МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра кібербезпеки

ПІДЛИСЬКИЙ Дмитро Андрійович

Інтегрована платформа розвідки загроз на основі плагіну Kibana / Integrated Threat Intelligence Platform Based on the Kibana Plugin

спеціальність: 125 – Кібербезпека та захист інформації

освітньо-професійна програма –Кібербезпека

Кваліфікаційна робота

Виконав студент групи КБм -21
Д.А. Підлиський

_____

Науковий керівник
д.т.н., професор В.В. Яцків

_____

Кваліфікаційну роботу допущено
до захисту:

«____» _____ ___2025 р.

Завідувач кафедри

_____ В.В.Яцків

ТЕРНОПІЛЬ - 2025

Факультет комп'ютерних інформаційних технологій
Кафедра кібербезпеки
Освітній ступінь «магістр»
спеціальність: 125 - Кібербезпека та захист інформації
освітньо-професійна програма –Кібербезпека

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____ В.В.Яцків
____" _____ 2024 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

ПІДЛИСЬКОМУ Дмитру Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи:
Інтегрована платформа розвідки загроз на основі плагіну Kibana / Integrated Threat Intelligence Platform Based on the Kibana Plugin
керівник роботи д.т.н., професор В.В. Яцків

затверджені наказом по університету від 20 грудня 2024 року № 938

2. Строк подання студентом закінченої кваліфікаційної роботи 5 грудня 2025року.
3. Вихідні дані до кваліфікаційної роботи: завдання на випускну кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:
– проаналізувати можливості Elastic Stack та плагіну Kibana для розвідки загроз;
– дослідити сучасні методи та інструменти побудови платформ розвідки загороз;
– розробити архітектуру та реалізувати інтегровану платформу розвідки загороз;
– провести тестування працездатності створеної платформи та оцінити її ефективність.

5. Перелік графічного матеріалу у роботі.
– схема взаємодії компонентів платформу розвідки загороз;
– інтерактивна панель платформи розвідки загороз;
– графічне представлення індикаторів компрометації.

6. Консультанти розділів кваліфікаційної роботи

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

7. Дата видачі завдання 20 грудня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строки виконання етапів кваліфікаційної роботи	Примітка
1	Аналіз платформи Kibana як інструмента аналітики та візуалізації	12.2024 р. – 03.2025 р.	
2	Реалізація інтегрованої платформи розвідки загроз	03.2025 р. – 05.2025 р.	
3	Дослідження інтегрованої платформи розвідки загроз на основі плагіну Kibana	05.2025 р. – 11.2025 р.	

Студент _____ Д.А. Підлиський
(підпис)

Керівник роботи _____ д.т.н., професор Яцків В.В.
(підпис)

АНОТАЦІЯ

Підлиський Д.А. Інтегрована платформа розвідки загроз на основі плагіну Kibana. – Рукопис.

Дослідження на здобуття освітнього ступеня «магістр» за спеціальністю 125 «Кібербезпека та захист інформації», освітньо-професійна програма «Кібербезпека». – Західноукраїнський національний університет, Тернопіль, 2025.

У роботі досліджено сучасні методи та інструменти розвідки загроз, проаналізовано функціональні можливості Elastic Stack і плагіну Kibana для оброблення індикаторів компрометації. Запропоновано архітектуру інтегрованої платформи розвідки загроз, що поєднує Elasticsearch, Kibana та модуль автоматизованого імпорту даних. Реалізовано прототип платформи, який забезпечує збір, індексацію, кореляцію та візуалізацію індикаторів загроз у режимі, наближеному до реального часу, та досліджено його ефективність у задачах моніторингу й аналітики загроз.

Ключові слова: ІНДИКАТОР КОМПРОМЕТАЦІЇ, РОЗВІДКА ЗАГРОЗ, ВІЗУАЛІЗАЦІЯ ДАНИХ, АНАЛІТИЧНІ ПЛАТФОРМИ, МОНІТОРИНГ БЕЗПЕКИ.

ABSTRACT

Pidlyskyi D.A. Integrated Threat Intelligence Platform Based on the Kibana Plugin. – Manuscript.

Doctoral studies for the education level «Master» with the title 125 «Cybersecurity and Information Protection». – West Ukrainian National University, Ternopil, 2025.

The work explores modern methods and tools for threat intelligence, analyzes the functional capabilities of Elastic Stack and the Kibana plugin for processing indicators of compromise. An integrated threat intelligence platform architecture is proposed, combining Elasticsearch, Kibana, and an automated data import module. A prototype of the platform has been implemented, enabling the collection, indexing, correlation, and visualization of threat indicators in near real-time, and its effectiveness in cyber monitoring and threat analytics tasks has been evaluated.

Keywords: INDICATOR OF COMPROMISE, THREAT INTELLIGENCE, DATA VISUALIZATION, ANALYTICAL PLATFORMS, SECURITY MONITORING.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ПРЗ – платформа розвідки загроз;

IoC – індикатор компрометації;

РЗ – розвідка загроз;

ІТ – інформаційні технології;

IDS/IPS – системи виявлення та запобігання вторгнень;

SIEM – система управління інформаційною безпекою;

SOC – операційний центр безпеки;

ECS - Elastic Common Schema;

ВСТУП

Актуальність теми. Сучасний розвиток IT супроводжується зростанням кількості кіберзагроз, які стають більш складними, динамічними та організованими [1-4]. Злочинці активно використовують бот-мережі, фішингові кампанії, вразливості нульового дня, автоматизовані сканери та інші інструменти атаки [5-8].

У таких умовах класичні засоби кіберзахисту такі як антивірусні системи, файрволи, IDS/IPS тощо, перестають бути достатніми для своєчасного виявлення та реагування [9-12]. Це зумовлює потребу у впровадженні платформ розвідки загроз (ПРЗ), які дозволяють отримувати індикатори компрометації (IoC), аналізувати їх та використовувати для попередження атак [13-17].

Інструменти розвідки загроз (РЗ) забезпечують збір, структурування та візуалізацію інформації щодо потенційно небезпечних IP-адрес, доменів, файлів, поведінкових моделей та інших ознак злочинної активності [18-21]. Однією з найпоширеніших технологічних платформ для побудови таких рішень є відкритий стек Elastic Stack [22], що включає Elasticsearch і Kibana та надає розширені можливості для швидкого дослідження даних, побудови аналітичних інструментів, формування звітності та виявлення аномалій.

Актуальність теми роботи визначається необхідністю створення доступних, гнучких та інтегрованих інструментів РЗ, які можуть використовуватися у корпоративному секторі. Особливого значення набувають рішення, що ґрунтуються на відкритих технологіях і не потребують дорогих комерційних ліцензій. Elastic Stack відповідає цим вимогам та дозволяє побудувати працездатну ПРЗ з можливістю подальшого розширення.

Мета і завдання дослідження. Метою роботи є розроблення інтегрованої ПРЗ на основі плагіну Kibana, яка забезпечує автоматизоване завантаження IoC, їх зберігання, оброблення та аналітичну візуалізацію.

Для досягнення поставленої мети необхідно виконати такі завдання:

- проаналізувати можливості Elastic Stack та плагіну Kibana для РЗ;
- дослідити сучасні підходи та інструменти побудови ПРЗ;
- розробити архітектуру інтегрованої платформи;
- реалізувати інтегровану платформу РЗ;
- провести тестування працездатності створеної платформи та оцінити її ефективність.

Об'єкт дослідження - процес моніторингу та аналізу кіберзагроз у інформаційних системах.

Предмет дослідження – засоби РЗ та візуалізації IoC.

Методи досліджеь: аналітичні методи для дослідження наукових джерел, стандартів та вимог до систем моніторингу; методи системного аналізу для формування архітектури та логічної моделі побудови ПРЗ; методи візуальної аналітики для побудови графіків і дашбордів у Kibana; експериментальні методи для тестування та підтвердження працездатності створеного прототипу.

Наукова новизна отриманих результатів полягає у розробці інтегрованої архітектури ПРЗ, побудованої на основі поєднанні механізмів Elasticsearch, плагіну Kibana та модуля імпорту даних, що дозволяє сформувати єдиний технологічний цикл оброблення IoC.

Практичне значення отриманих результатів. Запропонована інтегрована ПРЗ може бути використана для формування практичного інструментарію аналізу загроз у Kibana, а ткаож як основа для побудови системи РЗ в організаціях різних масштабів.

Публікації та апробація кваліфікаційної роботи.

1. Підлиський Д. Дослідження можливостей використання плагіну Kibana для розвідки кіберзагроз.- Збірник матеріалів науково-практичної конференції молодих вчених, аспірантів та студентів «Кібербезпека та комп'ютерно-інтегровані технології» (КБКІТ - 2025), Тернопіль, 2025.

2. Підлиський Д., Давлетова А. Платформа моніторингу інформаційної безпеки на базі Kibana.- Матеріали науково-практичного симпозіуму «Захист інформації», Тернопіль, 2025.

1. АНАЛІЗ ПЛАТФОРМИ KIBANA ЯК ІНСТРУМЕНТА АНАЛІТИКИ ТА ВІЗУАЛІЗАЦІЇ

1.1 Характеристики та функціональні можливості

Kibana - це аналітична платформа візуалізації та дослідження даних, що є складовою Elastic Stack. Вона забезпечує користувацький доступ до інформації, збереженої в сховищах Elastic [23, 24].

Основною характеристикою Kibana є її здатність забезпечувати візуальну аналітику великих масивів даних у реальному часі. Платформа підтримує побудову діаграм, графів, таблиць, карт, часових рядів, метрик та дашбордів, що дозволяє ефективно досліджувати тенденції, аномалії та закономірності. Завдяки інтерфейсу користувач може створювати складні візуалізації за допомогою механізму перетягування, що спрощує процес аналітики та робить його доступним навіть для користувачів без глибоких технічних знань (рисунок 1.1).

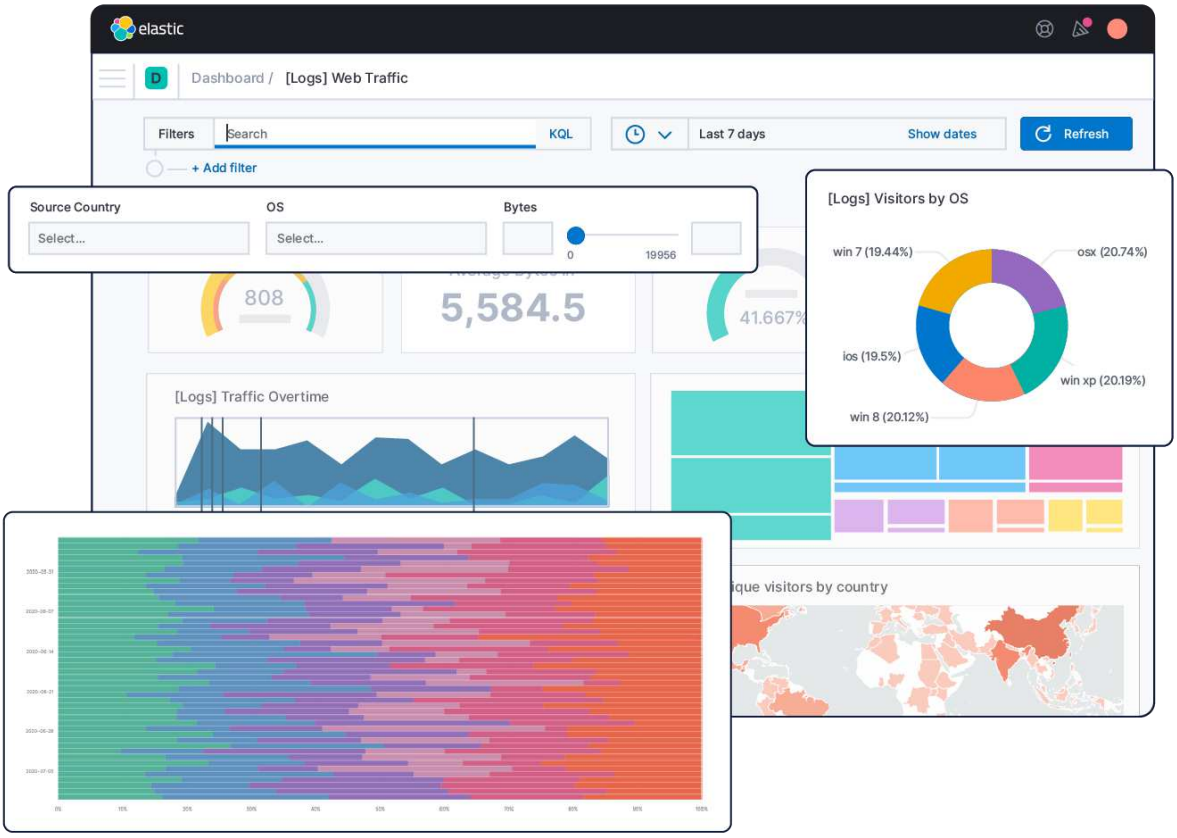


Рисунок 1.1 – Приклад інформаційних панелей

Основіні можливості платформи включають:

– підключення модуля РЗ через агент для збору індикаторів та їх індексації, наприклад IP, домени, хеші, тощо;

– створення ефективних інформаційних панелей для візуалізації, фільтрації та кореляції даних індикаторів та подій;

– використання правил, що дозволяє генерувати попередження.

Kibana також включає інструменти для глибокого аналізу даних, що забезпечують виконання пошукових запитів та перегляд документів у реальному часі. Платформа підтримує інтеграцію з різними джерелами даних і може використовуватися як складова частина систем безпеки, аналітичних платформ, SIEM-систем або рішень для РЗ.

Рішення на базі Elasticsearch сімейства дозволяє обробляти великі обсяги даних з високою швидкістю індексації та пошуку [25-27]. Таке рішення дозволяє об'єднати збір логів, їх обробку, індексацію і зберігання, пошук та візуалізацію загроз в єдиний процес (рисунок 1.2).

