OpenWrite(outFile);
    using var cs = new CryptoStream(fsOut, aes.CreateEncryptor(), CryptoStreamMode.Write);
    fsIn.CopyTo(cs);
}

0 references
public void DecryptFile(string inFile, string outFile)
{
    var key = Encoding.UTF8.GetBytes(_cfg["Crypto:Key"].PadRight(32).Substring(0, 32));
    var iv = new byte[16];
    using var aes = Aes.Create();
    aes.Key = key; aes.IV = iv;
    using var fsIn = File.OpenRead(inFile);
    using var fsOut = File.OpenWrite(outFile);
    using var cs = new CryptoStream(fsIn, aes.CreateDecryptor(), CryptoStreamMode.Read);
    cs.CopyTo(fsOut);
}
```

Рисунок 3.6 – Лістинг програмної реалізація шифрування і дешифрування

HashChecker це модуль контролю цілісності даних, який періодично обчислює хеш-суми SHA-256 файлів та порівнює їх з еталонними значеннями. Клас SHA256 у .NET дозволяє обчислювати хеш SHA-256 для вхідних даних.

Метод CheckHash зчитує файли, обчислює байтовий хеш, а потім порівнює результати з еталонними значеннями. У разі невідповідності модуль сигналізує про потенційну зміну даних та надсилає повідомлення до головного модуля реалізація наведена на рисинку 3.7 лістинг програмної реалізація перевірки хеш-суми SHA-256.

```
private async Task CheckHash(string path)
{
    if (!File.Exists(path)) return;
    using var sha = SHA256.Create();
    using var fs = File.OpenRead(path);
    var hash = BitConverter.ToString(sha.ComputeHash(fs)).Replace("-", string.Empty);
    var known = _cfg.GetSection("KnownHashes").Get<string[]>();
    if (!known.Contains(hash, StringComparer.OrdinalIgnoreCase))
        await OnMismatch?.Invoke($"Hash mismatch for {path}");
}
```

Рисунок 3.7 – Лістинг програмної реалізація перевірки хеш-суми SHA-256

LoggerService це служба реєстрації подій безпеки, яка реалізує метод LogAlert для запису інформації про виявлені загрози або підозрілі події. Записи

можуть зберігатися у файлі журналу, базі даних або системі реєстрації через інтерфейс ILogger. Централізоване ведення журналу дозволяє вести історію подій та полегшує аналіз інцидентів реалізація наведена на рисинку 3.8 лістинг програмної реалізація служби реєстрації подій безпеки.

```
public void LogAlert(string msg)
{
    var line = $"[{DateTime.Now}] ALERT: {msg}{Environment.NewLine}";

    try
    {
        lock (_logLock)
        {
            File.AppendAllText(_logPath, line);
        }
    }
    catch (IOException ex)
    {
        Console.Error.WriteLine("Не вдалося записати лог: " + ex.Message);
    }
}
```

Рисунок 3.8 – Лістинг програмної реалізація служби реєстрації подій безпеки

ProcessMonitor це модуль моніторингу процесів, який відстежує активність системних процесів. Його методи Start та CheckProcesses реалізують перерахування та аналіз списку запущених процесів реалізація наведена на рисинку 3.9 лістинг програмної реалізація моніторингу процесів.

```
private void CheckProcesses(object state)
{
    foreach (var proc in System.Diagnostics.Process.GetProcesses())
    {
        try
        {
            var name = Path.GetFileName(proc.MainModule.FileName);

            if (!_allowed.Contains(name, StringComparer.OrdinalIgnoreCase))
            {
                proc.Kill();
                OnThreatDetected?.Invoke($"Blocked process: {name}");
            }
        }
        catch (AccessViolationException ex)
        {
            Console.WriteLine($"Access error with process: {proc.ProcessName} ({proc.Id}) - {ex.Message}");
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Error while processing {proc.ProcessName} ({proc.Id}): {ex.Message}");
        }
    }
}
```

Рисунок 3.9 – Лістинг програмної реалізація моніторингу процесів

FileSystemWatcher також можна використовувати для моніторингу змін у файловій системі. Якщо виявлено підозрілі процеси, ProcessMonitor генерує подію або викликає відповідний метод у SecurityWorker для початку обробки загроз.

SecurityWorker це основна фонова служба, реалізована як клас, похідний від BackgroundService, яка запускається під час запуску програми. Метод ExecuteAsync періодично ініціює перевірки, очікуючи сигналів загрози.

Якщо такий сигнал отримано, виконується метод HandleThreatAsync, який послідовно викликає EncryptFile, LogAlert та SendAsync, забезпечуючи шифрування даних, ведення журналу подій та сповіщення адміністратора реалізація наведена на рисинку 3.10 лістинг програмної реалізація ведення журналу подій та сповіщення адміністратора.

```
protected override async Task ExecuteAsync(CancellationToken stoppingToken)
{
    _pm.Start();
    _hc.Start();
    _pm.OnThreatDetected += async (msg) => await HandleThreatAsync(msg);
    _hc.OnMismatch += async (msg) => await HandleThreatAsync(msg);

