

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки**

«На правах рукопису»
УДК _____

До захисту допущено
Завідувач кафедри

_____ Дмитро ЛАНДЕ
(підпис)

«_____» _____ 2025 р.

Магістерська дисертація

на здобуття ступеня магістра

**за освітньо-професійною програмою «Системи, технології та
математичні методи кібербезпеки»
спеціальності 125 «Кібербезпека та захист інформації»**

на тему: Автоматизація звітності про кіберінциденти з використанням LLM-експерта

Виконав: здобувач вищої освіти **VI** курсу, групи ФБ-42мп

Андреєв Данило Юрійович

(підпис)

Науковий керівник доцент кафедри ІБ к.т.н. Стьопочкіна Ірина Валеріївна

(підпис)

Рецензент доцент кафедри ОТ к.т.н. Волокита Артем Миколайович

(підпис)

Засвідчую, що у цій магістерській дисертації немає
запозичень з праць інших авторів без відповідних
посилань.

Здобувач вищої освіти _____
(підпис)

Київ – 2025 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки**

Рівень вищої освіти – другий (магістерський)
Спеціальність – 125 «Кібербезпека та захист інформації»
Освітньо-професійна програма «Системи, технології та математичні методи кібербезпеки»

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____ Дмитро ЛАНДЕ
(підпис)

« ____ » _____ 2025 р.

**ЗАВДАННЯ
на магістерську дисертацію студенту**

Андрєєву Данилу Юрійовичу

1. Тема дисертації: Автоматизація звітності про кіберінциденти з використанням LLM-експерта,

Науковий керівник дисертації к.т.н., доцент Стьопочкіна Ірина Валеріївна,

затверджені наказом по університету від 4 листопада 2025 р. № 4797-с

2. Термін подання студентом дисертації 16 грудня 2025 р.

3. Об'єкт дослідження: обмін інформацією про загрози

4. Вихідні дані до роботи: формалізовані моделі звітів згідно рекомендацій України та ЄС, програмне рішення автоматизації створення PDF звітів та подання індикаторів компрометації до MISF

5. Завдання дослідження: 1. Дослідити рекомендації та вимоги законодавства України та ЄС щодо звітності про кіберінциденти. 2. Побудувати моделі звітів на основі рекомендацій для подальшої автоматизації. 3. Створити векторну базу даних із стандартами для збагачення запиту контекстом. 4. Створити програму для генерації PDF звітів та подання подій у MISF на основі створених моделей. 5. Порівняти популярні LLM в задачі створення звіту на основі опису інциденту.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу (із зазначенням плакатів, презентацій тощо): Презентація на тему: «Автоматизація звітності про кіберінциденти з використанням LLM-експерта»

7. Орієнтовний перелік публікацій: Андреев Д.Ю., Стьопочкіна І.В., Бібіков А.Ю. АВТОМАТИЗАЦІЯ ЗВІТНОСТІ ПРО КІБЕРІНЦИДЕНТИ З ВИКОРИСТАННЯМ LLM ЕКСПЕРТА / Міжнародна мультидисциплінарна наукова конференція «Світ наукових досліджень». Випуск 45.

8. Дата видачі завдання: 20 вересня 2025 року

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Визначення тематики роботи	05.09.2025 - 15.09.2025	виконано
2	Отримання завдання	20.09.2025	виконано
3	Опрацювання законів та стандартів України	25.09.2025 - 09.10.2025	виконано
4	Опрацювання рекомендацій ЄС	10.10.2025 - 23.10.2025	виконано
5	Створення моделей звітів на основі рекомендацій	25.10.2025 - 17.11.2025	виконано
6	Написання програми та інтеграція з LLM	18.11.2025 - 30.11.2025	виконано
7	Тестування рішення з різними моделями	30.11.2025 - 03.12.2025	виконано
8	Створення ілюстративних матеріалів	04.12.2025 - 14.12.2025	виконано
9	Передзахист роботи	17.12.2025	виконано
10	Захист роботи	24.12.2025	

Здобувач вищої освіти

_____ (підпис)

Данило Андреев

Керівник роботи

_____ (підпис)

Ірина Стьопочкіна

РЕФЕРАТ

Робота складається з трьох розділів, містить 144 сторінки, 26 ілюстрацій, 13 таблиць, 1 додаток та 34 посилання на джерела.

Метою даної роботи є автоматизація звітності за кіберінцидентами, з метою обміну даними згідно вимог Threat Intelligence.

Методи дослідження включають в себе аналіз діючих стандартів звітності про кіберінциденти України та ЄС, виділення ключових вимог та рекомендацій із цих документів, аналіз сучасних методів роботи з великими мовними моделями.

Результатом роботи стало програмне рішення автоматизації звітності, а саме генерування PDF звітів для подання на CERT та CSIRT, а також створення автоматичне створення подій з індикаторами компрометації на платформі MISP з підтримкою всіх сучасних та популярних мовних моделей.

Створені розробки допоможуть компаніям полегшити приєднання до спільнот обміну інформацією про загрози, а також до взаємодії з державними структурами, що відповідають за реагування на загрози. За допомогою автоматизованого рішення організації зможуть значно зменшити витрати часу та ресурсів на інтеграцію культури обміну даними у свої процеси.

Ключові слова: Threat Intelligence, обмін даними про загрози, CERT, MISP, звітність про кіберінциденти

ABSTRACT

The work consists of three sections, contains 144 pages, 26 illustrations, 13 tables, 1 appendix and 34 references to sources.

The purpose of this work is to automate reporting on cyber incidents for the purpose of data exchange in accordance with Threat Intelligence requirements.

Research methods include analysis of current standards for reporting cyber incidents in Ukraine and the EU, identification of key requirements and recommendations from these documents, and analysis of current methods for working with large language models.

The result of the work was a software solution for reporting automation, namely the generation of PDF reports for submission to CERT and CSIRT, as well as the automatic creation of events with indicators of compromise on the MISP platform with support for all modern and popular language models.

The developed solutions will help companies facilitate joining threat intelligence sharing communities and interacting with government agencies responsible for responding to threats. With the help of an automated solution, organizations will be able to significantly reduce the time and resources spent on integrating a data sharing culture into their processes.

Keywords: Threat Intelligence, threat data sharing, CERT, MISP, cyber incident reporting

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	7
ВСТУП.....	9
1 ОГЛЯД МЕТОДІВ, ТЕХНОЛОГІЙ ТА СТАНДАРТІВ ОБМІНУ ІНФОРМАЦІЄЮ ПРО ІНЦИДЕНТИ.....	12
1.1 Значення та досвід обміну інформацією про інциденти.....	12
1.2 Методи та інструменти обміну даними між організаціями.....	21
1.3 Національні стандарти та рекомендації щодо ведення звітності про кіберінциденти.....	38
1.4 Рекомендації, фреймворки та документи ЄС, що регулюють звітність про кіберінциденти.....	44
Висновки до розділу 1.....	52
2 ПРОЕКТУВАННЯ ЗАСТОСУНКУ ТА СТВОРЕННЯ МОДЕЛЕЙ ЗВІТІВ.....	54
2.1 Визначення проблеми.....	55
2.2 Створення моделей звітів на основі вимог законодавства.....	56
2.3 Застосування LLM для уніфікації даних.....	73
2.4 Аналіз існуючих рішень.....	76
2.5 Компоненти застосунку.....	77
2.6 Побудова архітектури застосунку.....	82
2.7 Оцінка результатів роботи застосунку.....	86
Висновки до розділу 2.....	90
3 РЕАЛІЗАЦІЯ РІШЕННЯ СТВОРЕННЯ ЗВІТІВ ТА MISR ПОДІЙ З ВИКОРИСТАННЯМ LLM ТА MCR.....	91
3.1 Програмна реалізація застосунку.....	91
3.2 Застосування автоматизації на тестових інцидентах та оцінювання результатів.....	104
3.3 Порівняння ефективності та вартості моделей.....	109
Висновки до розділу 3.....	111
ВИСНОВКИ.....	113
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	115
ДОДАТОК А.....	120

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

TI (Threat Intelligence) — процес збору аналітики про загрози, тактики та техніки атакуючих

IoC (Indicator of Compromise) — індикатор (IP-адреса, хеш, домен), присутність якого вказує на потенційну компрометацію системи

LLM (Large Language Model) — велика мовна модель штучного інтелекту для генерації тексту

RAG (Retrieval-Augmented Generation) — генерація даних з урахуванням отриманого ззовні контексту

MISP (Malware Information Sharing Platform) — open-source платформа обміну даними про загрози

STIX (Structured Threat Information Exchange) — один з форматів структурованого опису загроз

TAXII (Trusted Automated Exchange of Intelligence Information) — протокол для передавання даних про загрози, часто в форматі STIX

CYBEX (The Cybersecurity Information Exchange Framework) — фреймворк для уніфікації обміну даними про загрози

CERT (Computer Emergency Response Team) — команда реагування на кіберінциденти, як правило національна

CSIRT (Computer Security Incident Response Team) — команда реагування на інциденти кібербезпеки

AI (Artificial Intelligence) — штучний інтелект

TLP (Traffic Light Protocol) — маркування даних з метою позначення, як їх можна поширювати

SIEM (Security Information and Event Management) — інструмент для збор та обробки логів з багатьох джерел

SOC (Security Operation Center) — центр реагування на інциденти

TTP (Tactics, Techniques and Procedures) — опис поведінки, типовий тактик та інструментів зловмисників

MCP (Model Context Protocol) — новий протокол для уніфікації взаємодії між LLM та зовнішніми інструментами

IDS/IPS (Intrusion Detection/Prevention System) — системи виявлення та блокування атак в мережі

OSINT (Open-source Intelligence) — розвідка з відкритих джерел інформації

EDR (Endpoint Detection and Response) — інструмент для виявлення та запобігання загрозам на хостах (пристроях)

ВСТУП

Важливою складовою успішного incident response є Threat Intelligence. Цей термін включає в себе проактивну розвідку з відкритих джерел, збір індикаторів компрометації та атаки, їхнє «збагачення» (enrichment) — тобто знаходження додаткових фактів та контексту щодо певного індикатора. Правильно зібраний та оформлений контекст дозволяє аналітикам та іншим фахівцям ефективніше працювати з індикаторами компрометації при розслідуванні інцидентів, підвищуючи швидкість та якість обробки.

Важливим складником Threat Intelligence є обмін інформацією про інциденти між організаціями та культура обміну даними загалом. Для цього використовуються спеціалізовані платформи, зокрема MISP. Також до цього можна віднести ведення звітності про кіберінциденти, для подальшого аналізу та відправки цих звітів до уповноважених структур, таких як CERT-UA.

Проте організації часто нехтують обміном даними, через що витрачають зайві кошти на підписки провайдерів threat intelligence. Причиною цього є необхідність витрат часу, коштів та людських ресурсів на впровадження культури обміну даними у власні процеси. Проте стрімкий розвиток LLM та AI-агентів з наданням ШІ повноцінних інструментів може значно скоротити витрати на звітність та ведення власного MISP, спонукаючи підсилення колективного кіберзахисту.

Актуальність роботи: компанії часто нехтують або відкладають впровадження MISP та обміну даними про загрози до своїх процесів, тому що це вимагає значних зусиль, в тому числі адаптації власних процесів та інструментів. Крім того, якщо організація хоче відправляти дані наприклад до CERT, вона також повинна дотримуватися вимог законодавства щод звітності про кіберінциденти. Тому автоматизація звітності та подання подій у MISP згідно чинного законодавства та рекомендацій є актуальною задачею.

Мета дослідження: автоматизація звітності за кіберінцидентами, з метою обміну даними згідно вимог Threat Intelligence.

Мета досягається за умови вирішення таких завдань:

1. Дослідити рекомендації та вимоги законодавства України та ЄС щодо звітності про кіберінциденти.
2. Побудувати моделі звітів на основі рекомендацій для подальшої автоматизації.
3. Створити векторну базу даних із стандартами для збагачення запиту контекстом.
4. Створити програму для створення PDF звітів та подання подій у MISP на основі створених моделей.
5. Порівняти популярні LLM в задачі створення звіту на основі опису інциденту.

Об’єкт дослідження: обмін інформацією про загрози

Предмет дослідження: методи автоматизації процесів звітності за інцидентами безпеки із застосуванням LLM та AI-агентів

Методи дослідження: аналіз діючих стандартів та рекомендацій щодо звітності про кіберінциденти України та ЄС, документації по роботі з LLM та бібліотекою Langgraph, тестування та порівняння отриманих результатів для різних мовних моделей.

Новизна одержаних результатів: створення архітектурного рішення автоматизації подання звітності у форматі PDF, а також MISP подій на основі вимог законодавства, що надаються в якості контексту за допомогою векторної БД, розробка унікального HTML парсера, створеного спеціально під структуру законів, розміщених на порталі Верховної Ради України.

Практичне значення одержаних результатів: спрощення впровадження культури обміну даними для організацій за допомогою автоматизованого рішення, покращення процесів Threat Intelligence та економія коштів на платні підписки.

Апробація результатів роботи: результати роботи представлено на Міжнародній мультидисциплінарній науковій конференції «Світ наукових досліджень». Випуск 45.

Публікації: результати роботи було опубліковано у збірнику матеріалів Міжнародної мультидисциплінарної наукової конференції «Світ наукових досліджень». Випуск 45.

1 ОГЛЯД МЕТОДІВ, ТЕХНОЛОГІЙ ТА СТАНДАРТІВ ОБМІНУ ІНФОРМАЦІЄЮ ПРО ІНЦИДЕНТИ

1.1 Значення та досвід обміну інформацією про інциденти

1.1.1 Поняття і типи Threat Intelligence та його місце в системі кібербезпеки

У зв'язку із збільшенням обсягів логів та кількості алертів в SIEM системах, підвищенням складності атак, SOC центри активно впроваджують Threat Intelligence для боротьби з цими проблемами. При чому якість впровадження може сильно відрізнятися в різних організаціях від простого збору індикаторів компрометації та використання онлайн сервісів до виділення окремої команди для аналізу потенційних загроз та підготовки до них.

Але проблема полягає в тому, що більшість команд використовують можливості Threat Intelligence вкрай обмежено, не розкриваючи потенціал цього інструменту. Найчастіше все зводиться до використання онлайн ресурсів для перевірки індикаторів, що трапляються в логах, а також збору та зберігання власних індикаторів з відкритих джерел або платних закритих фідів від різноманітних вендорів.

Ключове значення ТІ зростає і на фоні системних проблем, із якими стикаються фахівці у сфері кібербезпеки. До них можна віднести збільшення складності способів злому, постійний потік інцидентів, що призводить до так званого “alert fatigue” у SOC центрах, а також дефіцит кваліфікованих працівників з кіберзахисту.

Навіть збільшення фінансування не вирішує усіх проблем, тому що на практиці компанії стикаються з іншими перешкодами [1]:

- У 75% випадків для реагування не вистачає кваліфікованих кадрів;
- У 44% випадків інциденти, що надходять, залишають без повноцінного аналізу.

Крім того, із розвитком технологій та автоматизованих систем, зростає і складність кібератак, і, відповідно, складність їх детектування та впровадження засобів захисту. Компаніям важко покривати усі можливі сценарії та вектори атак, використовуючи лише реактивний підхід. А також розпорошуючи увагу на всі потенційні загрози, без урахування ймовірності їх застосування зловмисниками.

ТІ у цьому контексті можна розглядати як сукупність знань, яка дозволить захисникам застосовувати проактивний підхід, правильно пріоритизувати загрози, та вибудовувати захист, спираючись на знання про актуальні і реальні загрози для конкретної компанії або організації. Використовуючи на ці дані, аналітики формують контекст, який включає наступну інформацію [2]:

- Хто здійснює атаку
- Якою є мета та мотивація зловмисників
- Які технічні засоби використовуються
- Якими є тактики, техніки та процедури зловмисників (TTP), в тому числі по MITRE ATT&CK
- На які індикатори компрометації (Indicators of Compromise) необхідно звертати увагу під час моніторингу

На основі цього фахівці зможуть приймати виважені рішення, спрямовані на захист інфраструктури. Причому користь від використання ТІ отримує не тільки SOC центр. Інші процеси, які включає кіберзахист, також можуть покращувати результати та пріоритизувати напрямки роботи більш ефективно за допомогою ТІ, наприклад Vulnerability Management. ТІ дає зовнішній контекст та аналітику, щоб відрізнити реальні загрози для конкретної компанії від загальних потенційних ризиків.

Результати, отримані в ході Threat Intelligence можуть бути цінними для будь-яких співробітників у сфері інформаційної безпеки.

ТІ в залежності від масштабу розподіляють на 3 види - стратегічний, тактичний та оперативний [3]. Вони відрізняються джерелами даних для вивчення, цільовою аудиторією та форматом інформації, що надається.

Поділ на типи обумовлено тим, що для різних функцій інформаційної безпеки необхідні різні рівні та масштаби аналізу, різні підходи до отримання інформації про цілі та засоби зловмисників. При цьому ТІ повинен залишатися прикладним та орієнтованим на практичне застосування. Наприклад, дані, що необхідні командам моніторингу відрізнятимуться від тих, що потребує керівник з інформаційної безпеки (CISO) для прийняття стратегічних рішень. Отже, форма та зміст аналітичної інформації будуть також видозмінюватись.

Оперативний Threat Intelligence

Включає інформацію про поточні кібератаки, що надходять в інцидентах, які відбуваються в компаніях. Зібрана аналітика забезпечує підрозділи захисту даними, які допомагають їм зрозуміти характер того, що відбувається, якою є мета конкретної атаки та які її часові характеристики. Найчастіше ці відомості генеруються за допомогою технічних засобів [4].

Оперативний ТІ також іноді називають технічним. Це пов'язано з характером наданих даних. Інформацію використовують працівники, які безпосередньо захищають інформаційну систему. Наприклад, адміністратори чи співробітники підрозділу кіберзахисту.

Завдяки оперативному Threat Intelligence підвищується ефективність засобів захисту, що використовуються, а також процесів реагування на загрози, скорочується час відповіді на інциденти.

Тактичний Threat Intelligence

Тактичний ТІ представляє собою інформацію, яка допомагає командам безпеки виявляти загрози в своїх середовищах і реагувати на них. Він зосереджується на поточних загрозах і надає дані про нові кампанії, нову інфраструктуру зловмисників та поширені варіанти шкідливого програмного забезпечення. Тактична розвідка доповнює стратегічний ТІ, який більше орієнтований на довгострокові ризики та прийняття рішень.

Тактичний ТІ передбачає збір, обробку та використання конкретних індикаторів компрометації:

- Хеші файлів;

- Доменні імена;
- IP-адреси.

Фахівці з кібербезпеки можуть використовувати їх, щоб виявляти сліди несанкціонованої активності. Цей вид Threat Intelligence активно застосовують для проактивного пошуку загроз, реагування на інциденти, мережевого аналізу.

Стратегічний Threat Intelligence

Цей вид аналітики дозволяє узагальнити інформацію та скласти уявлення про ландшафт загроз для конкретної компанії. На основі таких даних керівники можуть приймати управлінські рішення щодо стратегії подальшого розвитку та пріоритетних напрямків кіберзахисту організації. Дані надходять у вигляді аналітичних звітів та оглядів, які формуються не лише за допомогою технічних засобів, а й за участю експертів-аналітиків.

Необхідність включення до процесу людини пов'язана з тим, що лише експерт може оцінити нові тактики злову, ТТР зловмисників. Причому зробити це з погляду на засоби захисту, що використовуються для конкретної інфраструктури. Для цього потрібне аналітичне мислення, здатність до інтерпретації даних. Деякі етапи і процеси можна автоматизувати, але вирішальну роль відіграє експерт.

Ефективний стратегічний Threat Intelligence дозволяє розуміти ризики, що виникають під час прийняття управлінських рішень, виявляти закономірності у тактиці атакуючих, враховувати геополітичні події та інші чинники.

З яких джерел можна отримувати інформацію для стратегічного ТІ:

- нормативно-правові документи державних органів та міжнародних організацій;
- засоби масової інформації, публікації у галузевих виданнях, експертні коментарі фахівців;
- аналітичні та дослідні звіти, підготовлені організаціями, що займаються інформаційною безпекою.

Автоматизація може використовуватися на всіх рівнях threat intelligence, в залежності від процесів та засобів, що наявні в конкретній організації.

1.1.2. Джерела Threat Intelligence та їх класифікація

Для розуміння спектру завдань, які можна вирішити за допомогою ТІ, необхідно визначити джерела даних, з яких фахівці можуть збирати дані та індикатори. Виділяють кілька ключових категорій, які зображено на рис. 1.1 [5].



Рисунок 1.1 - Типи джерел Threat Intelligence [6]

Відкриті джерела (OSINT)

Відкритими вважаються будь-які загальнодоступні джерела даних, включаючи:

- новинні ресурси;
- особисті блоги;
- соціальні мережі;
- тематичні форуми;
- відкриті бази даних.

У 2023 році джерелом інформації для ТІ в більшості випадків були відкриті зовнішні джерела. Наприклад, публікації у засобах масової інформації чи на професійних сайтах.

Сьогодні аналітики можуть використовувати велику кількість відкритих фідів (джерел) з індикаторами, які часто є доступними навіть для комерційного застосування. При впровадженні TIR (платформи інформаційної безпеки) компанія може обробляти великі масиви IoC із мінімальними зусиллями. Такий підхід сприяє появі колективної аналітичної діяльності та культури обміну даними, а також дозволяє автоматизовувати процеси за рахунок стандартизованих даних, які отримують у спеціальних форматах, найпопулярнішими серед яких є STIX та OpenIoC.

Внутрішній ТІ (In-House Intelligence)

Внутрішню аналітику формують команди реагування на інциденти, співробітники центрів моніторингу (SOC) та штатні аналітики організації. Джерелами даних є засоби моніторингу (SIEM, EDR), у тому числі журнали подій файрволів, системи виявлення вторгнень (IDS) та інші інструменти.

Комерційні сервіси Threat Intelligence

У 2022 році майже 80% компаній використовували ті чи інші платні послуги для отримання даних для ТІ. Такі платформи надають агреговані та постійно оновлювані дані про виявлені загрози, вразливості систем та кампанії атак.

Дані на комерційній основі надають десятки постачальників, потрібно лише підбирати їх в залежності від потреб організації. Найкращі вендори надають дуже якісні відомості, які гарантують мінімальну кількість помилкових спрацювань, особливо у порівнянні з відкритими джерелами інформації.

Аналітика загроз у даркнеті (Dark Web Intelligence)

Даркнет є частиною Інтернету, яку не індексують звичайні пошукові системи. Його часто використовують для нелегальної діяльності, але через це він має користь і для фахівців кіберзахисту. Джерелом даних можуть бути

спеціалізовані закриті форуми та торгові майданчики, куди можна отримати доступ за допомогою спеціального програмного забезпечення. [8]

Проте в більшості випадків організації не розкривають весь потенціал Threat Intelligence, зупиняючись лише на кількох найпопулярніших відкритих джерелах. Наприклад, згідно з результатами опитування SANS Institute, більшість респондентів використовують ТІ для блокування шкідливих доменів та ІР-адрес, а також для поповнення власних баз даних, які застосовуються при розслідуванні інцидентів. Способи використання ТІ компаніями наведено на рис. 1.2.

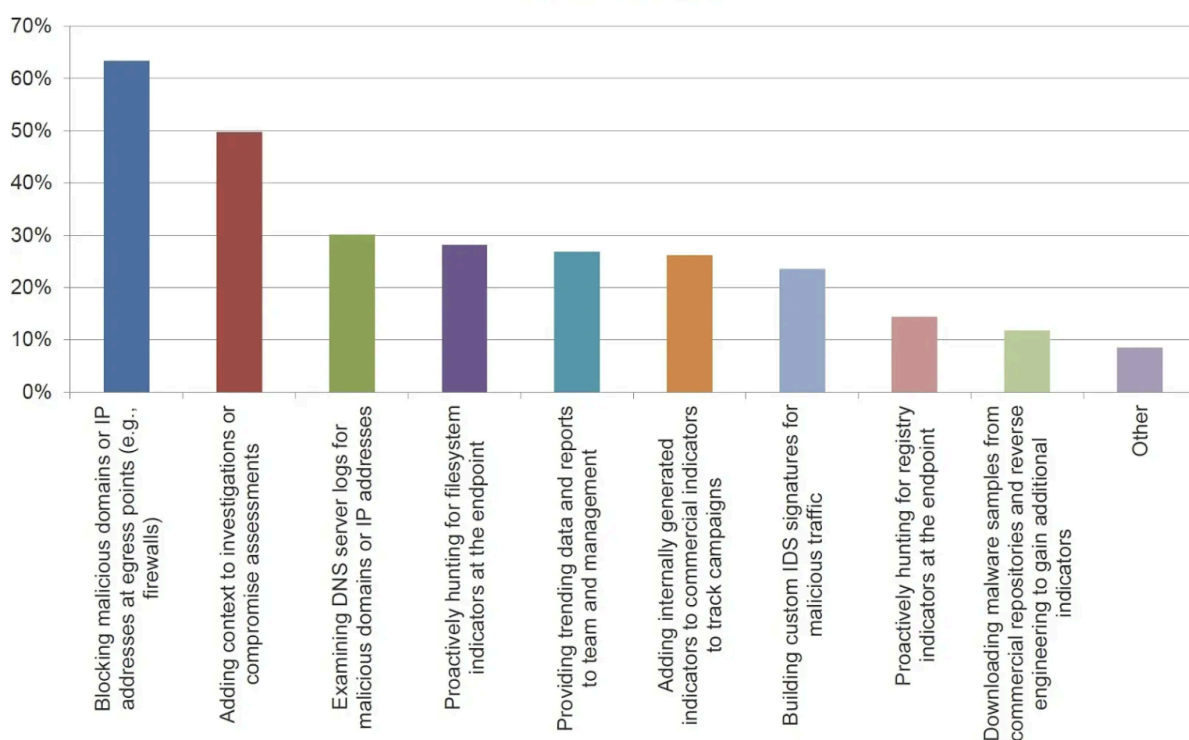


Рисунок 1.2 - Як компанії використовують ТІ [8]

Але при цьому лише деякі виходять за межі базових сценаріїв, що видно з графіка. Причин тому кілька:

- Якщо компанії використовують обмежений набір потоків індикаторів компрометації, вони так і будуть їх використовувати, не розкриваючи справжній потенціал ТІ.

- Більш складні і, як наслідок, більш корисні форми використання аналітики загроз сприймаються як важкодоступні з точки зору отримання значущих даних та їх практичного застосування.

У багатьох випадках впровадження ТІ починають із встановлення базової платформи та використання відкритих джерел. Такий підхід найменш витратний та зручний на початковому етапі.

Однак ТІ, який ґрунтується на одному типі джерела, приносить мінімум результатів. Підрозділам захисту доводиться використовувати реактивну модель, обробляючи оповіщення, що надходять, без контексту.

Основна проблема в тому, що у компанії найчастіше застосовують безперервні, але розрізнені потоки індикаторів, через які багато переваг ТІ стають недоступними.