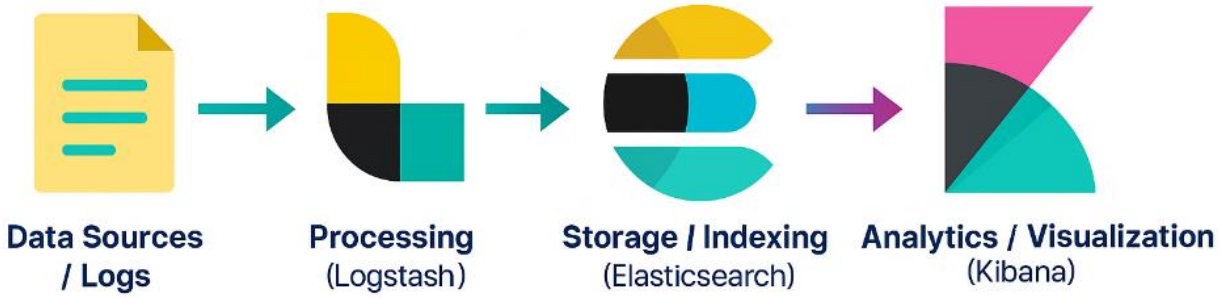


Рисунок 1.2 - Схема обробки та візуалізації даних у Elastic Stack

Офіційна документація вказує, що інтерфейси плагінів змінюються, і немає гарантії сумісності з новими версіями [23]. Це викликає необхідність проведення спеціалізованих налаштувань, підключення модуля РЗ, налаштування індексів, правил, API-ключів тощо потребує технічної експертизи. Активація великої кількості правил виявлення IoC може значно підвищити навантаження на стек та вимагає значних ресурсів. Для деяких випадкі потрібно створювати власні візуалізації, оскільки стандартні рішення можуть бути надто загальними [27-29].

Платформа побудована на модульній архітектурі плагінів, де кожен модуль відповідає за окремий набір функцій: пошук, фільтрацію, аналітику, побудову графіків, моніторинг показників, керування індексами та інші операції. Така архітектура дає змогу розширювати можливості Elastic Stack і адаптувати його до потреб в різних сферах від ведення журналів системних подій до моніторингу кіберзагроз [23, 24].

Kibana є універсальним інструментом, що забезпечує гнучку основу для побудови ПРЗ. Завдяки інтеграції з Elastic Stack він дає змогу об’єднати збір, обробку, індексацію та візуалізацію даних IoC. Проте її впровадження передбачає детальне планування, налаштування, значних обчислювальних ресурсів та уваги до сумісності.

1.2 Архітектура та структурні компоненти

Архітектура Kibana побудована за принципом модульності та тісної взаємодії з Elasticsearch, що забезпечує ефективну роботу з великими обсягами даних у реальному часі. Kibana функціонує як веб-клієнт, який надає користувачам інструменти для виконання аналітичних запитів, створення візуалізацій та керування даними, що зберігаються у сховищі Elastic Stack. Компонентна структура платформи орієнтована на масштабованість, розширюваність та інтеграцію з різними джерелами інформації [23-26].

У центрі архітектури Kibana знаходиться ядро, яке відповідає за маршрутизацію запитів, управління конфігураціями, аутентифікацію, інтеграцію з API Elasticsearch та обробку інтерфейсних компонентів. Це ядро забезпечує взаємодію між вбудованими плагінами та зовнішніми сервісами Elastic Stack, надаючи платформі цілісність та узгодженість у роботі.

Навколо ядра будується функціональна частина Kibana реалізована у вигляді модулів, кожен з яких відповідає за окремі можливості системивід пошуку та аналітики до РЗ, машинного навчання та геопросторової

візуалізації. На рисунку 1.3 узагальнену архітектуру Kibana.

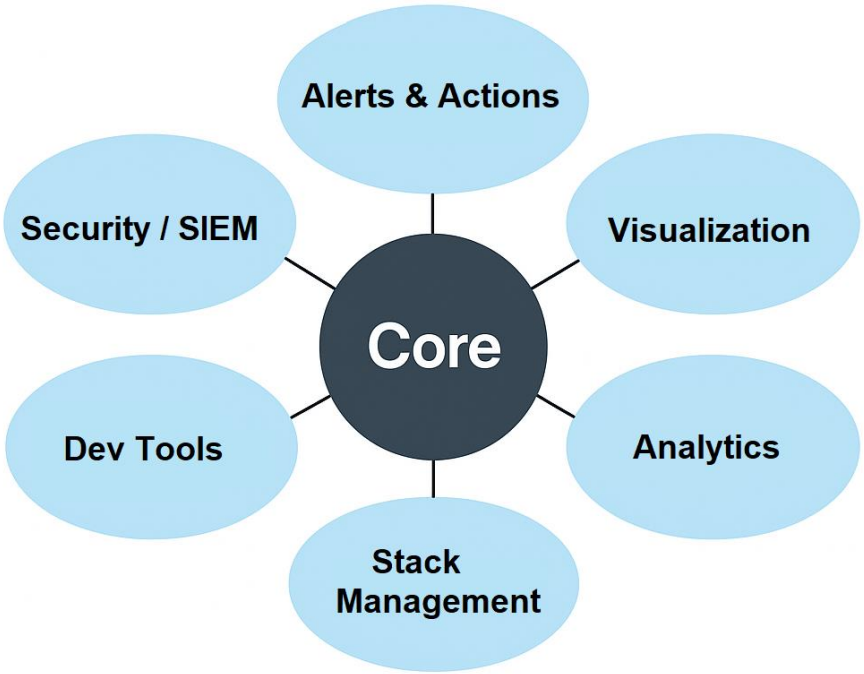


Рисунок 1.3 – Архітектура Kibana

До основних модулів належать [26, 29-32]:

- інструмент для пошуку та початкового аналізу документів, що дозволяє переглядати поля, виконувати фільтрацію та формувати запити у реальному часі.
- модулі побудови візуальних представлень даних, що включають конструктори візуалізацій, які дають змогу створювати графіки, діаграми, метрики та інші аналітичні компоненти;
- механізм інтеграції кількох візуалізацій в єдину аналітичну панель, що оновлюється в реальному часу, дозволяють створювати панелі моніторингу;
- інструменти керування індексами, шаблонами, політиками життєвого циклу даних та іншими системними конфігураціями;
- середовище для виконання запитів до Elasticsearch через REST API та діагностики даних;
- модулі, що надають розширену функціональність для роботи з подіями безпеки, кореляційними правилами та аналітикою загроз;

Архітектура Kibana спроектована таким чином, щоб забезпечити максимальну ефективність взаємодії з Elasticsearch (рисунок 1.4).

Рисунок 1.4 - Можливості Kibana для роботи з даними

Усі запити, що формуються користувачем під час пошуку або побудови візуалізацій, передаються через API Elasticsearch, після чого дані повертаються у Kibana у вигляді відповідей пошукового движка. Це дозволяє платформі працювати з високою продуктивністю та забезпечувати низький час відгуку навіть при великих обсягах структурованих або неструктурованих даних.

Завдяки взаємодії з Elasticsearch, аналітики отримують можливість швидко здійснювати розвідку, виявляти підозрілі активності, аналізувати часові ряди подій та формувати узагальнені огляди ситуації. Kibana підсилює

процеси моніторингу, збору доказів та прийняття рішень у межах ПРЗ.

Важливою характеристикою архітектури Kibana є можливість розширення за рахунок додаткових плагінів. Передбачена можливість створювати власні модулі для інтеграції з зовнішніми сервісами, додавання нових типів візуалізацій або реалізації специфічних сценаріїв аналітики, що робить платформу гнучкою у впровадженні рішень для кібербезпеки, моніторингу систем та ПРЗ.

1.3 Можливості Kibana для розвідки загроз

У сучасних системах кібербезпеки РЗ є комплексним процесом збору та аналізу даних про потенційні або актуальні кіберзагрози з метою підвищення рівня ситуаційної обізнаності та забезпечення ефективного реагування на інциденти. Основними етапами РЗ є збирання IoC, їх структуризація, збагачення, кореляція з подіями безпеки та подальше використання у процесах моніторингу та виявлення аномалій. РЗ допомагає завчасно виявляти атаки та скорочувати час реагування.

Ефективна ПРЗ повинна забезпечувати оперативне опрацювання великого обсягу даних, підтримку аналітичних інструментів та можливість інтеграції з іншими компонентами системи моніторингу.

В процесах РЗ основні операції технічно виконуються рушієм Elasticsearch та сторонніми інтеграціями, проте платформа Kibana відіграє значну роль у забезпеченні взаємодії аналітиків з даними. Вона виступає аналітично-візуалізаційним засобом, що забезпечує інструменти для інтерпретації IoC, оцінювання контексту загроз, пошуку закономірностей та побудови ситуаційної обізнаності.

У систем РЗ Kibana виконує роль верхнього рівня, який забезпечує:

- інтерактивність та доступність даних для аналітиків SOC і аналітиків РЗ;
- консолідовану подачу інформації з множини джерел;

– інструментарій для дослідження, порівняння та інтерпретації IoC;

– формування аналітичних звітів і дашбордів для прийняття рішень.

На рисунку 1.5 схема, що відображає основні можливості Kibana, які застосовуються під час аналізу IoC, формування сповіщень, кореляції подій та візуалізації результатів [23, 25, 29].

Рисунок 1.5 - Функціональні можливості Kibana для РЗ

У межах інтегрованих ПРЗ Kibana виконує функції:

– інтерфейсу доступу до структурованих та неструктурованих TI-даних;

– інструменту кореляції подій та IoC;

– засобу візуальної аналітики загроз;

– компонента, що підтримує інтеграцію з плагінами аналізу

безпеки;

- середовища побудови власних TI-рішень.

Kibana працює з Elasticsearch, забезпечуючи швидкий доступ до великих масивів індексованих IoC, логів, телеметрії чи аналітичних подій. Пошук, фільтрація, агрегації та побудова запитів виконуються у реальному часі. За допомогою таких модулів як Discover, Lens, Dashboard та Security аналітики можуть співставляти дані РЗ із даними з журналів безпеки, кінцевих точок, мережевої активності чи зовнішніх фідів РЗ.

Kibana дозволяє створювати аналітичні панелі для відображення:

- статистики IoC за типами загроз,
- частоти появи індикаторів у часовому вимірі,
- джерел надходження TI-даних,
- взаємозв'язків між подіями та IoC.

Ці механізми дозволяють узагальнювати інформацію про загрози та виявляти аномалії або тренди.

У складі Elastic Security Kibana забезпечує доступ до модулів виявлення загроз і попереджень, case-менеджменту та базових можливостей класифікації активностей за моделлю MITRE ATT&CK mapping [13]. Таким чином, навіть без комерційних модулів Kibana може бути основним інтерфейсом для експертного аналізу загроз на основі користувацьких IoC.

Гнучкість моделі індексації Elasticsearch та доступність API дозволяють створювати кастомні рішення для автоматизованого імпорту IoC, для нормалізації та оновлення IoC, формування TI-панелей під конкретні сценарії, наприклад C2-взаємодія, ботнет-активність, сканування портів тощо.

На основі проведеного аналізу функціональних можливостей і архітектурних особливостей Kibana можна стверджувати, що ця платформа містить усі необхідні механізми для підтримки повного циклу РЗ. Завдяки інтеграції з Elasticsearch та використанню спеціалізованих модулів Kibana виступає базовим компонентом інтегрованої ПРЗ [22].

У процесі формування інтегрованої ПРЗ Kibana може виконувати роль центрального аналітичного центру, який надає інструменти оперативного аналізу, побудови інтерактивних візуалізацій та гнучкої фільтрації даних. Це дозволить підсилити можливості своєчасного виявлення ризиків і підтримувати прийняття обґрунтованих рішень у сфері кіберзахисту.

2. РЕАЛІЗАЦІЯ ІНТЕГРОВАНОЇ ПЛАТФОРМИ РОЗВІДКИ ЗАГРОЗ

2.1 Дослідження підходів до побудови інтегрованої платформи розвідки загроз

Підхід до реалізації програмного засобу РЗ ґрунтується на інтеграції засобів збирання, оброблення, агрегування та візуалізації IoC у єдиному інформаційному середовищі Elastic Stack. Побудована ПРЗ повинна відповідати принципам сучасних систем РЗ, зокрема:

- автоматизоване отримання IoC з різних джерел;
- збагачення одержаних даних додатковими атрибутами, що підвищують інформативність та аналітичну цінність IoC;
- формування спеціалізованого сховища IoC, оптимізованого для швидкого пошуку та кореляції;
- застосування аналітичних інструментів для створення візуалізацій, інтерактивного моніторингу та оперативної підтримки рішень фахівця з кібербезпеки.

Застосований методологічний підхід включає наступні етапи:

- розгортання програмного середовища для збору, зберігання та аналітики даних, яке забезпечує приймання, індексацію та подальше опрацювання відомостей про потенційні загрози;
- створення структури для впорядкованого зберігання записів про загрози, що передбачає визначення шаблонів, правил збереження та логічної організації даних у сховищі;
- формування моделі даних РЗ, яка охоплює основіні характеристики, що дають змогу ідентифікувати, класифікувати та описувати виявлені IoC;
- розроблення модуля для автоматичного додавання нових відомостей про загрози до сховища, що дасть змогу регулярно оновлювати інформаційну базу без ручного втручання;
- налаштування оброблення мережевих адрес під час надходження

даних, зокрема автоматичного визначення країни, належності до мережевого провайдера за допомогою спеціалізованого блоку оброблення;

− створення у засобі аналітики спеціального представлення даних, яке забезпечить доступ до IoC, їх структури та значень для подальшого аналізу.

− побудова набору візуальних елементів, що може включати графіки, таблиці, діаграми, тощо та формування інтегрованої аналітичної панелі огляду загроз;

− перевірка працездатності побудованої моделі на тестовому наборі IoC, створених за допомогою допоміжних скриптів та підготовлених файлів зі зразками даних.