    await Task.Delay(Timeout.Infinite, stoppingToken);
}

2 references
private async Task HandleThreatAsync(string message)
{
    _lg.LogAlert(message);
    await _al.SendAsync("Security Alert", message);
}
```

Рисунок 3.10 – Лістинг програмної реалізація ведення журналу подій та сповіщення адміністратора

Під час ініціалізації SecurityWorker завантажує системні параметри з файлів конфігурації через Microsoft.Extensions.Configuration та встановлює інтервали опитування.На рисинку 3.11 наведено структуру програмного забезпечення системи захисту конфіденційних даних.

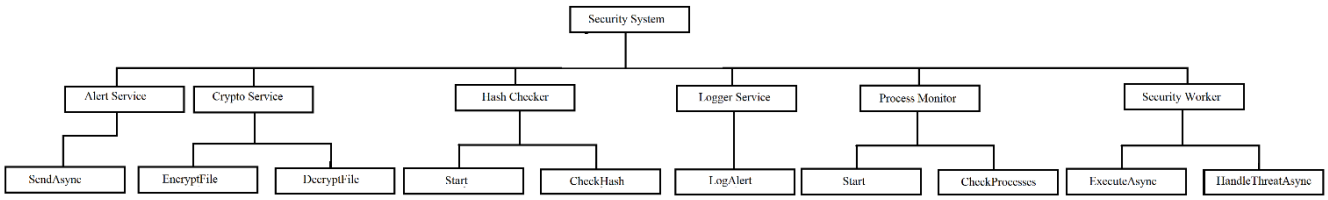


Рисунок 3.11 – Структура програмного забезпечення системи захисту конфіденційних даних

Під час роботи системи компоненти взаємодіють відповідно до внутрішньої логіки. При запуску SecurityWorker запускаються служби моніторингу та контролю хешу.

Якщо ProcessMonitor виявляє новий процес, він передає інформацію SecurityWorker, який оцінює його на наявність загрози. Аналогічно, HashChecker повідомляє про зміну хешу.

У цьому випадку SecurityWorker викликає HandleThreatAsync, де виконується послідовна відповідь: шифрування файлу EncryptFile, ведення журналу LogAlert та надсилання повідомлення SendAsync. Таким чином, система забезпечує постійний моніторинг стану даних та оперативно реагує на загрози, зберігаючи конфіденційність інформації.

Модуль ProcessMonitor ініціалізує таймер або цикл перевірки, який регулярно викликає метод CheckProcesses(). Цей метод використовує клас Process з простору імен System.Diagnostics для отримання списку запущених процесів:

Перевірка виконується шляхом зіставлення відомих шкідливих процесів. Ви також можете використовувати FileSystemWatcher для моніторингу змін у каталогах, таких як Program Files. Коли виявляється шкідливий процес, ProcessMonitor генерує подію або викликає метод у SecurityWorker.

Результати порівнюються з еталонними хешами. У разі розбіжності HashChecker надсилає повідомлення SecurityWorker.

Сервіс CryptoService реалізує методи EncryptFile та DecryptFile, використовуючи AesCryptoServiceProvider або Aes.Create() для генерації ключа та

IV. Після генерації файл шифрується або дешифрується відповідно до алгоритму AES. Ці дії виконуються автоматично у відповідь на виявлену загрозу.

Таким чином, усі компоненти тісно співпрацюють, забезпечуючи багаторівневу систему захисту конфіденційних даних з функціями моніторингу, виявлення та реагування. На рисинку 3.12 наведено схему підприємства.

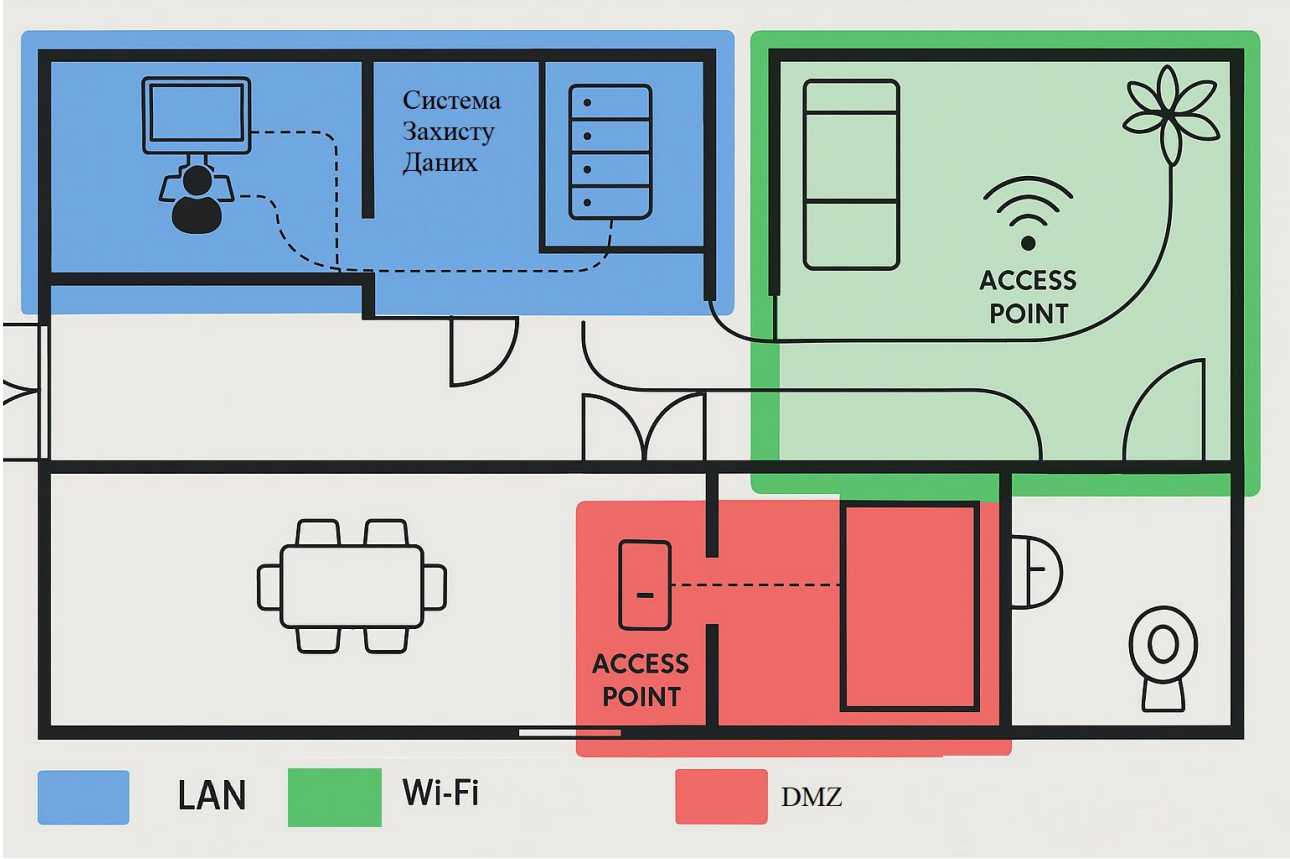


Рисунок 3.12 – Схема підприємства

Приміщення умовно поділено на кілька функціональних зон: кабінет адміністратора, серверна, зона рецепції клієнтів, кімната відпочинку, кухня та санвузол. Для зручності сприйняття логічні сегменти комп'ютерної мережі позначені кольоровими контурами.

Синьою рамкою позначено локальну мережу, яка об'єднує всі комп'ютери та сервери в єдине захищене середовище.

Червоною рамкою позначено демілітаризовану зону ізольований сегмент, призначений для розміщення серверів, до яких можна отримати доступ із зовнішньої мережі.

Зелена зона на схемі відображає зону покриття бездротової мережі Wi-Fi, яка охоплює зону очікування клієнтів, кімнату відпочинку та частину інших приміщень підприємства.

У верхній лівій частині схеми розташований комп'ютер адміністратора, який є основним елементом керування всією системою безпеки. Саме з цього комп'ютера адміністратор запускає фонову службу SecurityWorker, контролює поточний стан системи, обробляє сповіщення про загрози та має доступ до журналів подій.

Цей комп'ютер підключений до внутрішньої ЛОМ та оснащений програмними засобами керування, що запобігають несанкціонованому доступу. Поруч із робочим місцем адміністратора розташована серверна кімната, в якій розміщено сервер, що містить базу даних, системні журнали, резервні копії та іншу критично важливу інформацію.

Сервер фізично ізольований і має обмежений доступ лише для уповноважених осіб. Логічно він може належати до DMZ, якщо виконує функції, доступні ззовні, наприклад, надає API або розміщує веб-сервіси.

Система захисту даних побудована за модульною архітектурою. Візуально це відображається на схемі пунктирними лініями, які позначають зв'язки між її компонентами. Хоча більша частина логіки реалізована програмно, апаратна інфраструктура забезпечує захищене середовище для виконання її функцій.

Основні модулі CryptoService, HashChecker, LoggerService, ProcessMonitor, AlertService взаємодіють один з одним через внутрішню захищену мережу, використовуючи зашифровані канали зв'язку, що унеможливлює перехоплення або зміну даних під час процесу обміну.

Зона рецепції клієнтів обладнана бездротовим доступом до мережі, що дозволяє відвідувачам або співробітникам підключатися за допомогою мобільних пристроїв.

Цей доступ реалізовано через окрему гостьову мережу Wi-Fi, яка ізольована від основної локальної мережі за допомогою механізмів VLAN. Таким чином, навіть у разі компрометації гостьового сегмента, критично важливі внутрішні ресурси залишаються захищеними.

У вітальні та на кухні також є покриття Wi-Fi, що забезпечує зручність для персоналу. Однак усі підключення в цих зонах проходять через брандмауери та системи брандмауерів, які фільтрують трафік з неперевірених джерел. У разі порушення політик безпеки або виявлення підозрілої активності спрацьовують відповідні модулі програмного забезпечення для моніторингу.

Загалом, вся мережа підприємства побудована на принципах багаторівневого захисту. Кожна функціональна зона має певний рівень доступу до мережевих ресурсів.

Серверна зона максимально ізольована, тоді як бездротові мережі використовуються виключно для некритичних завдань. Всі пристрої підключені через керовані мережеві комутатори з підтримкою VLAN, що дозволяє розділяти трафік на гостьовий та внутрішній, забезпечуючи додатковий рівень безпеки.

Інформаційні потоки в системі забезпечують безперервний моніторинг стану програмного та апаратного середовища. Служба ProcessMonitor постійно сканує активні процеси в операційній системі, тоді як FileSystemWatcher відстежує зміни у файловій системі.

У разі виявлення загроз або аномальної поведінки ініціюється відповідна команда для шифрування вразливих файлів через CryptoService, реєстрації інциденту в журналі через LoggerService та негайного повідомлення адміністратора через AlertService.

Такий рівень взаємодії між компонентами дозволяє швидко реагувати на інциденти та мінімізувати ризики втрати або витоку конфіденційної

інформації.Всі ці логічні зв'язки між елементами інформаційної системи належним чином відображаються на схемі підприємства.

3.3 Налаштування системи захисту конфіденційних даних від шкідливого програмного забезпечення

Для забезпечення ефективної роботи системи захисту конфіденційних даних від шкідливих програм реалізовано механізм конфігурації, який дозволяє гнучко налаштовувати основні параметри роботи програми за допомогою конфігураційного файлу. На рисунку 3.13 показано загальну схему конфігурації системи, яка включає основні елементи конфігурації:

- Allowed Executables список файлів, які дозволено виконувати,
- Monitor Folder каталог для моніторингу,
- Crypto налаштування шифрування і розшифрування файлів,
- LogFilePath шлях до файлу журналу подій,
- Telegram параметри надсилання повідомлень у Telegram-бот.

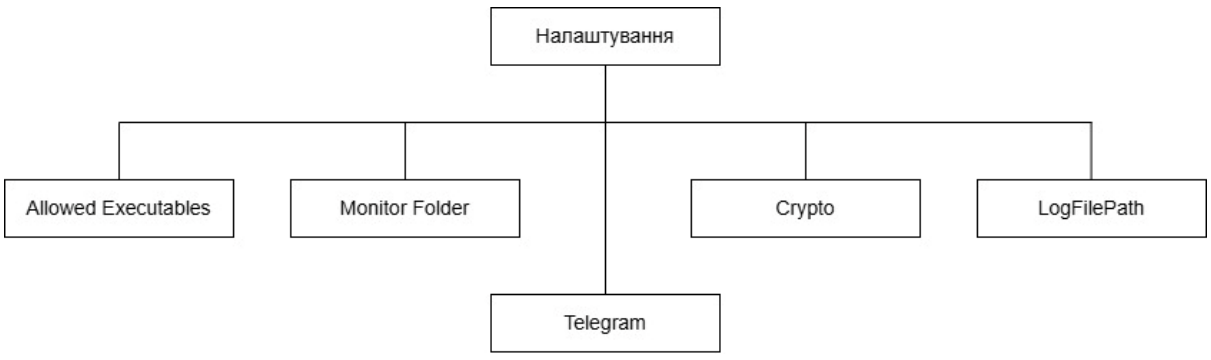


Рисунок 3.13 – Схема налаштування

Праметри задаються в конфігураційному файлі appsettings.json, який є стандартним способом зберігання конфігурації для додатків на платформі .NET.На рисинку 3.14 лістинг програмної реалізація appsettings.json.

Рисунок 3.14 – Лістинг програмної реалізація appsettings.json

AllowedExecutables список програм, яким дозволено запускатися з теки, що моніториться. Всі інші виконувані файли блокуються або попереджуються. Це запобігає несанкціонованому запуску невідомих або потенційно шкідливих програм.

MonitorFolder визначає шлях до папки, вміст якої постійно контролюється. Система реагує на додавання, зміну або видалення файлів у цій теці, ініціюючи відповідні дії, такі як шифрування, ведення журналу або сповіщення.

Crypto містить параметри шифрування, які використовуються для забезпечення конфіденційності даних. Зокрема, ключ і вектор ініціалізації (IV) закодовані у форматі Base64. Це дозволяє шифрувати і розшифровувати файли, які з'являються в папці моніторингу, за допомогою симетричного алгоритму.

LogFilePath шлях до текстового файлу, в якому система записує всі дії, які були виконані: запуск, моніторинг, шифрування, повідомлення, помилки тощо. Цей лог корисний для аудиту та діагностики.

Telegram містить дані для інтеграції з Telegram-ботом. Наявність такого каналу дозволяє оперативно інформувати адміністратора про підозрілу активність або спрацювання захисних механізмів. Використання Telegram як додаткового каналу підвищує мобільність і швидкість реагування.

Приклад вихідних даних системи у вигляді записів у журналі показано на рисунку 3.15. Як видно з ілюстрації, система записує кожну дію у вигляді позначки часу, опису події та імені користувача. Це дозволяє зберігати повну історію подій.

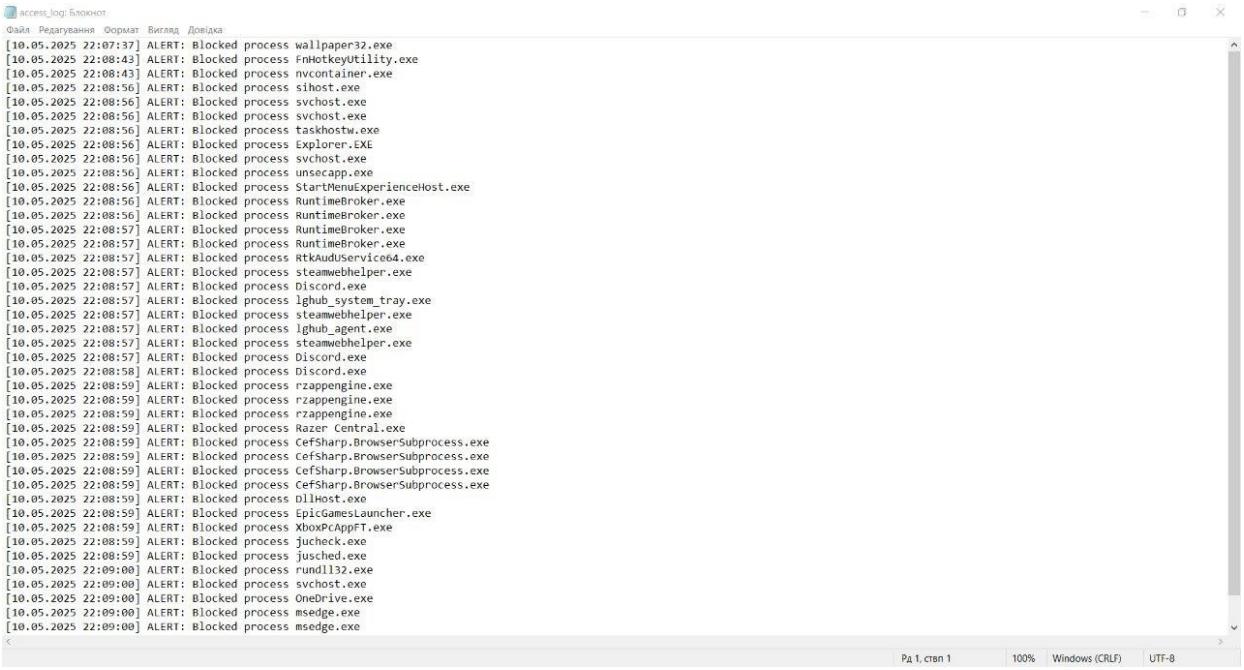


Рисунок 3.15 – Результат роботи програми у фалі з логами

Крім локального логу, система також надсилає повідомлення в Telegram-канал адміністратора, що значно прискорює процес оповіщення та дозволяє швидко реагувати навіть за відсутності доступу до консолей сервера.

У разі виявлення несанкціонованого файлу, запущеного з незареєстрованого каталогу, або при підозрі на зловмисну активність бот миттєво генерує повідомлення, що містить ім'я файлу, повний шлях до нього, часову мітку події, а також ім'я користувача або процесу, який ініціював запуск.

Крім того, додаткове поле містить рекомендації щодо подальших дій. Така структура повідомлення не тільки інформує, але й допомагає адміністратору швидко зорієнтуватися в ситуації та вжити необхідних заходів. Такий результат показано на рисунку 3.16.

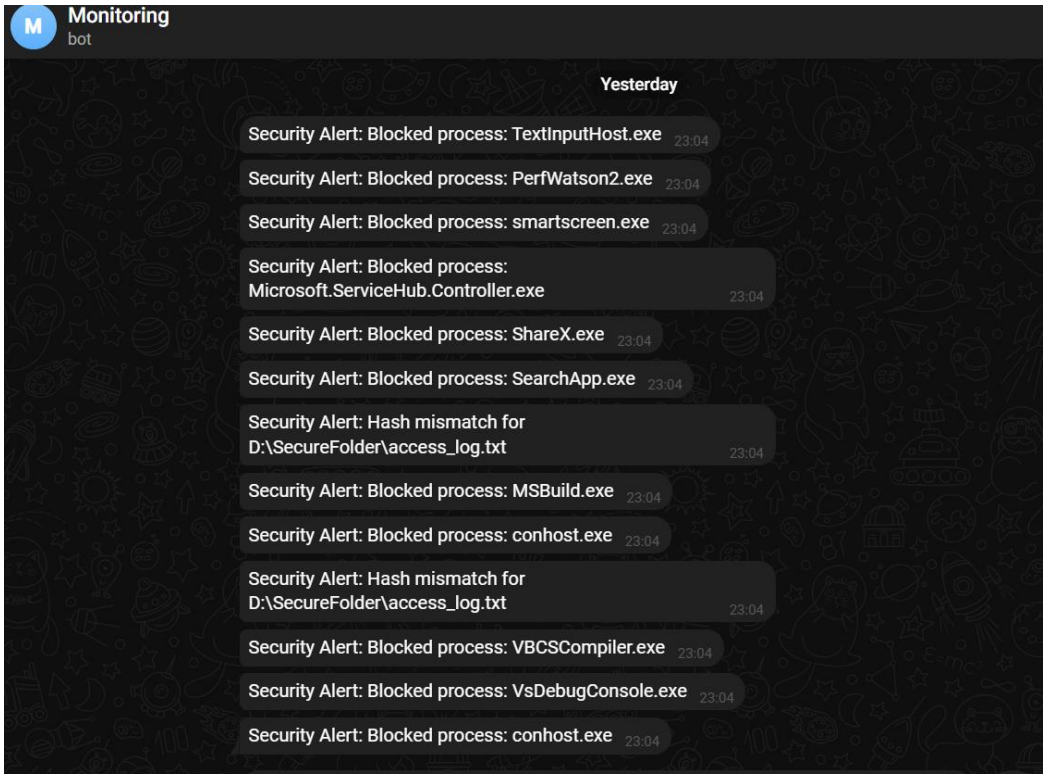


Рисунок 3.16 – Результат роботи програми у телеграм

Після завантаження налаштувань з конфігураційного файлу основна програма ініціалізує відповідні сервіси: систему моніторингу файлової системи FileSystemWatcher, механізм шифрування на основі ключів за допомогою Crypto, систему логування в файл і телеграм-сервіс. Всі ці компоненти працюють у фоновому режимі і дозволяють виявляти підозрілу активність у реальному часі.

Зокрема, при виявленні нового файлу в контрольованій папці перевіряється його ім'я та розширення. Якщо файл не входить до списку дозволених, система блокує його відкриття, шифрує або відправляє на карантин, додає запис до журналу, а також надсилає повідомлення в Telegram. Таким чином, впроваджена система дозволяє не тільки фіксувати події, а й оперативно реагувати на загрози,

мінімізуючи ймовірність витоку конфіденційних даних або запуску шкідливого програмного забезпечення.

Перевагою такої архітектури є легкість масштабування: нові компоненти, такі як підтримка електронної пошти, централізоване логування або інтеграція з SIEM-системами, можуть бути додані шляхом розширення відповідних розділів конфігурації в файлі appsettings.json без зміни основного коду програми.

Підсумовуючи, можна сказати, що розроблена система конфігурації дозволяє створити гнучке, надійне та адаптивне середовище для захисту важливої інформації. Її перевагами є повна автоматизація реагування на інциденти, централізоване налаштування та простота впровадження для різних користувачів та середовищ.

3.4 Висновки

Впровадження розробленої системи захисту конфіденційних даних від шкідливого програмного забезпечення. Розробка охоплює повний цикл від побудови блок-схеми алгоритму до створення повнофункціонального прототипу.

Система складається з кількох взаємопов'язаних модулів моніторингу процесів, перевірки хешів, логування, шифрування та сповіщення які разом забезпечують цілісний та ефективний механізм захисту.

Кожен модуль реалізовано як окремий клас у просторі імен SecuritySystem, що дозволяє забезпечити високий рівень структурованості коду, гнучкості та зручності обслуговування.

Параметри конфігурації системи зберігаються у файлі appsettings.json та включають: список дозволених процесів, список заблокованих хешів, шлях до контрольованих каталогів та параметри сповіщень Telegram та електронної пошти.

Система успішно реалізує функції, визначені на етапі визначення завдання: виявлення шкідливих процесів, реагування на виявлені загрози за допомогою шифрування або сповіщення, моніторинг змін файлів за допомогою перевірки хешів та ведення повного журналу подій.

Таким чином, розроблений програмний продукт підтверджує свою ефективність як засобу комплексного захисту конфіденційної інформації та відповідає всім заданим вимогам.

Висновки

Проведене дослідження підтвердило високу актуальність розробки ефективної системи захисту конфіденційних даних від шкідливого програмного забезпечення.

У сучасних умовах стрімкого зростання кібератак, що спрямовані на викрадення особистої та службової інформації, питання інформаційної безпеки набуває стратегічного значення як для окремих користувачів, так і для організацій.

У ході роботи було визначено об’єкт і мету системи захисту, яка повинна не лише забезпечувати виявлення та нейтралізацію шкідливого ПЗ, але й активно запобігати несанкціонованому доступу до даних.

Основними компонентами запропонованого підходу стали: постійний моніторинг процесів, шифрування та ізоляція критичної інформації, а також застосування сучасних технологій штучного інтелекту й машинного навчання для аналізу поведінки програм.

Особливу увагу приділено питанням зручності використання та збереження продуктивності системи, що є критично важливим фактором для її впровадження в реальних умовах експлуатації. Забезпечення балансу між високим рівнем захисту та комфортною роботою користувача стало ключовим завданням проєкту.

Таким чином, результати дослідження та розробки демонструють можливість створення дієвої, інтелектуальної системи захисту, здатної ефективно протистояти як відомим, так і новим кіберзагрозам, забезпечуючи надійне збереження конфіденційної інформації.