1.1.3. Роль Threat Intelligence в системах моніторингу та реагування

Інтеграція Threat Intelligence з системами моніторингу (SIEM, EDR, IDS тощо) дозволяє організаціям посилити кіберзахист своєї інфраструктури. ТІ надає інформацію про потенційні загрози, а SIEM пропонує централізовану платформу, за допомогою якої можна проводити моніторинг, аналізувати та реагувати на інциденти. Вони працюють у синергії, що розширює можливості для виявлення, потенційних загроз та відповіді на них.

Рішення SIEM дозволяють збирати, аналізувати та зіставляти дані відразу з багатьох джерел логів. З їхньою допомогою можна моніторити та аналізувати події безпеки в реальному часі. Цей інструмент дозволяє оперативно виявляти та реагувати на інциденти.

Інтеграція SIEM, що використовується в компанії, з інструментами Threat Intelligence може принести ряд переваг [9]:

Поліпшене виявлення загроз

Інтеграція TI із SIEM розширює можливості аналітиків. Потоки даних надають системам SIEM актуальну інформацію про відомі індикатори компрометації та дозволяють точно виявляти підозрілі дії. У результаті фахівці можуть зіставити внутрішні події із зовнішніми даними про загрози.

Наприклад, SOC аналітик бачить спрацювання правила детектування, яке сигналізує про кілька невдалих спроб входу до облікового запису з певної IP адреси. Без threat intelligence його можна віднести до незначних подій. Але інтеграція дозволяє визначити, що IP-адреса пов'язана з ботнетом, що підвищує серйозність інциденту і вимагає оперативного реагування.

Проактивний захист

Проактивний аналіз даних, отриманих за допомогою threat intelligence дає змогу організаціям передбачати загрози та атаки, з якими вони можуть стикнутися в найближчій перспективі. Таке прогнозування дає змогу застосовувати превентивні заходи, у тому числі змінювати налаштування фаєрволів, усувати вразливості чи посилювати моніторинг високоризикових ресурсів (assets) організації.

Наприклад, результат аналізу TI вказав на підвищення кількості атак типу Ransomware. Detection інженери можуть використати отримані дані та IoC для створення додаткових правил реагування з метою збільшення ймовірності успішного виявлення таких атак.

Поліпшене реагування на інциденти

Зв'язок TI із SIEM оптимізує процеси захисту та реагування на події. У разі виникнення інциденту аналітична платформа надає контекстну інформацію про загрозу, у тому числі вказує на ймовірне джерело атаки, потенційний вплив та техніки, що використовуються. Завдяки аналізу у контексті можна ефективніше відреагувати на інцидент, мінімізувати його наслідки.

Комплексне розуміння загроз

Інтеграція TI та SIEM дозволяє повніше і глибше зрозуміти ландшафт загроз. В результаті можна досягти комплексної прозорості, що допомагає організації виявити та нейтралізувати атаки, які в іншому випадку могли

залишитися непоміченими. Якщо постійно підтримувати SIEM в актуальному стані, організація завжди залишатиметься в курсі потенційних небезпек і може налаштовувати свій захист відповідно.

Наприклад, фахівці центрів моніторингу можуть корелювати між собою різні інциденти, що з'являються в організації і будувати загальну картину на основі інформації з ТІ, що дозволить виявити складну комплексну атаку, що задіює одразу декілька векторів, технік та технічних засобів.

1.2 Методи та інструменти обміну даними між організаціями

1.2.1 Автоматизовані платформи обміну кіберзагрозами

Для ефективного захисту від кіберзагроз критично важливо обмінюватися даними про них між організаціями. Цю функцію беруть на себе автоматизовані платформи обміну інформацією про загрози. Одним з найбільш популярних інструментів є MISP або Malware Information Sharing Platform. З його допомогою можна збирати і зберігати дані, а також відправляти їх іншим зацікавленим учасникам всередині організації і за її межами.

MISP є платформою з відкритим вихідним кодом. Її використовують відділи із захисту від кіберзагроз для взаємодії між співробітниками, іншими підрозділами та компаніями. З її допомогою можна поширювати індикатори компрометації та контекстну інформацію про них, щоб підвищити ефективність виявлення атак. Основні можливості MISP [10]:

- Обмін даними про загрози з перевіреними спільнотами та зовнішніми організаціями;
- Виявлення взаємозв'язків між індикаторами компрометації для глибокого аналізу та розуміння інцидентів;
- Інтеграція з іншими інструментами безпеки, в тому числі SIEM для автоматизації управління IoC;

- Візуалізація процесів за допомогою панелей моніторингу, що спрощує відстеження активності та визначення тенденцій;
- Можливість налаштування інструменту під себе, в тому числі шляхом модифікації відкритого вихідного коду.

MISP широко застосовується фахівцями завдяки наступним перевагам:

- Централізоване управління отриманими даними про загрози з можливістю їх структурування та аналізу;
- Безпечний обмін інформацією про інциденти за допомогою механізмів детального обміну та синхронізації в онлайн-режимі;
- Можливість зіставлення індикаторів компрометації та автоматизації робочих процесів з метою підвищення ефективності виявлення, аналізу та реагування на події;
- Легка інтеграція з існуючою IT-інфраструктурою через API, модульні розширення та завдяки підтримці популярних форматів обміну даними (STIX або власний від MISP).

Фахівці отримують досить гнучку і зручну у використанні платформу, яка дозволяє організувати взаємодію між будь-якими підрозділами та організаціями для колективного захисту від кіберзагроз. Збережені дані можна надсилати колегам та партнерам, генерувати за допомогою них правила для популярних IDS (Snort, Suricata, Bro, Zeek), експортувати у STIX, OpenIOC, CSV.

Завдяки такому обміну виключається ситуація, при якій одна група аналітиків проводить розбір інциденту і розуміє, що колеги або інша організація вже працювали над подібною загрозою і підготували відповідний звіт.

При додаванні нових даних в MISP аналітик відразу бачить взаємозв'язок з іншими індикаторами та сутностями, що збільшує ефективність його аналізу. Фахівець отримує повне уявлення про TTP, пов'язані атрибуції та кампанії. Передбачена функція, яка дозволяє обговорювати інцидент в рамках платформи.

Альтернативним варіантом може бути OpenCTI. Це також платформа з відкритим вихідним кодом, яка допомагає керувати даними про загрози, візуалізувати та реагувати на інциденти. Цей сервіс надає командам захисту необхідний контекст, взаємозв'язки та значення кожного елемента — IP-адреси, типу шкідливого ПЗ, методу атаки тощо.

1.2.2 Протоколи та формати передавання даних у системах кіберзахисту

Протоколи та формати передачі інформації у сфері кібербезпеки ще перебувають на стадії формування та стандартизації, але найбільш популярними та актуальними у 2025 році є стандарти MISP та STIX. Існують також менш поширені варіанти, які використовуються досить рідко.

Формат STIX використовується вже більше 10 років і на даний момент актуальною є версія 2.1, хоча в 2026 році планують прийняти версію 3.0, яка зробить його ще більш гнучким і зручним для фахівців з кібербезпеки. Стандарт підтримує OASIS, до складу комітету цієї організації з Threat Intelligence входить близько 50 компаній, які давно працюють у сфері ТІ. Як наслідок, в ньому втілені найкращі практики досвідчених фахівців у цій сфері. З його допомогою можна:

- Детально описати загрозу;
- Вказати всі взаємозв'язки;
- Відзначити технічні артефакти.

При цьому отриманий результат може аналізувати як людина вручну, так і різні засоби автоматизації.

STIX є стандартизованою мовою, за допомогою якої обмінюються даними в ТІ [11]. Він вводить набір сутностей і взаємозв'язки між ними. У специфікаціях його творці називають протокол *serialize-agnostic*, проте фактично застосовується формат JSON. В якості транспорту можна використовувати будь-який інструмент,

але OASIS пропонує використовувати стандарт TAXII з протоколом HTTPS в якості фундаменту.

У STIX дані описані у вигляді графа, де в якості вузлів використовуються об'єкти SDO (Domain Objects), а ребрами є SRO (Relationship objects). Стандарт вводить наступні сутності [12]:

- Схема атаки або Attack pattern;
- Шкідлива кампанія або Campaign;
- План дій або Course of action;
- Особистість або Identity;
- Індикатор або Indicator;
- Набір поведінкових ознак або Intrusion set;
- Шкідливе ПЗ або Malware;
- Об'єкт спостереження або Observed data;
- Звіт або Report;
- Зловмисники або Threat actor;
- Інструмент або Tool;
- Вразливість або Vulnerability.

Завдяки набору зазначених сутностей і визначенню взаємозв'язків між ними фахівець відділу кібербезпеки може зрозуміло описати дані, придатні для подальшої автоматизації за допомогою стороннього ПЗ. Далі наведено декілька прикладів опису індикаторів та загроз за допомогою STIX.

У цьому прикладі описано індикатор компрометації <http://x4z9arb.cn/4712/> та його атрибуцію зі шкідливим програмним забезпеченням x4z9arb backdoor. Роль індикатора відіграє сайт, з якого завантажується ПЗ. Аналітик на основі цієї інформації може виявити ІоС в інфраструктурі компанії і зробити висновок про те, що в неї міг потрапити вірус. Подальші дії він може планувати виходячи з цієї інформації і на основі загальнодоступних даних, які є у відкритих БД. Граф наведено на рис. 1.3.

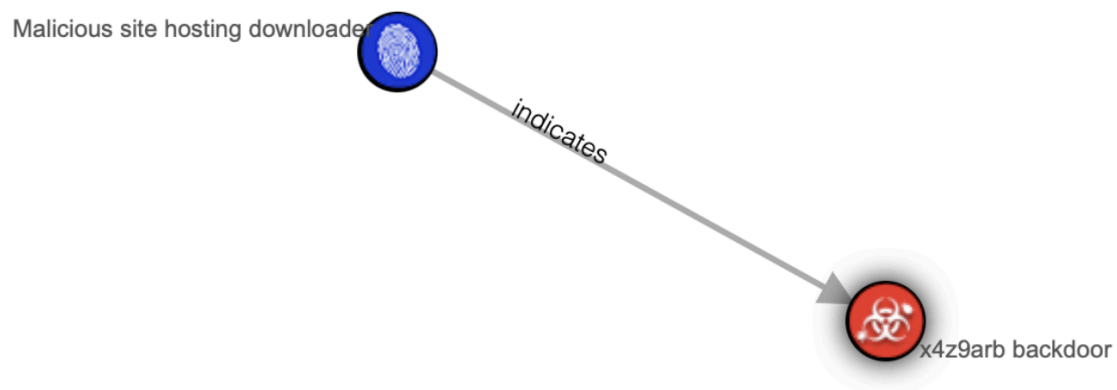


Рисунок 1.3 - Опис індикатора у STIX [13]

У цьому прикладі чітко видно зв'язок між Threat actor на ім'я Adversary Bravo, використовуваним ним Attack pattern — фішингом і шкідливим програмним забезпеченням Poison Ivy Variant d1c6. Якщо аналітик бачить наявність вірусу в інформаційній інфраструктурі, він розуміє, хто його використовував і яким чином він потрапив на комп'ютери користувачів. Приклад графу наведено на рис. 1.4.

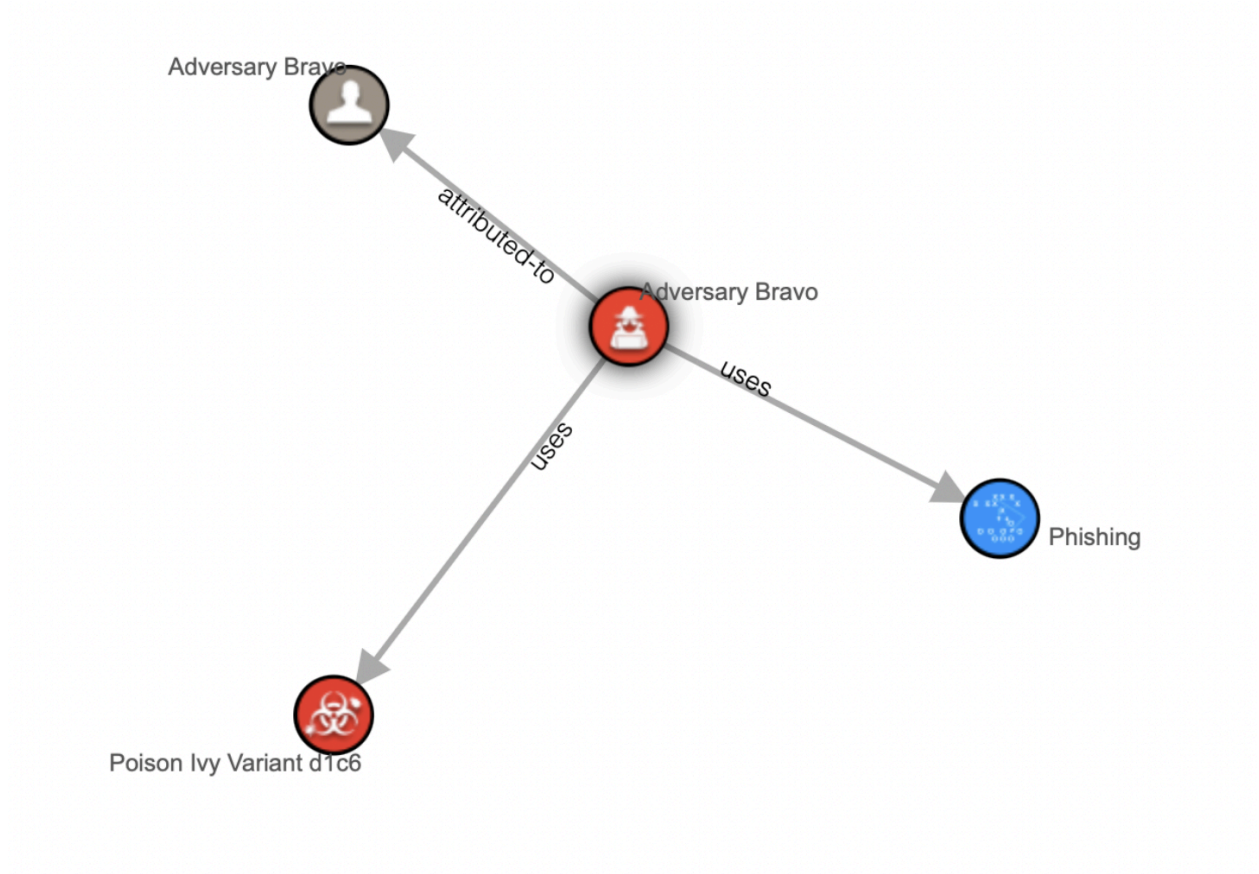


Рисунок 1.4 - Опис зловмисника у STIX [13]

У реальній роботі використовуються набагато складніші графи. Наприклад, за допомогою STIX можна описувати дуже об'ємні дослідження, в тому числі діяльність шкідливих угруповань. На рис. 1.5 представлений приклад звіту про роботу угруповання APT1.

- Атрибути події або Event attributes;
- Об'єкт або Object;
- Тег або Tag;
- Виявлення або Sighting;
- Галактика або Galaxy.

MISP дозволяє дуже детально описати загрози, для цього в ньому є безліч атрибутів, таксономії для розподілу за категоріями, галатики для кластеризації. До мінусів можна віднести відсутність достатньої строгості і типізованих зв'язків, які існують в STIX. Але це не говорить про те, що MISP чимось йому поступається.

Крім перерахованих вище де-факто стандартів галузі, є й інші формати обміну даними в Threat Intelligence. Вони передбачають створення файлів у форматах CSV або TXT. Існують і інші, але вони використовуються досить рідко. Вибір на користь STIX та MISP зроблений через їх простоту. У більшості випадків використовується мінімальний контекст або просто вказуються індикатори компрометації без опису контексту. Це не дозволяє повноцінно аналізувати події і використовувати дані в нових розслідуваннях.

Найчастіше у форматі простого тексту фахівець отримує список IP-адрес, доменів, URL та деяких інших даних. Але без контексту їм набагато складніше довіряти та інтерпретувати дані. Аналітик не зрозуміє, про які загрози свідчать індикатори, наскільки ці загрози зберегли свою актуальність і яку небезпеку становлять.

Фіди про події в CSV-форматі складаються з декількох колонок, в яких вказані типи індикаторів, значення і час. Іноді додають опис шкідливого програмного забезпечення або вразливостей, пов'язаних з конкретним індикатором компрометації. Користь від csv-залежить від ситуації.

MISP і STIX виділяються тим, що на відміну від них дані в текстовому і CSV-форматах не приведені до єдиної форми, тому їх складно пов'язувати і використовувати в засобах автоматизації. Наприклад, в текстовому файлі можуть міститися коментарі або використовуватися різні способи опису індикаторів

компрометації. Тобто парсити їх буде дуже складно. У CSV проблема парсингу полягає в неконсистентності роздільників даних або з інших причин.

Гетерогенність форматів є викликом для аналітиків, які потребують узагальненої та актуальної інформації для своєї роботи. Їм доводиться вручну вивчати джерела, виконувати попередню обробку даних, щоб не втрачати важливу інформацію. Наприклад, деякі дані не можна розмістити в полях моделі даних інструменту, що використовується.

1.2.3 CYBEX - Cybersecurity Information Exchange Framework

CYBEX — це набір стандартизованих технік обміну інформацією між різними учасниками, що працюють у сфері кібербезпеки. Він дає можливість надсилати та отримувати дані операторам IT-інфраструктури, провайдерам, розробникам програмного забезпечення тощо. У CYBEX визначено типи інформації для обміну та їхні структури. [15]

Мета створення CYBEX — стандартизувати обмін даними з урахуванням ступеня їх надійності та сумісності. Цей фреймворк створює технічну, логічну та організаційну основу для взаємодії між учасниками. Для реалізації цієї мети в CYBEX створено шість так званих exchange clusters, які відповідають за різні аспекти обміну даними [16]:

1. Weakness, Vulnerability and State — відомості про вразливості інформаційних систем, конфігурації і використовуване програмне забезпечення. Завдяки йому можна створювати загальну базу даних, в якій будуть міститися вся інформація про потенційні вразливості.
2. Event, Incident and Heuristics — дані про події, що відбулися, інциденти, що відбуваються, сигнатури та інші дані, які відображають спроби атак, компрометації систем тощо.

3. Information Exchange Policy — опис політик обміну інформацією. Включає в себе умови, обмеження, домовленості між сторонами, узгодження прав доступу і рівнів довіри, категоризацію даних.
4. Identification, Discovery and Query — способи ідентифікації організацій, джерел і типів інформації, можливості для пошуку і виявлення даних в розподілених системах. Цей кластер потрібен для спрощення навігації та пошуку у різних репозиторіях.
5. Identity Assurance — кластер забезпечення довіри. Потрібен для перевірки автентичності відправника, гарантій цілісності та незмінності даних, автентифікації користувачів і сертифікації суб'єктів, які беруть участь в обміні.
6. Exchange Protocol — транспортні протоколи, за якими відбувається обмін даними. Описані формати повідомлень, методи пакування та способи транспортування, інфраструктура маршрутизації, сумісність з іншими стандартами.

Кластери можуть виступати як самостійні одиниці або в будь-якій комбінації один з одним. Підбирати варіант використання може конкретна компанія або спільнота користувачів.

Однак фреймворк не регламентує правила отримання цієї інформації учасниками. Вона може надходити з систем моніторингу, журналів, від діяльності SOC та іншими способами. Розглянемо основні завдання, які вирішує CYBEX:

- Інтероперабельність — фреймворк усуває роз'єднаність різних систем, які можуть перебувати в різних організаціях і країнах. Він забезпечує сумісність форматів і знижує фрагментацію кіберзахисту. Це особливо важливо для створення глобальної системи обміну даними.
- Прискорення обміну даними та реагування на інциденти — стандартизація обміну інформацією дозволяє швидше передавати відомості про виявлені вразливості, щоб інші учасники могли вживати заходів захисту.

- Підвищення надійності та обміну — кластери фреймворку полегшують встановлення довірчих відносин між учасниками та гарантують автентичність даних, що передаються.
- Глобальна співпраця — за допомогою CYBEX організації з різних країн можуть обмінюватися інформацією незалежно від галузі роботи та юрисдикції. Завдяки цьому спрощується організація колективного захисту і знижується небезпека транснаціональних кіберзагроз.
- Гнучкість і розширюваність — в рамках фреймворку можна комбінувати різні техніки, розширювати кластери та інтегрувати інші стандарти, залишаючи незмінною базову модель.

Але при цьому фреймворк не вимагає обов'язкової передачі інформації. Організації беруть участь в обміні добровільно з урахуванням вимог чинного законодавства та домовленостей між учасниками. Також CYBEX не нав'язує інструменти реагування або конкретні рішення. Він лише надає інфраструктуру, на яку можуть спиратися учасники при передачі інформації один одному.

CYBEX — це перевірена система, фреймворк використовують як теоретичну і практичну основу для розробки власних платформ обміну даними про загрози, в тому числі між організаціями. Завдяки модульній і кластерній структурі він забезпечує сумісність з іншими стандартами. На основі фреймворку можна побудувати міжнародні CERT/CSIRT-спільноти, портали для обміну індикаторами, репозиторії та платформи для спільного реагування або дослідження загроз.

1.2.4 Протокол TLP

Traffic Light Protocol або TLP необхідний для того, щоб маркувати чутливу інформацію в рамках взаємодії між організаціями у сфері кібербезпеки, аналітиками, групами реагування та спільнотами. [17]

TLP не є формальною класифікацією (на кшталт «таємно», «цілком таємно», «особливої важливості»), не замінює правові регламенти. Це механізм, заснований на довірі між сторонами і добровільному дотриманні встановлених вимог.

Застосування цього протоколу полегшує обмін корисною та чутливою інформацією про інциденти, нові атаки та індикатори. Але при цьому можна обмежувати поширення даних і знизити ризик несанкціонованого доступу до них.

У чому переваги протоколу TLP:

- Стандартизація обміну. Протокол зрозуміло описує правила поширення даних. За рахунок цього спрощується обмін інформацією між організаціями, галузями та країнами.
- Безпека обміну. Одержувач точно знає, як можна використовувати і поширювати отримані дані. Наприклад, в яких випадках заборонено публікувати дані або передавати їх третім особам. В результаті можна створити безпечні канали обміну інформацією для потреб Threat Intelligence.
- Гнучкість і простота. Для використання TLP не потрібні спеціальні технічні засоби. Його застосування базується тільки на угоді між сторонами і спільній системі маркування.
- Баланс між безпекою і публічністю. Виділення декількох рівнів доступу дозволяє поширювати інформацію на конкретному рівні в залежності від її змісту, контексту, мети використання, чутливості.
- Спільні зусилля з реагування на інциденти та їх запобігання. TLP дає можливість координувати потоки інформації між організаціями і мінімізувати при цьому ризик несанкціонованого поширення службових або чутливих даних за межі спільноти, для якої вони призначені.

У протоколі TLP використовується п'ять рівнів інформації (позначені кольорами), для кожного з яких є свої правила поширення. Ці рівні встановлює джерело або відправник даних. Одержувач, у свою чергу, зобов'язується дотримуватися накладених обмежень. Типи міток TLP наведено в таблиці 1.1. [18]

Таблиця 1.1 - Мітки TLP та їх пояснення

Мітка	Призначення інформації	Умови поширення
RED	Вкрай чутливі дані, розкриття за межами встановленого кола осіб може стати причиною значної шкоди.	Призначена тільки одержувачу. Не можна ділитися даними з іншими особами без явного дозволу від власника.
AMBER+STRICT	Поширення обмежене межами організації, інформація використовується тільки для службових цілей.	Можна поширювати всередині організації-одержувача без передачі зовнішнім партнерам і клієнтам, якщо це не погоджено відправником (джерелом).
AMBER	Інформація, отримання якої вимагає відповідних дій від клієнтів. Наприклад, партнерів або одержувачів послуг.	Можна поширювати всередині організації та серед клієнтів, якщо це необхідно для вжиття заходів реагування на загрозу.
GREEN	Інформація, яка буде корисною для великої кількості партнерів, спільнот, колективних структур реагування.	Дозволено поширювати всередині довіреної спільноти, між організаціями та партнерами, але без використання публічних каналів з відкритим доступом.
CLEAR	Дані без обмежень, їх публічне поширення несе мінімальний ризик.	Обмежень на поширення немає, за винятком вимог законодавства про авторське право та інших державних нормативів.

Мітку присвоює відправник даних, виходячи з власних уявлень про можливі ризики. Одержувач повинен дотримуватися правил, але має право запросити дозвіл у джерела, щоб вийти за межі обмеження.

Потрібно зауважити, що TLP не замінює прийняту в Україні систему роботи з обмеженою інформацією, яку вводить закон «Про державну таємницю» та підзаконні нормативні акти.

Розглянемо приклади використання протоколу:

- Під час спільної роботи CERT/CSIRT та оператори інфраструктури передають один одному індикатори вразливості та звіти про інциденти. Маркування TLP мітками дозволяє іншим знати, що з переданого можна публікувати, а що використовувати тільки всередині організації.
- У процесі Threat Intelligence галузеві та регіональні спільноти обмінюються між собою індикаторами та контекстом, позначаючи цю інформацію TLP мітками, щоб обмежити або розширити охоплення членів спільноти без ризику витоку чутливих даних.
- При публікації звітів про вразливості, бюлетенів і рекомендацій для фахівців з кібербезпеки організації маркують дані як Green або Clear, щоб донести дані до якомога більшої кількості людей.

TLP можна використовувати з іншими правилами. Наприклад, з Chatham House Rule, щоб поширити дані, але не вказати джерело їх отримання.

Використання протоколу можливе за дотримання кількох умов:

- Всі учасники повинні бути проінформовані про правила TLP, рівні та зобов'язання, а також розуміти, що мітка є обов'язковою до виконання згідно з домовленістю між членами спільноти.
- Мітку необхідно наносити на інформацію явно в заголовок повідомлення або перед початком тексту, щоб одержувач відразу розумів, яким чином її можна поширювати.
- Інформацію можна поширювати на ширше коло, тобто виходити за обмеження, але для цього потрібно явно запитувати дозвіл у джерела даних.

TLP — це один з основних інструментів, за допомогою якого можна створити безпечну та гнучку мережу обміну даними між учасниками спільноти. Він дозволяє окреслити межі та мінімізувати ризик витоку чутливої інформації, але при цьому й адаптувати вимоги під конкретні завдання.

Протокол є вкрай важливим для Threat Intelligence, щоб учасники могли обмінюватися індикаторами компрометації, контекстом і попередженнями про загрози. TLP дозволяє передавати ці дані, але гарантує, що вони не потраплять у відкритий доступ до того, як будуть повністю проаналізовані, узгоджені, а небезпека негативних наслідків мінімізована. Протокол систематизує та уніфікує цей процес.

1.2.5 Моделі обміну даними між організаціями

Міжорганізаційний обмін даними про загрози та інциденти вважається ключовим елементом колективної кібербезпеки. Для координації обміну даними створено багато центрів CERT або Computer Emergency Response Team, а також CSIRT або Computer Security Incident Response Team [19]. Хоча ці дві команди мають схожі завдання, але вони відрізняються сферою застосування, а також особливостями роботи. Проте обидві необхідні для обміну інформацією між організаціями, секторами та державами.

Поняття CERT виникло в 1988 році, коли в Університеті Карнегі створили команду реагування після інциденту з хробаком Morris. Пізніше цей термін став застосовуватися ширше.