Запропонована методологія дозволяє отримати повністю автоматизований та відтворюваний процес формування ПРЗ, який може бути розширений під реальні джерела IoC.

2.2 Архітектура проектованої платформи розвідки загроз

Архітектура реалізованої ПРЗ побудована на основі принципів модульності, контейнеризації та централізованого зберігання даних. Під час розробки використовувалися технічні підходи, що відповідають сучасним вимогам до систем обробки IoC та принципам Elastic Stack.

У процесі розробки було використано наступні технічні підходи:

− контейнеризація сервісів, що забезпечує ізольованість компонентів і простоту розгортання;

− централізоване зберігання даних на основі Elasticsearch Data Stream;

− автоматизований імпорт IoC за допомогою Python-модуля завантаження;

− аналітичні панелі Kibana Dashboard для візуалізації та дослідження загроз;

− використання плагінів Kibana, зокрема Discover, Lens, Alerts, Intelligence, для пошуку, аналітики та обробки IoC.

Застосування цих принципів дозволило сформувати гнучку архітектуру, орієнтовану на розширюваність та подальшу інтеграцію нових джерел даних.

Архітектура інтегрованої платформи включає компоненти, кожен з яких виконує окрему роль у процесі відтворення повного циклу роботи платформи РЗ.

Джерелом IoC у проектованій ПРЗ використано тестовий CSV-файл із полями, які містять дані про IP-адреси, типи загроз та джерелами походження. Це дозволяє змоделювати структуру типової стрічки РЗ та забезпечує можливість перевірити коректність обробки IoC.

Модуль імпорту IoC реалізований як Python-модуль імпорту IoC, що автоматично завантажує IoC до Elastic Data Stream. Він виконує читання CSV-файлу, трансформацію даних у формат ECS та надсилання IoC до Elasticsearch через REST API. Цей компонент дозволяє розширювати систему новими джерелами без змін в інфраструктурі.Т акий підхід забезпечує можливість автоматизованого надходження IoC у систему.

Elasticsearch використано в якості сховища, у якому виконується індексація документів на основі шаблону logs-ti-template. Компонент також виконує функції бробки запитів і агрегацій, а також забезпечує централізований доступ до всіх даних розвідки. Фактично Elasticsearch є базовим компонентом ПРЗ, що забезпечую високу продуктивність пошуку та аналітичних операцій.

Data stream використовується як основне сховище IoC із можливістю автоматичної ротації backing-індексів. З його допомогою забезпечується пошук та агрегація полів, а акож підтримується структура, визначена index template для РЗ. У шаблоні у data stream logs-ti визначена структура полів, що забезпечує коректну роботу і повну сумісність зі стеками аналітики Kibana.

Kibana забезпечує інструменти візуального моделювання загроз, аналітики, пошуку, візуалізації та створення РЗ-дашбордів, а також інтерфейсу користувача. Kibana служить аналітичним ядром ПРЗ та забезпечує перегляд IoC, дослідження даних, побудову інтерактивних графіків, створення дашбордів РЗ та можливість використання правил. В процесі проєктування використано наступні функції [30-32]:

- Discover – для початкового перегляду індикаторів;
- Lens – для створення аналітичних графіків;
- Dashboard – для побудови інтерактивних панелей огляду та аналізу загроз;
- Management – для налаштування Data Views.

Архітектура реалізованої ПРЗ охоплює всі етапи циклу РЗ та роботи з даними від імпорту та структурованого зберігання IoC до їх аналітичної обробки та візуалізації.

В таблиці 2.1 наведено компоненти фізичної архітектури проектованої ПРЗ та їх розміщення.

Таблиця 2.1 – Фізична архітектура

Компонент	Розміщення	Технологія
Elasticsearch	Docker-контейнер	Elastic Stack
Kibana	Docker-контейнер	Elastic Stack
Python IoC Loader	Host OS	Python 3
Data Stream	Elasticsearch storage	ES Indexing Engine
Dashboard	Kibana UI	Lens, Dashboards

Архітектурне рішення ПРЗ спроєктовано відповідно до вимог:

- централізованого зберігання IoC;
- можливості масштабування;
- інтеграції даних з різних джерел;
- формування інтерактивних панелей аналітики;

– забезпечення гнучкого механізму імпорту, оброблення та візуалізації IoC.

На рисунку 2.1 наведено спрощену схему логічної взаємодії компонентів ПРЗ, зовнішні джерела IoC або передають дані у Python-модуль, який перетворює та надсилає IoC у вигляді JSON-запитів через REST API до Elasticsearch.

Elasticsearch отримує документи, поміщає їх у data stream logs-ti, зберігає та індексує. Kibana отримує доступ до даних через Data View logs-ti* і забезпечує: пошук, фільтрацію, побудову графіків, формування інтерактивного Dashboard.

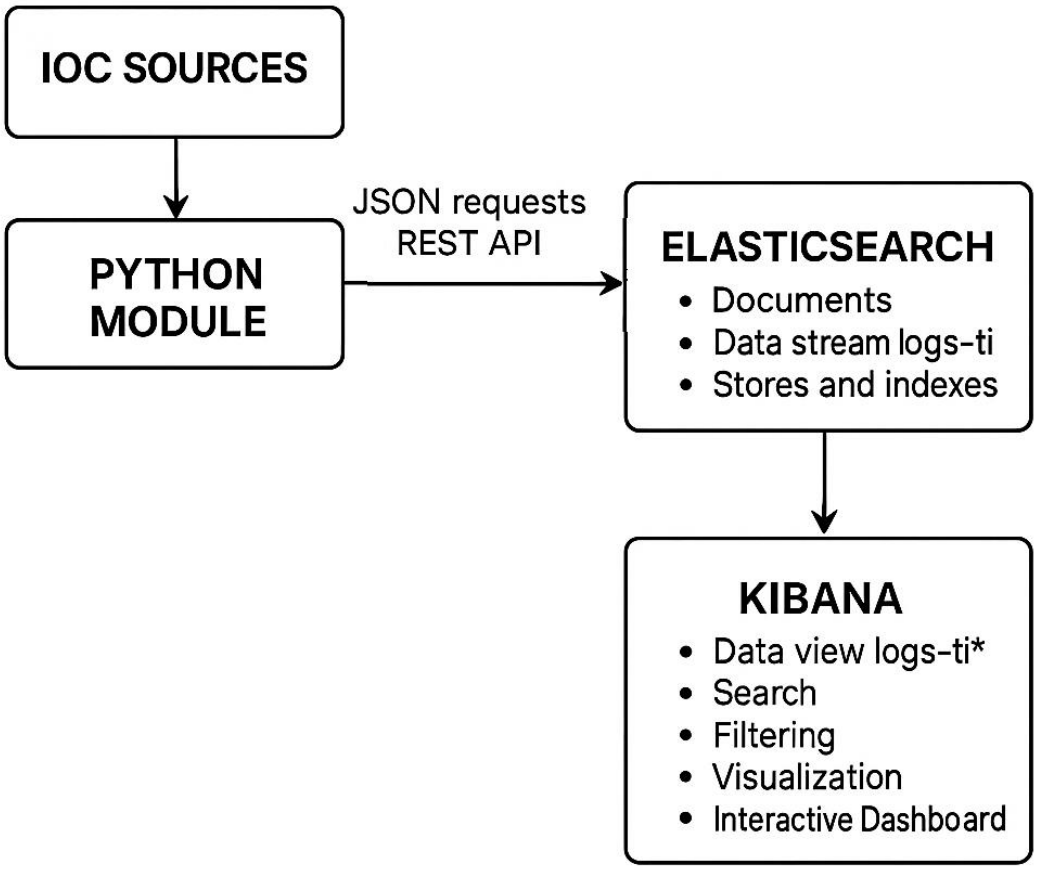


Рисунок 2.1 – Схема взаємодії компонентів ПРЗ

Такий підхід забезпечує реалізацію повного циклу роботи ПРЗ – від первинного надходження даних до їх аналітичної інтерпретації.

2.3 Розгортання середовища

Розробка інтегрованої ПРЗ ґрунтується на застосуванні програмного стека Elastic, який забезпечує можливості централізованого збирання, зберігання, пошуку та аналітичної обробки IoC.

У межах роботи було розгорнуто локальне середовище Elastic Stack, яке складається з контейнерів Elasticsearch та Kibana. Розгортання виконано за допомогою Docker [33], що забезпечує ізольованість компонентів, повторюваність конфігурацій та простоту масштабування. Він повністю сумісний з Threat Intel, Indicator Match, Kibana Security, не вимагає додаткових налаштувань та створює окремі вузли для компонентів (рисунок 2.2).

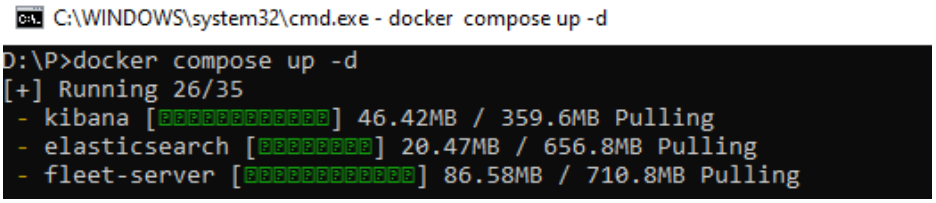


Рисунок 2.2 – Розгортання середовища

На рисунку 2.3 видно, що образ Elasticsearch та Kibana образ успішно завантажуються, відсутні будь помилки overlayfs чи input/output, Docker Engine стабільний, мережа та WSL2 працюють.

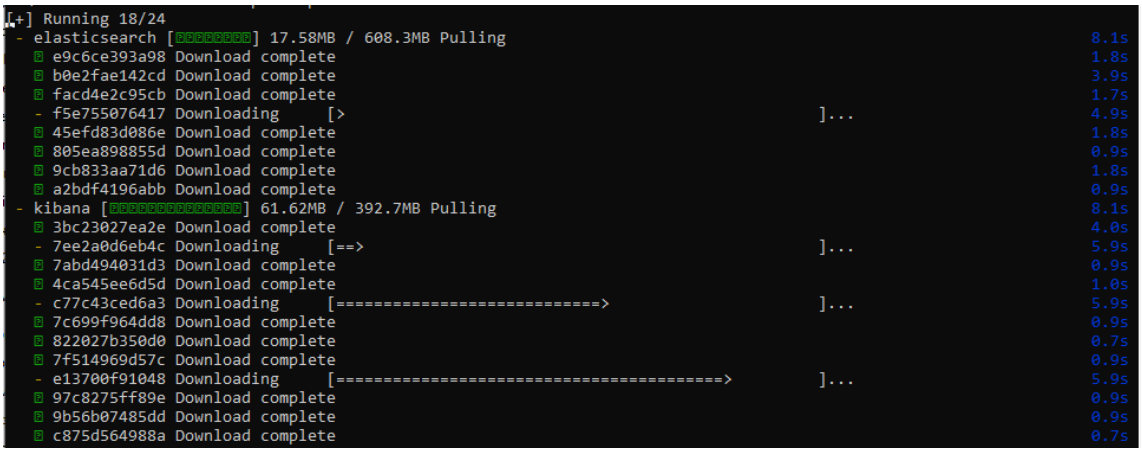


Рисунок 2.3 – Встановлення компонентів

Після завершення завантаження образів система автоматично запустить контейнери (рисунок 2.4).

Рисунок 2.4 – Результат завантаження

В результаті перевірки статусу контейнерів видно (рисунок 2.5) запущені сервіси середовища Elastic Stack. У контейнеризованій інфраструктурі одночасно працюють два основних компоненти. Обидва контейнери перебувають у стані Up, що підтверджує їх коректну ініціалізацію та готовність до подальшої взаємодії в рамках розгортання системи аналітики та РЗ.

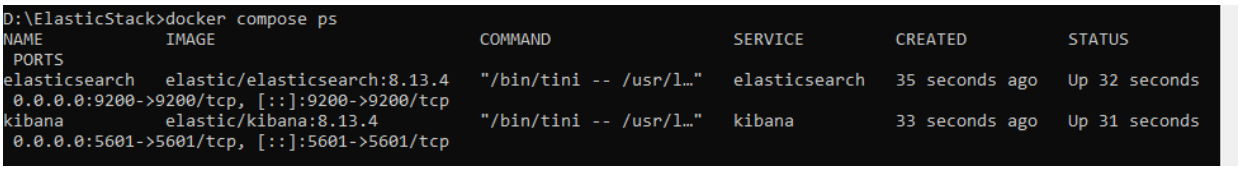


Рисунок 2.5 - Результат розгортання середовища

Сервіс Elasticsearch запущений у контейнері, створеному на основі образу elastic/elasticsearch:8.13.4, і доступний на порту 9200/tcp (рисунок 2.6).