Перелік джерел посилання

1. Прогнози кібербезпеки яких загроз очікувати у 2025 році CyberCalm URL: https://cybercalm.org/novyny/prognozi-kiberbezpeki-2025 (дата звернення: 15.02.2025).

2. Класифікація загроз безпеці даних у комп'ютерних системах Tausoft URL: https://tausoft.com.ua/klasyfikacziya-zagroz-bezpeczi-ta-poshkodzhennya-danyh-u-kompyuternyh-systemah/ (дата звернення: 19.02.2025).

3. System security assurance a systematic literature review Sciencedirect URL: https://www.sciencedirect.com/science/article/pii/S1574013722000338 (дата звернення: 21.02.2025).

4. Cisco Secure Malware Analytics (Threat Grid) Cisco URL: https://www.cisco.com/c/en/us/products/security/secure-malware-analytics-threat-grid/index.html (дата звернення: 23.02.2025).

5. Cyber Security White Papers SANS Institute URL: https://www.sans.org/white-papers/ (дата звернення: 24.02.2025).

6. Home Page CISA URL: https://www.cisa.gov/ (дата звернення: 25.02.2025).

7. ISO/IEC 27005:2018 – Information security risk management ISO URL: https://www.iso.org/standard/75281.html (дата зверення: 26.02.2025).

8. NIST SP 800-30 Rev. 1: Guide for Conducting Risk Assessments NIST URL: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf (дата зверення: 27.02.2025).

9. ENISA Threat Landscape 2025 ENISA URL: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2025 (дата зверення: 01.03.2025).

10. SANS 2025 Cyber Threat Predictions SANS Institute URL: https://www.sans.org/white-papers/4014/ (дата зверення: 05.03.2025).

11. MITRE ATT&CK™ Framework MITRE URL: https://attack.mitre.org/ (дата звернення: 10.03.2025).

12. Security and Privacy Controls for Information Systems and Organizations (NIST SP 800-53 Rev. 5) NIST URL: https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final (дата звернення: 12.03.2025).

13. ISO/IEC 27001:2013 — Інформаційна безпека. Системи управління безпекою інформації. Вимоги ISO URL: https://www.iso.org/standard/54534.html (дата звернення: 15.03.2025).

14. ISO/IEC 27002:2022 — Інформаційна безпека. Кодекс практики контролів інформаційної безпеки ISO URL: https://www.iso.org/standard/75652.html (дата звернення: 17.03.2025).

15. OWASP Top 10 2021 OWASP URL: https://owasp.org/www-project-top-ten/ (дата звернення: 18.03.2025).

16. Top Strategic Technology Trends for 2025 Gartner URL: https://www.gartner.com/en/documents/3987890 (дата звернення: 20.03.2025).

17. Microsoft Digital Defense Report 2025 Microsoft URL: https://www.microsoft.com/security/business/microsoft-digital-defense-report-2025 (дата звернення: 22.03.2025).

18. Cisco Annual Cybersecurity Report 2025 Cisco URL: https://www.cisco.com/c/en/us/products/security/security-reports.html (дата звернення: 24.03.2025).

19. IBM X-Force Threat Intelligence Index 2025 IBM URL: https://www.ibm.com/security/data-breach/threat-intelligence-index/2025 (дата звернення: 26.03.2025).

20. CrowdStrike Global Threat Report 2025 CrowdStrike URL: https://www.crowdstrike.com/resources/reports/global-threat-report-2025/ (дата звернення: 28.03.2025).

21. Trend Micro Security Predictions 2025 Trend Micro URL: https://www.trendmicro.com/vinfo/us/security/research-and-analytics/predictions/2025 (дата звернення: 30.03.2025).

22. M-Trends 2024 Mandiant (FireEye) URL: https://www.mandiant.com/resources/m-trends-2024-report (дата звернення: 06.04.2025).

23. Check Point Research Cyber Security Report 2025 Check Point URL: https://research.checkpoint.com/2025/ (дата звернення: 08.04.2025).

24. Verizon 2024 Data Breach Investigations Report Verizon URL: https://www.verizon.com/business/resources/reports/dbir/2024/ (дата звернення: 10.04.2025).

25. Unit 42 Cloud Threat Report 2025 Palo Alto Networks URL: https://unit42.paloaltonetworks.com/cloud-threat-report-2025/ (дата звернення: 12.04.2025).

26. Fortinet 2025 Threat Landscape Report Fortinet URL: https://www.fortinet.com/content/dam/fortinet/assets/threat-reports/threat-landscape-report-2025.pdf (дата звернення: 14.04.2025).

27. Akamai State of the Internet Security Report Q1 2025 Akamai URL: https://www.akamai.com/state-of-the-internet-security-q1-2025 (дата звернення: 16.04.2025).

28. Falcon X Threat Intelligence Report 2024 CrowdStrike URL: https://www.crowdstrike.com/resources/reports/falcon-x-threat-intel-2024/ (дата звернення: 18.04.2025).

29. Global Threat Index 2024 Check Point URL: https://research.checkpoint.com/2024/global-threat-index/ (дата звернення: 20.04.2025).

30. QRadar SIEM Technical White Paper IBM URL: https://www.ibm.com/downloads/cas/QRADAR-WHITEPAPER (дата звернення: 22.04.2025).

31. Security Intelligence Report 2024 Microsoft URL: https://www.microsoft.com/security/sir-2024 (дата звернення: 24.04.2025).

32. Internet Security Threat Report 2024 Broadcom URL: https://www.broadcom.com/products/cyber-security/threat-reports/internet-security-threat-report-2024 (дата звернення: 26.04.2025).

33. Internet Crime Report 2024 FBI IC3 URL: https://www.ic3.gov/Media/PDF/AnnualReport/2024_IC3Report.pdf (дата звернення: 28.04.2025).

34. Enterprise Security Threat Intelligence Guide Splunk URL: https://www.splunk.com/en_us/form/splunk-enterprise-security-threat-intelligence.html (дата звернення: 30.04.2025).

35. BlackBerry Cylance Threat Report 2025 BlackBerry URL: https://www.blackberry.com/us/en/forms/cylance/cylance-threat-report-2025 (дата звернення: 02.05.2025).

36. X-Ops Threat Report 2025 Sophos URL: https://www.sophos.com/en-us/security-news-trends/sophos-threatexpert-report-2025 (дата звернення: 04.05.2025).

37. DNS Threat Report 2024 F5 URL: https://www.f5.com/labs/articles/dns-threat-report-2024 (дата звернення: 06.05.2025).

38. The Human Factor 2025 Proofpoint URL: https://www.proofpoint.com/us/resources/threat-reports/human-factor-2025 (дата звернення: 08.05.2025).

39. IoT Threat Report 2024 Unit 42 (Palo Alto) URL: https://unit42.paloaltonetworks.com/iot-threat-report-2024/ (дата звернення: 10.05.2025).

40. Overwatch Report 2024 CrowdStrike URL: https://www.crowdstrike.com/resources/overwatch-report-2024/ (дата звернення: 10.05.2025).

41. Google Cloud Security Whitepaper Google URL: https://cloud.google.com/security/whitepaper (дата звернення: 08.05.2025).

42. AWS Security Best Practices AWS URL: https://docs.aws.amazon.com/whitepapers/latest/aws-security-best-practices/aws-security-best-practices.pdf (дата звернення: 09.05.2025).

43. Azure Security Documentation Microsoft URL: https://learn.microsoft.com/en-us/azure/security/ (дата звернення: 10.05.2025).

44. Threat Analysis Reports BleepingComputer URL: https://www.bleepingcomputer.com/category/security/ (дата звернення: 09.05.2025).

45. Annual Review 2024 NCSC (UK) URL: https://www.ncsc.gov.uk/annual-review-2024 (дата звернення: 08.05.2025).

Додаток А
(Обов’язковий)
Програмний код вузла програмного забезпечення системи

Клас для відправки сповіщень про інциденти безпеки:

```
using Microsoft.Extensions.Configuration;
using MimeKit;
using MailKit.Net.Smtp;
using Telegram.Bot;
using Telegram.Bot.Types;

namespace SecuritySystem.Modules
{
   public class AlertService
   {
      private readonly IConfiguration _cfg;
      private readonly ITelegramBotClient _bot;
      public AlertService(IConfiguration cfg)
      {
         _cfg = cfg;
         _bot = new TelegramBotClient(_cfg["Telegram:Token"]);
      }
      public async Task SendAsync(string subject, string body)
      {
         // Telegram
         var chatId = new ChatId(long.Parse(_cfg["Telegram:ChatId"]));
         await _bot.SendTextMessageAsync(chatId, $"{subject}: {body}");
      }
   }
}
```

Клас для виконання всіх криптографічних операцій (шифрування і дешифрування файлів):

```
using Microsoft.Extensions.Configuration;
using System.Security.Cryptography;
using System.Text;

namespace SecuritySystem.Modules
{
   public class CryptoService
   {
      private readonly IConfiguration _cfg;
```

```
        public CryptoService(IConfiguration cfg)
        {
            _cfg = cfg;
        }

        public void EncryptFile(string inFile, string outFile)
        {
            var key =
Encoding.UTF8.GetBytes(_cfg["Crypto:Key"].PadRight(32).Substring(0, 32));
            var iv = new byte[16];
            using var aes = Aes.Create();
            aes.Key = key; aes.IV = iv;
            using var fsIn = File.OpenRead(inFile);
            using var fsOut = File.OpenWrite(outFile);
            using var cs = new CryptoStream(fsOut, aes.CreateEncryptor(),
CryptoStreamMode.Write);
            fsIn.CopyTo(cs);
        }

        public void DecryptFile(string inFile, string outFile)
        {
            var key =
Encoding.UTF8.GetBytes(_cfg["Crypto:Key"].PadRight(32).Substring(0, 32));
            var iv = new byte[16];
            using var aes = Aes.Create();
            aes.Key = key; aes.IV = iv;
            using var fsIn = File.OpenRead(inFile);
            using var fsOut = File.OpenWrite(outFile);
            using var cs = new CryptoStream(fsIn, aes.CreateDecryptor(),
CryptoStreamMode.Read);
            cs.CopyTo(fsOut);
        }
    }
}
```

Клас для контролю хешів файлів у моніторинговій папці:

```
using Microsoft.Extensions.Configuration;
using System.Security.Cryptography;

namespace SecuritySystem.Modules
{
    public class HashChecker
```

```
    {
        private FileSystemWatcher _watcher;
        private readonly string _folder;
        private readonly IConfiguration _cfg;
        public event Func<string, Task> OnMismatch;

        public HashChecker(IConfiguration cfg)
        {
            _cfg = cfg;
            _folder = cfg["MonitorFolder"];
        }

        public void Start()
        {
            _watcher = new FileSystemWatcher(_folder)
            {
                IncludeSubdirectories = true,
                EnableRaisingEvents = true
            };
            _watcher.Created += async (s, e) => await CheckHash(e.FullPath);
            _watcher.Changed += async (s, e) => await CheckHash(e.FullPath);
        }

        private async Task CheckHash(string path)
        {
            if (!File.Exists(path)) return;
            using var sha = SHA256.Create();
            using var fs = File.OpenRead(path);
            var hash = BitConverter.ToString(sha.ComputeHash(fs)).Replace("-",
string.Empty);
            var known = _cfg.GetSection("KnownHashes").Get<string[]>();
            if (!known.Contains(hash, StringComparer.OrdinalIgnoreCase))
                await OnMismatch?.Invoke($"Hash mismatch for {path}");
        }
    }
}
```

Клас для запису повідомлень про загрози до журналу логування:

```
using Microsoft.Extensions.Configuration;

namespace SecuritySystem.Modules
{
    public class LoggerService
```

```
    {
        private readonly string _logPath;
        private static readonly object _logLock = new();
        public LoggerService(IConfiguration cfg)
        {
            _logPath = cfg["LogFilePath"];
        }
        public void LogAlert(string msg)
        {
            var line = $"[{DateTime.Now}] ALERT: {msg}{Environment.NewLine}";
            try
            {
                lock (_logLock)
                {
                    File.AppendAllText(_logPath, line);
                }
            }
            catch (IOException ex)
            {
                Console.Error.WriteLine("Не вдалося записати лог: " + ex.Message);
            }
        }
    }
}
```

Клас для моніторингу запущених процесів та їх зупинки, якщо вони не в білому списку:

```
using Microsoft.Extensions.Configuration;

namespace SecuritySystem.Modules
{
    public class ProcessMonitor
    {
        private readonly string[] _allowed;
        private Timer _timer;
        public event Func<string, Task> OnThreatDetected;
        public ProcessMonitor(IConfiguration cfg)
        {
            _allowed = cfg.GetSection("AllowedExecutables").Get<string[]>();
        }
        public void Start()
        {
```

```
            _timer = new Timer(CheckProcesses, null, TimeSpan.Zero,
TimeSpan.FromMinutes(1));
        }
        private void CheckProcesses(object state)
        {
            foreach (var proc in System.Diagnostics.Process.GetProcesses())
            {
                try
                {
                    var name = Path.GetFileName(proc.MainModule.FileName);
                    if (!_allowed.Contains(name, StringComparer.OrdinalIgnoreCase))
                    {
                        proc.Kill();
                        OnThreatDetected?.Invoke($"Blocked process: {name}");
                    }
                }
                catch (Exception ex)
                {
                    Console.WriteLine($"Error while processing {proc.ProcessName}
({proc.Id}): {ex.Message}");
                }
            }
        }
    }
}
```

Клас, що реалізує фонову службу — обробник подій безпеки:

```
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.Hosting;

namespace SecuritySystem
{
    public class SecurityWorker : BackgroundService
    {
        private readonly ProcessMonitor _pm;
        private readonly HashChecker _hc;
        private readonly CryptoService _cs;
        private readonly LoggerService _lg;
        private readonly AlertService _al;
        private readonly IConfiguration _cfg;
        public SecurityWorker(ProcessMonitor pm, HashChecker hc, CryptoService cs,
LoggerService lg, AlertService al, IConfiguration cfg)
        {
```

```
            _pm = pm;
            _hc = hc;
            _cs = cs;
            _lg = lg;
            _al = al;
            _cfg = cfg;
        }

        protected override async Task ExecuteAsync(CancellationToken stoppingToken)
        {
            _pm.Start();
            _hc.Start();
            _pm.OnThreatDetected += async (msg) => await HandleThreatAsync(msg);
            _hc.OnMismatch += async (msg) => await HandleThreatAsync(msg);

            await Task.Delay(Timeout.Infinite, stoppingToken);
        }

        private async Task HandleThreatAsync(string message)
        {
            _lg.LogAlert(message);
            await _al.SendAsync("Security Alert", message);
        }
    }
}
```

Клас з точкою входу програми:

```
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
using Microsoft.Extensions.Logging;
using SecuritySystem;
using SecuritySystem.Modules;

namespace SecurityDemo
{
    class Program
    {
        public static async Task Main(string[] args)
        {
            var host = Host.CreateDefaultBuilder(args)
                .ConfigureAppConfiguration((context, config) =>
                {
```

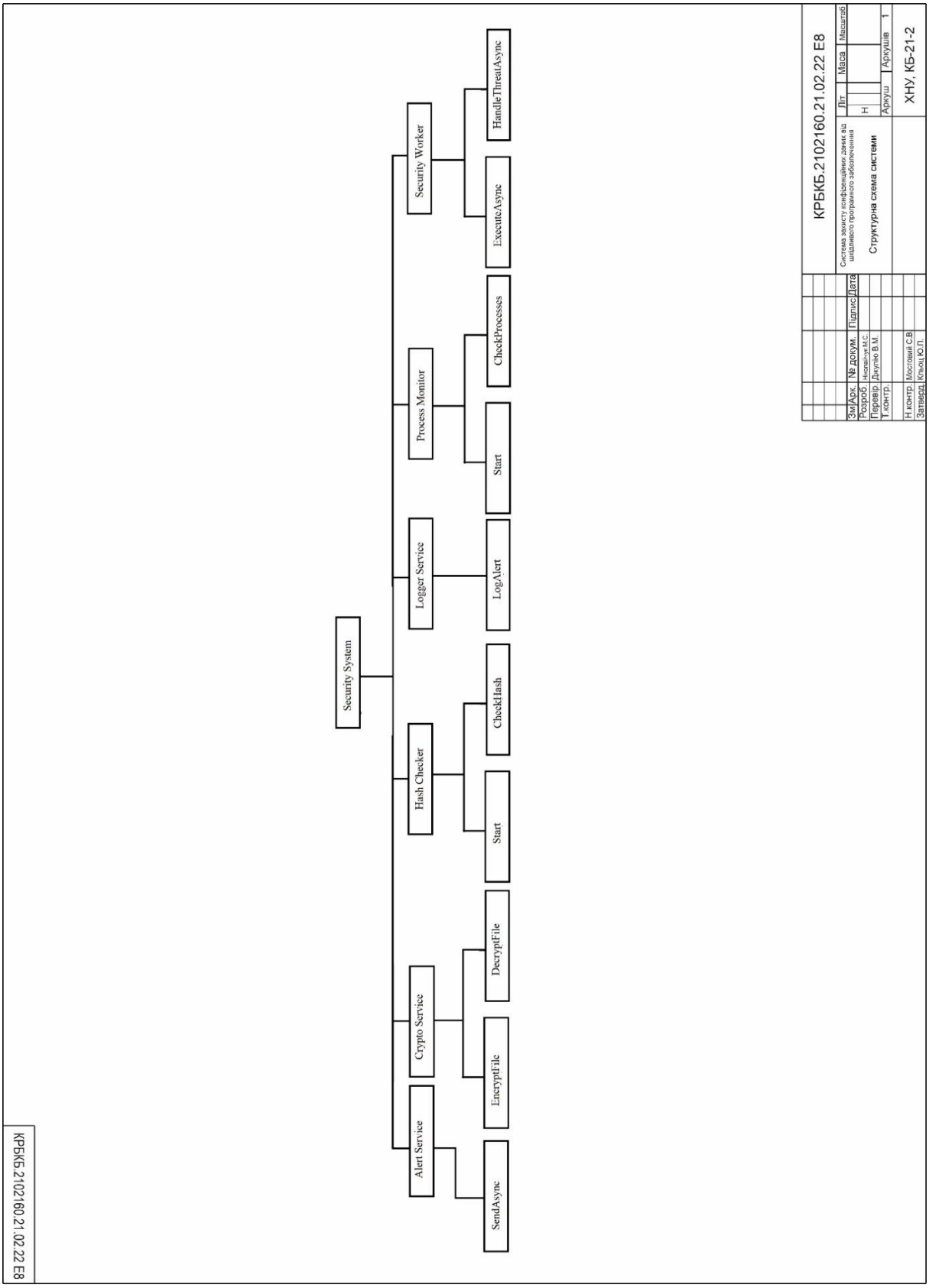
```
                config.AddJsonFile("appsettings.json", optional: false, reloadOnChange:
true);
            })
            .ConfigureServices((context, services) =>
            {
                services.AddSingleton<ProcessMonitor>();
                services.AddSingleton<HashChecker>();
                services.AddSingleton<CryptoService>();
                services.AddSingleton<LoggerService>();
                services.AddSingleton<AlertService>();
                services.AddHostedService<SecurityWorker>();
            })
            .ConfigureLogging(logging =>
            {
                logging.ClearProviders();logging.AddConsole();
            })
            .Build();
        await host.RunAsync();
    }
  }
}
```

Файл конфігурації appsettings.json, що містить параметри для всіх модулів:

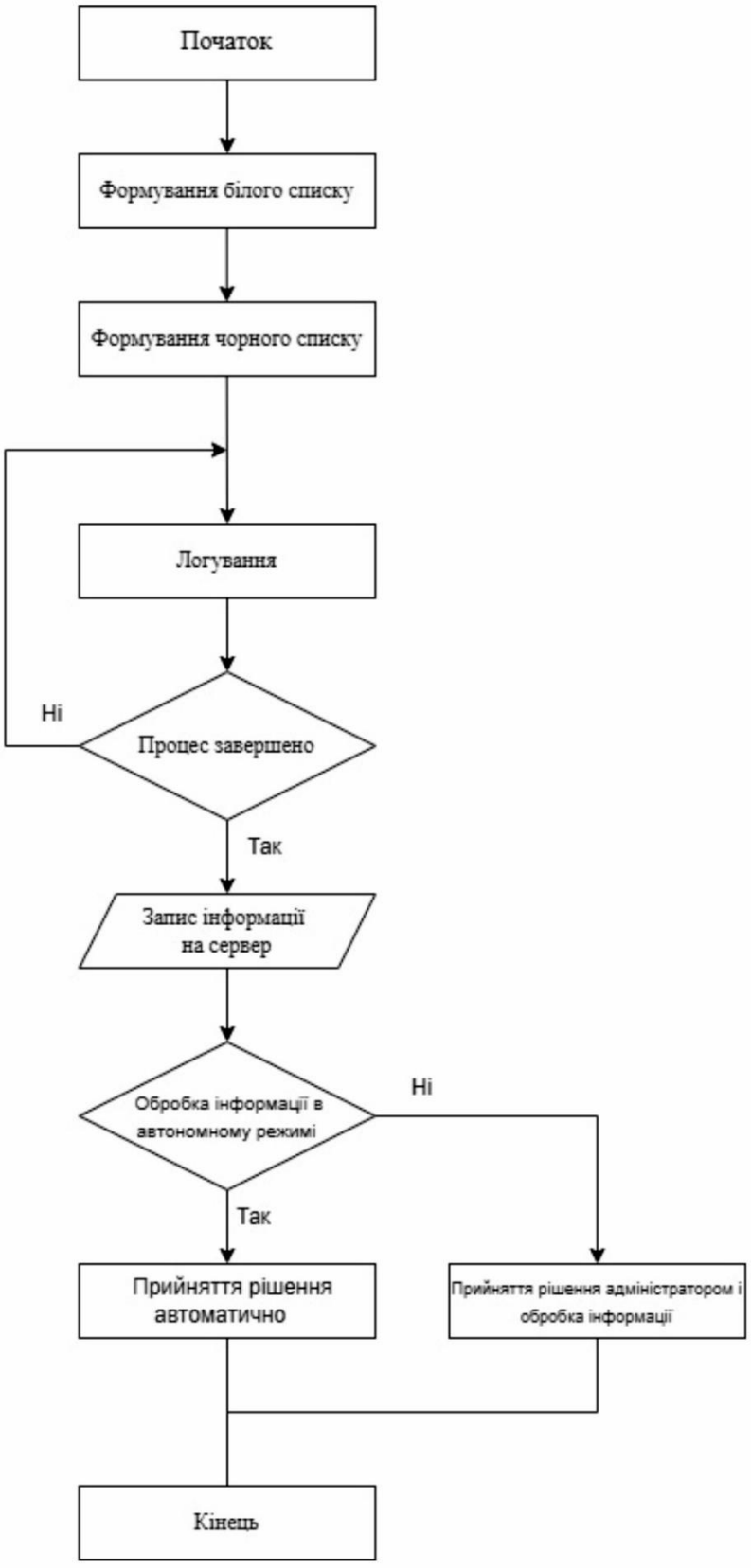
```
{
 "AllowedExecutables": ["explorer.exe", "svchost.exe", "devenv.exe"],
 "MonitorFolder": "D:\\SecureFolder",
 "KnownHashes": ["e3b0c...", "2d711..."],
 "Crypto": {
  "Key": "MySuperSecretEncryptionKey123!"
 },
 "LogFilePath": "D:\\SecureFolder\\access_log.txt",
 "Telegram": {
  "Token": "<bot_token>",
  "ChatId": "<chat_id>"
}
}
```

Додаток Б

(Обов’язковий)
Копії графічної частини



					КРБКБ.2102160.21.02.22 Е8		
Зм	Арк.	№ докум.	Підпис	Дата	Система захисту конфіденційних даних від шкідливого програмного забезпечення	Літ	Маса / Масштаб
Розроб.		Ніколайчук М.С.			Схема процесу проведення оглядового дослідження	Н	
Перевір.		Джулію В.М.				Аркуш	Аркушів 1
Т.контр.							
Н.контр.		Мостовий С.В.				ХНУ, КБ-21-2	
Затверд.		Кльоц Ю.П.					



					КРКБ.2102160.21.02.22 Е8			
Зм	Арк.	№ докум.	Підпис	Дата	Система захисту конфіденційних даних від шкідливого програмного забезпечення	Літ	Маса	Масштаб
Розроб.		Ніконенко М.С.				Н		
Перевір.		Джуній В.М.			Алгоритм роботи системи	Аркуш	Аркушів	1
Т.контр.								
Н.контр.		Мостовий С.В.				ХНУ, КБ-21-2		
Затверд.		Кльоп Ю.П.						

Додаток В

(Обов’язковий)
Копії наукових публікацій

Міністерство освіти і науки України
Хмельницький національний університет

ЗБІРНИК НАУКОВИХ ПРАЦЬ
за матеріалами XVI Всеукраїнської науково-практичної конференції
«Актуальні проблеми комп’ютерних наук АПКН-2024»

15-16 листопада 2024

Хмельницький 2024

Денисенко В.О., Мельников О.Ю.
Вдосконалення наявного додатку для оброблення інформації про лісові насадження 175

Дидо Р.А., Мазурець О.В., Кліменко В.І.
Інформаційна система для нейромережевої інтерактивної ідентифікації особистості за зображенням обличчя 180

Дідур В.О.
Нейромережева класифікація залишків будівництва 187

Діхтяр М.О., Радюк П.М., Скрипник Т.К.
Метод інтерпретування результатів виявлення патологій серця за зображенням МРТ 189

Драган Т.С., Галка А.О., Ніколайчук М.С., Джулій В.М.
Алгоритм передачі конфіденційної інформації без спотворення растрового зображення 192

Жайворон Д.О., Пасічник О.А., Скрипник Т.К., Манзюк Е.А.
Метод ідентифікації лікарських рослин за аналізом зображень нейромережевими засобами 196

Жарновський О.В., Казмірчук Я.М., Собко О.В., Мазурець О.В.
Практична реалізація методу ідентифікації згенерованих штучним інтелектом зображень людей засобами машинного навчання 198

Жарчинський С.М.
Система надійного зберігання даних на основі Openstack Object Storage 205

Жук Д.І., Мазурець О.В., Кадинська В.Д., Тищенко О.О.
Підхід до визначення сумісності клієнтів шлюбних агентств за інтелектуальним аналізом анкетних даних 208

Загребельний В.В., Кльоц Ю.П.
Роль OSINT у протидії кіберзлочинності та веденні інформаційної війни 215

Зайцев І.О., Федоров Є.Є.
Кіберфізична система для інтерактивного відображення доступності міської інфраструктури 218

Залуцька О.О.
Метод автоматизованого оцінювання відповідності тональності відгуків на товари в інтернет-магазинах до їх користувацької оцінки з використанням нейромереж 221

УДК 004.891

Драган Т.С., Галка А.О., Ніколайчук М.С., Джулій В.М.

Хмельницький національний університет

АЛГОРИТМ ПЕРЕДАЧІ КОНФІДЕНЦІЙНОЇ ІНФОРМАЦІЇ БЕЗ СПОТВОРЕННЯ РАСТРОВОГО ЗОБРАЖЕННЯ

Запропоновано алгоритм передачі конфіденційної інформації без спотворення растрового зображення. Наведено поняття прихованого каналу передачі інформації, класифікація методів і завдань прихованої передачі інформації. Запропонований алгоритм, на відміну від розглянутих, має високу пропускну здатність. Файл-ключ пікселів стискається і зберігається в форматі результуючого растрового зображення.

An algorithm for transmitting confidential information without distortion of the raster image is proposed. The concept of a hidden channel of information transmission, the classification of methods and tasks of hidden information transmission are given. The proposed algorithm, unlike the considered ones, has a high bandwidth. The pixel key file is compressed and saved in the format of the resulting bitmap image.

В області комунікації безпека є однією із головних проблем сучасного світу. Стеганографічні програмні засоби приховування конфіденційної інформації забезпечують перевагу перед іншими видами програмної інформаційної безпеки, оскільки текст ховається у зображенні, яке не сприймається як носій текстової інформації. Стеганографія та криптографія – це способи захисту інформації від несанкціонованого доступу, але ці технології окремо не досконалі, і можуть бути скомпрометовані. Як тільки наявність прихованої інформації виявляється, або якщо виникне якась підозра, то стеганографія частково зазнає поразки. Ефективність стеганографії можна підсилити шляхом об'єднання її з криптографією [1,2].

Алгоритми приховування конфіденційної інформації в графічних файлах орієнтовані на формати файлів з втратою, наприклад, JPEG. На відміну від LSB вони більш стійкі до геометричних перетворень. Це відбувається за рахунок варіювання в широкому діапазоні якості зображення, що призводить до неможливості визначення джерела зображення.