Незважаючи на спільні риси, є деяка різниця в термінології:

- CERT — початковий термін, який зараз використовують для команд реагування на міждержавному, національному або галузевому рівні;
- CSIRT — загальний термін для команд реагування всередині державних або приватних організацій.

На практиці обидві аббревіатури використовують разом і замінюють ними одна одну. Основна мета CERT/CSIRT — це забезпечення своєчасного, ефективного та структурованого реагування на інциденти та створення системи обміну інформацією про загрози.

Команди CERT/CSIRT можуть мати різні організаційні форми залежно від поставлених завдань, рівнів відповідальності та масштабу організації. Зокрема:

- Корпоративні CSIRT. Їх створюють всередині компанії або установи. Вони добре знають внутрішню інфраструктуру організації, можуть швидко реагувати на загрози, адаптувати бізнес-процеси для зменшення вразливості. Але такі команди мають обмежені ресурси, не мають інформації про загрози за межами організації і не обмінюються контекстом.
- Галузеві CERT. Формуються з декількох галузевих організацій. Наприклад, у фінансовому, телекомунікаційному, енергетичному секторі. Забезпечують колективний захист і обмінюються внутрішньогалузевою інформацією, можуть уніфікувати механізми реагування та забезпечити доступ до надійної експертизи. Плюс у тому, що вони мають більше можливостей для раннього попередження загроз, оскільки інцидент в одній організації стає відомий усім учасникам.
- Національні CERT. Працюють на державному рівні і координуються міністерствами або спеціально створеними незалежними структурами. Вони потрібні для захисту критичної інфраструктури, реагування на великі інциденти, організації міжнародної взаємодії та створення національної політики безпеки. Як приклад можна навести CERT-EU або CERT-UA.
- Міжнародні та міжмережеві моделі. Забезпечують обмін даними між країнами та глобальний аналіз загроз, а також реагування на транскордонні атаки. До міжнародних спільнот належать ENISA, APCERT, TF-CSIRT, в рамках яких об'єднані сотні команд реагування.
- Гібридні та аутсорсингові CSIRT. Якщо компанія не може створити повноцінний CSIRT через обмеженість ресурсів, вона користується

послугами спільних команд, сервісів MSSP або галузевих центрів обміну інформацією. Це дозволяє отримати всі можливості CSIRT, але без великих витрат на створення власної інфраструктури.

Взаємодія CERT і CSIRT між собою та з організаціями будується на декількох принципах:

- Стандартизація обміну даними з використанням структурованих форматів даних. З їх допомогою можна автоматизувати аналіз загроз та інтегрувати отримані дані у свої SIEM системи.
- Дотримання правил конфіденційності, збереження цілісності та достовірності інформації, що передається. Це дає можливість сформувати довірчу екосистему.
- Своєчасність обміну, оперативна передача інформації про атаки та індикатори компрометації у мінімальні терміни, що дає можливість уникнути поширення загрози.
- Виявлення масштабних інцидентів для реалізації розподіленої моделі реагування, коли подія в одній організації може бути частиною великої атаки.

CERT/CSIRT можуть взаємодіяти не тільки між собою, але й з юридичними підрозділами, державними регуляторами, органами влади, провайдерами інформаційних послуг, центрами SOC. CERT/CSIRT можна назвати сполучною ланкою між технічним аналізом кіберзагроз і процесом поширення інформації про них між організаціями та галузями.

1.3 Національні стандарти та рекомендації щодо ведення звітності про кіберінциденти

1.3.1 Наказ ДССЗЗІУ №570 від 03.07.2023 «Про затвердження Методичних рекомендацій щодо реагування суб'єктами забезпечення кібербезпеки на різні види подій у кіберпросторі»

Наказ №570 від 3 липня 2023 року уніфікує процеси виявлення інцидентів, їх класифікації, фіксації та реагування. Документ розроблений відповідно до закону «Про основні засади забезпечення кібербезпеки України» і створює єдині правила для команд, які працюють у державному секторі, на об'єктах критичної інфраструктури та для інших суб'єктів, які займаються цією діяльністю. [20]

Цей документ є обов'язковим лише для команд, які захищають структуру державного управління або критично важливі інформаційні ресурси. Для інших він має рекомендаційний характер, але приватні структури користуються ним, щоб брати участь у забезпеченні загальної національної кіберстійкості. Документ виконує кілька завдань:

- Уніфікація процесів реагування на кіберзагрози в державних органах, спеціальних службах, на об'єктах критичної інфраструктури;
- Зменшення часу реакції на інциденти за рахунок чіткої формалізації дій учасників;
- Забезпечення обміну інформацією між організаціями та національним центром реагування CERT-UA;
- Підвищення готовності організацій до реагування на кібератаки, включаючи цільові та високотехнологічні;
- Створення стандарту звітності про виявлені інциденти.

У Наказі розписано повний цикл реагування на подію. Умовно документ розділений на кілька блоків, які наведено в таблиці 1.2. [20]

Таблиця 1.2 - Блоки реагування на подію з описом

Блок	Опис
Виявлення події	Які події свідчать про загрози, які індикатори потрібно відстежувати (підозріла активність, аномалії, мережеве сканування).
Класифікація подій	Події класифікують за певними критеріями, щоб розділяти інциденти на групи залежно від небезпеки, масштабу дій, ймовірності розвитку. Описується, які події вимагають оперативного реагування, а також вводиться структура повідомлення держорганів про інцидент.
Первинні заходи реагування	Рекомендації для кожного класу подій, які можуть мінімізувати збитки. Вони включають локалізацію, ізоляцію заражених систем, запобігання поширенню, збереження доказів і стабілізацію роботи критичних сервісів.
Процедура повідомлення CERT-UA та інших організацій	Порядок повідомлення CERT-UA. Встановлюється термін повідомлення, перелік обов'язкової інформації, канали зв'язку та формат. За рахунок цього забезпечується повнота і своєчасність передачі даних.
Документування інцидентів	Вимоги до ведення журналів і складання звітів, в яких фіксуються події та етапи його обробки, збираються всі артефакти і формуються звіти.
Аналіз, відновлення та запобігання атакам	Порядок аналізу причин події, оцінки збитків, усунення вразливостей, перевірки відновлення системи, оновлення політики безпеки, поліпшення захисних механізмів.

Документ інтегрований у національну систему захисту, зокрема систему державного реагування на інциденти та діяльність центру CERT-UA. Іншою його особливістю є чітка регламентація звітності, зокрема він визначає поняття інциденту, про який необхідно обов'язково повідомляти CERT-UA.

Уніфікація звітності забезпечує порівнюваність даних, автоматизацію аналізу, спрощує створення централізованих баз даних загроз і формування рекомендацій для всіх учасників системи кібербезпеки України.

1.3.2 Стратегія кібербезпеки України, затверджена Указом ПрУ № 447/2021 від 26.08.2021

Стратегія кібербезпеки, затверджена Указом Президента України 447/2021 від 26.08.2021, є документом стратегічного планування, який визначає пріоритети розвитку на найближчі роки, а також принципи і механізми реалізації поставлених завдань. [21] Документ прийнятий з урахуванням Стратегії національної безпеки, закону «Про основні принципи забезпечення кібербезпеки України», а також стратегій Європейського союзу, окремих європейських держав і НАТО.

Завдяки стратегії в Україні створюється цілісна і стійка національна система безпеки, яка забезпечує роботу національних ресурсів, захищає інтереси особистості і суспільства, підтримує соціальний і економічний розвиток, а також оборону країни, створює умови для цифрової трансформації держави.

Якщо розглядати прикладне значення документа, то він вирішує такі завдання:

1. Визначає потенційні кіберзагрози для української держави;
2. Формулює стратегічні цілі та завдання для суб'єктів системи кібербезпеки України, до якої входять державні органи, оборонний сектор, оператори критичної інфраструктури та приватний сектор;
3. Встановлює принципи розвитку кібероборони та забезпечення кіберстійкості;
4. Слугує фундаментом для розробки підзаконних нормативних актів та інших документів, у тому числі Методичних рекомендацій, введених наказом ДССЗЗІ № 570.

У документі структуровані заходи кібербезпеки навколо шести ключових цілей. Цілі являють собою наступний перелік [21]:

1. У Міністерстві оборони створюються кібервійська і розробляється план кіберзахисту як частини загального планування захисту країни, а також посилення Збройних сил України.
2. Передбачається створення переліку об'єктів критичної інформаційної інфраструктури та встановлення вимог до їх захисту, а також розвиток системи галузевого моніторингу та виявлення вразливостей, реагування на інциденти.
3. Акцент на поглибленні взаємодії правоохоронних органів, розвитку міжнародного співробітництва у сфері боротьби з кіберзлочинністю, а також вдосконаленні локального кримінального законодавства.
4. Розвиток інституційного та кадрового потенціалу, в тому числі підвищення кваліфікації кадрів у секторі безпеки, розвиток системи профільної освіти, формування національної школи експертизи та аналітики.
5. Проведення заходів, спрямованих на поширення знань про загрози серед українських громадян, бізнесу, органів влади.
6. Розвиток внутрішнього партнерства між державою і приватним сектором, а також міжнародного співробітництва. Створення реального партнерства, залучення бізнесу та професійних спільнот до вирішення завдань державного масштабу.

Документ виступає в якості нормативного документа верхнього рівня, який встановлює необхідність системного обліку та аналізу кіберінцидентів, вимогу до оперативного обміну інформацією та важливість координації зусиль усіх учасників спільноти.

1.3.3 Закон України «Про основні засади забезпечення кібербезпеки України» № 2163-VIII від 5.10.2017 із змінами від 27.03.2025

Закон України «Про основні засади забезпечення кібербезпеки України» № 2163-VIII від 5.10.2017 із змінами від 27.03.2025 є основним нормативним

документом, який встановлює правові та організаційні засади забезпечення кібербезпеки країни. [22] Він є фундаментом, на якому розробляються підзаконні нормативно-правові акти, регламенти та стратегії. Основні цілі прийняття законодавчого акту:

- Формулювання принципів державної політики у сфері кібербезпеки;
- Визначення повноважень органів влади та розподіл відповідальності між суб'єктами кібербезпеки та оборони;
- Створення правових основ для захисту інформаційної інфраструктури держави;
- Регламентація процесу реагування на кіберінциденти, в тому числі обміну інформацією та координації дій всіх учасників;
- Гарантія прав громадян на захист персональної інформації.

Закон можна назвати «Конституцією» для учасників системи кіберзахисту країни, який визначає загальні правила і механізми. А розширюють їх вже інші нормативні акти.

Згідно зі статтями цього документа до Національної системи кібербезпеки входять Державна служба спеціального зв'язку та захисту інформації (ДССЗІ), СБУ, Міноборони та Генеральний штаб, Національна поліція, Нацбанк України, інші державні органи, які захищають критично важливі дані та інфраструктуру. Кожен з цих учасників має свої завдання, ролі та зони відповідальності.

Закон вводить нове поняття «критичної інформаційної інфраструктури». До неї належать інформаційні та комунікаційні системи, порушення роботи яких вплине на національну безпеку, обороноздатність, економіку або суспільну стабільність. [22]

Суб'єкти критичної інформаційної інфраструктури (КІІ) повинні забезпечити належний захист об'єктів від кіберзагроз, впровадити для цього відповідні організаційно-технічні заходи.

Крім цього, вони мають забезпечити своєчасне інформування про інциденти уповноважені органи, а також брати участь у створенні та

функціонуванні національних систем моніторингу та обміну інформацією. У 2024 та 2025 роках до закону були внесені зміни, які уточнили критерії віднесення до КІІ та посилили вимоги щодо обміну інформацією.

Як доповнення, закон регламентує партнерство між державою та приватним сектором, а також партнерами. Його норми передбачають участь України в міжнародних ініціативах та обміні інформацією з союзниками про можливі загрози.

Закон відіграє важливу роль у забезпеченні національної кіберстійкості, оскільки на його основі формується екосистема нормативно-правових актів і визначаються ролі учасників процесів. Він систематизує регламенти реагування та запроваджує правові механізми кіберзахисту, а також сприяє формуванню національної політики в галузі кібербезпеки.

1.3.4 Постанова КМУ №299 від 04.04.2023 «Деякі питання реагування суб'єктами забезпечення кібербезпеки на різні види подій у кіберпросторі»

Постанова Кабінету Міністрів №299 встановлює нормативні вимоги для процедур реагування на інциденти. Цей документ узгоджує між собою дії всіх учасників національної системи захисту від атак на інформаційну інфраструктуру. [23] Він є обов'язковим для виконання державними органами, операторами критичної інфраструктури, оборонними компаніями, ІТ-підрозділами інших організацій.

Головне завдання документу — регламентувати порядок розслідувань інцидентів, у тому числі визначити послідовність дій фахівців.

Його прийняття мало кілька цілей:

- Уніфікувати процедуру, зокрема запровадити оцінку критичності події, способи локалізації загроз та заходи з відновлення працездатності систем;

- Створити основу для документування розслідування інцидентів, їх класифікації;
- Формалізувати вимоги до моніторингу та звітності про події, щоб зміцнити кіберстійкість держави.

У постанові, на відміну від Закону №2163-VIII та Указу № № 447/2021, наведено конкретні вимоги для суб'єктів, щоб зробити процедуру реагування прозорою, уникнути неузгоджених або несвоєчасних дій. Вона покликана підвищити ефективність розслідувань і скоротити кількість інцидентів, а також створити загальну національну систему кіберзахисту.

1.4 Рекомендації, фреймворки та документи ЄС, що регулюють звітність про кіберінциденти

1.4.1 NIST 800-61 rev 2

Документ NIST Special Publication 800-61 Revision 2 «Computer Security Incident Handling Guide» вважається де-факто стандартом і широко використовується в різних країнах світу для організації процесів реагування на інциденти в кіберпросторі. [24]

Він розроблений і опублікований Національним інститутом стандартів і технологій (NIST) і використовується як основа, на базі якої розробляють свої регламенти державні структури, оператори критичної інфраструктури, інші компанії та організації.

У посібнику описано основні принципи реагування та формалізовано порядок обробки інцидентів. Також у ньому наведено практичні рекомендації, які фахівці у сфері кібербезпеки можуть адаптувати під особливості власних технологічних процесів.

Основні цілі документа:

- Створення загальної моделі життєвого циклу реагування на інциденти;

- Опис структури команд реагування та ролей учасників у їхній роботі;
- Надання рекомендацій щодо підготовки, виявлення, аналізу, ескалації, реагування та відновлення після інцидентів;
- Створення методологічної бази, якою можуть користуватися у своїй діяльності великі організації та малі компанії;
- Створення та вдосконалення системи обміну інформацією про інциденти всередині організації та за її межами.

Документ є рекомендацією, але через свою універсальність використовується величезною кількістю команд і фактично відіграє роль галузевого стандарту.

NIST SP 800-61 Rev. 2 вводить чотириступеневу модель життєвого циклу реагування на інциденти. Вона вважається складається з таких етапів: [24]

Preparation. У стандарті цей етап названий ключовим для успішного реагування на інциденти. Кожна організація, згідно з правилами, повинна створити свою політику обробки інцидентів, розподілити ролі та повноваження учасників команди і розвивати її, впровадити моніторинг і ведення журналів. Крім того, необхідно встановити процедури обміну даними із зовнішніми компаніями, в тому числі з CERT. Також підготовчий етап включає створення класифікації інцидентів і побудову схем повідомлень інших суб'єктів.

Detection & Analysis. Ця фаза включає моніторинг журналів подій і виявлення аномалій, використання SIEM, IDS/IPS для виявлення атак, аналіз підозрілої діяльності. Також вона передбачає ідентифікацію інцидентів та їх пріоритезацію. У документі представлені детальні рекомендації для того, щоб зменшити кількість помилкових спрацьовувань і зібрати достатньо артефактів для подальшого розслідування та експертизи. Окремий акцент зроблено на централізації логування та документування подій.

Containment, Eradication & Recovery. Ця фаза передбачає заходи щодо обмеження поширення ШПЗ, ізоляцію систем, видалення шкідливих артефактів та відновлення. Фахівці повинні відновити систему до нормального стану і

провалідувати цілісність даних. NIST наполягає на балансі між швидкістю реакції на події та мінімізацією збитків з його документуванням.

Post-Incident Activity. На цій фазі фахівці аналізують першопричини інциденту, оцінюють ефективність реагування. При необхідності оновлюють політики і процеси захисту, в тому числі технічні, а також поповнюють базу знань для поліпшення майбутнього реагування. Акцентується увага на обміні інформацією із зовнішніми партнерами.

Цей документ легко адаптується для використання в будь-якій галузі та організації будь-якого масштабу. Він тісно пов'язаний з іншими документами NIST, досить деталізований і надає необхідні приклади, списки інцидентів, рекомендації щодо комунікацій та звітності. Причому в ньому зроблено акцент на процесах, що дозволяє швидко впровадити його рекомендації в роботу компанії.

Оскільки в NIST акцентують увагу на координації дій усіх учасників, він повністю відповідає сучасним вимогам щодо співпраці між організаціями та створення систем колективного кіберзахисту.

1.4.2 ENISA Technical guideline on incident reporting under the EEC

ENISA Technical Guideline on Incident Reporting under the European Electronic Communications Code (EECC) — це європейський документ, який стандартизує процедуру повідомлення про значущі інциденти у сфері кібербезпеки. [25] Технічне керівництво відіграє роль інструкції для операторів телекомунікаційних послуг і регламентує формат, структуру та критерії звітності.

У керівництві встановлені критерії класифікації, за якими потрібно відносити інциденти до істотних і інформувати про них місцевих регуляторів. До них відносяться:

- Масштаб впливу;
- Тривалість порушення роботи системи в годинах;
- Вплив на критичні сервіси — екстрений зв'язок, інтернет, голосові послуги;

- Поширеність по територіях;
- Ступінь порушення конфіденційності та цілісності послуг.

Документ вводить порогові значення, при перевищенні яких подія вважається істотною.

- Також керівництво вводить додаткову класифікацію інцидентів за їх типом:
- Компрометація активів або даних — Security Breaches;
- Відмова інфраструктури — System Failures;
- Відмови або атаки через постачальників телекомунікаційних послуг — Third-party Failures;
- Шкідливе ПЗ, експлойти та DDoS-атаки — Malicious Attacks.

Крім того, в ENISA представлена структура повідомлення про інцидент. Воно повинно складатися з таких елементів [25]:

- Initial Notification — індикативний звіт, який подають відразу і включають в нього початкові відомості про подію: її тип, активи, час виявлення та оцінку критичності.
- Intermediate Report — уточнені дані про причини події, вектор атаки, рівень впливу та уражені системи;
- Final Report — детальний звіт з розбором ситуації, аналізом причин події, застосовуваними методами реагування та порядком відновлення сервісів, рекомендаціями щодо запобігання повторним атакам.

Для зручності фахівців у технічному керівництві наведено уніфіковані терміни, завдяки чому звіти про інциденти стають зрозумілими спеціалістам у всіх країнах Європейського Союзу.

Таким чином, ENISA Technical guideline on incident reporting under the ЕЕСС створює єдиний формат звітності. Він забезпечує коректну інтеграцію суб'єктів кіберзахисту з національними центрами моніторингу. Це необхідно для оптимізації процесів управління інцидентами.

1.4.3 ENISA Incident Management Guide

Технічний посібник ENISA Incident Management Guide стандартизує процеси управління інцидентами в організаціях. Норми цього документа дозволяють створити комплексну модель incident management, яка включає в себе технічні, організаційні та процедурні компоненти. Воно служить фундаментом для роботи CERT/CSIRT, а також операторів критичної інфраструктури, держорганів і приватних компаній в ЄС. [26]

Посібник вводить наступну архітектуру системи управління інцидентами:

- Incident Detection & Logging — встановлює вимоги до інструментів моніторингу, складання журналів, реєстрації мережеских і системних вимог. У тому числі регулює застосування SIEM, IDS/IPS, засобів аналізу поведінки UEBA.
- Incident Classification & Prioritization — посібник вводить класифікацію інцидентів, яка базується на типі впливу (DoS, malware, intrusion, data breach, misconfiguration), критичності інфраструктури, що атакується, типі впливу (integrity, availability, confidentiality impact), потенціалі для подальшого поширення.
- Incident Analysis — опис методик збору технічних артефактів і forensic-процедур, які потрібні як докази. Документ описує формати даних, механізми і кореляції та засоби підтвердження компрометації.
- Containment, Eradication & Recovery — регламент визначає технічні стратегії локалізації, методи усунення загроз і вимоги до відновлення сервісів.
- Reporting & Communication — ENISA Incident Management Guide містить у тексті рекомендації щодо складання стандартизованих відповідей. Вони включають первинне повідомлення, оновлення даних під час розслідування, підсумковий технічний звіт. У регламенті окремо згадують про те, що потрібно використовувати уніфіковані терміни та параметри звітів, які

сумісні з національними CERT та прийнятими в Європі структурами обміну даними.

- Lessons Learned & Continuous Improvement — у регламенті називають пост-інцидентну фазу обробки подій обов'язковою частиною циклу, яка також включає розробку коригувальних дій, оновлення політик безпеки та архітектури моніторингу.

Також документ встановлює вимоги до CSIRT і описує взаємодію суб'єктів, рівні ескалації, вимоги до засобів автоматизації процесів управління інцидентами та інтеграції з платформами обміну даними Threat Intelligence (STIX/TAXII). Це одне з найдетальніших керівництв, яке забезпечує стійкість і відтворюваність процесів реагування на інциденти на рівні окремих компаній або глобальних державних структур.

1.4.4 NIS2

Директива Європейського Союзу 2022/2555 NIS2 є другим поколінням нормативної бази, яка регулює діяльність у сфері кібербезпеки. Цей документ встановлює розширені вимоги до управління ризиками, а також до обміну інформацією та складання обов'язкової звітності про кіберінциденти. [27]

NIS2 замінює попередній документ NIS (2016), посилює та уніфікує діючі технічні стандарти для держав, що входять до ЄС, та операторів критичної інфраструктури.

Ця директива є обов'язковою для двох класів суб'єктів:

- essential entities — до неї входять енергетичний сектор, транспорт, банки, охорона здоров'я та цифрова інфраструктура;
- important entities — пошта, виробництво критично важливих товарів та провайдери телекомунікаційних послуг.

Ключові положення NIS2 представлені в таблиці 1.3.

Таблиця 1.3 - Положення NIS2 з описом

Положення	Опис
Risk Management Measures	Організації повинні впроваджувати у своїй діяльності технічні та організаційні заходи для моніторингу систем, управління вразливостями. Вони включають багатофакторну аутентифікацію, сегментацію мереж, криптографічний захист, логування та журналювання з можливістю подальшого forensic-аналізу.
Mandatory Incident Reporting	Визначаються терміни подання звітності. Initial Notification — інформування протягом 24 годин після виявлення значущого інциденту. Incident Notification — 72 години на уточнення технічних характеристик події, зокрема типу інциденту, впливу, векторів атаки та індикаторів компрометації. Final Report — 1 місяць для надання root cause analysis, переліку вжитих заходів та оцінки залишкових ризиків. Значущим називають інцидент, який впливає на доступність та цілісність критичних послуг, спричиняє істотні економічні або соціальні наслідки, зачіпає критичні сервіси або велику кількість користувачів.
Supply Chain Security	Обов'язкові заходи для оцінки ризиків ланцюгів поставок, в тому числі оцінку їх безпеки, контроль зовнішніх сервісів і вимоги до менеджменту інцидентів.
Enforcement & Penalties	Відповідальність аж до штрафів у розмірі 2% від річного обороту за порушення, а також суворі вимоги щодо технічного аудиту ресурсів.

NIS2 формалізує та деталізує режими звітності про інциденти в Європі. Він стандартизує повідомлення, зміст звіту, вводить критерії значущості та набір технічних параметрів. На його основі створюються автоматизовані системи складання звітів, платформи обміну інцидентами та модулі інтеграції з SIEM/SOAR і TIP.

1.4.5 EU Cyber Solidarity Act

Регламент ЄС EU Cyber Solidarity Act набув чинності з 4 лютого 2025 року. Цей документ прийнятий для посилення здатності держав та організацій в ЄС виявляти та реагувати на інциденти, а також відновлювати інфраструктуру після значних і масштабних атак.

Основні цілі прийняття регламенту [28]:

- Створення загальної європейської інфраструктури, яка потрібна для виявлення та попередження загроз у кіберпросторі;
- Розробка механізму екстреного реагування на події;
- Забезпечення спільних дій та взаємодії між державами та операторами критичної інфраструктури, а також постачальниками послуг безпеки.

Документ побудований навколо трьох модулів, пов'язаних між собою:

1. European Cybersecurity Alert System (ECAS) — мережа національних і міжнародних кібер-хабів у різних країнах Європи. Вони повинні збирати, аналізувати і передавати інформацію про загрози, в тому числі з використанням сучасної аналітики та машинного навчання.
2. Cybersecurity Emergency Mechanism — механізм термінового реагування на загрози, за допомогою якого можна мобілізувати необхідні ресурси. Він включає резерв сертифікованих провайдерів, служб реагування, надання підтримки постраждалим державам і компаніям, координоване відновлення після атак.
3. Cybersecurity Incident Review Mechanism — механізм аналізу та оцінки значущих або масштабних інцидентів. Передбачає формулювання висновків після розслідування, видачу рекомендацій щодо вдосконалення захисту на рівні всього ЄС, а також рекомендації щодо запобігання майбутнім атакам.

Документ передбачає взаємодію хабів ECAS між собою та їх синхронну роботу, щоб забезпечити ситуаційну обізнаність на рівні ЄС. Це необхідно для збільшення швидкості виявлення атак, їх локалізації та масштабування реакції.

Регламент описує повний цикл реагування на інциденти від первинного оповіщення до аналізу наслідків і відновлення. Причому всі дії повинні відбуватися всередині регламентованої архітектури. Документ надає достатню гнучкість для приватних організацій. Він не є обов'язковим для всіх суб'єктів, але надає необхідні механізми для них.

Cyber Solidarity Act доповнює існуючі нормативно-правові акти, в тому числі NIS2 Directive, але робить акцент на колективному захисті кіберпростору. Він створює основу для єдиного обміну даними під час threat intelligence і стандартизує процедуру повідомлення, забезпечує доступ до резервів реагування і передбачає аналіз подій на рівні ЄС з метою подальшого перенесення в локальні практики.

Висновки до розділу 1

В даному розділі було наведено основні теоретичні відомості щодо культури обміну інформацією про загрози між організаціями, важливість цього процесу. Розглянуто інструменти, що використовуються для реалізації такого обміну, їхні переваги та задачі, які вони вирішують.

Стало зрозуміло, що використовуючи сучасні інструменти обміну даними про загрози, такі як автоматизована платформа MISP в поєднанні з відкритими джерелами індикаторів компрометації та відомостями, отриманими від партнерів може значно посилити процеси Threat Intelligence всередині організації, що дозволить покращувати репутацію, заощаджувати гроші, отримувати якісну контекстну інформацію про загрози та прискорити аналіз та реагування на інциденти.

Також розглянуто нормативно-правову базу, яка регулює процеси реагування та звітування на кіберінциденти, а також звітування на CERT/CSIRT. Ці документи будуть використані в наступних розділах при взаємодії з LLM для надання додаткового контексту та рекомендацій, отримання кращого результату та

генерації звітів, що будуть відповідати законодавчим вимогам України та Європейського Союзу.