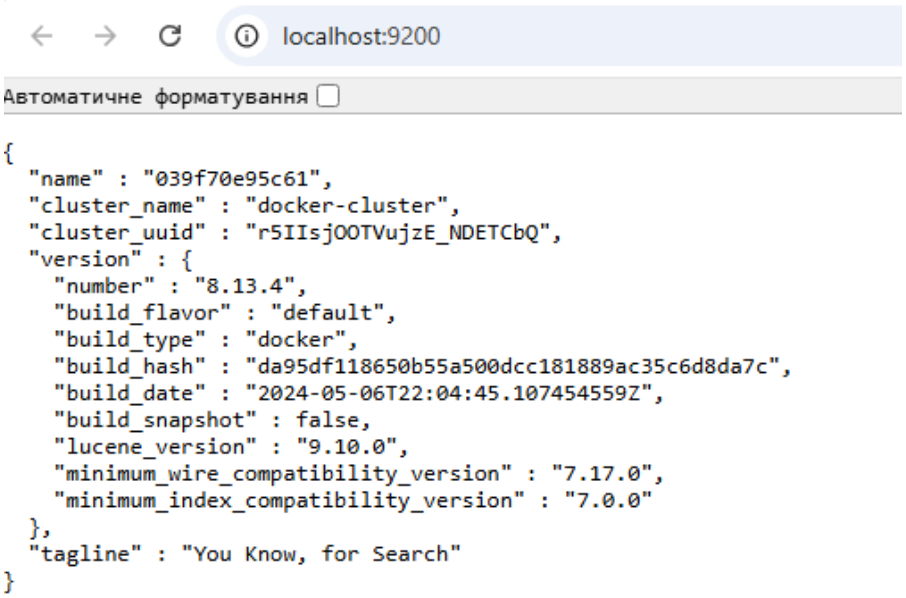


Рисунок 2.6 – Запуск Elasticsearch

На рисунку 2.6 представлено відповідь сервера Elasticsearch, отриману під час звернення до вузла за адресою localhost:9200. У вікні браузера відображено JSON-повідомлення, яке повертається після успішного запуску. У вихідних даних наведені основні метадані кластера: ім’я вузла (name), назва кластера (cluster_name), унікальний ідентифікатор (cluster_uuid), а також детальна інформація про, тип збірки (docker), дату компіляції та сумісність із версіями Lucene.

Поява цієї відповіді підтверджує, що контейнер з Elasticsearch успішно запущено, кластер ініціалізовано, служба доступна на порту 9200 та коректно обробляє HTTP-запити. Це також засвідчує працездатність середовища Docker і стабільність інфраструктури, у якій розгортається Elastic Stack.

Сервіс Kibana працює в окремому контейнері на базі образу elastic/kibana:8.13.4 з відкритим портом 5601/tcp (рисунок 2.7), що забезпечує доступ до веб-інтерфейсу платформи.

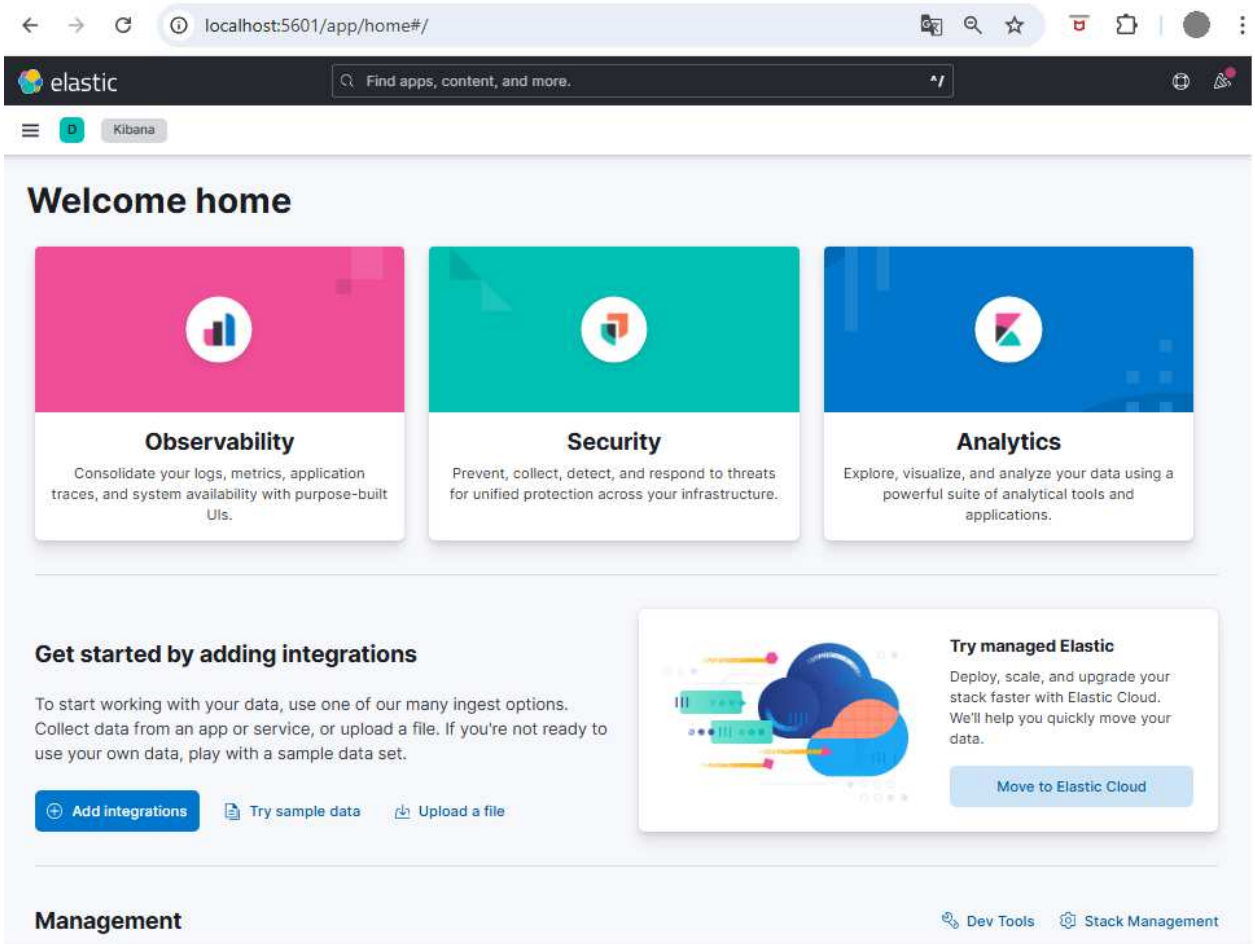


Рисунок 2.7 - Головна сторінка Kibana

2.4 Активація механізмів розвідки загроз

Після успішного розгортання сервісів Elasticsearch та Kibana наступним кроком є активація механізмів РЗ та підготовка інфраструктури для роботи з IoC. Для цього необхідно послідовно виконати ряд налаштувань.

2.4.1 Налаштування модуля безпеки

Після запуску Elastic Stack активується модуль Security, який містить вбудований функціонал РЗ. надає можливість переглядати, аналізувати та корелювати IoC з подіями безпеки.

На рисунку 2.8 представлено сторінку інтерфейсу модуля безпеки у складі Elastic Stack, який відкривається після його активації. У верхній частині відображається панель роботи з модулем, що підтверджує успішне завантаження всіх компонентів.

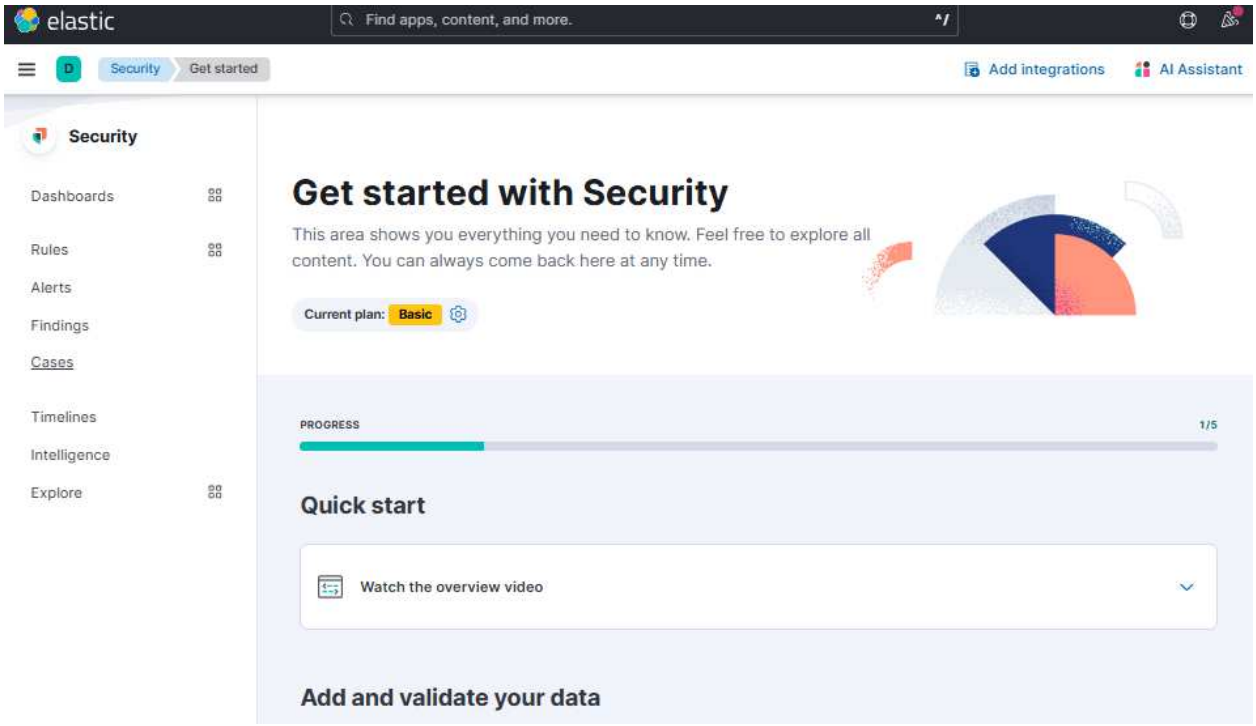


Рисунок 2.8 - Інтерфейс Kibana Security

Доступні основні інструменти безпеки, такі як Dashboards, Rules, Alerts, Cases, Timelines та вкладка Intelligence, яка відповідає за функціонал РЗ.

Наявність цих елементів інтерфейсу свідчить про успішну активацію модуля безпеки та готовність системи до роботи з IoC.

Для коректної роботи необхідне створення спеціального індексу для зберігання IoC, що відповідає структурі Elastic Common Schema (ECS), оскільки РЗ працює з даними у структурованому форматі.

2.4.2 Створення шаблону для індикаторів загроз

На наступному етапі для подальшої аналітики у РЗ необхідно перейти до створення індексів, налаштування шаблонів та завантаження IoC-даних.

Для роботи з РЗ-фідами створюється index template, що визначає структуру індексів, типи полів та прив'язку до ECS (рисунок 2.9).

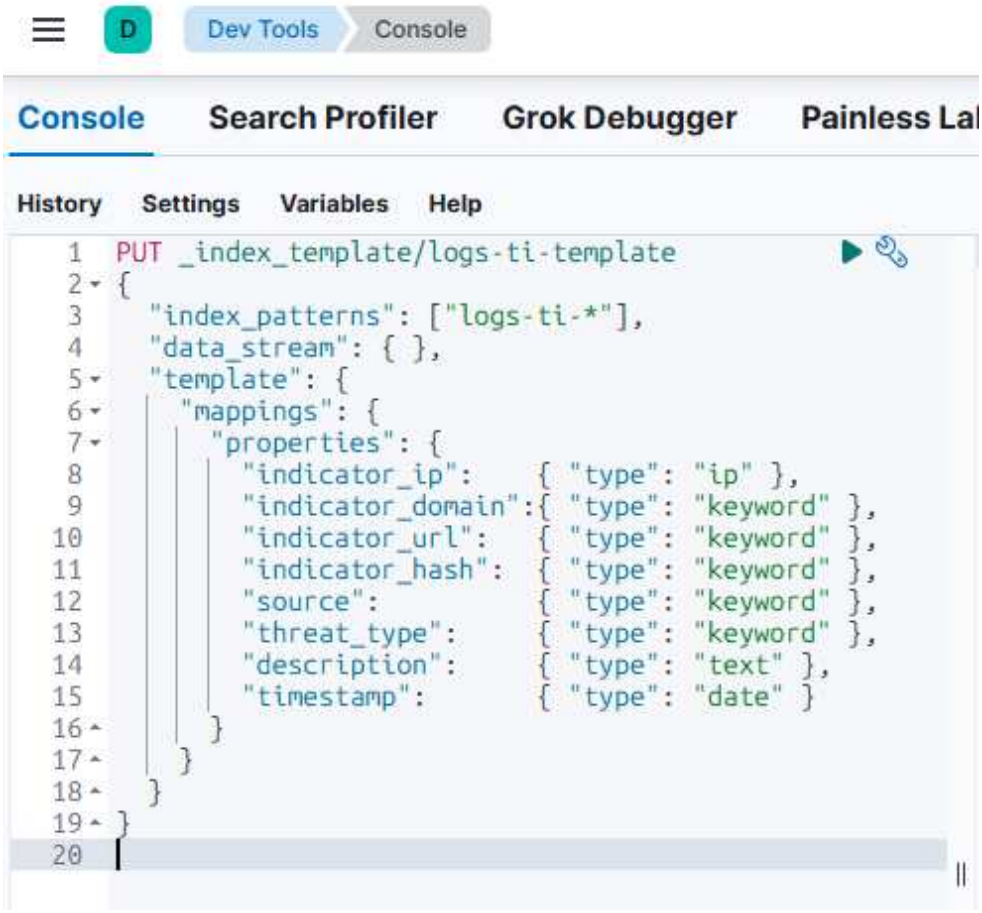


Рисунок 2.9 – Створення index template

На рисунку 2.9 наведено процес створення шаблону logs-ti для РЗ у консолі Dev Tools платформи Kibana. У вікні Console виконується команда

PUT _index_template/logs-ti-template, яка визначає шаблон індексів для майбутніх даних РЗ.

Наступним етапом є формування Data View logs-ti* У тілі запиту задається патерн індексів, що забезпечує автоматичне застосування цього шаблону до всіх індексів і data stream, призначених для зберігання IoC. У секції template описано структуру полів (mappings), що відповідає формату ECS.

Data View включає ключові поля, що наведені в таблиці 2.2.

Таблиця 2.2 – Основні поля logs-ti*

Поле	Опис
indicator_ip	IP-адреса індикатора компрометації
source	джерело отримання індикатора
threat_type	тип загрози (botnet, dns, scanner тощо)
description	опис загрози
@timestamp	час отримання індикатора

На рисунку 2.10 наведено відповідь Elasticsearch, отриману після виконання запиту на створення шаблону. Повідомлення у форматі JSON містить ключ "acknowledged": true, що означає успішне застосування створеного шаблону. Код відповіді 200-OK підтверджує, що операція виконана без помилок, а індексний шаблон коректно збережено в конфігурації Elasticsearch..