Стеганографічна система - сукупність засобів і методів, за допомогою яких створюється прихований канал передачі інформації. При побудові стегосистеми повинні враховуватися наступні положення: супротивник має повне представлення про стеганографічні системи і деталі їх реалізації. Єдиною інформацією, яка залишається невідомою потенційному супротивникові, є ключ, за допомогою якого тільки його власник може встановити факт присутності і зміст прихованого повідомлення. Потенційний супротивник повинен бути позбавлений будь-яких

технічних та інших переваг у розпізнаванні або розкритті змісту таємних повідомлень. На рисунку 1 представлена узагальнена схема стегосистеми [1,2].

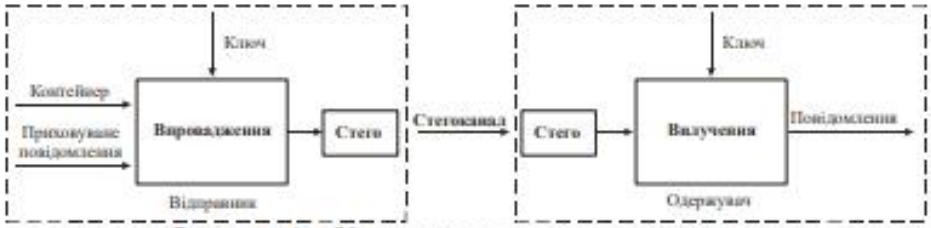


Рисунок 1 – Узагальнена схема стегосистеми

Широке розповсюдження растрових та мультимедійних технологій дало імпульс розвитку нових і вдосконаленню існуючих методів приховування інформації, а також сприяло виникненню більш складних методів організації прихованих каналів зв'язку, в основу яких були покладені особливості подання інформації в комп'ютерних файлах, пристроях, обчислювальних мережах і т.п. [2-5]. Класифікація методів стеганографії представлена на рисунку 2.

Рисунок 2 – Класифікація методів стеганографії

Алгоритм передачі прихованої конфіденційної інформації без спотворення растрового зображення дозволяє із зображення і файла даних, який необхідно приховати, отримати файл-ключ, з використанням якого можна буде витягнути

файл даних із початкового зображення, без його спотворення [2,3]. Для того, щоб приховати конфіденційні дані, необхідно на вхід подати растрове зображення у форматі .PNG і файл з даними. А на виході отримаємо файл-ключ у форматі *.PNG (рисунок 3а). Для того, щоб витягнути файл даних із растрового зображення, необхідно на вхід подати початкове зображення і файл-ключ, отриманий в процесі шифрування, на виході отримаємо файл даних (рисунок 3б).

а) б)

Рисунок 3 – Загальна схема формування даних: а) формування файла - ключа; б) витягнення файла даних із растрового зображення

Алгоритм роботи системи передачі інформації без спотворення растрового зображення представлений на рисунку 4.

Рисунок 4 – Алгоритм роботи системи передачі конфіденційної інформації без спотворення растрового зображення

На основі проведеного аналізу наведено поняття прихованого каналу передачі інформації, а також класифікацію методів і завдань прихованої передачі інформації. Проведено огляд існуючих методів впровадження інформації в растрові зображення. Розглянуті можливі області застосування стеганографії, зокрема-стеганографія може бути використана для зберігання і розподілення ключів в мережах зв'язку. Виявлені недоліки методів впровадження інформації в растрові зображення і їх програмні реалізації не дозволяють в повній мірі використовувати їх для безпечної передачі інформації. Для усунення виявлених недоліків запропонований алгоритм передачі прихованої інформації без спотворення растрового зображення, який дозволяє із зображення і файла даних, отримати файл-ключ пікселів, за допомогою якого можна витягнути файл даних із початкового растрового зображення. без спотворення растрового зображення.

Для організації прихованого каналу зв'язку, розподілу і передачі ключової інформації запропонований метод, на відміну від розглянутих, має високу пропускну здатність. Файл-ключ пікселів стискається і зберігається в форматі результуючого растрового зображення. Через низьку ентропію файл-ключа пікселів, він буде мати набагато менший розмір, ніж початкове растрове зображення. При необхідності існує можливість шифрування файл-ключа пікселів методом симетричного шифрування або RSA, що значно підвищить стійкість до атак - впроваджена інформація може піддаватися злому, видаленню чи атакам. Стійкість є головною вимогою, що висувається до будь яких стеганографічних методів, а також забезпечить секретність вбудованої інформації. Розроблені методи впровадження інформації в растрові зображення можуть застосовуватися для ефективного захисту авторських прав інтелектуальної власності.

Перелік посилань

1. Конахович Г.Ф. Комп'ютерна стеганографічна обробка й аналіз мультимедійних даних : навчальна література/ Г.Ф. Конахович, Д.О. Прогонов, О.Ю. Пузиренко – Центр навч. літ., 2018р. – 560 с.
2. Koptyra, K., Ogiela, M.R. Steganography in IoT: Information Hiding with Joystick and Touch Sensors. Sensors 2023, https://doi.org/10.3390/s23063288
3. Лук'янов, Б. В. Комп'ютерний аналіз даних / Б. В. Лук'янов. К. : Академія, 2017р. – 345с.
4. Лавров, Є. А. Математичні методи дослідження операцій: підручник / Є. А. Лавров, Л. П. Перхун, В. В. Шендрик – Суми : Сумський державний університет, 2017. – 212 с
5. Лєнков С.В. Концептуальна схема системи інтелектуальної обробки даних / С.В. Лєнков, В.М. Джулій, О.М. Горбатюк, Н.М. Берназ // Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка. – К.: ВІКНУ, 2014. – Вип. № 46. – С.181-190

Завідувачу кафедри кібербезпеки
к.т.н., доц. Кльоцу Ю.П.
Ніколайчук Мар'ян Сергійович
ПІБ здобувача вищої освіти

Студента ФІТ, 4 курсу, групи КБ-21-2

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 31.08.2023, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

28.05.2025 Н.Мич.

Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 1.0%

Dictionaries check: en_US, ru_RU, ua_UA. **Errors in the documents: 8%**

ID: 242466 Title: Система захисту конфіденційних даних від шкідливого програмного забезпечення Added in a DB: 2025-05-29 Authors: Ніколайчук Мар'ян Сергійович Heads: Джулій В.М. Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	73857	1087	992 (1%)	12 (1%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Ніколайчук Мар'ян Сергійович

Співавтор:

Назва: Система захисту конфіденційних даних від шкідливого програмного забезпечення

Науковий керівник:

Підрозділ: Кафедра кібербезпеки

Коефіцієнт подібності 1:0.8%

Коефіцієнт подібності 2:0.2%

Мікропробіли: 0

Заміна букв: 0

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2025-05-29 22:26:47.0

Після аналізу Звіту подібності констатую наступне:

☐ **Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.**

☐ **Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.**