Документи схожі між собою рекомендаціями, що надаються, кроками реагування на інциденти та порядком звітності. Проте щоб створювати правильні compliant звіти за допомогою AI-агента, не можна нехтувати відмінностями та особливостями стандартів різних країн та враховувати всі викладені вимоги.

2 ПРОЕКТУВАННЯ ЗАСТОСУНКУ ТА СТВОРЕННЯ МОДЕЛЕЙ ЗВІТІВ

Обмін інформацією про загрози між організаціями та культура обміну даними в цілому - це вкрай важливий процес, який допомагає фахівцям з кіберзахисту централізовувати свої знання, дані та результати досліджень. Таким чином навіть невеликим командам без власних ТІ спеціалістів вдається своєчасно отримувати дані про актуальні загрози та адаптуватися до нових прийомів, що використовуються зловмисниками.

Крім цього, активна участь у обміні даними дозволяє компаніям в деяких випадках заощаджувати значні кошти. Адже в такому випадку вони можуть відмовлятися від платних джерел даних, які часто коштують досить дорого і при цьому можуть бути замінені даними з відкритих джерел. Ще однією перевагою можна виділити пришвидшення обробки та реагування на інциденти. Якщо аналітик матиме доступ до бази знань, де до індикаторів компрометації надається хоча б мінімальний контекст та enrichment, це може значно збільшити швидкість роботи, особливо коли мова йде про комплексні атаки з використанням декількох векторів та інструментів.

Наприклад, якщо EDR виявив шкідливе програмне забезпечення, аналітик може перевірити його хеш у MISP і побачити, що його колеги з іншої організації з'ясували, що цей файл поширюється через скомпрометований веб-сайт. Тоді фахівець зможе одразу переключити свою увагу на логи інтернет підключень, які виконував користувач, не витрачаючи час на перегляд логів поштового сервера з метою перевірки гіпотези про фішинговий лист.

Не менш важливим є питання репутації та соціальної відповідальності. Цей аспект є актуальним для багатьох комерційних компаній. В деяких випадках це закладено в цінності та філософію організації, деяких цікавить вплив репутації на фінансові показники. Активна участь в процесах обміну даними про загрози може покращити репутацію компанії в очах партнерів та держави, про це можна згадувати на конференціях та презентаціях тощо. Це може опосередковано збільшити доходи чи допомогти отримати інші переваги.

2.1 Визначення проблеми

Для того, щоб брати участь в обміні інформацією про загрози, компанії необхідно вкладати у це власні сили, час та ресурси. Це включає в себе:

1. Розгортання інфраструктури. Потрібно підготувати інфраструктуру для розгортання MISP або іншої платформи для обміну інформацією про загрози. Це може бути внутрішній сервер компанії або хмарне середовище, як от AWS чи Azure. В обох випадках необхідно докладати додаткових зусиль для налаштування віртуальної машини, забезпечення доступу до платформи ззовні.
2. Проектування та реалізація інтеграції. Після розгортання платформи необхідно інтегрувати з нею інструменти, що використовуються в компанії. Це можуть бути засоби моніторингу, бази даних, системи оповіщення (наприклад, пошта чи Slack), системи тікетів (Jira або аналоги). Це вимагає витрат часу та ресурсів на вивчення API, написання та тестування коду.
3. Адаптування внутрішніх процесів. Необхідно модифікувати внутрішні процеси обробки інцидентів з урахуванням подання даних на платформу обміну. Наприклад, якщо раніше аналітики розслідували інцидент, після чого закривали його, то тепер вони матимуть зібрати індикатори компрометації, додати контекст і подати дані на платформу обміну. За відсутності автоматизації все це буде робитись вручну через графічний інтерфейс платформи.
4. Подання звітності. Організації можуть подавати звіти про кіберінциденти, які відбулися в їхній інфраструктурі в CERT-UA або інші структури. Деякі підприємства зобов'язані це робити, наприклад об'єкти критичної інфраструктури. Інші можуть робити це за бажанням, в разі необхідності допомоги чи координації з CERT-UA.

Створення такого звіту також потребує часу на заповнення шаблонів та надання даних, які вимагаються законодавством. Формати та структура даних. В різних організаціях підхід до обробки інцидентів може відрізнятися. Деякі використовують техніку «5 Ws», деякі описують інцидент згідно вимог стандартів або виходячи з особливостей власних інструментів. Це викликає проблеми при інтеграції з платформами обміну даними і поданні звітів, оскільки формат опису інцидентів часто відрізняється від схеми, що застосовується, наприклад, у MISP або регламентах, таких як NIST 800-61 Rev.2 [24]. В таких випадках доводиться виконувати додаткову обробку та перетворення даних.

Через всі вищезазначені проблеми компанії часто нехтують участю в обміні даними про загрози, або відкладають роботи по впровадженню на довгі терміни. Це негативно впливає на ефективність захисту, швидкість реагування, витрати на threat intelligence.

Проте стрімкий розвиток AI-агентів дозволяє гнучко застосовувати штучний інтелект в рішеннях автоматизації. Тобто можливо створити програму, яка на основі ШІ зможе не лише генерувати тексти, а й приймати рішення про виконання тих чи інших операцій та самостійно їх виконувати. Це дозволяє реалізувати рішення автоматичного подання звітності та роботи з MISP на основі наявних даних про кіберінциденти згідно законодавства та актуальних стандартів.

2.2 Створення моделей звітів на основі вимог законодавства

Використання LLM для генерації даних передбачає певну нестабільність у результатах роботи. Навіть при однакових вхідних даних, ШІ може подавати на вихід дані, що будуть дещо відрізнятися один від одного. Це відома особливість технології, вже існують способи, як можна вирішувати цю проблему та контролювати результат, який видає мовна модель.

Задача автоматизації звітності про кіберінциденти та створення сутностей у MISP вимагає точних значень полів (в тому числі з певного переліку), дотримання заданого формату та схеми даних. Модель буде генерувати значення, які потім будуть застосовуватись для заповнення шаблону звіту, тому важливо обмежити «творчість» моделі та змусити її діяти в певних рамках.

Великі провайдери мовних моделей вже пропонують власні рішення цієї задачі. Здебільшого вони схожі між собою та носять назву «Structured output» [29]. Суть полягає в тому, щоб одночасно із запитом подати в LLM схему даних, яку користувач бажає отримати у відповідь.

В мові програмування Python для цього використовується бібліотека Pydantic. Вона дозволяє задавати власні схеми даних за допомогою класів, прописувати власні обмеження для полів і виконувати валідацію даних на предмет відповідності схемі. Можливо навіть задавати власні функції валідації. Наприклад, програма видаватиме помилку, якщо поле `IoC_type` не матиме значення зі списку (`hash`, `domain`, `IP`).

Отже, для того щоб налаштувати автоматичну генерацію звітів згідно законодавства, необхідно перевести вимоги стандартів в набір Pydantic моделей. Виконувати цю задачу доведеться вручну, оскільки вона вимагає ретельного аналізу законодавчих норм та рекомендації, пошуку шаблонів звітності та їх інтерпретації в межах можливості бібліотеки, що використовуватиметься в роботі.

Ще однією перевагою такого підходу є те, що інша бібліотека - Langchain, яка застосовується для побудови AI-агентів та застосунків у вигляді графів, має модуль, який дозволяє автоматично генерувати інструкції (промпт) для LLM на основі Pydantic моделі. Це дозволяє пришвидшити роботу та згенерувати запит за допомогою лише однієї функції за умови готової моделі.

Автоматизація буде працювати у 2 режимах:

- для українського законодавства;
- для законодавства ЄС.

Для кожного з режимів підібрано законодавчу базу у вигляді набору документів, які містять вимоги та рекомендації щодо звітності, обов'язкові дані,

що має містити звіт тощо. На основі цієї інформації будуватимуться моделі RYDANTIC. Ці документи також завантажаться в базу даних ChromaDB, що буде розглянуто пізніше.

2.2.1 Модель звіту на базі українського законодавства

Для створення моделі звіту на базі українського законодавства, було використано ряд документів, які регулюють порядок звітності про кіберінциденти. Їх було розглянуто у розділі 1.3 цієї роботи. Основним документом є Наказ ДССЗІУ №570 від 03.07.2023 «Про затвердження Методичних рекомендацій щодо реагування суб'єктами забезпечення кібербезпеки на різні види подій у кіберпросторі» [20]. Зокрема, в додатках до нього наявна Картка інформування про кіберінцидент/кібератаку.

Цей документ представляє собою PDF шаблон, призначений для заповнення фахівцями з команди реагування на інциденти. Він містить текстові поля для заповнення, а також поля, де необхідно вибрати один з варіантів. Ці поля дозволяють вичерпно описати інцидент, в тому числі повідомити про його критичність, негативний вплив на інфраструктуру, відношення інциденту до України, виявлені індикатори компрометації, сектор промисловості тощо. Надані методичні рекомендації можна успішно використати для формування моделі, а також як шаблон для автоматичної генерації PDF звітів. Тобто, налаштувавши автоматичне заповнення такої картки, можна отримати звіт, який відповідає чинному українському законодавству.

Всі поля картки, їх пояснення, можливі значення, а також назва відповідного поля у моделі, наведено у таблиці 2.1.

Таблиця 2.1 - Співставлення картки та моделі Pydantic для Українського звіту

Назва поля у картці	Відповідне поле у моделі звіту	Можливі значення (якщо обмежені)	Пояснення до поля
TLP	tlp	RED, AMBER, AMBER+STRICT, GREEN, CLEAR	Маркування за Traffic light protocol, позначення аудиторії розповсюдження конфіденційної інформації
Дата та час виявлення кіберінциденту	detection_datetime		Час, коли було виявлено інцидент (разом з часовим поясом)
Заявник	reporter		Дані особи, що подає звіт. Включає ПІБ, посаду, пошту, телефон
Джерело виявлення кіберінциденту	detection_source, detection_source_other	Адміністратор, Користувач, Антивірусне ПЗ, EDR, IDS/IPS, Інше (вказати)	Яким чином було виявлено інцидент
Запропонована категорія (рівень) критичності кіберінциденту	criticality	0, некритичний (білий), 1, низький (зелений), 2, середній (жовтий), 3, високий (помаранчевий), 4, критичний (червоний), 5, надзвичайний (чорний), Не вдалося визначити	Критичність інциденту
Отримана інформація	general_summary		Короткий опис інциденту

Продовження таблиці 2.1

Назва поля у картці	Відповідне поле у моделі звіту	Можливі значення (якщо обмежені)	Пояснення до поля
Постраждалий суб'єкт забезпечення кібербезпеки	victim		Постраждалий суб'єкт, включає юридичну назву, адресу, пошту, контактних осіб
Сектор (галузь) атакованого об'єкта	sectors, sectors_other	Сектор безпеки і оборони, Органи державної влади, Органи місцевого самоврядування, Енергетичний сектор тощо	Галузь об'єкта, який було атаковано
Доменна зона	domains	UAGOV Український державний, UACOM Український недержавний, FGOV Закордонний державний, FCOM Закордонний недержавний	В якому домені знаходиться атакований об'єкт
Відношення України до кіберінциденту	Relation_to_ukraine, relation_to_ukraine_other	Україна - об'єкт атаки, Україна - джерело атаки, Україна - елемент механізму атаки, не стосується території України	Як Україна відноситься до інциденту
Категорія та тип кіберінциденту (відповідно до Переліку категорій та типів кіберінцидентів)	category, incident_type	Відповідно до Переліку категорій та типів кіберінцидентів	Категорія та тип інциденту, відповідно до офіційного переліку

Продовження таблиці 2.1

Назва поля у картці	Відповідне поле у моделі звіту	Можливі значення (якщо обмежені)	Пояснення до поля
Дата та час початку кіберінциденту	incident_begin_datetime		Час початку кіберінциденту
Чи пов'язаний цей (ця) кіберінцидент з попередніми?	incident_relation, relation_code, related_incident_id	Так, Ні, ID пов'язаного кіберінциденту, Невідомо	Чи пов'язаний поточним інцидент з іншим, що був виявлений в організації
Короткий опис кіберінциденту	related_incident_summary		Короткий опис пов'язаного кіберінциденту (якщо є)
Вплив на функціонування систем/мереж, сервіси (послуги)	impact_code	Немає впливу взагалі, Мінімальний/значний вплив/втрата доступності на некритичні/критичні сервіси (послуги)	Ступінь впливу на сервіси або послуги, що надає атакований об'єкт
Кількість скомпрометованих систем/мереж (ЕОМ)	amount_code, amount_other	1-10, 10-50, Інше (вказати), Невідомо	Кількість систем, що було скомпрометовано
Тип скомпрометованої системи/мережі за функціоналом	system_type_code, system_type_other	Робочі станції, Сервери додатків, Сервери баз даних, Вебсервери, Сервери доменних імен тощо	Тип систем(и), що було скомпрометовано

Продовження таблиці 2.1

Назва поля у картці	Відповідне поле у моделі звіту	Можливі значення (якщо обмежені)	Пояснення до поля
Об'єкт кібератаки	attack_objects		Включає такі поля, як тип, ім'я, ОС, часова зона, мережеві налаштування, облікові записи, CVE (якщо є), висновок
Чи вирішено кіберінцидент/кібератаку?	incident_solved	Так, Ні	Чи завершено обробку інциденту
Чи потрібна допомога CERT-UA?	cert_ua_help_needed	Так, Ні	Чи потребується допомога CERT-UA у вирішенні інциденту
Чи повідомлялося про кіберінцидент/кібератаку іншим основним суб'єктам забезпечення кібербезпеки? Яким саме?	reported_to_other_subjects, other_subjects_list, other_subjects_comment	Так, Ні Служба безпеки України, Міністерство оборони України та Генеральний штаб Збройних Сил України, Розвідувальні органи тощо	Чи було повідомлено про інцидент в інші структури в Україні
Чи залучалися сторонні організації до вирішення кіберінциденту/кібератаки?	other_orgs_engaged, other_orgs_list, other_orgs_other, other_orgs_comment	Так, Ні Інший CERT, CSIRT, SOC, Антивірусні компанії, Обслуговуюча компанія, інтегратор, представник вендора, Інше (вказати)	Чи було залучено інші сторони до вирішення інциденту

Кінець таблиці 2.1

Назва поля у картці	Відповідне поле у моделі звіту	Можливі значення (якщо обмежені)	Пояснення до поля
Результат впливу	impact_result_code, impact_result_other	Витік даних, Злам (компрометація) системи, Втрата функціональності систем/сервісів, Інше (вказати)	Вплив інциденту на доступність/конфіденційність
Від потенційного впливу кіберінциденту на державному рівні під загрозою	impact_target_code, impact_target_other	Національна безпека, Сталість економіки, Національний імідж, Функціонування Уряду, Безпека персональних даних Громадян, Інше	Вплив інциденту на критичні функції держави
Індикатори компрометації	iocs	Мережеві, Хостові, Файлові	Знайдені індикатори компрометації (значення та опис)
Перелік отриманих даних	generated_data		Документи, що було згенеровано в ході розслідування

Таким чином, ми отримуємо готову Pydantic модель, яку можна використовувати для подальшого створення рішення автоматизації. Для валідації полів, що мають обмежений вибір можливих значень, використовується тип `Literal` з подальшим створенням власної функції валідації всередині моделі, яка перевірятиме, чи входить поточне значення поля до списку можливих варіантів. Нижче наведено фрагмент коду визначення моделі.

```
class UAIncidentReport(BaseModel):
    tlp: Optional[TLP] = Field(None,
description="TLP: RED / AMBER / AMBER+STRICT / GREEN / CLEAR")
```

```

detection_datetime: Optional[str] = Field(
    None,
    description="Detection datetime (e.g.
'YYYY-MM-DD HH:MM:SS', ideally in UTC)."
)
reporter: Optional[Reporter] = None
detection_source: Optional[DetectionSource] = None
detection_source_other: Optional[str] = Field(
    None,
    description="If detection_source == 'other',
explain here."
)
criticality: Optional[CriticalityLevel] = None
general_summary: Optional[str] = Field(
    None,
    description="General short summary of the
incident.")

```

Тепер таку модель можна використовувати для вирішення одразу 3 задач:

1. Генерація інструкцій для LLM для отримання результатів у заданому форматі
2. Валідація відповідей від LLM, генерація помилок у випадку невалідних значень полів.
3. Використання отриманого JSON для автоматичного заповнення шаблону та створення готового PDF звіту.

LLM буде отримувати інструкції, промпт та текстовий опис кіберінциденту. Далі вона аналізуватиме його та заповнюватиме поля відповідно до схеми. Також передбачено окремий механізм для полів, що мають варіант «Інше». Якщо LLM вирішує, що задані значення не підходять для поля, вона виставляє значення «Інше» та заповнює поле `fieldname_other`

на власний розсуд. Наприклад, якщо інцидент було виявлено SIEM системою (якщо про це йдеться в описі інциденту), модель вкаже цю інформацію в полі `detection_source_other`.

2.2.2 Модель звіту на базі законодавства ЄС

Після створення моделі для українського законодавства, необхідно розробити аналогічну модель з урахуванням вимог та стандартів ЄС. Складність створення моделі для ЄС полягає в тому, що в документах викладаються лише загальні вимоги та рекомендації для країн-членів, на основі яких вони повинні створювати власні стандарти. Тобто не можна розробити єдиний шаблон звіту, який підійшов би для будь-якої країни-учасниці Європейського Союзу. Адже кожна країна має свої структури, що відповідають за реагування на кіберінциденти (наприклад, CERT-FR для Франції).

Такі структури мають свої процедури оповіщення, побудовані процеси та вимоги. Наприклад, в багатьох країнах застосовуються веб-форми, де заявник повинен заповнювати поля. Після надсилання така форма автоматично відправиться в уповноважені структури.

Тому було прийнято рішення створити загальну модель, яка буде містити поля, рекомендовані документом NIST 800-61 Rev.2. В додатку В до цього документу перераховуються поля (в документі - «data elements») які мають бути зібрані для кіберінциденту. Хоча в документі зазначено, що список не є вичерпним і кожна організація має створювати власний набір полів, спираючись на особливості своїх процесів, він може стати основою для майбутніх модулів та розширень автоматизованого рішення.

Оскільки автоматизація використовує в своїй основі бібліотеку Langgraph, вона являє собою граф, де на кожному вузлі виконується певна операція, про що піде мова далі. Таким чином, можна створити вузол генерації основи звіту, а далі розширювати граф, додаючи вузли для окремих країн або організацій,

доповнюючи та змінюючи JSON з попереднього кроку з урахуванням вже більш конкретних вимог.

В документі поля поділено на 2 групи: Basic Data Elements та Incident Handler Data Elements. Перший містить основну інформацію про кіберінцидент, а другий блок призначений для заповнення фахівцем, що займався безпосередньою обробкою інциденту і містить такі поля, як поточний стан інциденту, вартість (яких збитків завдано), причина інциденту тощо. Буде створено 2 моделі, що описують ці групи, проте під час роботи програми, для зручності їх буде об'єднано в один JSON, з яким буде відбуватися робота. Співставлення полів моделі з рекомендаціями, а також пояснення для групи Basic Data Elements наведено у таблиці 2.2.

Таблиця 2.2 - Співставлення картки та моделі Pydantic для звіту ЄС (Basic Data Elements)

Назва поля у документі	Відповідне поле у моделі звіту	Пояснення до поля
Contact Information for the Incident Reporter and Handler	Окрема модель EUContact (name, role, org_unit, affiliation, email phone, location)	Контактні дані про заявника та обробника інциденту (включає ім'я, посаду, email тощо)
Status change date/timestamps (including time zone)	Окрема модель EUIncidentTimestamps (incident_start, incident_discovered, incident_reported тощо)	Часові мітки інциденту (коли почався, коли виявлено, коли вирішено тощо)
Physical location of the incident	physical_location	Де відбувся інцидент (місто, штат тощо)
Current status of the incident	current_status	Поточний статус інциденту
Source/cause of the incident	source_cause_description, source_hostnames, source_ip_addresses	Причина або джерело виникнення інциденту

Кінець таблиці 2.2

Назва поля у документі	Відповідне поле у моделі звіту	Пояснення до поля
Description of the incident	incident_description	Короткий опис інциденту
Description of affected resources	Окрема модель EUAffectedResource (resource_type, function, hostname, ip_address, description)	Опис асетів, на які вплинув інцидент або кібератака
If known, incident category, vectors of attack associated with the incident, and indicators related to the incident	incident_category, окрема модель attack_vectors (vector_description, indicators)	Вектори атаки та виявлені індикатори компрометації (IoC)
Prioritization factors	Окрема модель prioritization_factors (functional_impact, information_impact, recoverability, overall_severity, other_factors)	Фактори, що впливають на критичність інциденту та швидкість його обробки
Mitigating factors	mitigating_factors	Що могло пом'якшити наслідки інциденту
Response actions performed	Окрема модель response_actions	Які дії було виконано у відповідь на інцидент
Other organizations contacted	other_organizations_contacted	Які ще організації було проінформовано
General Comments	general_comments	Загальні додаткові коментарі

Фрагмент коду моделі наведено нижче.

```
class EUIncidentBasicData(BaseModel):
    reporters: List[EUContact] = Field(
        default_factory=list,
```

```

        description="Contact information for incident
reporter(s)."
    )
    handlers: List[EUContact] = Field(
        default_factory=list,
        description="Contact information for incident
handler(s), if different from reporters."
    )

    timestamps: Optional[EUIncidentTimestamps] =
Field(
    None,
    description="Key timestamps for incident
start, discovery, reporting, resolution, etc."
)

    physical_location: Optional[str] = Field(
        None,
        description="Physical location of the
incident (e.g., city, region, country)."
    )

```

Співставлення полів моделі з полями, зазначеними в документі для групи Incident Handler Data Elements наведено у таблиці 2.3.

Таблиця 2.3 - Співставлення картки та моделі Pydantic для звіту ЄС (Incident Handler Data Elements)

Назва поля у документі	Відповідне поле у моделі звіту	Пояснення до поля
Current Status of the Incident Response	current_response_status (not_started, in_progress, monitoring, completed, closed, unknown)	Поточний етап обробки інциденту

Кінець таблиці 2.3

Назва поля у документі	Відповідне поле у моделі звіту	Пояснення до поля
Summary of the Incident	incident_summary	Короткий опис інциденту
Incident Handling Actions (Log of actions taken by all handlers, Contact information for all involved parties, List of evidence gathered)	Окрема модель handling_actions_log (action, timestamp, performed_by, notes), involved_parties (name, role, org_unit, affiliation, email, phone, location), evidence (description, location, collected_by, integrity_reference)	Які дії було виконано під час обробки, які сторони було залучено, які докази зібрано
Incident Handler Comments	handler_comments	Коментарі обробника інциденту
Cause of the Incident	cause_of_incident	Причина інциденту
Cost of the Incident	cost_of_incident (currency, direct_cost, indirect_cost, description)	Вартість інциденту (які втрати понесе компанія)
Business Impact of the Incident	business_impact (impact_summary, operational_impact, reputational_impact, legal_regulatory_impact, safety_impact, other_impact_details)	Вплив інциденту на бізнес

Фрагмент коду моделі наведено нижче.

```
class EUIncidentHandlerData(BaseModel):
    current_response_status:
Optional[IncidentResponseStatus] = Field(
    None,
    description="Current status of the incident
response process."
```

```

)

incident_summary: Optional[str] = Field(
    None,
    description="Short summary of the incident from
the handler's perspective."
)

handling_actions_log: List[EUResponseAction] = Field(
    default_factory=list,
    description="Log of actions taken by all handlers
(ordered chronologically if possible).")

```

Тепер, маючи готові моделі, ми можемо генерувати готові JSON з цими полями за допомогою LLM в автоматизованому режимі. Але не менш важливим є правильне написання промпту, про що піде мова далі у роботі.

2.2.3 Модель події MISP

Для того, щоб LLM мала можливість автоматично створювати події у MISP на основі отриманих даних та ІоС, необхідно також створити відповідну модель. Оскільки просунуті можливості MISP не використовуватимуться в роботі, достатньо буде описати 2 сутності: Event та Attribute. Всі особливості цих сутностей, їхнє призначення та можливі значення полів прописані в документації до платформи. Процес створення аналогічний до моделей, що розглядалися вище. Далі наведено фрагмент коду моделі.

```

class MispEventCreate(BaseModel):
    info: str = Field(

```

```

        ...,
        description="Short human-readable description
of the event."
    )

    date: Optional[str] = Field(
        None,
        description="YYYY-MM-DD. If None, server can
default to 'today'."
    )

    distribution: DistributionLevel = Field(
        "1",
        description="MISP distribution level: 0-5 as
string."
    )

    threat_level_id: ThreatLevel = Field(
        "4",
        description="MISP threat level: 1 High, 2
Medium, 3 Low, 4 Undefined."
    )

    analysis: AnalysisLevel = Field(
        "0",
        description="Analysis level: 0 Initial, 1
Ongoing, 2 Complete."
    )

    published: bool = Field(

```

```

        False,
        description="Should the event be published
immediately?"
    )

    tags: List[str] = Field(
        default_factory=list,
        description="List of tag names to apply to
the event (e.g. TLP, sector, malware-family)."
    )

    attributes: List[MispAttribute] = Field(
        default_factory=list,
        description="List of initial attributes to
create inside the event."
    )

    extends_uuid: Optional[str] = Field(
        None,
        description="If this event should extend an
existing one, put its UUID here."
    )

    comment: Optional[str] = Field(
        None,
        description="Optional free-form comment about
the event as a whole."
    )

    source_incident_id: Optional[str] = Field(

```

```

        None,
        description="Internal ID of the original
incident in your system."
    )

```

Ця модель буде використовуватись аналогічно до попередніх. На основі отриманого JSON буде автоматично створюватися MISP подія за допомогою окремо написаних функцій та MCP сервера. Усі знайдені IoC будуть додані до події в якості атрибутів. Використання моделей в застосунку буде детально описане в наступних розділах.

2.3 Застосування LLM для уніфікації даних

Однією з основних причин того, що багато компаній нехтують обміном інформацією про загрози, полягає в необхідності адаптовувати власні процеси та інструменти для роботи з засобами Threat Intelligence, зокерма MISP, а також для відповідності законодавству тої країни, в правовому поля якої працює організація.

Незважаючи на існування стандартизованих загальноприйнятих форматів та протоколів обміну даними, такими як STIX, MISP та інші, компаніям доводиться докладати додаткових зусиль, щоб трансформувати свої дані в ці формати. Адже структура опису інциденту може кардинально відрізнятися в залежності від процесів, цілей компанії, а також інструментів, що використовуються.

Наприклад, SOC-аналітики компанії користуються принципом «5 Ws» для опису інцидентів, що виникли всередині організації. Цей підхід пропонує досить зручну структуру, дозволяє виділити IoC в окремий блок і зрозуміло описати інцидент, наприклад, в тікеті (якщо використовується Jira або аналогічний сервіс), проте для того, щоб автоматично завантажити всі знайдені індикатори компрометації з такого звіту, необхідно створювати власний парсер і писати

додатковий програмний код, або шукати готове рішення для конкретної ситуації, наприклад, на Github або від приватних компаній-вендорів.