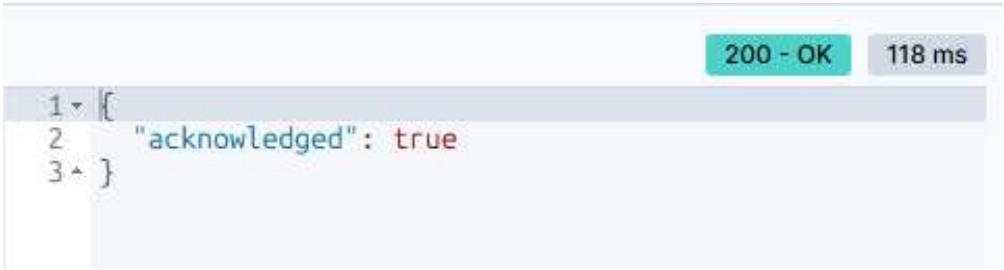


Рисунок 2.10 – Результат створення шаблону

На основі цього шаблону даних РЗ створюється data stream, який забезпечує безперервне надходження та зберігання IoC. На рисунку 2.11 наведено результат виконання команди PUT _data_stream/logs-ti, що створює відповідний data stream у Elasticsearch.

Рисунок 2.11 - – Створення data stream

У відповідь система повертає підтвердження у вигляді JSON-повідомлення "acknowledged": true, що свідчить про успішне створення потоку даних (рисунок 2.12).

Рисунок 2.12 – Результат створення data stream

Data stream використовується для масштабованого опрацювання, оновлення та утримання актуальності IoC у середовищі Elasticsearch. Він дозволяє зберігати TI-дані у хронологічному порядку, забезпечує їх ефективне індексування та оптимізує роботу механізмів Threat Intelligence у Kibana

2.4.3 Додавання індикаторів компрометації

Після створення data stream виконується додавання тестового IoC, який індексується в Elasticsearch та стає доступним для перегляду через інтерфейс

Kibana в модулі безпеки у вкладці Intelligence.

На рисунку 2.13 наведено процес створення першого IoC до створеного раніше data stream logs-ti.

У консолі Dev Tools виконуєтьсявідповідий запит, який створює новий запис у data stream, що підтверджується відповіддю Elasticsearch зі статусом 201 – Created (рисунок 2.14).

У відповіді наведено дані про створений документ – ім’я backing-index (.ds-logs-ti-...), ідентифікатор документа (_id), номер версії та інформацію про розподіл по shard’ах.

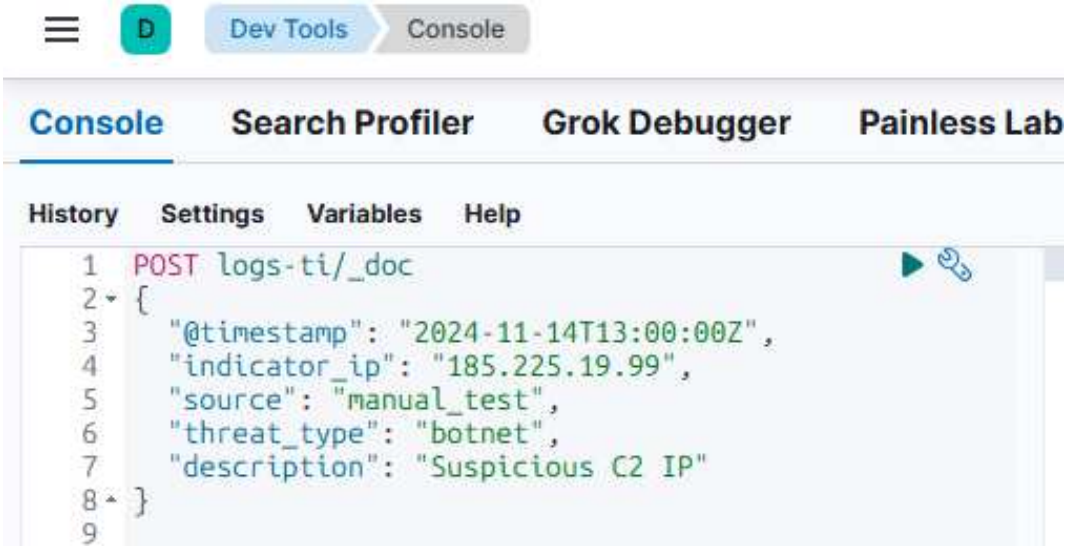


Рисунок 2.13 – Додавання IoC у data stream logs-ti

Рисунок 2.14 – Результат створення IoC

Додавання тестового IoC підтверджує коректність структури шаблону, працездатність data stream та готовність Elasticsearch до приймання даних РЗ.

Після індексації такий IoC автоматично стає доступним для перегляду у Kibana, яка автоматично зчитує метадані IoC, забезпечує його візуалізацію, збагачення та можливість використання у кореляційних правилах типу Indicator Match. Це підтверджує правильну роботу механізмів РЗ та можливість подальшої інтеграції реальних РЗ-фідів у платформу.

3. ДОСЛІДЖЕННЯ ІНТЕГРОВАНОЇ ПЛАТФОРМИ РОЗВІДКИ ЗАГРОЗ НА ОСНОВІ ПЛАГІНУ KIBANA

3.1 Розробка модуля імпорту даних

Для моделювання та тестування функціонування розробленої інтегрованої ПРЗ на основі стеку Elasticsearch та Kibana було створено окремий модуль імпорту даних, який забезпечує автоматизоване завантаження, попередню обробку та структурування IoC перед їх подальшим опрацюванням у системі. Застосування цього модуля дозволяє відтворити реальний цикл надходження IoC, що характерний для промислових платформ, та забезпечити коректність експериментального оцінювання роботи ПРЗ.

Розроблений модуль складається з таких частин:

- імпорт CSV-файлів - зчитує IoC із файлу indicators.csv та забезпечує перевірку на коректність полів.
- формування JSON-запитів - перетворює кожен запис у формат документа Elasticsearch тадодає службове поле @timestamp.
- REST-клієнт для Elasticsearch - використовує бібліотеки requests або urllib3 і надсилає дані через Bulk API або окремими документами.
- блок логування - відображає статус завантаження іповідомляє про помилки з’єднання або некоректні записи.
- блок конфігурації - визначає адресу Elasticsearch, містить параметри авторизації та дозволяє змінювати назву Data Stream.

Алгоритм роботи розробленого Python-модуля включає наступні кроки.

Крок 1. Зчитування вхідних даних. Виконується відкриття CSV-файлу та зчитування його вмісту за допомогою структурованого словника:

```
import csv

with open("indicators.csv", newline="") as f:
    reader = csv.DictReader(f)
    indicators = list(reader)
```

Модуль очікує, що кожен запис міститиме чотири інформаційні поля: indicator_ip, source, threat_type, description.

Крок 2. Формування структури даних для індексації Elasticsearch. Для кожного рядка CSV формується окремий документ у стандартизованому вигляді. До нього автоматично додається мітка часу (@timestamp), що відповідає моменту обробки IoC:

```
doc = {
    "indicator_ip": row["indicator_ip"],
    "source": row["source"],
    "threat_type": row["threat_type"],
    "description": row["description"],
    "@timestamp": datetime.utcnow().isoformat() + "Z"
}
```

Після чого дані стають придатними для подальшої індексації та аналітичної обробки у платформі.

Крок 3. Передавання IoC до сховища даних. Документ надсилається до спеціалізованого сховища IoC через HTTP-запит:

```
import requests, json

resp = requests.post(
    "http://localhost:9200/logs-ti/_doc",
    headers={"Content-Type": "application/json"},
    data=json.dumps(doc)
)
```

Використання REST-інтерфейсу забезпечує незалежність модуля від внутрішньої реалізації платформи та дозволяє інтегрувати його з будь-яким сумісним бекендом.

Крок 4. Оброблення результату та звіт про виконання. Після надсилання даних модуль аналізує код відповіді сервера:

```
if resp.status_code == 201:
    print("Document indexed")
else:
    print("Error:", resp.text)
```

Це забезпечує контроль коректності імпорту та дозволяє фіксувати можливі помилки або відхилення.

В додатку А наведено повний скрипт Python-модуля, що обув

використаний для моделювання роботи ПРЗ. Алгоритм його роботи відповідає початковим етапам життєвого циклу РЗ, отримання IoC та попереднє опрацювання зібраної інформаці, що дозволяє використовувати модуль як у тестовому середовищі, так і в реальних процесах РЗ.

У процесі тестування та валідції роботи модуля було перевірено такі параметри:

- коректність зчитування CSV-файлів;
- стабільність відправлення HTTP-запитів;
- відповідність структури документів індексному шаблону logs-ti-template;
- відображення IoC у Kibana Discover;
- можливість побудови візуалізацій на основі завантажених IoC.

Усі завантажені IoC були успішно проіндексовані та відображені у Data Stream.

Цей компонент дозволяє розширювати систему та інтегрувати її з зовнішніми службами РЗ у майбутньому (MISP API, AbuseIPDB API, OTX API тощо). Модуль може виконувати роль базового конектора даних у реальних процесах кіберзахисту, забезпечуючи оперативне оновлення та централізоване управління IoC.

3.2 Моделювання роботи платформи

3.2.1 Імпорт індикаторів компрометації

Для перевірки працездатності створеної ПРЗ було проведено моделювання процесу автоматизованого завантаження IoC та подальшої їх обробки в інтегрованому середовищі Elastic Stack.

Основним елементом цього етапу є розроблений Python-модуль import_ioc.py, що виконує функції прикладного завантажувача даних та забезпечує тестування компонентів проєктованої ПРЗ.

Для моделювання створюється вхідний CSV-файл indicators.csv, який

містить структуровану інформацію про тестові IoC. До набору включено чотири атрибути:

```
indicator_ip,source,threat_type,description
185.225.19.99,manual_test,botnet,Suspicious C2 server
8.8.8.8,external_feed,dns,Known DNS resolver
45.155.205.233,abuseipdb,scanner,Brute-force activity detected
```

Такий набір дозволяє змоделювати різні типи індикаторів і оцінити коректність подальшої обробки та відображення даних у Kibana.

На рисунку 3.1 наведено фрагмент консольного виводу, отриманого після запуску модуля.

```
D:\ElasticStack>python import_ioc.py
Importing IOC from indicators.csv ...
[201] 185.225.19.99 -> {"_index":".ds-logs-ti-2025.11.13-000001","_id":"rEz3fZoBOzwUjnZj7sbb","_version":1,"result":"created","_shards":{"total":2,"successful":1,"failed":0},"_seq_no":1,"_primary_term":1}
[201] 8.8.8.8 -> {"_index":".ds-logs-ti-2025.11.13-000001","_id":"rUz3fZoBOzwUjnZj7sbb","_version":1,"result":"created","_shards":{"total":2,"successful":1,"failed":0},"_seq_no":2,"_primary_term":1}
[201] 45.155.205.233 -> {"_index":".ds-logs-ti-2025.11.13-000001","_id":"rkz3fZoBOzwUjnZj7sbb","_version":1,"result":"created","_shards":{"total":2,"successful":1,"failed":0},"_seq_no":3,"_primary_term":1}
Finished!
```

Рисунок 3.1 – Результат імпорту даних

Результати свідчать про успішне завантаження всіх трьох тестових IoC до потоку даних logs-ti. Для кожного запису відображено:

- код відповіді 201 (Created), що підтверджує коректну індексацію документа;
- ідентифікатор створеного документа (_id);
- автоматично сформовану назву data-stream сегмента (наприклад, .ds-logs-ti-2025.11.13-000001);
- відомості про успішність реплікації ("successful":1, "failed":0).

Завершальне повідомлення Finished! підтверджує, що модуль завершив обробку всіх рядків CSV-файлу без помилок, що демонструє працездатність механізму імпорту IoC та коректну взаємодію Python-модуля з підсистемою зберігання даних Elastic.

Після імпорту для перевірки стану створеного сховища використано

вбудовані інструменти Kibana та Dev Tools. Для цього виконано запитGET _data_stream/logs-ti (рисунок 3.2), що повертає інформацію про структуру data stream:

- наявність часової мітки @timestamp;
- створений перший сегмент .ds-logs-ti-2025.11.13-000001;
- статус YELLOW, що є типовим для одновузлових кластерів;
- прив’язаний шаблон logs-ti-template.