☐ **Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.**

Обгрунтування:

30.05.2025р

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРИ КІБЕРБЕЗПЕКИ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Система захисту конфіденційних даних від шкідливого програмного забезпечення

Автор: Ніколайчук Мар`ян Сергійович

Спеціальність: 125 – Кібербезпека

Освітня програма:Кібербезпека

Науковий керівник:Володимир ДЖУЛІЙ, канд. техн. наук, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних). Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) запозичення розміщені в розділах аналізу існуючих методів та технологій, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості, складає 1%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи _____ Володимир ДЖУЛІЙ

Завідувач кафедри Кб _____ Юрій КЛЬОЦ

Гарант ОП _____ Віктор ЧЕШУН

Дата:

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

освітнього ступеня «бакалавр»

Студент Ніколайчук Мар'ян Сергійович

Тема Система захисту конфіденційних даних від шкідливого програмного забезпечення

Спеціальність 125 – Кібербезпека

Обсяг кваліфікаційної роботи освітньо-кваліфікаційного рівня «бакалавр»:

кількість листів креслень 3; кількість сторінок записки 72.

1. Короткий зміст роботи та прийнятих рішень У кваліфікаційній роботі була розроблена система захисту конфіденційних даних від шкідливого програмного забезпечення, основана на модульній архітектурі з центральним компонентом SecurityWorker, який координує фонова обробка даних та взаємодію з іншими сервісами. Ця система забезпечує своєчасне виявлення та нейтралізацію загроз шляхом моніторингу процесів, контролю цілісності, шифрування даних і генерації сповіщень. У процесі проєктування були розроблені такі компоненти: сервіс оповіщень AlertService, криптосервіс CryptoService, модуль перевірки хешів HashChecker, сервіс ведення журналу LoggerService та компонент моніторингу процесів ProcessMonitor. Крім того, для гнучкого зберігання параметрів конфігурації використано бібліотеку Microsoft.Extensions.Configuration, а реалізація здійснена на платформі .NET із використанням асинхронних інтерфейсів для високої продуктивності. Надані рекомендації щодо розгортання рішення в корпоративній мережі та проведення локального тестування з репрезентативним набором зразків шкідливого ПЗ.

2. Висновок про відповідність кваліфікаційної роботи завданню.У кваліфікаційній роботі було повністю реалізовано поставлені завдання як у теоретичній, так і в практичній частинах. Теоретичний розділ охоплює аналіз загроз, оцінку ризиків і огляд сучасних підходів до захисту конфіденційних даних, тоді як практична частина передбачає проєктування, реалізацію та тестування модульної системи з центральним компонентом Security Worker. Всі вимоги до функціональності системи виконані: виявлення, блокування та логування підозрілих подій, шифрування та перевірка цілісності даних.

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У роботі послідовно вирішено низку завдань із застосуванням найновіших наукових і технічних розробок. У першому розділі здійснено системний аналіз загроз та сучасного шкідливого ПЗ, проведено кількісну оцінку ризиків із використанням матриці ризиків і чітко сформульовано завдання проєктування. У другому розділі обґрунтовано принципи захисту — моніторинг процесів, аналіз мережевого трафіку, симетричне шифрування та ізоляцію даних — описано алгоритми збору й кореляції подій та розроблено модульну архітектуру із п'ятьма сервісами AlertService, CryptoService, HashChecker, LoggerService, ProcessMonitor. У третьому розділі спроектовано детальний алгоритм роботи системи на базі асинхронних інтерфейсів .NET, реалізовано програмні компоненти із застосуванням Microsoft Extensions Configuration для гнучкої конфігурації, TensorFlow для евристичного аналізу поведінки процесів, ELK-стека для централізованого логування, а також налаштовано тестове середовище з репрезентативним набором зразків шкідливого ПЗ. Такий підхід забезпечив комплексність рішення, гнучкість архітектури та високу ефективність захисту без істотного впливу на продуктивність.

4. Позитивні сторони роботи Система вирізняється збалансованим багаторівневим підходом до захисту, який поєднує поведінковий аналіз процесів, контроль цілісності даних та симетричне шифрування, при цьому для виявлення нових, раніше невідомих загроз застосовано методи машинного навчання. Модульна архітектура системи забезпечує високу гнучкість, простоту інтеграції та масштабованість, а асинхронна реалізація сервісів на платформі .NET гарантує високу продуктивність із мінімальним впливом на ресурси операційної системи.

5. Негативні сторони роботи У системі відсутній графічний інтерфейс для керування системою, що ускладнює її використання некваліфікованими користувачами.Не передбачено автоматизованого резервного копіювання зашифрованих даних та конфігураційних контейнерів, що підвищує ризики при відмові обладнання або витоку даних.Крім того, підтримка обмежена лише ОС Windows Server, а відсутність версії для Linux/Unix знижує універсальність та можливість інтеграції в різноманітні корпоративні середовища.

6. Оцінка графічного оформлення та пояснювальної записки роботи. Графічне оформлення кваліфікаційної роботи відповідає темі роботи та виконане з дотриманням стандартів. Графічне оформлення є чітким та якісним, а пояснювальна записка відповідає нормам оформлення університету.

7. Відгук про роботу в цілому Кваліфікаційна робота заслуговує високої оцінки завдяки чіткій структурі та послідовному викладенню матеріалу. Усі розділи логічно пов'язані між собою, що сприяє глибшому розумінню теми, а графічні ілюстрації наочно демонструють доцільність і ефективність запропонованих рішень.

8. Інші зауваження У переліку використаних джерел присутні неконсолідовані ресурси .Рекомендується замінити їх на наукові статті, профільні стандарти ISO/IEC та публікації.

9. Оцінка кваліфікаційної роботи Враховуючи всебічний аналіз, застосування сучасних технологій і якісну реалізацію, робота заслуговує оцінки «відмінно».

РЕЦЕНЗЕНТ (прізвище, ім'я, по батькові, посада, місце роботи)
Бойко Юлій Миколайович,
доктор технічних наук, професор кафедри ТМІТ

« 3 » 06 2025. (підпис)