Частково цю проблему вирішує наявність готових інтеграцій для популярних інструментів. Наприклад, такі інструменти як SentinelOne та Torq мають готові рішення для інтеграції з платформою MISP [30]. SentinelOne може автоматично збирати індикатори компрометації, наявні в платформі, і перевіряти дані з логів на наявність збігів в реальному часі. Можна також налаштувати генерування інциденту у випадку знаходження збігів. Проте такі інтеграції наявні не в усіх рішеннях. І навіть якщо їх реалізовано, вони часто обмежуються лише збором індикаторів, без урахування активної взаємодії з платформою та додавання ІоС із нових інцидентів. В таких випадках залишається лише робити feature request до вендора з проханням реалізувати необхідні функції, або робити це самостійно, використовуючи API.

Вирішити цю проблему можна за допомогою останніх досягнень в області великих мовних моделей. Сучасні мовні моделі можуть працювати з багатьма мовами, в тому числі українською, аналізувати контекст, «роздумувати», знаходячи найкращу відповідь. Такі моделі надають всі великі провайдери, такі як OpenAI, XAI, Anthropic та інші. Крім того, моделі працюють на серверах вендора і доступ до них можна отримувати за допомогою API.

Недоліком такого рішення є певна залежність від вендора, а також необхідність оплачувати використання API відповідно до використаних токенів. Ще одним мінусом можна назвати передачу чутливих даних про інциденти третім сторонам, проте це велика кількість вендорів вже пропонують вирішення цієї проблеми. Адже при використанні дорогих тарифів класу «Enterprise» дані, які надає користувач не будуть зберігатися та використовуватися для подальшого навчання моделі.

До недавнього часу великі мовні моделі могли лише генерувати тексти у відповідь на запит користувача, що було великою перешкодою при інтеграції штучного інтелекту з інструментами та процесами компанії. Все змінилося разом

з появою інструментів (Tools), а далі з популяризацією MCP, або model context protocol.

Це відкритий стандарт, що було розроблено компанією Anthropic в 2024 році. Головна мета MCP - створення зв'язку між LLM, зовнішніми джерелами даних та інструментами [31]. MCP дозволяє уніфікувати визначення викликів інтерфейсу для доступу до різних інструментів. Протокол є розвитком рішень, які до його появи пропонували вендори для вирішення таких проблем. Зокрема, підхід function calling, або tool calling. Сама компанія називає протокол еквівалентом порта USB-C для агентських систем. Ілюстрацію цього зображено на рис. 2.1.

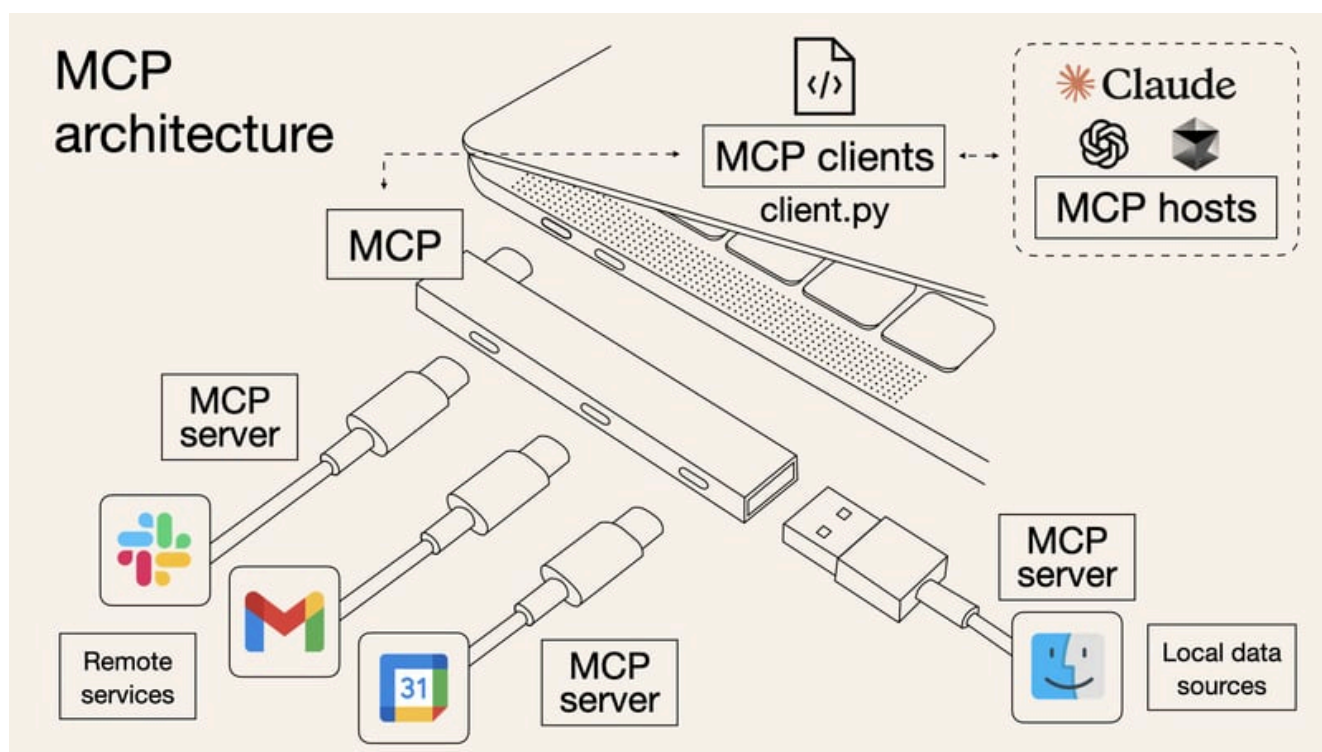


Рисунок 2.1 - Принцип роботи MCP [32]

Таким чином, усім сторонам взаємодії доведеться реалізовувати тільки один формат підключення для взаємодії з усіма зовнішніми інструментами з одного боку та усіма LLM з іншого.

Таким чином, використовуючи LLM та MCP, можна створити AI-агента, який вирішуватиме проблему різних форматів даних та особливостей обробки

інцидентів в різних організаціях. Достатньо лише передати опис інциденту в будь-якому форматі і модель згенерує дані, готові до створення звіту згідно законодавства України чи ЄС та до створення події в MISP з подальшим додаванням індикаторів компрометації. А за допомогою зовнішніх інструментів застосунок зможе виконати відповідні дії самостійно, тобто аналітику чи іншому фахівцю не доведеться, наприклад, створювати подію в MISP вручну, використовуючи дані, отримані від моделі.

2.4 Аналіз існуючих рішень

Стрімкий розвиток та популяризація великих мовних моделей та штучного інтелекту в цілому призвела до того, що сьогодні більшість розробників популярних сервісів та інструментів вже впровадили або знаходяться в процесі впровадження AI у свої інструменти. Ця тенденція торкнулася і рішень безпеки. AI інтегрується в EDR, SOAR та інші рішення і виконує різноманітні задачі від автоматизації до виявлення загроз.

Найближчими аналогами рішення, що розробляється в рамках виконання роботи, можна назвати:

1. Загальні платформи автоматизації з AI можливостями, такі як n8n або Make.com.
2. SOAR рішення які автоматизують процеси реагування на інциденти за допомогою штучного інтелекту, наприклад Torq.

Просунуті платформи, такі як Torq вже сьогодні можуть перетворювати дані в різні формати відповідно до заданих у промпті вимог, писати summary до інцидентів, збагачувати індикатори компрометації корисним контекстом, працювати з інцидентами тощо.

Проте, вищеназвані інструменти мають значні недоліки в порівнянні з розробленим рішенням:

1. Слабка підтримка RAG-пайплайнів та MCP інструментів. Torq підтримує використання RAG та MCP, проте лише у вигляді API з HTTP запитами. n8n більш гнучкий та дозволяє налаштувати RAG взаємодію з використанням моделі ембедингів, проте на налаштування такої взаємодії необхідно буде витратити додаткові ресурси і опції для конфігурації також обмежені тим, що змогли реалізувати розробники.
2. Відсутність можливостей для кастомізації та налаштувань. Часто розробник того чи іншого інструменту може не підтримувати ту модель або підхід, який необхідно реалізувати для досягнення задачі. Тоді доводиться або шукати обхідні шляхи, або чекати на реалізацію потрібних функцій розробниками.
3. Висока ціна. n8n може використовуватися безкоштовно для тестування та прототипування, проте для більш складних сценаріїв роботи та автоматизацій компанії доведеться витратити багато коштів. Особливо це стосується Torq, ціна на сервіси та інструменти цієї компанії зазвичай доволі висока і є зовсім великою особливо для стартапів та молодих SOC команд.

Враховуючи всі вищеназвані фактори можна зробити висновок, що рішення, створене в ході виконання роботи зможе краще виконувати задачу включення тієї чи іншої організації в процеси обміну інформацією про загрози та звітування державним структурам. Аналоги можуть потенційно його замінити, проте вони також мають свої недоліки.

2.5 Компоненти застосунку

Застосунок AI-агента складається з декількох частин, кожна з яких виконує свою функцію. Для досягнення максимальної ефективності роботи програми,

необхідно правильно підібрати технічні рішення, що будуть використовуватися, враховуючи сумісність з мовою програмування, що використовується. Для написання AI-агента буде використовуватися мова Python 3 з набором важливих бібліотек. Також застосовуватимуться наступні компоненти:

1. Векторна база даних.
2. Модель для створення ембедингів.
3. Велика мовна модель, що буде основою застосунку.
4. Ряд необхідних Python бібліотек.

Обґрунтування використання конкретних інструментів, а також порівняння різних варіантів баз даних та моделей ембедингів буде наведено в цьому розділі.

2.5.1 Векторна база даних

Оскільки в застосунку буде використовуватися RAG-пайплайн (Retrieval-Augmented Generation) для отримання даних з нормативно-правових актів України та ЄС, необхідно обрати рішення для зберігання ембедингів цих документів, що будуть генеруватися спеціальною моделлю.

RAG-пайплайн сьогодні є популярною технікою, яку часто використовують в поєднанні з LLM для підвищення ефективності та точності результатів, які генеруються моделлю. Вона складається з 3 кроків [31]:

1. Retrieval, під час якого модель звертається до зовнішнього джерела інформації, та забирає звідти необхідну інформацію.
2. Augmentation, додавання контексту до отриманої інформації, наприклад метадані.
3. Generation, який використовує отримані дані як додатковий контекст при генерації фінальних результатів.

Цей підхід також використовується в застосунку задля забезпечення кращої відповідності результатів вимогам нормативно-правових актів України та Європейського Союзу.

Під час роботи RAG-пайплайну LLM оперує ембедінгами. Вони являють собою представлення фрагментів тексту у вигляді вектора певної довжини. За перетворення тексту в ембедінги також відповідає окрема модель, яка називається моделлю ембедінгів. Таким чином можливо виконувати семантичний пошук і повертати результати, найбільш наближені за змістом до запиту.

У зв'язку з цим, до бази даних, яка буде зберігати та працювати з ембедінгами, висуваються спеціальні вимоги, що потребує застосування спеціалізованих під цю задачу рішень. Зі списку вимог можна виокремити наступні:

- Оптимізований ANN-пошук (Approximate Nearest Neighbors).
- Фільтрація за метаданими.
- Гібридний пошук (комбінація декількох типів пошуку та фільтрів)
- Переіндексація (повторне створення індекса з іншими параметрами)

Таким чином, не дивлячись на те, що сучасні реляційні та нереляційні бази даних підтримують зберігання векторів, вони не підходять для організації повноцінного RAG-пайплайну та взаємодії з великими мовними моделями. Тому було прийнято використовувати саме спеціалізовані векторні бази даних, які зможуть впоратися з цією задачею.

На сьогодні існує велика кількість векторних БД, що мають свої переваги та недоліки. Було проаналізовано кілька популярних рішень, результати наведено в таблиці 2.4.

Таблиця 2.4 - Порівняння векторних БД

Рішення	Переваги	Недоліки
Weaviate	гібридний пошук GraphQL API велика кількість модулів	високі вимоги до ресурсів складне розгортання
Milvus	підтримка багатьох індексів швидкодія масштабована	необхідне ручне налаштування потребує DevOps навичок

Кінець таблиці 2.4

Рішення	Переваги	Недоліки
ChromaDB	просте використання та інтеграція не потребує багато ресурсів вбудована фільтрація за тегами інтеграція з Python	обмежена масштабованість немає підтримки складних індексів немає можливостей кластеризації
Pinecone	хмарне рішення легке масштабування підтримка від вендора	пропрієтарність ціна мало можливостей для налаштування

Для застосунку було обрано базу даних ChromaDB. Хоча вона погано працює з великими об'ємами даних, масштабується і не має багатьох функцій, що є в інших рішеннях, вона дозволяє швидко розгортати її, гарно інтегрована з мовою програмування Python та бібліотекою LangChain. Ця БД гарно підходить для невеликої кількості даних, швидкого прототипування та PoC, що робить її гарним рішенням для розробки базового AI-агента.

2.5.2 Модель ембедингів

Ще одним критично важливим компонентом, що впливає на ефективність роботи RAG-пайплайну і як наслідок, моделі в цілому, є модель ембедингів. Вона виконує перетворення тексту у векторне представлення, яке потім буде зберігатися у векторній базі даних і по якій буде виконуватися пошук.

Моделі існують як пропрієтарні, які як правило працюють в хмарі і можуть застосовуватися через API, так і open-source, які можна встановити на виділений сервер і використовувати безкоштовно. Для максимального точного порівняння необхідно самостійно проводити тести та перевіряти моделі на власних даних та сценаріях. Це змушує витратити багато часу та ресурсів, особливо якщо обсяг текстів досить великий.

Проте існує велика кількість відкритих тестів, що перевіряють моделі на однакових масивах даних. Найпопулярнішим з таких тестів є MTEB (Massive Text Embedding Benchmark) [32]. Це великий бенчмарк, що використовується для перевірки моделей ембедингів. Він включає в себе різні типи завдань, що ставляться перед моделями, наприклад кластеризація, класифікація тощо.

Для задач застосунку найважливішими показниками є Retrieval, що показує ефективність моделі у задачах RAG-пайплайну та STS (Semantic Text Similarity), що показує точність відображення семантичного значення тексту. Для порівняння було відібрано декілька open-source та декілька пропрієтарних моделей. Результати порівняння наведено в таблиці 2.5.

Таблиця 2.5 - Порівняння моделей ембедингів

Модель	Категорія	Переваги	Недоліки	Retrieval	STS
text-embedding-3-large	Cloud API	Висока точність, стабільність	Залежність від вендора	59.32	71.68
text-embedding-3-small	Cloud API	Дешевша за large, швидша	Менш точна за large	50.69	69.42
paraphrase-multilingual-MiniLM-L12-v2	Open-source	Безкоштовна, дуже швидка, локальне розгортання	Менша розмірність, точність та стабільність	36.61	66.58
multilingual-e5-base	Open-source	Велика розмірність, приватність, локальне розгортання	Ручний тюнінг, необхідні потужності для розгортання	52.72	71.44
voyage-3.5	Cloud API	Краща розмірність векторів	Нова, недостатньо перевірена версія	64.01	69.97
gemini-embedding-001	Cloud API	Одна з кращих моделей за метриками, інтеграція з Google	Висока ціна, залежність від провайдера	67.71	79.40

В результаті для створення ембедингів було обрано модель multilingual-e5-base. Хоча дана модель не показує найкращих результатів за метриками бенчмарку MTEB, вона має низку переваг, що дозволяють ефективно використовувати дану модель у роботі:

1. Модель є open-source, тому її можна використовувати безкоштовно, розгорнувши на локальній машині. Це забезпечує приватність даних, що використовуються та незалежність від третіх сторін.
2. Розмір моделі є невеликим, що дозволяє розгортати її на локальній машині без значних витрат пам'яті.
3. Хоча модель і програє за метрикам варіантам від великих провайдерів, вона все ще показує хороші результати, які є кращими за, наприклад, модель paraphrase-multilingual-MiniLM-L12-v2.

Модель ембедингів multilingual-e5-base надає оптимальний баланс швидкості та якості роботи, є безкоштовною і має можливість легкого розгортання на локальній машині без значних витрат ресурсів, що вигідно вирізняє її для задач розроблення застосунку навіть на фоні моделей від великих вендорів, наприклад OpenAI.

2.6 Побудова архітектури застосунку

Коли основні компоненти застосунку вибрано та підготовлено до впровадження, необхідно повністю спроектувати програму, а також продумати потік даних (data flow), тобто яким чином інформація буде оброблятися та передаватися між компонентами. Оскільки буде застосовуватися типовий RAG-пайплайн, більшість кроків визначено одразу. Залишається лише врахувати взаємодію з MCP серверами та зовнішніми інструментами.

Схему застосунку наведено на рис. 2.2.

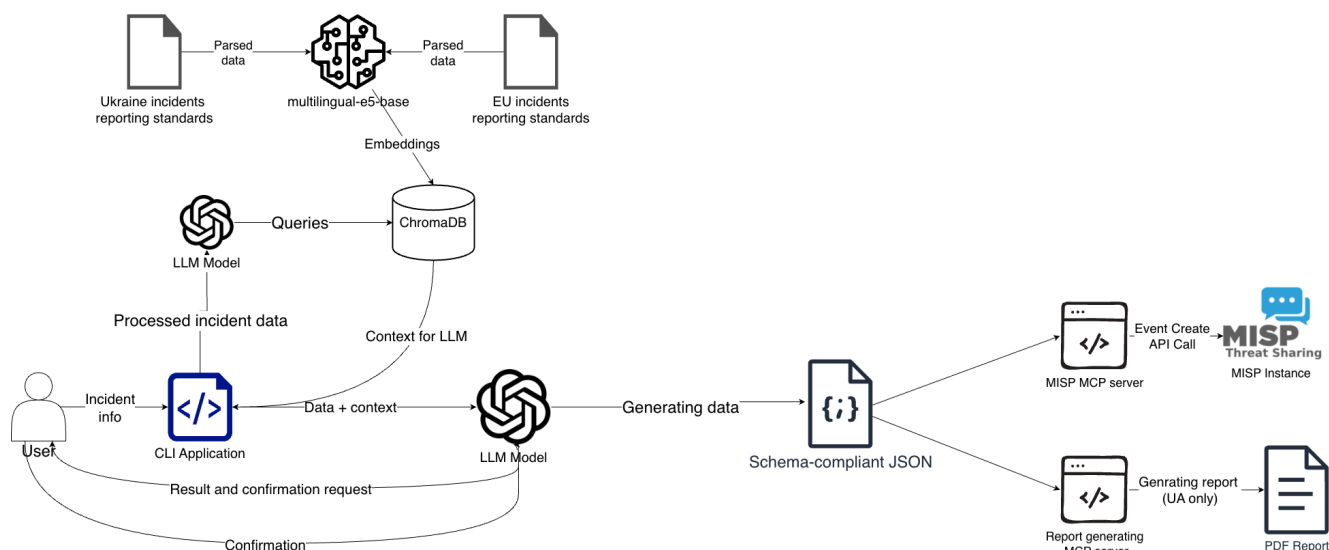


Рисунок 2.2 - Схема застосунку

Деякі компоненти застосунку було описано у попередніх розділах. Застосунок складається з наступних частин:

1. Нормативно-правова база для України та ЄС, представлена у вигляді PDF та HTML файлів та модель ембедінгів, яка перетворює тексти та зберігає у векторній базі даних. Ця процедура виконується лише один раз.
2. Векторна база даних, в який зберігаються ембедінги і яка грає ключову роль в RAG-пайплайні.
3. LLM, яка виконує генерацію фінального JSON, викликає інструменти. LLM обирається на початку виконання програми.
4. MCP сервери для генерації PDF звітів та MISP подій на основі JSON, побудованого за завчасно прописаною Pydantic моделлю.
5. Працюючий MISP інстанс, до якого є доступ з машини, на якій виконується код застосунку.
6. Допоміжний код, наприклад, парсери текстів стандартів для подальшого створення ембедінгів.

Типовий сценарій роботи застосунку виглядає наступним чином:

1. Користувач записує дані про кіберінцидент у файл, який потім читається програмою.

2. На початку роботи програми користувач обирає, згідно яких нормативно-правових актів необхідно генерувати звіт про інцидент (Україна або Європейський союз), а також LLM, яка буде використовуватися в якості основної. Вона буде отримувати дані, викликати інструменти і заповнювати JSON.
3. LLM генерує список запитів до векторної бази даних, які будуть використовуватися в RAG-пайплайні для отримання контексту з бази документів обраного законодавства.
4. Згенеровані на попередньому уроці запити відправляються до векторної бази даних, де виконується пошук і повернення найбільш семантично релевантних документів, які зберігаються в стані графу.
5. Опис інциденту від користувача, а також отриманий контекст надається LLM для генерування JSON, який має відповідати заданій схемі, в якій прописані можливі поля та їхні значення. Після генерації отриманий JSON проходить валідацію (без участі LLM).
6. Якщо валідацію провалено, повідомлення про знайдені помилки виводиться користувачу. Він вирішує, чи можна продовжувати виконання з отриманими даними або зупиняє виконання для проведення ще однієї спроби або використання іншої моделі LLM. В більшості випадків, помилки некритичні і звіт можна коректно згенерувати на основі отриманого JSON.
7. Після цього, потік виконання залежить від обраного законодавства. У випадку українського законодавства, формується PDF звіт. Якщо на початку було обрано нормативно-правову базу ЄС, цей крок пропускається і отриманий JSON просто виводиться користувачу.
8. Далі, незалежно від обраного законодавства, виконується генерація події MISP. Формується ще один JSON який відповідає завчасно створеній схемі для MISP подій та атрибутів. Далі на основі цього JSON за допомогою MCP сервера та бібліотеки PyMisp виконується з'єднання з MISP, після чого в ньому створюється подія та додаються атрибути та індикатори компрометації на основі JSON отриманого від великої мовної моделі.

Бібліотека Langgraph дозволяє візуалізувати граф виконання з вузлами та ребрами, які побудовано у коді. Цю візуалізацію наведено на рис. 2.3.

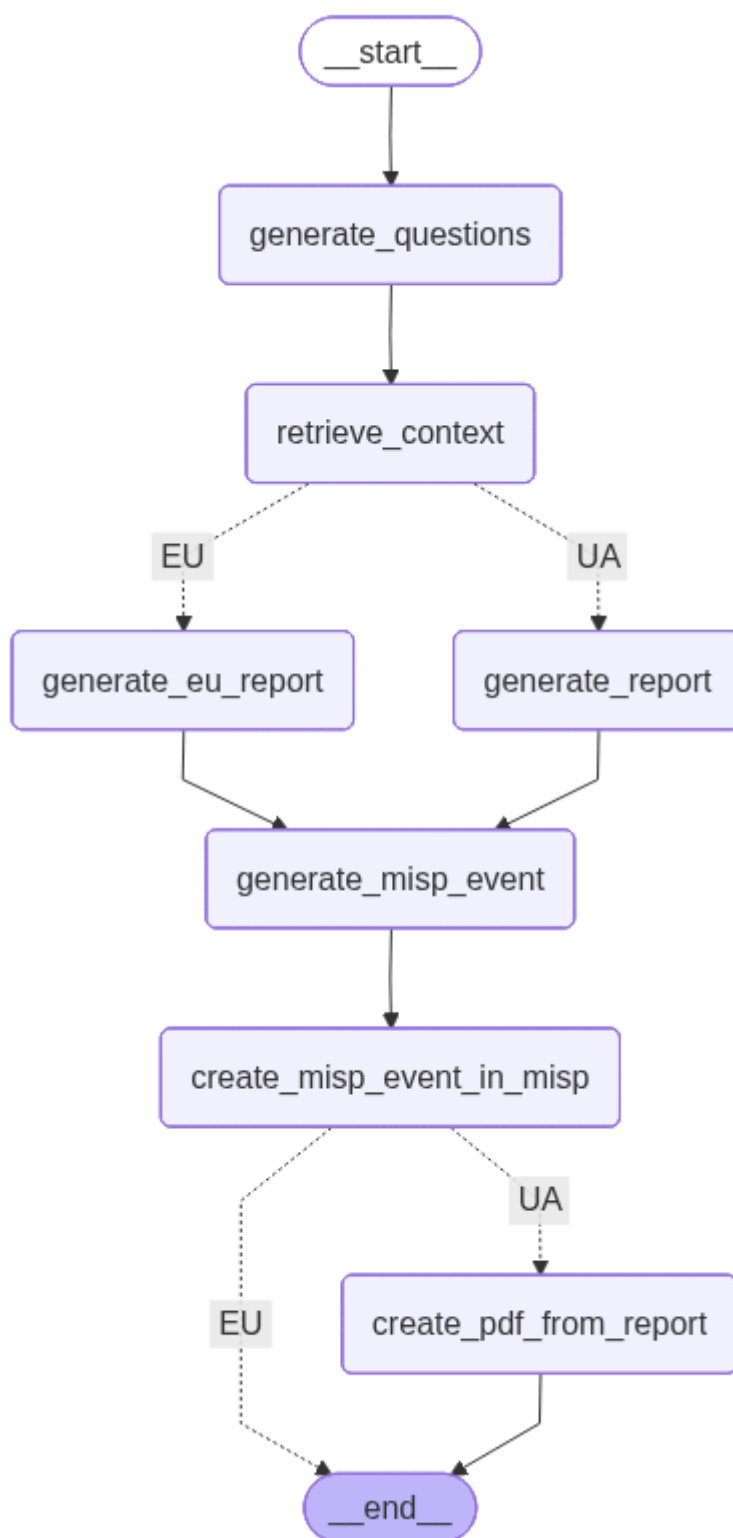


Рисунок 2.3 - Граф роботи програми

Таким чином, ми отримуємо готовий PDF звіт а також створену MISP подію з атрибутами та індикаторами компрометції. Всі дії під час виконання логуються, що дозволяє відстежити помилки при виконанні та оцінити отримані JSON, які згенерувала LLM.

2.7 Оцінка результатів роботи застосунку

Для того щоб оцінити якість роботи застосунку, а також порівняти між собою різні мовні моделі у задачі генерування звітів про кіберінциденти на основі опису користувача та контексту з векторної бази даних ChromaDB, необхідно ввести чіткі критерії визначення якості отриманого результату. Оцінюватися буде готовий PDF звіт для українського законодавства та згенерований JSON для нормативно-правової бази ЄС. Створена MISP подія разом з атрибутами та індикаторами компрометації буде оцінюватися для обох варіантів роботи програми.

Оцінка буде проводитися за наступними критеріями:

1. Якість заповнення звіту. До цього критерія відноситься правильність згенерованих даних, відсутність галюцинацій, точність передання всіх відомостей, викладених користувачем в описі кіберінциденту.
2. Повнота заповнення звіту. В цьому критерії оцінюється повнота розкриття інциденту в звіті. Мовна модель повинна робити висновки і заповнювати поля інциденту, дані про які не надано в описі у явному вигляді. Проте ті поля, які неможливо заповнити без відповідних даних, повинні залишатися пустими.
3. Правильність створення MISP події. В цей показник закладається правильність розпізнавання індикаторів компрометації у тексті, їх вид (мережні, хостові) та тип (IP-адреса, хеш тощо). Крім того, оцінюються правильність коментарів та нотаток, що прикріплюються до індикаторів, теги для MISP подій та інші дані, що заносяться у MISP в результаті роботи програми.