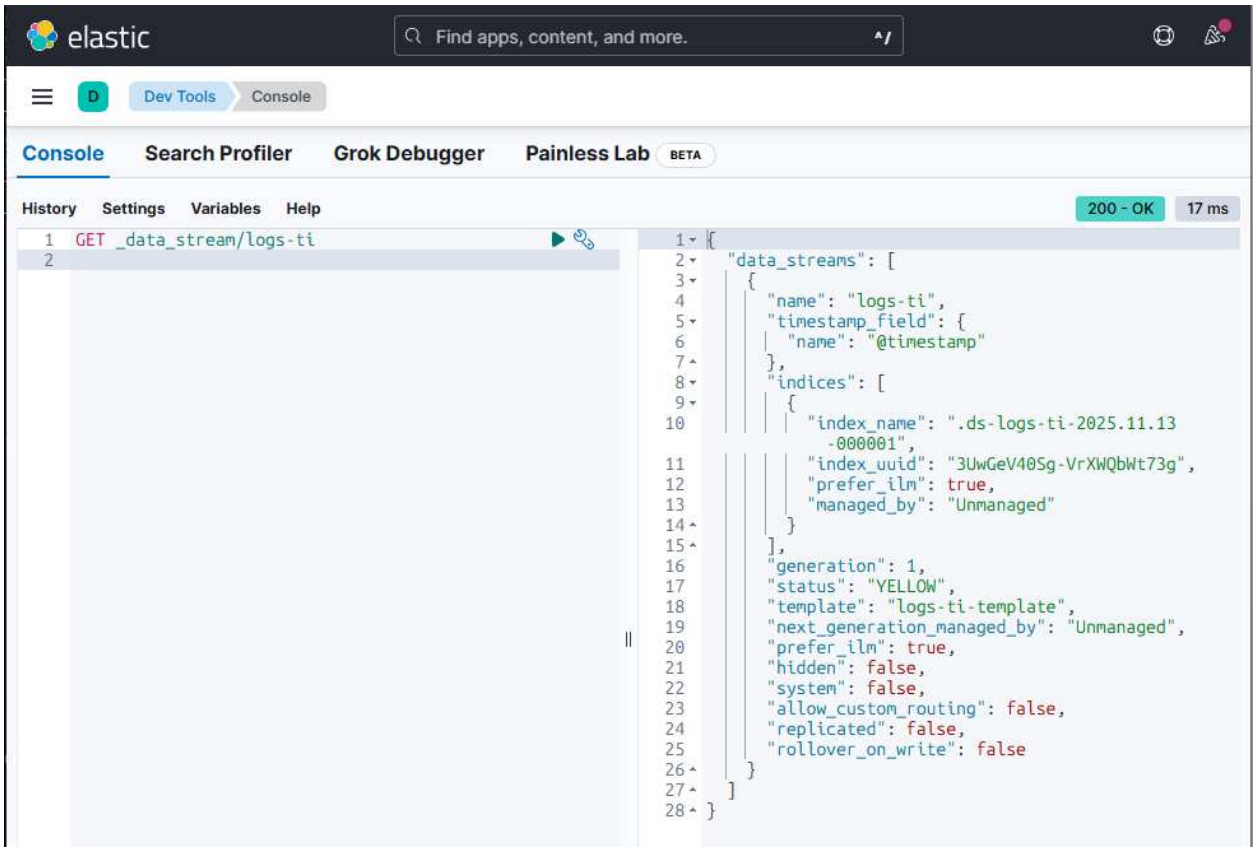


Рисунок 3.2 - Перевірка стану створеного сховища

Отримані дані свідча, що процес індексації працює коректно, а створена ПРЗ може виконувати подальші аналітичні сценарії.

Після успішної індексації IoC стають доступними також в аналітичних модулях Kibana:

- Discover – для перегляду кожного запису окремо;
- Dashboard – для побудови інтерактивних панелей моніторингу;
- Security / Intelligence / Indicators – для роботи з даними РЗ.

3.2.2 Аналіз даних у модулі Discover

Kibana Discover є базовим інструментом для первинного аналізу даних. У межах реалізованої ПЗР цей модуль використано для:

- перегляду завантажених індикаторів;
- оцінки коректності роботи Python-модуля завантаження;
- фільтрації за полями source, threat_type, indicator_ip;
- перевірки структури Data Stream;
- виконання пошуку за ключовими атрибутами IoC.

Для перевірки того, що дані доступні для подальшої аналітики виконано типові запити у Discover, наприклад:

- пошук за типом загрози: threat_type : "botnet"
- пошук за джерелом: source : "abuseipdb"
- пошук за IP-адресою: indicator_ip : "185.225.19.99"

Тестування підтверджує, що всі індикатори успішно проіндексовані, а поля автоматично розпізнані Kibana.

На рисунку 3.3 наведено інтерфейс модуля Discover, у якому відображаються дані, імпортовані до потоку logs-ti з використанням розробленого модуля завантаження IoC. Отримані результати свідчать про коректність функціонування всього ланцюга оброблення IoC:

- у стрічці документів відображається три індексовані записи, що відповідає кількості рядків у тестовому CSV-файлі;
- поля indicator_ip, source, threat_type, description зчитані і відображені коректно, що підтверджує правильність структури сформованих JSON-документів;
- значення @timestamp проставлені автоматично під час імпорту та коректно інтерпретуються системою, що забезпечує можливість подальшого часово-орієнтованого аналізу;
- модуль Discover успішно отримує дані саме з data stream logs-ti, що свідчить про правильність налаштування сховища та шаблону.

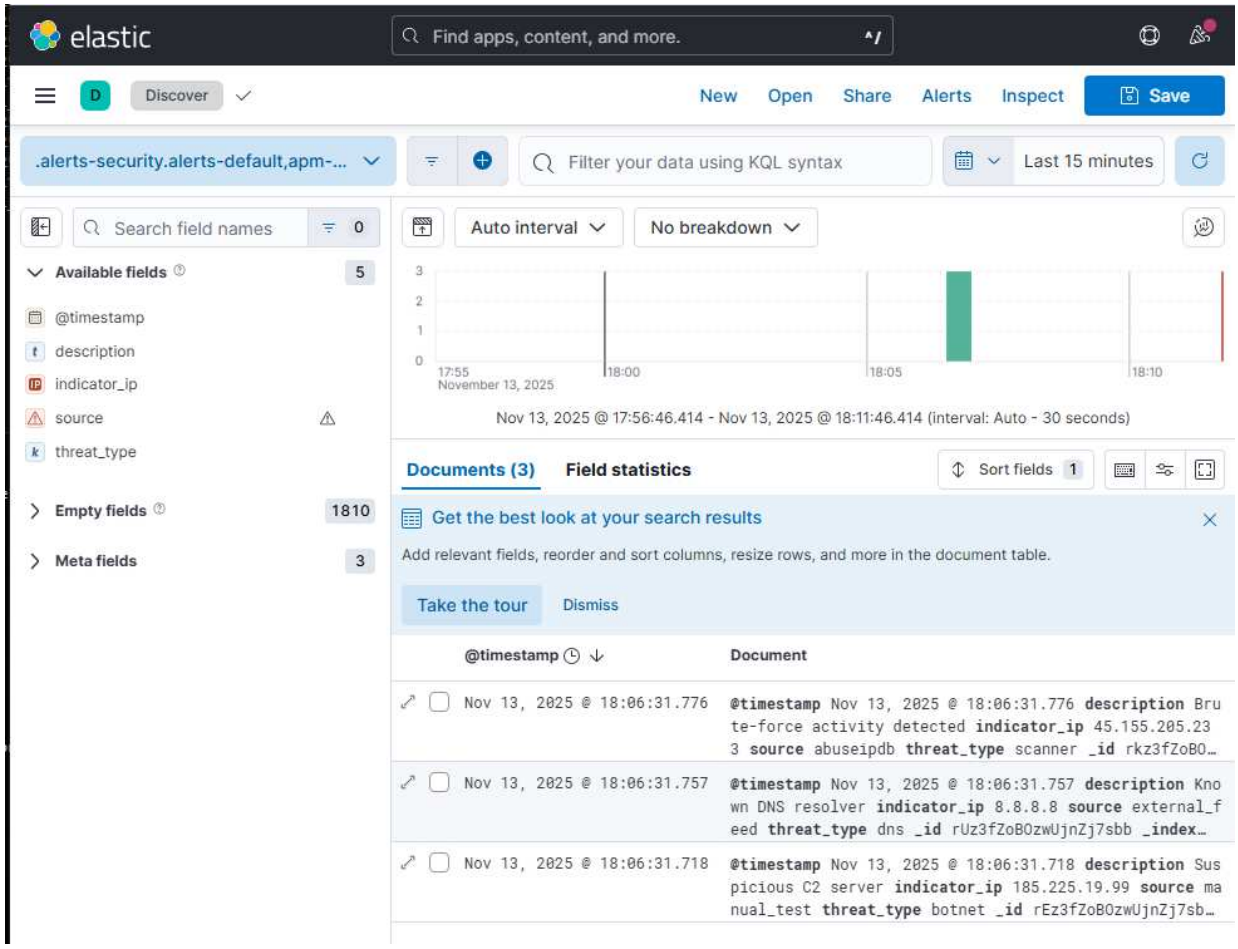


Рисунок 3.3 – Відображення даних РЗ

В результаті підтверджено повну працездатність створеного конвеєра завантаження IoC - індексація у сховищі - візуальний перегляд у Kibana, що забезпечує основу для побудови функціональної ПРЗ.

3.2.3 Представлення даних для Dashboards

Після завантаження IoC у сховище даних виникає потреба забезпечити їх подальший доступ для інструментів аналітики Kibana. Для цього створюється представлення даних (Data View), яке виконує роль логічного інтерфейсу між користувачем та даними, що зберігаються у data stream.

На рисунку 3.4 наведено початкове вікно керування представленнями даних, у якому відображаються доступні конфігурації та забезпечується можливість створення нового Data View.

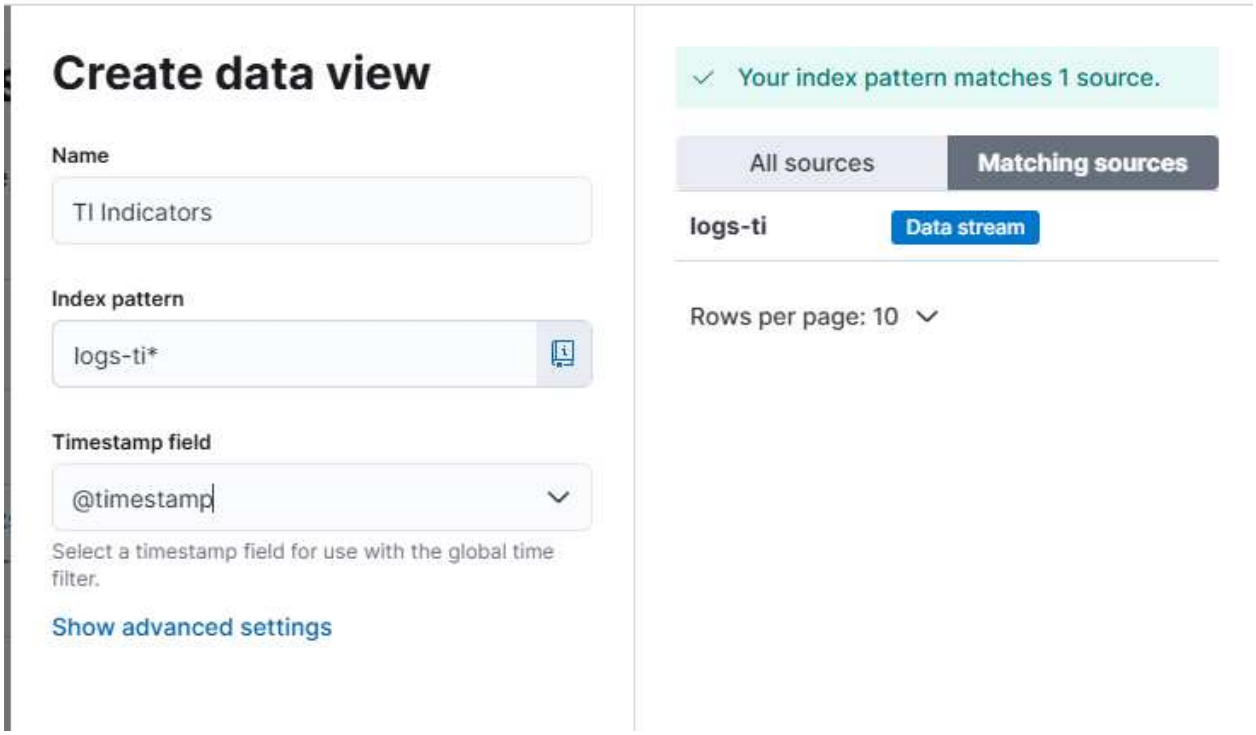


Рисунок 3.4 – Процес створення представлення даних

Процедура створення представлення даних TI Indicators включає виконання налаштування параметрів:

- Data view name: TI Indicators – назва набору даних, що використовуватиметься у Discover, Lens та Dashboard.
- Index pattern: logs-ti* – шаблон, що відповідає назві data stream (logs-ti), індекси якого генеруються автоматично у форматі .ds-logs-ti-YYYY.MM.DD-000001. Тому шаблон logs-ti* коректно покриває всі майбутні сегменти потоку даних.
- Timestamp field: @timestamp – часовий атрибут, необхідний для побудови часових фільтрів та хронологічного аналізу індикаторів.

Після підтвердження у Kibana створюється нове представлення даних, яке забезпечує доступ до всіх IoC, що зберігаються в data stream logs-ti.

На рисунку 3.5 наведено інтерфейс перегляду створеного представлення даних.