4. Якість JSON для ЄС. В цьому критерії оцінюється, наскільки правильно модель заповнила JSON на основі наданої їй Pydantic схеми та інструкцій згідно вимог, наведених у NIST 800-61 Rev.2. [24]. Враховується як правильність відображення відомостей, наведених в описі інциденту, так і правильність висновків, які можна зробити з цих даних для заповнення таких полів, як, наприклад, вплив на бізнес.
5. Якість MISP події для ЄС. Оцінюється повнота та правильність заповнення MISP події: чи всі індикатори компрометації було додано як атрибути, наскільки точні коментарі додано до індикаторів, чи було додано індикатори, які вказані неявно, наприклад, ідентифікатори CVE вразливостей, що було знайдено.

Використовуючи ці параметри, можна комплексно оцінити роботу моделі і порівняти LLM між собою в контексті виконуваної задачі.

Моделі будуть оцінюватися за критеріями, наведеними вище за шкалою від 1 до 5. Будуть оцінюватися 2 типи сутностей: PDF-звіт/JSON та MISP подія.

Пояснення до оцінок у шкалі для PDF-звіту/JSON:

- 1 - звіт заповнено некоректно або більшість полів пропущені. Наявні критичні помилки, галюцинації, звіт зовсім не відповідає опису або вимогам.
- 2 - частина полів заповнена, але вони містять значні неточності. Поля що не наведено прямо в описі, проігноровано.
- 3 - основні поля заповнено правильно на основі опису інциденту. Більшість полів, що потребують висновків від моделі, також заповнено правильно. Можуть бути неточності, але не критичні.
- 4 - все, що прямо описано в інциденті, відображено повністю правильно. Для нетипових значень використано поле «other» та записано коректне значення. Висновки зроблено правильно та більшість таких полів заповнено.
- 5 - вся інформація з опису інциденту відображена правильно. Всі поля, що потребують висновків від моделі, також заповнено

правильно. Для нетипових значень використано поле «other» та записано коректне значення. Всі поля з вільним введенням тексту заповнені коректно та детально.

Далі у списку наведено пояснення до оцінок у шкалі для MISP події та атрибутів:

- 1 - модель додала лише незначну частину індикаторів до події. Опис події не відображає суті, коментарі до індикаторів некоректні або відсутні.
- 2 - було додано більшість індикаторів, описи присутні, але невірні або вірні лише частково. Назва події частково відображає суть інциденту. Відсутні додаткові коментарі (external analysis або other).
- 3 - всі індикатори компрометації, що явно надано в описі інциденту, додані правильно. Коментарі присутні, проте не завжди достатньо інформативні. Назва події відображає суть інциденту. Додаткові коментарі присутні, але недостатньо інформативні.
- 4 - всі вказані в інциденті індикатори присутні з правильними коментарями. Назва події повністю відображає суть інциденту. Додаткові коментарі присутні та інформативні.
- 5 - всі індикатори, явно наведені в інциденті присутні та мають інформативні коментарі. Назва події повністю відображає суть інциденту. Також додано індикатори, про які зробила висновки модель, наприклад ідентифікатори вразливостей CVE. Присутні коментарі external analysis або other, що надають додатковий контекст.

В якості моделей для роботи було обрано найпопулярніші моделі від великих провайдерів останніх стабільних моделей. Повний перелік наведено в таблиці 2.6.

Таблиця 2.6 - LLM що використовуються в застосунку

Назва моделі в застосунку	Модель, що використовується
GPT-4.1	gpt-4.1 (non-reasoning)
GPT-5.2	gpt-5.2 (reasoning)
Claude Sonnet 4.5	claude-sonnet-4-5 (reasoning)
Claude 3.5 Haiku latest	claude-3-5-haiku-latest (non-reasoning)
Grok 4.1 (fast)	grok-4-1-fast-non-reasoning
Grok 4.1 (reasoning)	grok-4-1-fast-reasoning
LLaMA 4 Scout 17B	meta-llama/llama-4-scout-17b-16e-instruct
DeepSeek (fast)	deepseek-chat
DeepSeek (reasoning)	deepseek-reasoner
Gemini (fast)	gemini-2.5-flash (non-reasoning)
Gemini (reasoning)	gemini-2.5-pro (reasoning)

Кожна модель представлена в 2 варіаціях, швидка та «мисляча» (reasoning). Таким чином, окрім порівняння моделей між різними виробниками, результати також покажуть різницю між різними режимами роботи і продемонструють, чи варте очікування і більші витрати на використання тих результатів, які генеруватиме reasoning модель.

Оцінювання буде відбуватися експертним методом. Буде вручну створено 3 різних описи кіберінцидентів, що будуть відрізнятися деталізованістю, кількістю відомостей, індикаторами компрометації. Описи буде створено таким чином, щоб перевірити різні значення полів, різні типи інцидентів тощо.

Експерти будуть оцінювати згенеровані JSON а також події у MISP на основі опису про інцидент, до якого вони також матимуть доступ. Далі виставлятимуться оцінки від 1 до 5 за кожним критерієм, які було розписано вище. Також вибірково експерти надаватимуть коментарі, які будуть пояснювати

поставлену оцінку. Повну таблицю з результатами перевірок, тексти опису інцидентів буде наведено в наступному розділі.

Висновки до розділу 2

В даному розділі було повністю спроектовано та описану архітектуру застосунку, що реалізовано в роботі, обґрунтовано використання тих чи інших компонентів, рішень та моделей, побудовану схему програми та описано типовий сценарій роботи.

Також виконано порівняння з існуючими рішенням, виділено їх переваги та недоліки, описано відмінності між ними та програмою, що створюється в рамках виконання роботи. Описано також критерії та метод оцінювання, моделі що будуть брати участь в порівняннях.

Всі ці кроки були необхідними для забезпечення максимально можливого рівня ефективності застосунку та правильності згенерованих даних. Кожен компонент, включаючи обрану базу даних, модель ембедингів та велику мовну модель має значний вплив на результат роботи. Використання великої кількості моделей є важливим, оскільки дозволяє обрати найкращу модель для виконання задачі, продемонструвати відмінності роботи різних варіантів LLM. Таким чином, вдасться підібрати оптимальний варіант для постійного використання.

3 РЕАЛІЗАЦІЯ РІШЕННЯ СТВОРЕННЯ ЗВІТІВ ТА MISR ПОДІЙ З ВИКОРИСТАННЯМ LLM ТА MCP

В даному розділі наведено відомості про особливості роботи застосунку, подробиці щодо реалізації компонентів, в тому числі фрагменти коду мовою Python 3. В розділі наведено реалізацію MCP серверів та підключення до них, графу виконання, а також парсерів. HTML парсер документів було розроблено спеціально під формат нормативно-правових актів, розміщених на сайті Верховної Ради України і дозволяє отримати оптимальні фрагменти для створення ембедінгів та завантаження до векторної бази даних. Також буде розглянуто нюанси, що виникли під час розробки програми і шляхи їх подолання.

Крім цього, в рамках розділу буде проведено тестування роботи застосунку на основі ряду великих мовних моделей, як безкоштовних, так і платних. Детально буде розглянуто лише декілька тестів, інші через їхню велику кількість буде зведено в порівняльну таблицю за критеріями, наведеними в попередньому розділі. В кінці будуть підведені висновки, яка модель показала себе найкраще з точки зору співвідношення якості та ціни за один запуск застосунку.

3.1 Програмна реалізація застосунку

3.1.1 Парсери документів

Для створення RAG-пайплайну та наповнення векторної бази даних ембедінгами, спочатку необхідно отримати дані, які будуть використовуватися в якості контексту для LLM. Ці дані буде взято з нормативно-правових актів України та Європейського союзу, щоб результат, який генеруватиме модель, максимально відповідав вимогам та рекомендаціям законодавства.

Оскільки моделі ембедінгів та великі моделі мають обмеження на кількість токенів, що можуть оброблятися за раз, передавати весь текст документу цілком в якості контексту неможливо. Крім того, такий підхід навіть за відсутності обмежень має великі недоліки. Адже контекст необхідно надавати з урахуванням

задачі, що вирішується. В іншому випадку позитивного впливу на результат досягнути не вдасться.

Тому текст документів необхідно парсити та розбивати на менші фрагменти, перетворювати на ембедінги, після чого база даних буде виконувати пошук та повертати лише найбільш семантично наближені до запиту фрагменти. Бібліотеки, що використовуються в роботі, уже надають необхідні інструменти та функції для отримання готових фрагментів з PDF файлів. Їх будемо використовувати для законодавчих актів ЄС.

Для українських законів було розроблено HTML парсер, що враховує структуру законів для отримання кращих атомарних фрагментів без втрати змісту. Фрагмент коду реалізації HTML парсера зображено на рисунках 3.1 та 3.2.

```
def extract_structured_blocks(path: Union[str, Path],
                             jurisdiction: str,
                             min_chars: int = 400,
                             max_chars: int = 1200) -> List[Dict]:
    soup = fetch_html_from_file(path)
    header = extract_doc_header([soup])

    current_section_num: int | None = None
    current_section_name: str | None = None
    current_article: str | None = None
    current_clause: str | None = None

    blocks: List[Dict] = []
    p = Path(path)

    buffer: List[str] = []
    buffer_len = 0
    has_list_items = False

    def emit_block(force=False):
        nonlocal buffer, buffer_len, has_list_items
        if not buffer:
            return
        text = " ".join(buffer).strip()
        if not force and len(text) < max(80, min_chars // 3):
            return

        section_label = f"Розділ {current_section_num}" if current_section_num is not None else None
        parts = []
        if section_label: parts.append(section_label if not current_section_name else f"({section_label}): {current_section_name}")
        if current_article: parts.append(current_article)
        if current_clause: parts.append(f"n. {current_clause}")

        text_fmt = "; ".join([t.strip(" ;") for t in buffer]) if has_list_items else text
```

Рисунок 3.1 - Фрагмент коду HTML парсера

```

80 def extract_structured_blocks(path: Union[str, Path],
99     def emit_block(force=False):
117         jurisdiction_metadata, language_metadata = 'UA', 'uk'
118     elif jurisdiction == 'EU':
119         jurisdiction_metadata, language_metadata = 'EU', 'en'
120
121     block_meta = {
122         **header,
123         "section": current_section_num,
124         "section_name": current_section_name,
125         "clause": current_clause,
126         "headings_path": " > ".join(parts) or None,
127         "source": p.name,
128         "jurisdiction": jurisdiction_metadata,
129         "language": language_metadata
130     }
131     blocks.append({"text": text_fmt, "metadata": block_meta})
132
133     buffer = []
134     buffer_len = 0
135     has_list_items = False
136
137     for tag, text, _ in iter_text_nodes(soup):
138         m_sec = RX_SECTION.match(text) or RX_SECTION_ROMAN_ONLY.match(text)
139         if m_sec:
140             emit_block(force=True)
141             roman = m_sec.group(1)
142             title = (m_sec.group(2) or "").strip() or None
143             current_section_num = roman_to_int(roman)
144             current_section_name = title
145             current_article = None
146             current_clause = None
147             buffer.append(text); buffer_len += len(text)
148             emit_block(force=True)
149             continue
150

```

Рисунок 3.2 - Метадані HTML парсера

Такий парсер дозволяє розбивати текст документів з урахуванням пунктів законів, розділів та інших структурних елементів, включаючи правильну обробку списків. Отримані відомості також передаються в блок метаданих разом із кожним документом. Результат роботи парсера (приклад одного з отриманих блоків) зображено на рис. 3.3.

```
~/Documents/master_project > /Users/lingvvarm/Documents/master_project/llm-reporting/bin/python /Users/lingvvarm/Documents/master_project/parse_html.py
Всього створено документів: 92

Приклад тексту документу:
Проводиться моніторинг сповіщень, створених системою виявлення вразливостей (інцидентів), а також запроваджуються додаткові вимоги щодо журналювання (логування), зберігання та управління журналами (лог-файлами). 6. У реагуванні на кіберінциденти/кібератаки бере участь персонал суб'єкта забезпечення кібербезпеки з підтвердженою кваліфікацією та відповідними навичками. Запроваджується періодична перевірка компетенцій фахівців шляхом виконання практичних вправ у режимі реального часу з метою підвищення готовності до реагування на кіберінциденти/кібератаки. Важливим є залучення до навчання з реагування саме того персоналу суб'єкта забезпечення кібербезпеки, який буде здійснювати реагування на підозрілу поведінку (або кіберінциденти/кібератаки) на практиці, що дозволить зменшити витрати, підвищити компетенції та покращити навички спеціалістів (фахівців), які дійсно цього потребують.

Приклад метаданих документу
{'doc_title': 'Про затвердження Методичних рекомендацій щодо реагування суб'єктами забезпечення кібербезпеки на різні види подій у кіберпросторі | від 03.07.2023 № 570', 'doc_number': '570', 'section': 2, 'section_name': 'Етап підготовки', 'clause': '6', 'headings_path': 'Розділ 2: Етап підготовки > п. 6', 'source': 'methodics_recommendations_ir.htm', 'jurisdiction': 'UA', 'language': 'uk'}
```

Рисунок 3.3 - Приклад отриманого документа

На основі отриманих даних було створену векторну базу даних ChromaDB. Скрипт виконав парсинг всіх файлів у вказаних директоріях відповідно до типу і додав всі отримані документи до векторної БД. Хід цього процесу наведено на рис. 3.4.

```
~/Documents/master_project > /Users/lingvvarm/Documents/master_project/llm-reporting/bin/python /Users/lingvvarm/Documents/master_project/generate_chroma.py
Parsed file standards/ua/pdf_docs/no_html_analogue/card_template.pdf in 11 documents
Parsed file standards/ua/pdf_docs/no_html_analogue/interaction_principles_during_ir.pdf in 25 documents
Parsed file standards/ua/html_docs/about_main_principles.htm in 83 documents
Parsed file standards/ua/html_docs/some_questions.htm in 9 documents
Parsed file standards/ua/html_docs/banking.htm in 45 documents
Parsed file standards/ua/html_docs/methodics_recommendations_ir.htm in 92 documents
Parsed file standards/ua/html_docs/cybersec_strategy.htm in 48 documents
Parsed file standards/ua/html_docs/about_critical_infrastructure.htm in 104 documents
Parsed file standards/eu/pdf_docs/enisa_incident_management.pdf in 266 documents
Parsed file standards/eu/pdf_docs/enisa_ir.pdf in 84 documents
Parsed file standards/eu/pdf_docs/NIST_SP_800-61r2.pdf in 265 documents
Parsed file standards/eu/pdf_docs/nis2.pdf in 335 documents
Parsed file standards/eu/pdf_docs/cs_coalition_ir.pdf in 120 documents
/Users/lingvvarm/Documents/master_project/chroma_embedding.py:6: LangChainDeprecationWarning: The class `HuggingFaceEmbeddings` was deprecated in LangChain 0.2.2 and will be removed in 1.0. An updated version of the class exists in the `langchain-huggingface` package and should be used instead. To use it run `pip install -U langchain-huggingface` and import as `from langchain_huggingface import HuggingFaceEmbeddings`.
embeddings = HuggingFaceEmbeddings()
Saved 1487 chunks to chromadb/proddb
```

Рисунок 3.4 - Формування векторної бази даних

3.1.2 RAG-пайплайн та отримання контексту

Далі буде розглянуто код для взаємодії з векторною базою даних, формування запитів до неї за допомогою LLM, а також процес отримання готових документів. Оскільки ChromaDB має дуже хорошу інтеграцію з Python з коробки, всю взаємодію можна побудувати за допомогою бібліотеки Chroma. Процес складається з 2 частин: генерації запитів та використання цих запитів для пошуку в базі даних та отримання фрагментів. Код генерування питань наведено на рис. 3.5.

```

559 def generate_questions(state: IncidentState) -> IncidentState:
560     incident = state["incident_description"]
561     jurisdiction = state.get("jurisdiction", "UA").upper()
562     logger.info(
563         f"[generate_questions] jurisdiction={jurisdiction}, "
564         f"incident_len={len(incident)}"
565     )
566
567     if jurisdiction not in ("UA", "EU"):
568         logger.warning(f"[generate_questions] Unknown jurisdiction={jurisdiction}, defaulting to UA")
569         jurisdiction = "UA"
570
571     human_text = question_prompt.format(
572         incident_description=incident,
573         jurisdiction=jurisdiction,
574     )
575
576     messages = [
577         SystemMessage(content=QUESTION_SYSTEM_PROMPT.strip()),
578         HumanMessage(content=human_text),
579     ]
580
581     raw_text = llm.invoke(messages).content
582     logger.debug(f"[generate_questions] Raw questions block:\n{raw_text}")
583
584     queries: List[str] = []
585     for line in raw_text.splitlines():
586         q = line.strip()
587         if q:
588             queries.append(q)
589
590     logger.info(f"[generate_questions] Generated {len(queries)} semantic queries")
591     return {"questions": queries}

```

Рисунок 3.5 - Код функції створення пошукових запитів

Таким чином генеруються 5-7 запитів, які потім використовуються для семантичного пошуку фрагментів у векторній базі даних. Приклад запитів наведено на рис. 3.6.

```

2025-12-13 13:46:50.156 | DEBUG | __main__:generate_questions:643 - [generate_questions] Raw questions block:
вимоги до класифікації кіберінцидентів
рівні впливу на критичну інфраструктуру
обов'язки суб'єктів реагування на кіберінциденти
вимоги до опису та фіксації інциденту
технічні заходи захисту систем від несанкціонованого доступу
організаційні заходи щодо реагування на кіберінциденти
вимоги до збереження журналів подій та цифрових артефактів
відповідальність за порушення вимог кібербезпеки
порядок надання інформації про кіберінциденти
критерії оцінки наслідків кіберінцидентів для інфраструктури

```

Рисунок 3.6 - Приклад отриманих пошукових запитів

Коли запити отримано та збережено, виконання переходить до наступного вузла графа, в якому відбувається отримання фрагментів тексту з бази даних. Код функції наведено на рис. 3.7.

```

594 def retrieve_context(
620
621     for q in queries:
622         search_q = f"query: {q}"
623
624         docs_with_scores = chroma_db.similarity_search_with_score(
625             search_q,
626             k=k_per_query,
627             filter=filter_meta,
628         )
629
630         for d, score in docs_with_scores:
631             m = d.metadata or {}
632
633             key = (
634                 m.get("source"),
635                 m.get("page"),
636                 m.get("headings_path") or m.get("doc_title"),
637                 d.page_content[:80],
638             )
639             if key in seen_keys:
640                 continue
641             seen_keys.add(key)
642             collected.append((d, score))
643
644             if len(collected) >= max_docs:
645                 break
646
647             if len(collected) >= max_docs:
648                 break
649
650     context_blocks: List[str] = []
651     context_chunks: List[Dict[str, Any]] = []
652

```

Рисунок 3.7 - Фрагмент функції отримання контексту

Приклад отриманих фрагментів з векторної бази даних наведено на рис. 3.8. Вони будуть передаватись LLM під час генерування фінального результату в якості додаткового контексту.

```
[retrieve_context] Collected 20 unique documents
[retrieve_context] DOC 1 | score=0.23675116896629333 | metadata={"clause": "5", "section_name": "Етап виявлення та аналізу", "jurisdiction": "UA", "doc_number": "570", "source": "methodics_recommendations_ir.htm", "headings_path": "Розділ 2: Етап підготовки > п. 5"}
[retrieve_context] DOC 2 | score=0.24552994966506958 | metadata={"section": 3, "section_name": "Використання Правил", "doc_number": "570", "source": "methodics_recommendations_ir.htm", "headings_path": "Розділ 2: Етап підготовки > п. 5"}
[retrieve_context] DOC 3 | score=0.2479671835899353 | metadata={"jurisdiction": "UA", "clause": "5", "doc_title": "Про затвердження Методи", "source": "methodics_recommendations_ir.htm", "headings_path": "Розділ 2: Етап підготовки > п. 5"}
[retrieve_context] DOC 4 | score=0.25311943888664246 | metadata={"start_index": 930, "total_pages": 6, "page_label": "3", "jurisdiction": "UA", "doc_number": "570", "source": "methodics_recommendations_ir.htm", "headings_path": "Розділ 2: Етап підготовки > п. 5"}
[retrieve_context] DOC 5 | score=0.271808385848999 | metadata={"clause": "12", "jurisdiction": "UA", "language": "uk", "doc_number": "570", "source": "methodics_recommendations_ir.htm", "headings_path": "Розділ 2: Етап підготовки > п. 5"}
[retrieve_context] DOC 6 | score=0.27317672967910767 | metadata={"language": "uk", "source": "some_questions.htm", "jurisdiction": "UA", "doc_number": "570", "source": "methodics_recommendations_ir.htm", "headings_path": "Розділ 2: Етап підготовки > п. 5"}
[retrieve_context] DOC 7 | score=0.17455227673053741 | metadata={"source": "about_main_principles.htm", "doc_number": "45", "jurisdiction": "UA", "doc_title": "Про затвердження Методи", "source": "methodics_recommendations_ir.htm", "headings_path": "Розділ 2: Етап підготовки > п. 5"}
[retrieve_context] DOC 8 | score=0.1917685568332672 | metadata={"jurisdiction": "UA", "doc_title": "Деякі питання реагування суб'єктами за", "source": "methodics_recommendations_ir.htm", "headings_path": "Розділ 2: Етап підготовки > п. 5"}
[retrieve_context] DOC 9 | score=0.20313584804534912 | metadata={"page_label": "7", "jurisdiction": "UA", "author": "user", "source": "sta", "source": "methodics_recommendations_ir.htm", "headings_path": "Розділ 2: Етап підготовки > п. 5"}
[retrieve_context] DOC 10 | score=0.2958793640136719 | metadata={"jurisdiction": "UA", "doc_number": "570", "section": 1, "section_name": "Використання Правил", "source": "methodics_recommendations_ir.htm", "headings_path": "Розділ 2: Етап підготовки > п. 5"}
[retrieve_context] DOC 11 | score=0.30993324518203735 | metadata={"language": "uk", "start_index": 0, "jurisdiction": "UA", "page": 2, "to", "source": "methodics_recommendations_ir.htm", "headings_path": "Розділ 2: Етап підготовки > п. 5"}
[retrieve_context] DOC 12 | score=0.31002217531204224 | metadata={"source": "methodics_recommendations_ir.htm", "doc_title": "Про затвердж", "source": "methodics_recommendations_ir.htm", "headings_path": "Розділ 2: Етап підготовки > п. 5"}
[retrieve_context] DOC 13 | score=0.2740306258201599 | metadata={"headings_path": "Розділ 2: ОСНОВНІ ЗАСАДИ ДЕРЖАВНОЇ ПОЛІТИКИ У СФЕРІ ЗАХ", "source": "methodics_recommendations_ir.htm", "headings_path": "Розділ 2: Етап підготовки > п. 5"}
[retrieve_context] DOC 14 | score=0.2802676558494568 | metadata={"source": "methodics_recommendations_ir.htm", "clause": "4", "jurisdiction": "UA", "doc_number": "570", "source": "methodics_recommendations_ir.htm", "headings_path": "Розділ 2: Етап підготовки > п. 5"}
[retrieve_context] DOC 15 | score=0.28118428587913513 | metadata={"language": "uk", "section_name": "Загальні положення", "doc_number": "570", "source": "methodics_recommendations_ir.htm", "headings_path": "Розділ 2: Етап підготовки > п. 5"}
[retrieve_context] DOC 16 | score=0.19985556602478027 | metadata={"source": "methodics_recommendations_ir.htm", "doc_number": "570", "clau", "source": "methodics_recommendations_ir.htm", "headings_path": "Розділ 2: Етап підготовки > п. 5"}
[retrieve_context] DOC 17 | score=0.2027049958705902 | metadata={"jurisdiction": "UA", "headings_path": "Розділ 2: Етап підготовки > п. 5", "source": "methodics_recommendations_ir.htm", "headings_path": "Розділ 2: Етап підготовки > п. 5"}
[retrieve_context] DOC 18 | score=0.20539778470993042 | metadata={"jurisdiction": "UA", "clause": "9", "source": "methodics_recommendation", "source": "methodics_recommendations_ir.htm", "headings_path": "Розділ 2: Етап підготовки > п. 5"}
[retrieve_context] DOC 19 | score=0.30031710863113403 | metadata={"section": 3, "source": "methodics_recommendations_ir.htm", "jurisdictio", "source": "methodics_recommendations_ir.htm", "headings_path": "Розділ 2: Етап підготовки > п. 5"}
[retrieve_context] DOC 20 | score=0.30300837755203247 | metadata={"source": "methodics_recommendations_ir.htm", "headings_path": "Розділ 2: Етап підготовки > п. 5"}
```

Рисунок 3.8 - Приклад отриманих документів

3.1.2 Генерування фінального JSON з полями звіту

Тепер, коли отримано всі необхідні дані, їх можна передавати у LLM разом з описом інцидента від користувача. Для уточнення задачі та покращення результатів використовуються також системний промпт та звичайний промпт у вигляді повідомлення від людини (клас `HumanMessage`). В них прописуються всі вимоги, обмеження, а також інструкції з генерації JSON, які створюються автоматично з Pydantic моделі за допомогою функції `get_format_instructions`. Більш детально задання схеми фінальної відповіді було розглянуто в попередніх розділах. Код генерації JSON для варіанту з українським законодавством наведено на рис. 3.9.

```

700 def generate_report(state: IncidentState) -> IncidentState:
706     logger.info(
707         f"[generate_report] Start | jurisdiction={jurisdiction}, "
708         f"questions={len(questions)}, context_len={len(context)}"
709     )
710
711     if jurisdiction not in ("UA", "EU"):
712         logger.warning(f"[generate_report] Unknown jurisdiction={jurisdiction}, defaulting to UA")
713         jurisdiction = "UA"
714
715     questions_block = (
716         "\n".join(f"- {q}" for q in questions)
717         if questions else "(no semantic queries generated)"
718     )
719
720     chain = report_prompt | llm | report_parser
721
722     raw = chain.invoke({
723         "jurisdiction": jurisdiction,
724         "incident_description": incident,
725         "questions_block": questions_block,
726         "context": context,
727     })
728     logger.debug(f"[generate_report] Raw LLM+parser output type={type(raw)}")
729
730     if isinstance(raw, UAIncidentReport):
731         logger.info("[generate_report] Got UAIncidentReport object from JsonOutputParser")
732         report_obj = raw
733         clean_json = report_obj.model_dump_json(indent=2)
734         return {
735             "report": clean_json,
736             "report_json": clean_json,
737         }
738

```

Рисунок 3.9 - Фрагмент функції створення звіту

Після того, як LLM згенерувала фінальний JSON, він проходить валідацію на відповідність заданій схемі, оскільки для автоматизації створення PDF звіту та MISP події важливе дотримання правильної структури та значень полів. В разі виникнення помилок під час валідації, вони виводяться користувачу, який повинен оцінити можливість створення звіту та його якість з урахуванням помилок, допущених LLM. Якщо помилки некритичні, можна продовжити виконання програми. Якщо користувач вважає, що отриманий JSON зовсім не відповідає заданій схемі, виконання можна перервати і спробувати ще раз з іншою моделлю або відредагувати промпт чи свій опис інциденту. Код реалізації цього механізму наведено на рис. 3.10.

```

700 def generate_report(state: IncidentState) -> IncidentState:
764     }
765
766     except ValidationError as e:
767         logger.error("[generate_report] Validation error for UAIncidentReport")
768         print("\n" + "=" * 80)
769         print("UAIncidentReport Pydantic validation errors")
770         print("=" * 80)
771         print(e)
772         print("-" * 80)
773
774         errors_list = e.errors()
775         print(f"Total validation errors: {len(errors_list)}")
776         print("=" * 80 + "\n")
777
778         while True:
779             answer = input(
780                 "Validation failed. Do you want to CONTINUE with the UNVALIDATED JSON "
781                 "from the model? [y/n]: "
782             ).strip().lower()
783
784             if answer in ("y", "yes"):
785                 print("Continuing with UNVALIDATED JSON from the model...\n")
786                 return {
787                     "report": raw_json_text,
788                     "report_json": raw_json_text,
789                 }
790
791             if answer in ("n", "no"):
792                 logger.error("Aborting execution due to validation errors.\n")
793                 raise RuntimeError("User aborted due to UAIncidentReport validation errors.")
794
795         print("Please answer 'y' or 'n'.")

```

Рисунок 3.10 - Код валідації результату JSON

Таким чином JSON, отриманий в ході роботи вузла вже готовий до передавання в MCP інструменти для генерації PDF звіту MISP події з індикаторами компрометації. В разі роботи в режимі законодавства ЄС, створюється лише MISP подія, а JSON можна використовувати для написання звіту вже згідно рекомендацій конкретної країни (для цього можна створити нові вузли у графі).

3.1.3 MCP інструменти

Для того щоб за допомогою згенерованого JSON отримати PDF звіт та подію у MISP, необхідно надати LLM інструменти, які будуть виконувати

відповідні функції. Це було реалізовано за допомогою MCP, принцип роботи цього протоколу було розкрито в попередніх розділах.

Для реалізації взаємодії необхідно створити функції для виклику інструментів в основному графі, і функції, які безпосередньо будуть виконувати роботу. Розглянемо принцип роботи на прикладі використання MCP інструмента створення події у MISP.

1. Викликається основна функція вузла графа, яка ініціює процес створення події і передає JSON, згенерований на попередніх вузлах. Цю функцію наведено на рис. 3.11.

```

962 async def create_misp_event_in_misp(state: IncidentState) -> IncidentState:
963     logger.info('[create_misp_event_in_misp] function started')
964     event_json = state.get("misp_event_json")
965
966     if not event_json:
967         logger.error("[create_misp_event_in_misp] No misp_event_json in state")
968         return {
969             "misp_create_result": {
970                 "status": "error",
971                 "stage": "no_event_json",
972                 "error": "misp_event_json is missing in state",
973             }
974         }
975
976     logger.info("[create_misp_event_in_misp] Sending event to MISP via MCP")
977     try:
978         res = await call_misp_mcp(event_json, publish=False)
979     except Exception as e:
980         logger.exception("[create_misp_event_in_misp] Exception while calling MISP MCP")
981         res = {
982             "status": "error",
983             "stage": "client_exception",
984             "error": repr(e),
985         }
986
987     logger.info("[create_misp_event_in_misp] MISP MCP result received")
988     logger.debug(f"[create_misp_event_in_misp] Result: {res}")
989     return {"misp_create_result": res}

```

Рисунок 3.11 - Код створення MISP події у графі

2. Друга функція відповідає за звернення до MCP сервера. Вона створює екземпляр класу MultiServerMCPClient. Це адаптер, який дозволяє моделі підключатися до MCP серверів та отримувати інструменти, які знаходяться на підключених серверах. Далі шукається потрібна

функція за іменем і викликається з необхідними аргументами. Код функції наведено на рис. 3.12.

```

async def call_misp_mcp(event_json: str, publish: bool = False) -> Dict[str, Any]:
    logger.info(f"[call_misp_mcp] Calling MISP MCP server, publish={publish}")
    logger.debug(f"[call_misp_mcp] Event JSON preview:\n{event_json[:500]}")
    client = MultiServerMCPClient(
        {
            "misp": {
                "command": "python",
                "args": [MISP_SERVER_PATH],
                "transport": "stdio",
            }
        }
    )

    tools = await client.get_tools()
    logger.info(f"[call_misp_mcp] Available tools: {[t.name for t in tools]}")

    create_tool = next(
        (t for t in tools if "create_misp_event" in t.name),
        None
    )
    if create_tool is None:
        logger.error("[call_misp_mcp] Tool 'create_misp_event' not found")
        return {
            "status": "error",
            "stage": "tool_lookup",
            "error": "Tool 'create_misp_event' not found in MCP server.",
        }

    res = await create_tool.ainvoke({
        "event_json": event_json,
        "publish": publish,
    })
    logger.info("[call_misp_mcp] MCP call finished")
    logger.debug(f"[call_misp_mcp] MCP result: {res}")

```

Рисунок 3.12 - Код звернення до MCP сервера

3. Остання функція, яка власне виконує корисну роботу, тобто створює подію у MISP на основі отриманих даних за допомогою бібліотеки `rumisp`, знаходиться в окремому файлі MCP сервера. Для кожного MCP сервера з інструментами використовується окремий файл. Код функції наведено на рис. 3.13.

```

22  async def create_misp_event(event_json: str) -> Dict[str, Any]:
36      ,
37
38      event_dict = event_obj.model_dump(exclude_none=True)
39      attributes = event_dict.pop("attributes", [])
40
41      try:
42          misp = get_misp_client()
43          response = misp.add_event(event_dict)
44
45          if not isinstance(response, dict) or "Event" not in response:
46              return {
47                  "status": "error",
48                  "stage": "misp_add_event",
49                  "error": f"Unexpected response from MISP: {response!r}",
50              }
51
52          created_event = response["Event"]
53          event_id = created_event.get("id")
54          event_uuid = created_event.get("uuid")
55
56          added_attributes_info = []
57          if event_id and attributes:
58              for idx, attr in enumerate(attributes):
59                  try:
60                      attr_resp = misp.add_attribute(event_id, attr)
61                      if isinstance(attr_resp, dict) and "Attribute" in attr_resp:
62                          added_attributes_info.append({
63                              "index": idx,
64                              "status": "ok",
65                              "attribute_id": attr_resp["Attribute"].get("id"),
66                              "attribute_uuid": attr_resp["Attribute"].get("uuid"),
67                          })

```

Рисунок 3.13 - Код функції створення події на MCP сервері

Аналогічним чином все працює і для MCP сервера з інструментом створення PDF звіту. Відрізняється лише код, який виконує корисну роботу.

При реалізації цього інструменту виникла проблема заповнення PDF файлу значеннями з отриманого JSON документу, оскільки шаблон звіту має власну структуру і поля місця полів для заповнення відрізняються для різних полів. Для вирішення цього питання було шаблон було перероблено в так званий «fillable PDF». Тобто за допомогою редактора на місця полів у шаблоні було додано інтерактивні поля, куди користувач може вводити текст. Це можна робити вручну, відкривши PDF файл, а також існують Python бібліотеки, які дозволяють

Таким чином заповнюється весь PDF звіт, за допомогою використання Pydantic моделі та валідації можна зробити чіткий мапінг полів і уникнути помилок через неіснуюче значення. Всі варіанти виборів в шаблоні прописані в моделі як єдині можливі значення для конкретного поля.

3.2 Застосування автоматизації на тестових інцидентах та оцінювання результатів

Було виконано тести на моделях, наведених в попередніх розділах. Кожну модель було перевірено на 3 інцидентах різного рівня деталізованості. Опис одного з інцидентів, на яких проводилися тестування, наведено на рис. 3.16.

```
medium.txt
1 4 березня 2025 року близько 10:25 SOC аналітик держпідприємства в енергетичному секторі отримав сповіщення EDR про спробу запуску шкідливого виконуваного файлу upda
2 Хеш файлу (MD5): a7c3f9e4bd55ab10e3c4d98122a011ef.
3 Також EDR зафіксував мережеві з'єднання до зовнішньої IP-адреси 91.245.77.9 на порт 443, які класифіковано як можливий C2-трафік.
4 Перевірка системних журналів показала наявність уразливості CVE-2023-21752 у встановленій версії Windows, яка могла бути використана для підвищення привілеїв.
5 Постраждала організація:
6 ДП "ЕнергоСервіс", Київ, вул. Хрещатик 22.
7 Контактна особа: Олена Данилюк, фахівець ІБ, телефон +380501234567, email: o.danyliuk@example.com
8 Репортер інциденту:
9 Ігор Паламарчук, аналітик SOC, email: i.palamarchuk@example.com, телефон +380671112233.
10 Додаткові дії: SOC ізолював робочу станцію від мережі приблизно о 10:37. Дані про подальшу ескаляцію або інші уражені системи відсутні.
11 Інших організацій не залучали, CERT-UA не повідомляли.
12 Компрометацію вдалося локалізувати, але остаточний аналіз ще триває.]
```

Рисунок 3.16 - Приклад опису кіберінциденту

Збиралися фінальні JSON для української та європейської гілок. Приклад заповненого фрагменту JSON для української гілки наведено на рис. 3.17.

```
=====
INCIDENT REPORT JSON
=====
{
  "tlp": "AMBER",
  "detection_datetime": "2025-01-15 03:42:00",
  "reporter": {
    "first_name": "Олександр",
    "last_name": "Ситник",
    "position": "старший аналітик SOC",
    "email": "o.sytnyk@medsoft.ua",
    "phone": "+380931112233",
    "cybersecurity_subject_name": "ТОВ \"МедСофт Сервіс\""
  },
  "detection_source": "other",
  "detection_source_other": "Система моніторингу компанії (SOC/SIEM)",
  "criticality": "3",
  "general_summary": "15 січня 2025 року о 03:42 система моніторингу ТОВ \"МедСофт Сервіс\" зафіксувала аномалії у роботі сервери сервери: сервер застосунок, сервер баз даних та мережевий шлюз. Зловмисники використали відомі вразливості для віддаленого зміни конфігурації мережі, що призвело до порушення доступності електронної медичної системи та створило ризик витоку персональних даних.",
  "victim": {
    "legal_name": "ТОВ \"МедСофт Сервіс\"",
    "address": null,
    "email": null,
    "phone": null,
    "contact_persons": [
      {

```

Рисунок 3.17 - Приклад отриманого JSON

Приклад атрибутів, виділених моделлю з опису інциденту та додано в MISP подію в якості атрибутів, наведено на рис. 3.18.

				(windows server 2019), використана для віддаленого виконання коду									
☐	2025-12-13	External analysis	vulnerability	CVE-2022-26809			Уразливість на ems-app-01 (Windows Server 2019), використана для віддаленого виконання коду	☒	☐	Inherit		(0/)	
☐	2025-12-13	External analysis	vulnerability	CVE-2024-23897			Уразливість на db-med-02 (Ubuntu Server 22.04 LTS), експлуатована для компрометації сервера БД	☒	☐	Inherit		(0/)	
☐	2025-12-13	External analysis	vulnerability	CVE-2023-46604			Уразливість на db-med-02, використана для віддаленого виконання коду через брокер повідомлень	☒	☐	Inherit		(0/)	
☐	2025-12-13	External analysis	vulnerability	CVE-2022-40620			Уразливість на gateway-01 (Debian 11), пов'язана з помилками конфігурації та обходом фільтрації	☒	☐	Inherit		(0/)	
☐	2025-12-13	External analysis	vulnerability	CVE-2023-28712			Уразливість на gateway-01 (Debian 11), пов'язана з некоректною обробкою пакетів	☒	☐	Inherit		(0/)	
☐	2025-12-13	Other	comment	Скомпрометовані системи: ems-app-01.medsoft.local (172.16.10.21), db-med-02.medsoft.local (172.16.10.45), gateway-01 (192.168.1.1). Інцидент пов'язаний з попереднім MS-2024-1210-A від 10.12.2024.			Контекст інциденту: 3 скомпрометовані сервери, зв'язок з попередньою кампанією	☒	☐	Inherit		(0/)	
☐	2025-12-13	Network activity	ip-src	203.0.113.77			Джерело сканування портів внутрішньої мережі	☒	1684	☒	Inherit		(0/)

Рисунок 3.18 - Приклад створеної події у MISP

Всі результати та критерії було надано експертам для оцінки. Далі їх було агреговано в 3 таблиці, кожна з яких відповідає інциденту, з яким працювали моделі. Умовно їх можна назвати «малий», «середній» та «великий». Фінальні середні оцінки від усіх експертів для малого інциденту наведено в таблиці 3.1.

Таблиця 3.1 - Порівняння результатів для малого інциденту

Модель	Якість заповнення звіту	Повнота заповнення звіту	Правильність створення MISP події	Якість JSON для ЄС	Якість MISP події для ЄС
GPT-4.1	4	3.67	4.67	4.33	4.33
GPT-5.2	4	4	5	4	4.67

Кінець таблиці 3.1

Модель	Якість заповнення звіту	Повнота заповнення звіту	Правильність створення MISP події	Якість JSON для ЄС	Якість MISP події для ЄС
Claude Sonnet 4.5	4.33	4.67	4.67	4.33	4.67
Claude 3.5 Haiku latest	3.67	3.67	5	4	4.33
Grok 4.1 (fast)	3.67	3.33	4.67	4.33	5
Grok 4.1 (reasoning)	4	3.67	5	4.33	5
LLaMA 4 Scout 17B	2.67	2.67	3	2.67	3.33
DeepSeek chat	3.33	3.33	5	4.33	4.67
DeepSeek reasoner	3.33	4	5	4	4.67
Gemini (fast)	3.67	4.33	3.67	4	3.67
Gemini (reasoning)	4	4.33	4.67	4.67	4.67

Результати для середнього інциденту наведено в таблиці 3.2.

Таблиця 3.2 - Порівняння результатів для середнього інциденту

Модель	Якість заповнення звіту	Повнота заповнення звіту	Правильність створення MISP події	Якість JSON для ЄС	Якість MISP події для ЄС
GPT-4.1	4	4	4	4.33	4.67
GPT-5.2	4	4.67	4.67	4	4.33

Кінець таблиці 3.2

Модель	Якість заповнення звіту	Повнота заповнення звіту	Правильність створення MISP події	Якість JSON для ЄС	Якість MISP події для ЄС
Claude Sonnet 4.5	5	4.67	5	4.67	5
Claude 3.5 Haiku latest	3.33	3.67	5	3.33	3.67
Grok 4.1 (fast)	3	3	4	3.33	3.67
Grok 4.1 (reasoning)	4	4	4	4	4.67
LLaMA 4 Scout 17B	2	3	3.33	3	3.33
DeepSeek chat	3	3.33	4	4	4
DeepSeek reasoner	3.33	4	4.33	4	5
Gemini (fast)	4	4	4.33	3.67	4
Gemini (reasoning)	4	4.33	4.67	4.33	5

Агреговані результати для великого інциденту від усіх експертів наведено в таблиці 3.3.

Таблиця 3.3 - Порівняння результатів для великого інциденту

Модель	Якість заповнення звіту	Повнота заповнення звіту	Правильність створення MISP події	Якість JSON для ЄС	Якість MISP події для ЄС
GPT-4.1	4	3	4.33	4	4
GPT-5.2	3.67	3.33	4	4	4

Кінець таблиці 3.3

Модель	Якість заповнення звіту	Повнота заповнення звіту	Правильність створення MISP події	Якість JSON для ЄС	Якість MISP події для ЄС
Claude Sonnet 4.5	5	4	5	4.67	5
Claude 3.5 Haiku latest	3.67	3.67	4.67	3.67	4
Grok 4.1 (fast)	3.33	3.33	4.33	3.67	4
Grok 4.1 (reasoning)	4	3.67	4.33	4	4.67
LLaMA 4 Scout 17B	3	2.33	4	2.67	3
DeepSeek chat	3	3.33	4	3.67	4
DeepSeek reasoner	3.67	4	4.33	4	5
Gemini (fast)	3.67	3.67	4	3	4
Gemini (reasoning)	4	4	4.67	4	4.67

Таким чином, за допомогою експертного методу було отримано порівняльні таблиці з оцінками для моделей, на основі яких можна робити висновки та приймати рішення щодо основної моделі, яку можна використовувати для застосунку для досягнення максимальної якості результатів та мінімальних витрат.

В якості додаткових чинників було також зібрано дані щодо кількості помилок валідації даних, які генерувала кожна модель, а також загальних витрат, які пішли на прогін всіх тестових інцидентів. Кожна модель обробила рівну кількість даних. Ці відомості наведено в таблиці 3.4.

Таблиця 3.4 - Порівняння кількості помилок та витрат для моделей

Модель	Кількість помилок під час тестів	Загальні витрати на всі тести
GPT-4.1	2	\$1.37
GPT-5.2	0	
Claude 3.5 Haiku latest	2	\$0.21
Claude Sonnet 4.5	1	\$0.88
Grok 4.1 (fast)	1	\$0.0219
Grok 4.1 (reasoning)	1	\$0.0552
LLaMA 4 Scout 17B	3	\$0
DeepSeek chat	1	\$0.04
DeepSeek reasoner	1	\$0.06
Gemini (fast)	0	\$1.12
Gemini (reasoning)	0	

В комбінації з оцінками експертів ця таблиця дозволяє оцінити моделі в рамках задачі генерування звітів на основі опису інциденту від користувача. За результатами порівняння було з'ясовано, що кожна модель має свої переваги та недоліки. Фінальні висновки та порівняння наведено в наступному розділі.

3.3 Порівняння ефективності та вартості моделей

Використовуючи таблиці, наведені в попередньому розділі, ми можемо комплексно оцінити якість роботи моделей, а також виділити інші тренди, що спостерігаються за результатами тестів. Важливо оцінювати якість роботи моделі у контексті ціни її використання. Бо найдорожча модель може бути не виправданим вибором для багатьох організацій через обмеження витрат на threat

intelligence або незначний вплив застосунку та його результатів на роботу компанії.

Згідно таблиць, в чистій ефективності та якості найкращою виявилася модель Claude Sonnet 4.5. Вона надала повні звіти про інциденти, які відображали відомості, що були надані користувачем. Крім того, ця модель єдина з тих, що тестувалися, змогла ідентифікувати CVE, надані в описі в якості атрибутів і коректно додати їх у MISP. Всі інші моделі ігнорували ідентифікатори CVE в контексті роботи з MISP. Проте ця модель виявилася однією з найдорожчих у використанні.

Трохи гірше, але все одно дуже добре показали себе моделі GPT. Вони видали найменше помилок, високі показники за всіма критеріями. Проте з мінусів можна зазначити найвищу ціну, а також обмежене контекстне вікно для GPT 5.2. Ця модель не могла коректно працювати з великим інцидентом через його розмір, тому опис довелося зменшувати.

Хорошою альтернативою GPT є моделі Gemini. Вони показали результати майже на рівні GPT та Claude, не припустилися жодної помилки. Ціна цих моделей помітно вища за Deepseek та Grok, проте це виправдано якістю результатів. Якщо організація за якихось причин не може використовувати моделі від Anthropic, Gemini є чудовим варіантом.

Grok та Deepseek показали себе трохи гірше моделей, описаних вище, проте ціна їхнього використання значно менша ніж у конкурентів, що може зробити саме ці моделі початковим вибором для багатьох організацій, які потім зможуть переключитись на дорожчі варіанти.

Очікувано, найгірше себе показала безкоштовна модель LLaMA 4 Scout 17B. Вона гарно підходить для тестування та розробки застосунку, проте фінальні результати в тестах у неї найгірші. Вона припускається найбільшої кількості помилок, ігнорує інструкції в промптах. Навіть безкоштовність моделі не виправдовує її використання в процесах організації, що не відносяться до розробки та початкового тестування програми.

Із загальних тенденцій можна помітити, що reasoning моделі, очікувано, показують себе краще за fast або non-reasoning варіанти від того самого вендора. Проте працюють вони помітно повільніше, але для задач застосунку час виконання не є критичним показником. Тому єдиною перевагою fast, або non-reasoning моделей в цьому порівнянні є нижча ціна використання.

Також можна відзначити, що в середньому всі моделі краще впоралися з роботою з MISP та додаванням подій, тоді як в генерації JSON для обох варіантів законодавства (Україна та ЄС) моделі показали себе помітно гірше, бо і сама задача є складнішою.

Загалом, за співвідношенням ціни та якості результатів, найкращими виявилися Claude Sonnet 4.5 та Gemini 2.5 pro. З урахуванням того, що застосунок не буде використовуватися часто, бо не всі інциденти потребують ескалації на CERT або CSIRT, використання Claude Sonnet 4.5 є виправданим. В разі частішого використання можна розглянути моделі Grok або DeepSeek для заощадження коштів без значної втрати в якості.

Висновки до розділу 3

В даному розділі було наведено деталі щодо роботи автоматизованого рішення, наведено ключові фрагменти коду, що відповідають за основні функції. Також показано зразки даних, що передаються між компонентами під час роботи: пошукові запити для векторної бази даних, отримані з БД фрагменти тексту разом з метаданими, фінальні JSON, які генерує модель, приклад заповненого PDF шаблону звіту про інцидент та подію MISP разом з доданими атрибутами.

Було проведено тести автоматизації з популярними моделями, виконано порівняння з використанням експертного методу за 5 критеріями. Також було наведено окрему таблицю з кількістю помилок валідації та загальною вартістю тестів для кожної моделі.

За результатами тестування було з'ясовано, що найкращими моделями для використання в застосунку, який було створено в роботі, є Claude Sonnet 4.5 та

Gemini 2.5 Pro. Вибирати між ними потрібно в залежності від потреб компанії, грошових ресурсів, а також частоти використання рішення. Інші моделі або показують помітно гірші результати, або є значно дорожчими. Безкоштовну модель LLaMA 4 Scout 17B має сенс застосовувати виключно під час розробки та тестування в якості тимчасового варіанту.

ВИСНОВКИ

В ході роботи було виконано всі поставлені задачі задачі:

1. Шаблон, побудований за вимогами нормативно правових актів України та ЄС, дає змогу конкретизувати інструкції до LMM в частині наступних умов: необхідні та опціональні поля, допустимі значення полів.
2. Pydantic моделі для звітів, запропоновані в роботі, слугують для генерації та валідації JSON відповіді від мовної моделі, що дозволило автоматизувати валідацію, а також значно скоротити довжину промπτу для LLM зі збереженням схеми відповіді з 839 рядків до 296 рядків.
3. Використання RAG-пайплайну та перетворених в ембедінги нормативно-правових актів дозволило моделі більш чітко слідувати заданим законодавчим нормам з урахуванням даних, отриманих від користувача без необхідності вручну прописувати все в промπτі, що покращило якість звітів, які генеруються застосунком.
4. Програмна модель, створена для дослідження, дозволяє автоматично генерувати звіт, уникати ручного заповнення характеристик інциденту, ручної взаємодії з MISF, що забезпечує зниження часу створення звіту з годин до декількох хвилин і впровадження обміну даними про загрози в процеси організації без значних часових та грошових витрат.
5. В результаті порівняння роботи застосунку на основі останніх версій моделей від популярних провайдерів було визначено, що найбільш ефективною є модель Claude Sonnet 4.5, яка забезпечує максимальну якість звіту, проте її використання коштує дорого. Хорошою альтернативою є модель Gemini 2.5 pro, яка не значно поступається в якості. За ними за результатами тестів йде Grok, що видає результати трохи гірші за попередні дві моделі, проте вирізняється помітно меншою ціною. Гірше у співвідношенні ціни та якості себе показали моделі GPT та DeepSeek. GPT виявилися невиправдано дорогими, а

DeepSeek - надали результати, що є недостатньо якісними для впровадження в організації. Безкоштовна модель LLaма показала себе найгірше.

Результати роботи показують, як розвиток великих мовних моделей може впроваджуватися в існуючі процеси організації та при відносно невеликих витратах приносити значний результат, який також відобразиться на економічних показниках компанії. Продемонстровано, що вибір моделі, що використовується, значно впливає на результат, тому цьому необхідно приділяти значну увагу.

Подальші наукові дослідження в цьому напрямку є доцільними і будуть продовжуватися, оскільки тема машинного навчання сьогодні є вкрай популярною та актуальною, при правильному впровадженні LLM може повністю автоматизувати процеси в організаціях, причому не тільки в області Threat Intelligence.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Ahlberg C. The Threat Intelligence Handbook. Annapolis : CyberEdge Group, LLC. 141 с. URL:
[https://soclibrary.futa.edu.ng/books/The%20Threat%20Intelligence%20Handbook%20by%20Zane%20Pokornoy%20\(z-lib.org\).pdf](https://soclibrary.futa.edu.ng/books/The%20Threat%20Intelligence%20Handbook%20by%20Zane%20Pokornoy%20(z-lib.org).pdf) (дата звернення: 14.10.2025).
2. Cyber-Threat Intelligence for Security Decision-Making: A Review and Research Agenda for Practice / S. Ainslie та ін. Computers & Security. 2023. С. 16. URL: <https://doi.org/10.1016/j.cose.2023.103352> (дата звернення: 18.10.2025).
3. What are the Types of Cyberthreat Intelligence (CTI). URL:
<https://www.paloaltonetworks.com/cyberpedia/types-of-cyberthreat-intelligence> (дата звернення: 18.10.2025).
4. Cyber threat intelligence explained. URL:
<https://www.crowdstrike.com/en-us/cybersecurity-101/threat-intelligence/> (дата звернення: 16.10.2025).
5. The primary sources of cyberthreat intelligence. URL:
<https://www.anomali.com/blog/threat-intelligence-sources> (дата звернення: 20.10.2025).
6. Bunting D. Crushing False Positives: Supercharging SOC Efficiency with Smarter Threat Intel. URL:
<https://www.chaossearch.io/blog/false-positives-soc-threat-intelligence> (дата звернення: 23.11.2025).
7. Schröer S. L., Canevascini N., Pekaric I. The dark side of the web: towards understanding various data sources in cyber threat intelligence. 2025. 11 с. (Препринт. arXiv:2504.14235). URL:
<https://doi.org/10.48550/arXiv.2504.14235> (дата звернення: 17.10.2025).

8. What are Threat Intelligence Sources. URL:
<https://www.recordedfuture.com/threat-intelligence-101/intelligence-sources-collection/threat-intelligence-sources> (дата звернення: 23.10.2025).
9. J. Roberts S., Brown R. Intelligence-Driven Incident Response. Sebastopol : O'Reilly Media, Inc, 2017. 282 с.
10. Features of MISP, the open source threat sharing platform.
[misp-project.org. URL: https://www.misp-project.org/features/](https://www.misp-project.org/features/) (дата звернення: 30.10.2025).
11. Committee Specification 02. STIX Version 2.1. На заміну Committee Specification Draft 05. Вид. офіц. URL:
<https://docs.oasis-open.org/cti/stix/v2.1/cs02/stix-v2.1-cs02.html> (дата звернення: 30.10.2025).
12. STIX Introductory Walkthrough. URL:
<https://oasis-open.github.io/cti-documentation/stix/walkthrough> (дата звернення: 31.10.2025).
13. Threat Intelligence sorted out: understanding the standards of data exchange. URL:
<https://defensys.com/blog-posts/threat-intelligence-sorted-out-understanding-standards-data-exchange> (дата звернення: 31.10.2025).
14. Dulaunoy A., Iklody A. MISP core format. URL:
<https://www.misp-standard.org/rfc/misp-standard-core.html> (дата звернення: 30.11.2025).
15. Rutkowski A., Kadobayashi Y., Furey I. CYBEX The Cybersecurity Information Exchange Framework. ACM SIGCOMM Computer Communication Review. 2010. Т. 40, № 5. С. 6. URL:
https://www.scribd.com/document/824369625/CYBEX-the-cybersecurity-information-exchange-framework?language_settings_changed=English (дата звернення: 31.10.2025).
16. Sadique F., Astaburuaga I., Kaul R. Cybersecurity Information Exchange with Privacy (CYBEX-P) and TAHOE. 2021. 27 с. (Препринт.