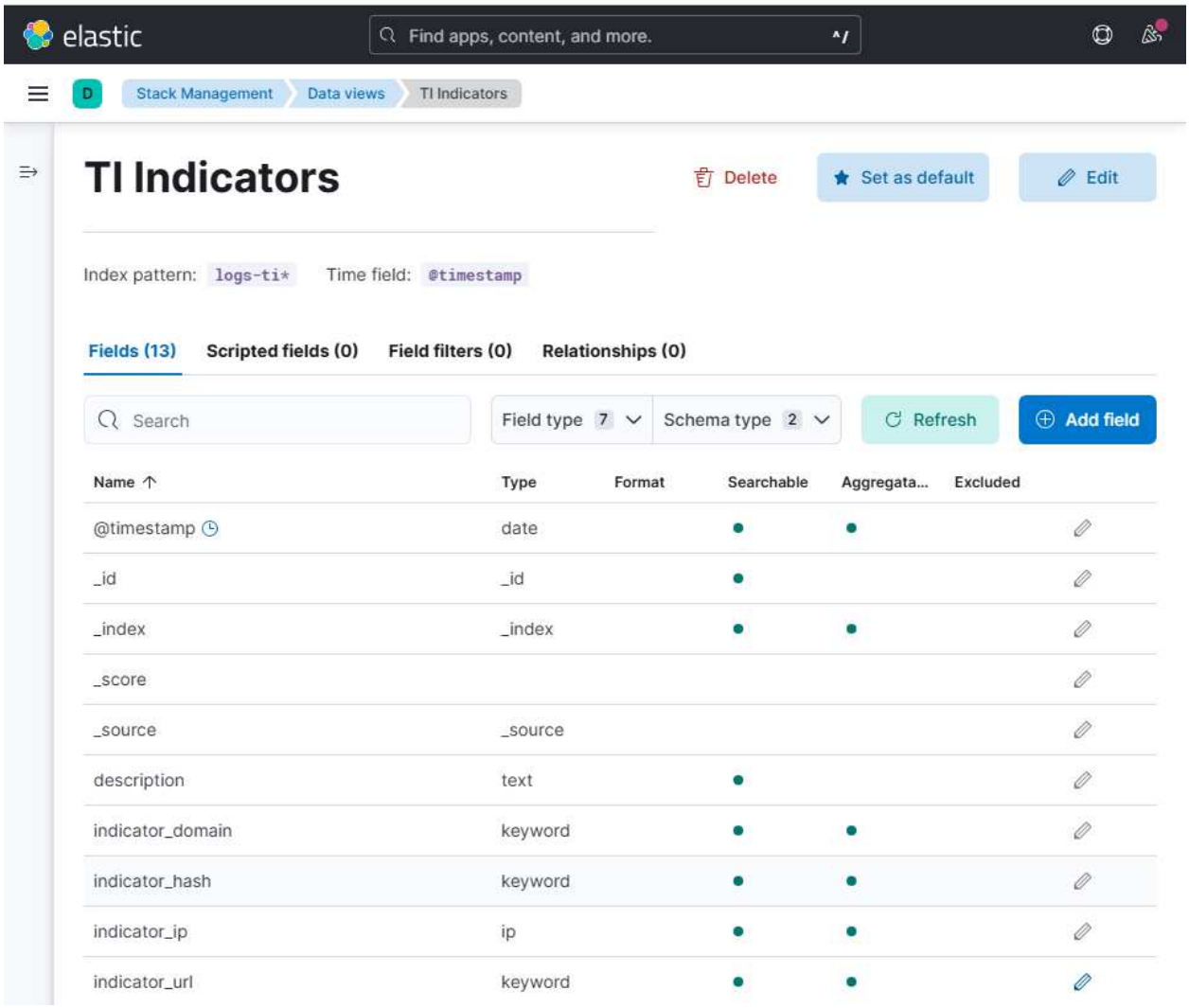


Рисунок 3.5 – Відображення створеного представлення даних

На рисунку 3.5 видно, що створення виконано коректно. Kibana успішно розпізнала всі наявні поля та автоматично визначила їх типи, що підтверджує правильність структури отриманих IoC та відповідність моделі даних вимогам до ПРЗ.

Представлення даних сформоване без помилок та повністю готове до подальшого використання в аналітичному модулі, зокрема:

- перегляду даних у режимі Discover;
- побудови графічних візуалізацій у Lens;
- формування інтегрованої аналітичної панелі (Dashboard

3.2.4 Імпорт та інтеграція збережених об’єктів Kibana

Для забезпечення подальшої візуалізації IoC та побудови аналітичної панелі було здійснено імпорт заздалегідь підготовлених збережених об’єктів Kibana (Saved Objects). До складу імпортованого набору входять два елементи, які формують основу інтерфейсу користувача для аналізу показників РЗ:

- збережений пошуковий запит (Saved Search) “TI Indicators Table”;
- дашборд “Threat Intelligence Overview”.

На рисунку 3.6 наведено список наявних збережених об’єктів перед виконанням імпорту. Інтерфейс Kibana дозволяє переглядати, змінювати та видаляти збережені конфігурації, що забезпечує централізоване керування елементами аналітичного середовища.

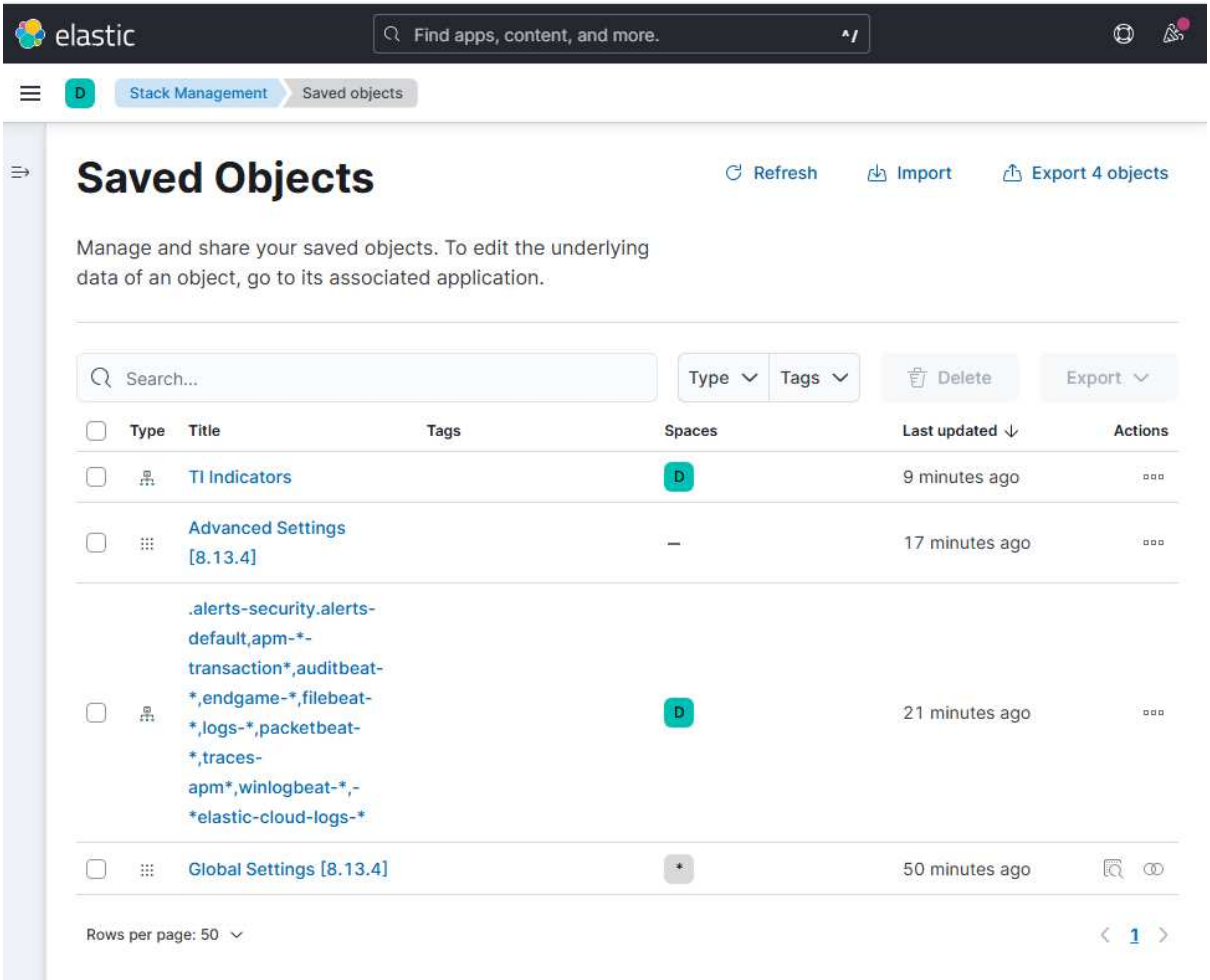
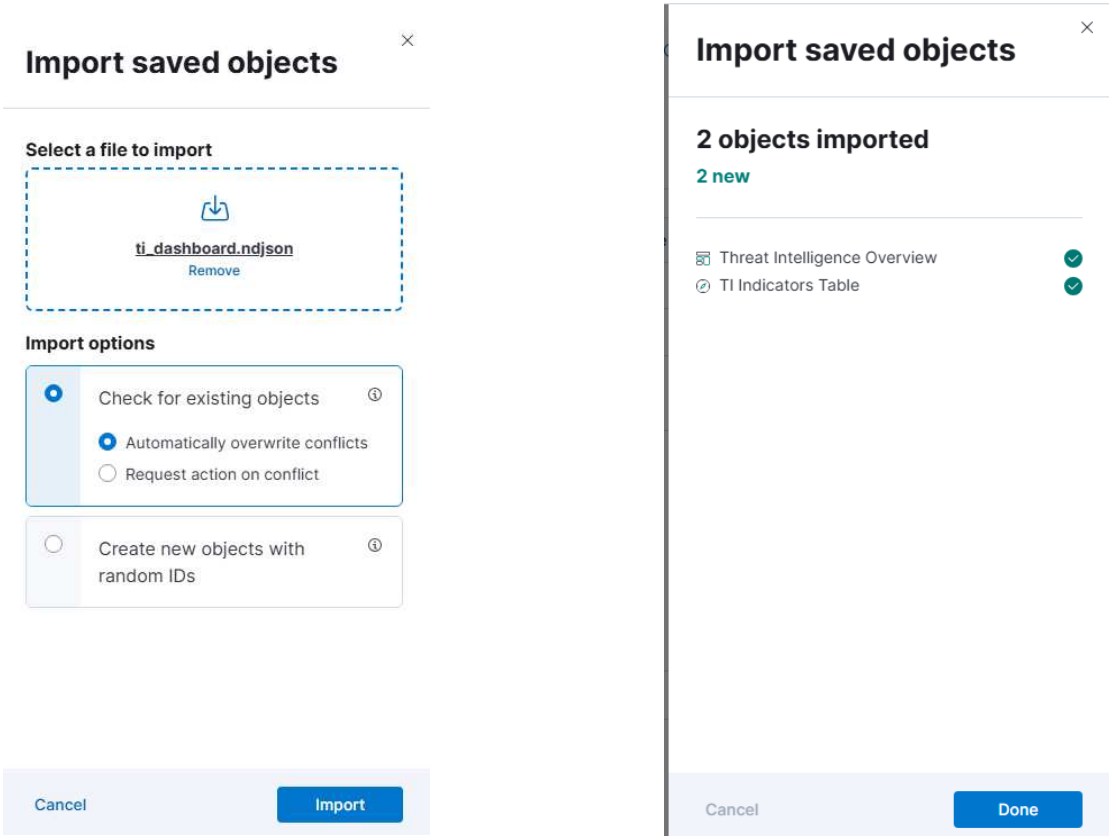


Рисунок 3.6 - Список наявних збережених об’єктів

На наступному етапі (рисунок 3.7 а) використано форму імпорту, де було завантажено файл ti_dashboard.ndjson. У налаштуваннях імпорту вибрано опцію автоматичного перезапису конфліктних об’єктів, що гарантує коректну інтеграцію елементів без дублювання та несумісностей.



а) б)

Рисунок 3.7 – Форми імпорту об’єктів

Після підтвердження виконання операції (рисунок 3.7б) Kibana повідомляє про успішне додавання двох нових об’єктів:

- Threat Intelligence Overview – інтегрована аналітична панель,
- TI Indicators Table – табличне представлення імпортованих індикаторів.

Після імпорту обидва об’єкти з’являються у загальному списку Saved Objects (рисунок 3.8), що підтверджує правильність виконання процедури та готовність інтерфейсу до подальшого використання.

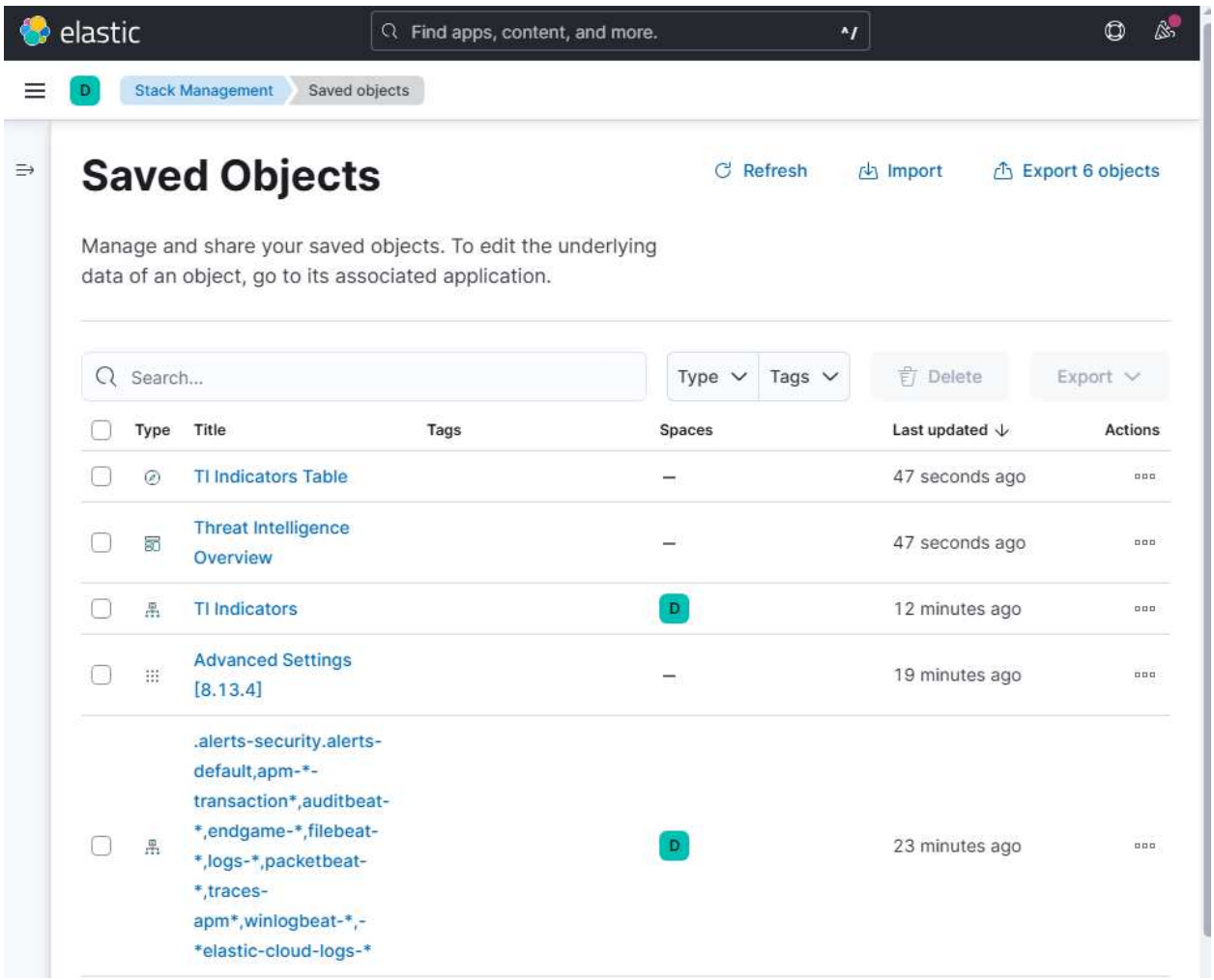


Рисунок 3.8 – Оновлений список наявних об’єктів

На рисунку 3.9 наведено приклад відображення збереженого пошукового запиту “TI Indicators Table” у модулі Discover. На цьому етапі Kibana коректно розпізнає структуру Data View та доступні атрибути індикаторів (indicator_ip, source, threat_type, description, @timestamp). За відсутності результатів у вибраному часовому діапазоні система повідомляє про необхідність його розширення, що є штатною поведінкою інструментів пошуку Kibana.