- arXiv:2106.01632 cybex). URL:
<https://doi.org/10.48550/arXiv.2106.01632> (дата звернення: 01.11.2025).
17. Traffic light protocol (tlp). URL: <https://www.first.org/tlp/> (дата звернення: 02.11.2025).
18. Traffic light protocol 2.0 user guide. 11 с. URL:
https://www.cisa.gov/sites/default/files/2023-02/tlp-2-0-user-guide_508c.pdf.
19. Nduhiu J. The computer security incident response team complete guide. URL:
https://www.splunk.com/en_us/blog/learn/csirt-computer-security-incident-response-team.html (дата звернення: 05.11.2025).
20. Про затвердження Методичних рекомендацій щодо реагування суб'єктами забезпечення кібербезпеки на різні види подій у кіберпросторі : Наказ від 03.07.2023 № 570. URL:
<https://zakon.rada.gov.ua/rada/show/v0570519-23#Text> (дата звернення: 04.11.2025).
21. Про рішення Ради національної безпеки і оборони України від 14 травня 2021 року "Про Стратегію кібербезпеки України" : Указ від 26.08.2021. URL: <https://zakon.rada.gov.ua/laws/show/447/2021#n12> (дата звернення: 13.11.2025).
22. Про основні засади забезпечення кібербезпеки України : Закон України від 05.10.2017 № 2163-VIII. URL:
<https://zakon.rada.gov.ua/laws/show/2163-19#Text> (дата звернення: 12.11.2025).
23. Деякі питання реагування суб'єктами забезпечення кібербезпеки на різні види подій у кіберпросторі : Постанова № 299. URL:
<https://zakon.rada.gov.ua/laws/show/299-2023-п#Text> (дата звернення: 28.11.2025).
24. Computer Security Incident Handling Guide : Recommendations of the National Institute of Standards and Technology / P. Cichonski та ін.

- National Institute of Standards and Technology, 2012. URL:
<https://doi.org/10.6028/nist.sp.800-61r2> (дата звернення: 14.11.2025).
25. Dekker M., Vytogianni E. Technical guideline on incident reporting under the eecr. 2021. 32 c. URL:
<https://www.enisa.europa.eu/publications/enisa-technical-guideline-on-incident-reporting-under-the-eecc> (дата звернення: 14.11.2025).
26. Maj MSc M., Reijers R., Stikvoort D. Good Practice Guide for Incident Management. 2010. URL:
<https://www.enisa.europa.eu/publications/good-practice-guide-for-incident-management> (дата звернення: 23.11.2025).
27. Directive (EU) 2022/2555 of the European parliament and of the council : Directive від 14.12.2022. URL:
<https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32022L2555> .
28. Cyber Solidarity Act : Regulation від 19.12.2024. URL:
<https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32025R0038> (дата звернення: 28.11.2025).
29. Structured model outputs. URL:
<https://platform.openai.com/docs/guides/structured-outputs> (дата звернення: 30.11.2025).
30. The Best Threat Intelligence Tools & How to Automate Alert Enrichment with Torq. URL: <https://torq.io/blog/threat-intelligence-tools/> (дата звернення: 29.11.2025).
31. Gutowska A. What is the Model Context Protocol (MCP)?. URL:
<https://www.ibm.com/think/topics/model-context-protocol> (дата звернення: 20.11.2025).
32. Sharma V. Introduction to MCP: Model Context Protocol. URL:
<https://www.linkedin.com/pulse/introduction-mcp-model-context-protocol-vikas-sharma-ofp9c/> (дата звернення: 03.12.2025).

- 33.Leto A., Aguerrebere C., Bhati I. Toward Optimal Search and Retrieval for RAG. NeurIPS 2024 Workshop, 2024. 19 с. (Препринт. arXiv:2411.07396). URL: <https://doi.org/10.48550/arXiv.2411.07396> (дата звернення: 02.12.2025).
- 34.MTEB Embedding Leaderboard. URL: <https://huggingface.co/spaces/mteb/leaderboard> (дата звернення: 10.12.2025).

ДОДАТОК А

Лістинг основних функцій програми

```

print("Choose LLM:")
for key, (_, label) in LLM_MENU.items():
    print(f"[{key}] {label}")
chosen = input("Model number: ").strip()
logger.info(f"User selected model option: {chosen!r}")

llm = get_llm_from_menu(chosen)
logger.success(f"LLM object initialised: {llm}")

logger.info("Initialising HuggingFaceEmbeddings: intfloat/multilingual-e5-base")
embeddings = HuggingFaceEmbeddings(
    model_name="intfloat/multilingual-e5-base"
)

logger.info("Initialising Chroma vector store at 'chromadb/proddb'")
chroma_db = Chroma(
    persist_directory="chromadb/proddb",
    embedding_function=embeddings,
)

def generate_questions(state: IncidentState) -> IncidentState:
    incident = state["incident_description"]
    jurisdiction = state.get("jurisdiction", "UA").upper()
    logger.info(
        f"[generate_questions] jurisdiction={jurisdiction}, "
        f"incident_len={len(incident)}"
    )

```

```

if jurisdiction not in ("UA", "EU"):
    logger.warning(f"[generate_questions] Unknown jurisdiction={jurisdiction},
defaulting to UA")
    jurisdiction = "UA"

human_text = question_prompt.format(
    incident_description=incident,
    jurisdiction=jurisdiction,
)

messages = [
    SystemMessage(content=QUESTION_SYSTEM_PROMPT.strip()),
    HumanMessage(content=human_text),
]

raw_text = llm.invoke(messages).content
logger.debug(f"[generate_questions] Raw questions block:\n{raw_text}")

queries: List[str] = []
for line in raw_text.splitlines():
    q = line.strip()
    if q:
        queries.append(q)

logger.info(f"[generate_questions] Generated {len(queries)} semantic queries")
return {"questions": queries}

def retrieve_context(

```

```

state: IncidentState,
k_per_query: int = 3,
max_docs: int = 20,
) -> IncidentState:
    queries = state.get("questions", []) or []
    jurisdiction = state.get("jurisdiction", "UA").upper()
    logger.info(
        f"[retrieve_context] jurisdiction={jurisdiction}, "
        f"queries_count={len(queries)}, k_per_query={k_per_query},
max_docs={max_docs}"
    )

    if jurisdiction not in ("UA", "EU"):
        jurisdiction = "UA"

    if not queries:
        logger.warning("[retrieve_context] No queries provided, returning empty context")
        return {
            "context": "",
            "context_chunks": [],
        }

    filter_meta: Dict[str, Any] = {"jurisdiction": jurisdiction}

    collected: List[tuple[Document, float]] = []
    seen_keys = set()

    for q in queries:
        search_q = f"query: {q}"

```

```

docs_with_scores = chroma_db.similarity_search_with_score(
    search_q,
    k=k_per_query,
    filter=filter_meta,
)

for d, score in docs_with_scores:
    m = d.metadata or {}

    key = (
        m.get("source"),
        m.get("page"),
        m.get("headings_path") or m.get("doc_title"),
        d.page_content[:80],
    )
    if key in seen_keys:
        continue
    seen_keys.add(key)
    collected.append((d, score))

    if len(collected) >= max_docs:
        break

if len(collected) >= max_docs:
    break

context_blocks: List[str] = []
context_chunks: List[Dict[str, Any]] = []

logger.success(f"[retrieve_context] Collected {len(collected)} unique documents")

```

```

for i, (d, score) in enumerate(collected, start=1):
    m = d.metadata or {}

    text_clean = (d.page_content or "").replace("\n", " ")
    logger.debug(
        f"[retrieve_context] DOC {i} | score={score} | metadata={json.dumps(m,
ensure_ascii=False)} | text={text_clean}"
    )

    meta_parts: List[str] = [f"[{i}]"]
    meta_parts.append(f"score: {score:.4f}")

    FIELD_LABELS = {
        "doc_title": "Title",
        "source": "Source",
        "page": "page",
        "headings_path": None,
        "section": "section",
        "clause": "clause",
        "jurisdiction": "jurisdiction",
        "language": "lang",
    }

    for key, label in FIELD_LABELS.items():
        val = m.get(key)
        if val in [None, "unknown"]:
            continue
        if label is None:
            meta_parts.append(str(val))
        else:

```

```

        meta_parts.append(f'{label}: {val}')

    meta_line = " | ".join(meta_parts)

    block_text = f'{meta_line}\n{d.page_content}'
    context_blocks.append(block_text)

    context_chunks.append({
        "index": i,
        "text": d.page_content,
        "metadata": m,
        "score": score,
    })

    context_text = "\n\n".join(context_blocks)

    logger.debug(f'[retrieve_context] Built context length={len(context_text)} chars')
    return {
        "context": context_text,
        "context_chunks": context_chunks,
    }

def generate_report(state: IncidentState) -> IncidentState:
    incident = state["incident_description"]
    jurisdiction = state.get("jurisdiction", "UA").upper()
    questions = state.get("questions", [])
    context = state.get("context", "")

    logger.info(
        f'[generate_report] Start | jurisdiction={jurisdiction}, "

```

```

        f"questions={len(questions)}, context_len={len(context)}"
    )

    if jurisdiction not in ("UA", "EU"):
        logger.warning(f"[generate_report] Unknown jurisdiction={jurisdiction},
defaulting to UA")
        jurisdiction = "UA"

    questions_block = (
        "\n".join(f"- {q}" for q in questions)
        if questions else "(no semantic queries generated)"
    )

    chain = report_prompt | llm | report_parser

    raw = chain.invoke({
        "jurisdiction": jurisdiction,
        "incident_description": incident,
        "questions_block": questions_block,
        "context": context,
    })

    logger.debug(f"[generate_report] Raw LLM+parser output type={type(raw)}")

    if isinstance(raw, UAIncidentReport):
        logger.info("[generate_report] Got UAIncidentReport object from
JsonOutputParser")
        report_obj = raw
        clean_json = report_obj.model_dump_json(indent=2)
        return {
            "report": clean_json,

```

```

    "report_json": clean_json,
}

```

```

if isinstance(raw, dict):

```

```

    raw_json_text = json.dumps(raw, ensure_ascii=False, indent=2)

```

```

elif isinstance(raw, str):

```

```

    raw_json_text = raw

```

```

elif hasattr(raw, "content"):

```

```

    raw_json_text = str(raw.content)

```

```

else:

```

```

    raw_json_text = str(raw)

```

```

try:

```

```

    if isinstance(raw, dict):

```

```

        logger.debug(raw)

```

```

        report_obj = UAIncidentReport.model_validate(raw)

```

```

    elif isinstance(raw, str):

```

```

        logger.debug(raw)

```

```

        report_obj = UAIncidentReport.model_validate_json(raw)

```

```

    else:

```

```

        report_obj = UAIncidentReport.model_validate_json(raw_json_text)

```

```

clean_json = report_obj.model_dump_json(indent=2)

```

```

logger.info("[generate_report] Pydantic validation OK for UAIncidentReport")

```

```

return {

```

```

    "report": clean_json,

```

```

    "report_json": clean_json,

```

```

}

```

```

except ValidationError as e:
    logger.error("[generate_report] Validation error for UAIncidentReport")
    print("\n" + "=" * 80)
    print("UAIncidentReport Pydantic validation errors")
    print("=" * 80)
    print(e)
    print("-" * 80)

    errors_list = e.errors()
    print(f"Total validation errors: {len(errors_list)}")
    print("=" * 80 + "\n")

    while True:
        answer = input(
            "Validation failed. Do you want to CONTINUE with the UNVALIDATED
JSON "
            "from the model? [y/n]: "
        ).strip().lower()

        if answer in ("y", "yes"):
            print("Continuing with UNVALIDATED JSON from the model...\n")
            return {
                "report": raw_json_text,
                "report_json": raw_json_text,
            }

        if answer in ("n", "no"):
            logger.error("Aborting execution due to validation errors.\n")
            raise RuntimeError("User aborted due to UAIncidentReport validation
errors.")

```

```
print("Please answer 'y' or 'n'.")
```

```

async def call_misp_mcp(event_json: str, publish: bool = False) -> Dict[str, Any]:
    logger.info(f"[call_misp_mcp] Calling MISP MCP server, publish={publish}")
    logger.debug(f"[call_misp_mcp] Event JSON preview:\n{event_json[:500]}")
    client = MultiServerMCPClient(
        {
            "misp": {
                "command": "python",
                "args": [MISP_SERVER_PATH],
                "transport": "stdio",
            }
        }
    )

    tools = await client.get_tools()
    logger.info(f"[call_misp_mcp] Available tools: {[t.name for t in tools]}")

    create_tool = next(
        (t for t in tools if "create_misp_event" in t.name),
        None
    )
    if create_tool is None:
        logger.error("[call_misp_mcp] Tool 'create_misp_event' not found")
        return {
            "status": "error",
            "stage": "tool_lookup",
            "error": "Tool 'create_misp_event' not found in MCP server.",

```

```

    }

res = await create_tool.ainvoke({
    "event_json": event_json,
    "publish": publish,
})

logger.info("[call_misp_mcp] MCP call finished")
logger.debug(f"[call_misp_mcp] MCP result: {res}")

if isinstance(res, dict):
    return res
if hasattr(res, "dict"):
    return res.dict()
return {"status": "error", "stage": "unexpected_result_type", "result_repr": repr(res)}

async def call_pdf_mcp(report_json: str, basename: str | None = None) -> Dict[str,
Any]:
    logger.info(f"[call_pdf_mcp] Calling PDF MCP server, basename={basename!r}")
    client = MultiServerMCPClient(
        {
            "ua_pdf": {
                "command": "python",
                "args": [PDF_SERVER_PATH],
                "transport": "stdio",
            }
        }
    )

    tools = await client.get_tools()
    pdf_tool = next(

```

```

        (t for t in tools if "create_incident_pdf" in t.name),
        None,
    )

```

```

if pdf_tool is None:

```

```

    return {
        "status": "error",
        "stage": "tool_lookup",
        "error": "Tool 'create_incident_pdf' not found in MCP server.",
    }

```

```

args: Dict[str, Any] = {"report_json": report_json}

```

```

if basename:

```

```

    args["basename"] = basename

```

```

res = await pdf_tool.ainvoke(args)

```

```

logger.info("[call_pdf_mcp] MCP call finished")

```

```

logger.debug(f"[call_pdf_mcp] MCP result: {res}")

```

```

if isinstance(res, dict):

```

```

    return res

```

```

if hasattr(res, "dict"):

```

```

    return res.dict()

```

```

return {

```

```

    "status": "error",
    "stage": "unexpected_result_type",
    "result_repr": repr(res),
}

```

```

def generate_misp_event(state: IncidentState) -> IncidentState:

```

```

logger.info("[generate_misp_event] Start")
incident = state["incident_description"]
context = state.get("context", "")

chain = misp_event_prompt | llm | misp_event_parser

raw = chain.invoke({
    "incident_description": incident,
    "legal_context": context or "(no extra context provided)",
})
logger.debug(f"[generate_misp_event] Raw output type={type(raw)}")

if isinstance(raw, MispEventCreate):
    logger.info("[generate_misp_event] Got MispEventCreate from
JsonOutputParser")
    misp_event_obj = raw
    misp_event_json = misp_event_obj.model_dump_json(indent=2)
    return {
        "misp_event_json": misp_event_json,
    }

if isinstance(raw, dict):
    raw_json_text = json.dumps(raw, ensure_ascii=False, indent=2)
elif isinstance(raw, str):
    raw_json_text = raw
elif hasattr(raw, "content"):
    raw_json_text = str(raw.content)
else:
    raw_json_text = str(raw)

```

```

try:
    if isinstance(raw, dict):
        logger.debug(raw)
        misp_event_obj = MispEventCreate.model_validate(raw)
    elif isinstance(raw, str):
        logger.debug(raw)
        misp_event_obj = MispEventCreate.model_validate_json(raw)
    elif hasattr(raw, "content"):
        text = raw.content
        logger.debug(text)
        misp_event_obj = MispEventCreate.model_validate_json(text)
    else:
        misp_event_obj = MispEventCreate.model_validate_json(raw_json_text)

    misp_event_json = misp_event_obj.model_dump_json(indent=2)

    return {
        "misp_event_json": misp_event_json,
    }

except ValidationError as e:
    logger.error("[generate_misp_event] Pydantic validation error")
    print("\n" + "=" * 80)
    print("MispEventCreate Pydantic validation errors")
    print("=" * 80)
    print(e)
    print("-" * 80)

    errors_list = e.errors()
    print(f"Total validation errors: {len(errors_list)}")

```

```

print("=" * 80 + "\n")

while True:
    answer = input(
        "MISP event validation failed. Do you want to CONTINUE with the "
        "UNVALIDATED JSON from the model? [y/n]: "
    ).strip().lower()

    if answer in ("y", "yes"):
        print("Continuing with UNVALIDATED MISP JSON from the model...\n")
        return {
            "misp_event_json": raw_json_text,
        }

    if answer in ("n", "no"):
        print("Aborting execution due to MispEventCreate validation errors.\n")
        raise RuntimeError("User aborted due to MispEventCreate validation errors.")

    print("Please answer 'y' or 'n'.")

async def create_misp_event_in_misp(state: IncidentState) -> IncidentState:
    logger.info("[create_misp_event_in_misp] function started")
    event_json = state.get("misp_event_json")

    if not event_json:
        logger.error("[create_misp_event_in_misp] No misp_event_json in state")
        return {
            "misp_create_result": {
                "status": "error",
                "stage": "no_event_json",
            }
        }

```

```

        "error": "misp_event_json is missing in state",
    }
}

logger.info("[create_misp_event_in_misp] Sending event to MISP via MCP")
try:
    res = await call_misp_mcp(event_json, publish=False)
except Exception as e:
    logger.exception("[create_misp_event_in_misp] Exception while calling MISP
MCP")
    res = {
        "status": "error",
        "stage": "client_exception",
        "error": repr(e),
    }

logger.info("[create_misp_event_in_misp] MISP MCP result received")
logger.debug(f"[create_misp_event_in_misp] Result: {res}")
return {"misp_create_result": res}

async def create_pdf_from_report(state: IncidentState) -> IncidentState:
    logger.info("[create_pdf_from_report] function started")
    report_json = state.get("report_json") or state.get("report")
    if not report_json:
        logger.error("[create_pdf_from_report] No report_json in state")
        return {
            "pdf_result": {
                "status": "error",
                "stage": "no_report_json",
                "error": "report_json missing in state",
            }
        }

```

```

    }
}

```

```

logger.info("[create_pdf_from_report] Calling PDF MCP to create incident PDF")

```

```

try:

```

```

    res = await call_pdf_mcp(report_json)

```

```

except Exception as e:

```

```

    logger.exception("[create_pdf_from_report] Exception while calling PDF MCP")

```

```

res = {

```

```

    "status": "error",

```

```

    "stage": "client_exception",

```

```

    "error": repr(e),

```

```

}

```

```

pdf_path = ""

```

```

if isinstance(res, dict) and res.get("status") == "ok":

```

```

    pdf_path = res.get("final_path") or ""

```

```

    logger.info(f"[create_pdf_from_report] PDF created at: {pdf_path}")

```

```

return {

```

```

    "pdf_result": res,

```

```

    "pdf_path": pdf_path,

```

```

}

```

```

def generate_eu_report(state: IncidentState) -> IncidentState:

```

```

    incident = state["incident_description"]

```

```

    jurisdiction = state.get("jurisdiction", "EU").upper()

```

```

    questions = state.get("questions", [])

```

```

    context = state.get("context", "")

```

```

logger.info(
    f"[generate_eu_report] Start | jurisdiction={jurisdiction}, "
    f"questions={len(questions)}, context_len={len(context)}"
)

if jurisdiction != "EU":
    logger.warning(
        f"[generate_eu_report] jurisdiction={jurisdiction}, "
        f"but EU node was called. Proceeding anyway."
    )

questions_block = (
    "\n".join(f"- {q}" for q in questions)
    if questions else "(no semantic queries generated)"
)

chain = eu_report_prompt | llm | eu_report_parser

raw = chain.invoke({
    "incident_description": incident,
    "questions_block": questions_block,
    "context": context,
})

logger.debug(f"[generate_eu_report] Raw LLM+parser output type={type(raw)}")

if isinstance(raw, EUIncidentFullReport):
    full_obj = raw
else:
    if isinstance(raw, dict):
        raw_json_text = json.dumps(raw, ensure_ascii=False, indent=2)

```

```

elif isinstance(raw, str):
    raw_json_text = raw
elif hasattr(raw, "content"):
    raw_json_text = str(raw.content)
else:
    raw_json_text = str(raw)

try:
    if isinstance(raw, dict):
        logger.debug(raw)
        full_obj = EUIncidentFullReport.model_validate(raw)
    else:
        logger.debug(raw_json_text)
        full_obj = EUIncidentFullReport.model_validate_json(raw_json_text)
except ValidationError as e:
    logger.error("[generate_eu_report] Validation error for EUIncidentFullReport")
    print("\n" + "=" * 80)
    print("EUIncidentFullReport Pydantic validation errors")
    print("=" * 80)
    print(e)
    print("-" * 80)

    errors_list = e.errors()
    print(f"Total validation errors: {len(errors_list)}")
    print("=" * 80 + "\n")

    while True:
        answer = input(
            "EU full incident validation failed. Continue with UNVALIDATED JSON?
[y/n]: "

```

```

).strip().lower()

if answer in ("y", "yes"):
    print("Continuing with UNVALIDATED EU full JSON...\n")
    return {
        "report": raw_json_text,
        "report_json": raw_json_text,
    }
if answer in ("n", "no"):
    logger.error("Aborting due to EUIncidentFullReport validation errors.\n")
    raise RuntimeError("User aborted due to EUIncidentFullReport validation
errors.")
print("Please answer 'y' or 'n'.")

def apply_env_reporter(full: EUIncidentFullReport) -> None:
    basic = full.basic
    if not basic.reporters:
        logger.warning("[apply_env_reporter] No reporters[] in basic, nothing to patch")
        return

    reporter = basic.reporters[0]

    field_map = {
        "name":    EU_REPORTER_NAME,
        "role":    EU_REPORTER_ROLE,
        "org_unit": EU_REPORTER_ORG_UNIT,
        "affiliation": EU_REPORTER_AFFILIATION,
        "email":    EU_REPORTER_EMAIL,
        "phone":    EU_REPORTER_PHONE,
        "location": EU_REPORTER_LOCATION,
    }

```

```

    }

    for attr, env_val in field_map.items():
        if not env_val:
            continue
        current = getattr(reporter, attr, None)
        if current is None or (isinstance(current, str) and not current.strip()):
            setattr(reporter, attr, env_val)

    apply_env_reporter(full_obj)

    full_json = full_obj.model_dump_json(indent=2)
    logger.info("[generate_eu_report] Pydantic validation OK for
EUIncidentFullReport")

    return {
        "report": full_json,
        "report_json": full_json,
    }

def route_after_context(state: IncidentState) -> str:
    j = state.get("jurisdiction", "UA")
    if not isinstance(j, str):
        j = "UA"
    j = j.upper()
    return "EU" if j == "EU" else "UA"

def route_after_misp(state: IncidentState) -> str:
    j = state.get("jurisdiction", "UA")

```

```

if not isinstance(j, str):
    j = "UA"
j = j.upper()
return "UA" if j == "UA" else "EU"

```

```
graph = StateGraph(IncidentState)
```

```

graph.add_node("generate_questions", generate_questions)
graph.add_node("retrieve_context", retrieve_context)
graph.add_node("generate_report", generate_report)
graph.add_node("generate_eu_report", generate_eu_report)
graph.add_node("generate_misp_event", generate_misp_event)
graph.add_node("create_misp_event_in_misp", create_misp_event_in_misp)
graph.add_node("create_pdf_from_report", create_pdf_from_report)

```

```

graph.add_edge(START, "generate_questions")
graph.add_edge("generate_questions", "retrieve_context")

```

```

graph.add_conditional_edges(
    "retrieve_context",
    route_after_context,
    {
        "UA": "generate_report",
        "EU": "generate_eu_report",
    },
)

```

```

graph.add_edge("generate_report", "generate_misp_event")
graph.add_edge("generate_eu_report", "generate_misp_event")
graph.add_edge("generate_misp_event", "create_misp_event_in_misp")

```

```
graph.add_conditional_edges(
    "create_misp_event_in_misp",
    route_after_misp,
    {
        "UA": "create_pdf_from_report",
        "EU": END,
    },
)
```

```
graph.add_edge("create_pdf_from_report", END)
```

```
app = graph.compile()
```

```
async def main():
```

```
    logger.info("=== Incident pipeline start ===")
```

```
    j_input = input("Choose jurisdiction [UA/EU] (default UA): ").strip().upper()
```

```
    if j_input not in ("UA", "EU"):
```

```
        j_input = "UA"
```

```
    logger.info(f"User selected jurisdiction: {j_input}")
```

```
    incident_path = Path(__file__).with_name("incident.txt")
```

```
    try:
```

```
        incident_text = incident_path.read_text(encoding="utf-8").strip()
```

```
        logger.info(f"incident description: {incident_text}")
```

```
    except FileNotFoundError:
```

```
        logger.error(f"Incident file not found: {incident_path!s}")
```

```
        raise
```

```

if not incident_text:
    logger.error(f"Incident file {incident_path!s} is empty")
    raise RuntimeError("Incident description is empty – please fill incident.txt")

initial_state: IncidentState = {
    "incident_description": incident_text,
    "jurisdiction": j_input,
}

logger.info(
    f"Initial state: jurisdiction={initial_state['jurisdiction']}, "
    f"incident_len={len(initial_state['incident_description'])}"
)

result = await app.ainvoke(initial_state)
logger.info("=== Incident pipeline finished successfully ===")
logger.debug(f"Final state: {result}")

report_str = result.get("report_json")
if report_str:
    try:
        report_obj = json.loads(report_str)
        pretty_report = json.dumps(report_obj, indent=2, ensure_ascii=False)
    except json.JSONDecodeError:
        pretty_report = report_str

logger.info(
    "\n" + "=" * 80 +
    "\nINCIDENT REPORT JSON\n" +
    "=" * 80 +

```

```

        "\n{}",
        pretty_report,
    )

misp_str = result.get("misp_event_json")
if misp_str:
    try:
        misp_obj = json.loads(misp_str)
        pretty_misp = json.dumps(misp_obj, indent=2, ensure_ascii=False)
    except json.JSONDecodeError:
        pretty_misp = misp_str

    logger.info(
        "\n" + "=" * 80 +
        "\nMISP EVENT JSON\n" +
        "=" * 80 +
        "\n{}",
        pretty_misp,
    )

if __name__ == "__main__":
    asyncio.run(main())

```