Здійснена процедура імпорту забезпечує готовність середовища до формування узагальненої аналітичної панелі та подальшої побудови візуалізацій, які використовуються при дослідженні роботи розробленого засобу РЗ.

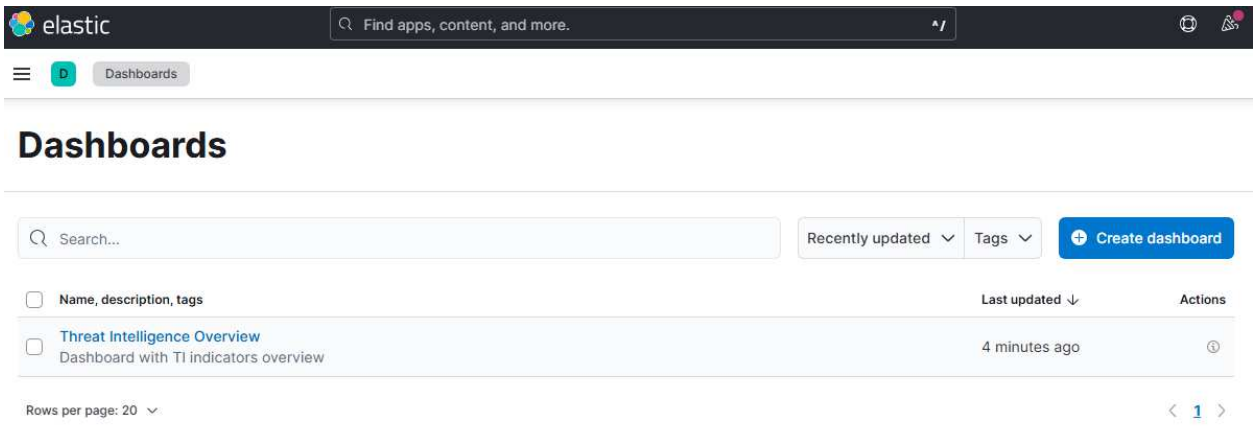


Рисунок 3.9 - Відображення збереженого об’єкту у модулі Discover

Після завершення етапів імпорту IoC та створення представлення даних було виконано побудову інтегрованої аналітичної панелі Threat Intelligence Overview. Створена панель забезпечує комплексний огляд завантажених індикаторів, їхніх характеристик та динаміки появи, що є ключовим елементом практичного застосування ПРЗ у процесах аналізу й моніторингу загроз.

На рисунку 3.10 наведено таблицю TI Indicators Table, сформовану на основі Data View TI Indicators.

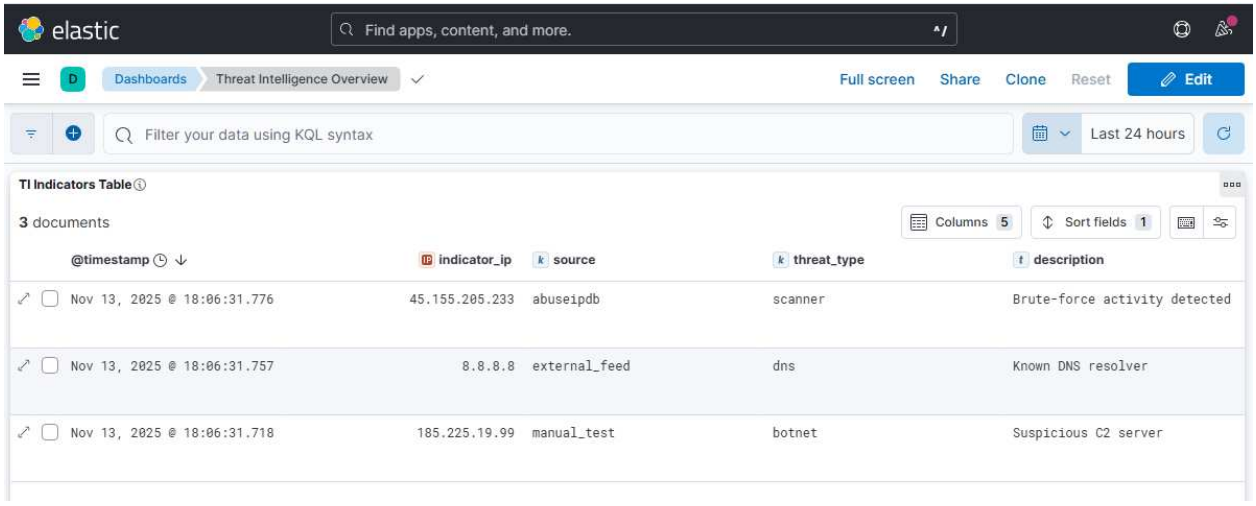


Рисунок 3.10 – Відображення даних індикаторів

У таблиці відображаються всі завантажені IoC з полями @timestamp, indicator_ip, source, threat_type та description. Така форма подання забезпечує

можливість оперативного перегляду нового набору IoC та швидку ідентифікацію їх типу та походження.

На рисунку 3.11 наведено фрагменти панелі, які демонструють часову динаміку надходження індикаторів.

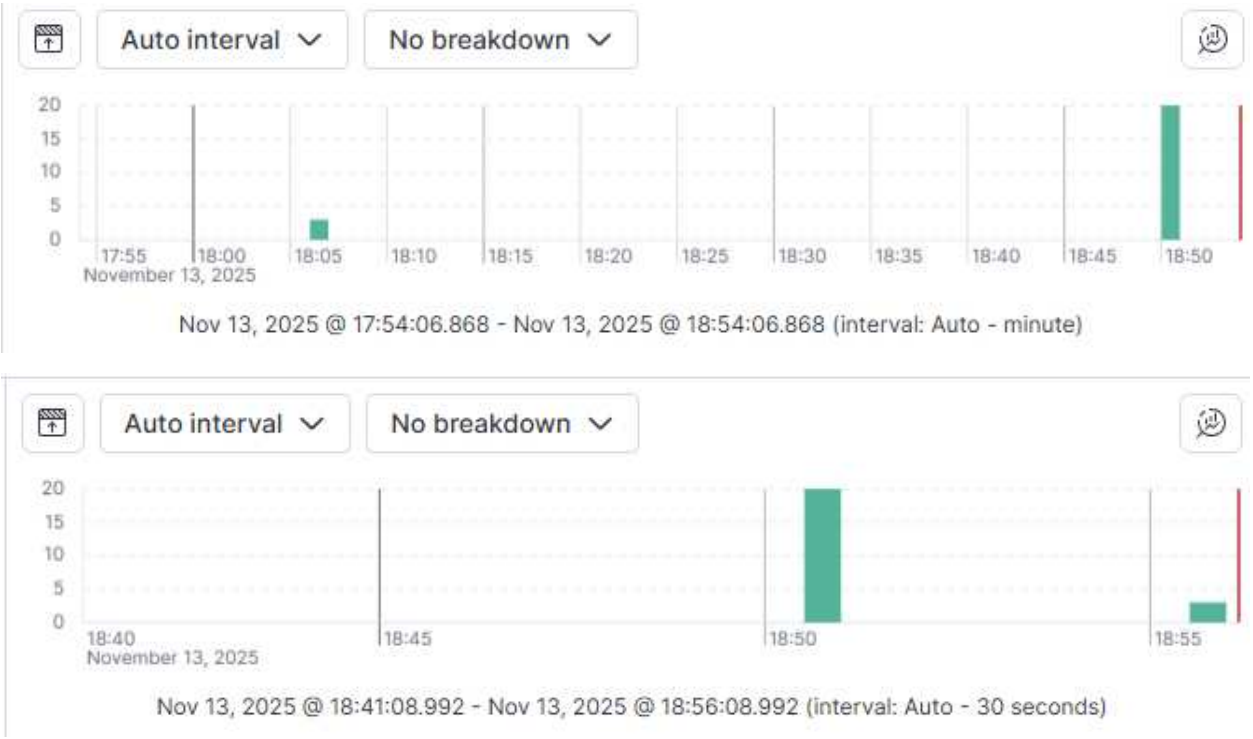


Рисунок 3.11 - Часова динаміка надходження IoC

Гістограми автоматично агрегують події за часовими інтервалами, надаючи візуальне уявлення про активність потоків загроз. Графіки коректно відображають три IoC, імпортовані на попередньому етапі, що підтверджує коректну роботу механізму зіставлення часових міток та прив’язки даних до глобального фільтра часу.

Для оцінки роботи системи у сценарії масового надходження даних було виконано імпорт розширеного набору тестових індикаторів. На рисунку 3.12 наведено фрагмент таблиці, що відображає отриманий набір, який містить різні типи загроз (scanner, malware, anonymizer, spyware, phishing, botnet тощо), а також IoC з різних джерел (manual_test, spamhaus, virus_total, tor_exit, internal_feed, alienvault та інші). Це демонструє можливість ПРЗ обробляти гетерогенні джерела IoC та коректно відображати

їх у уніфікованому аналітичному інтерфейсі.

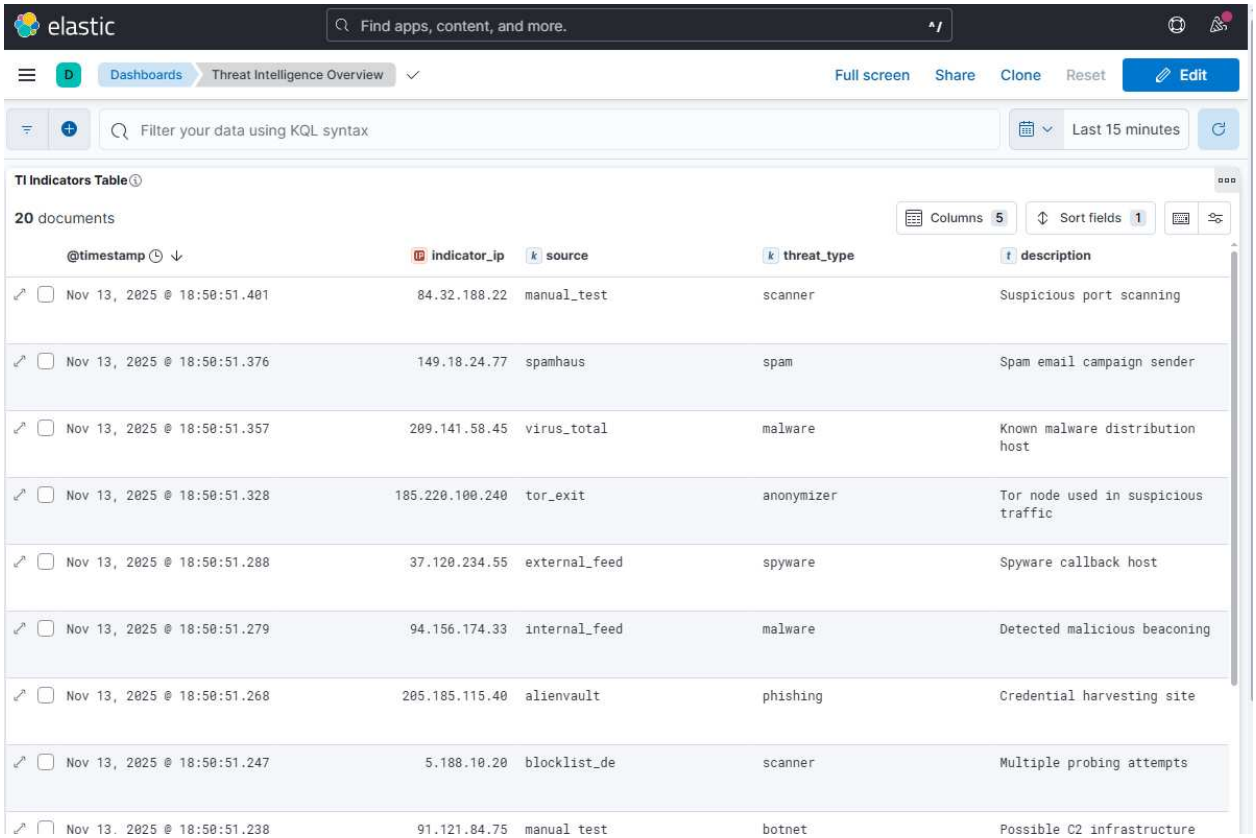


Рисунок 3.12 - Фрагмент таблиці відображення розширеного набору тестових індикаторів

Створена аналітична панель забезпечує наступні функціональні можливості ПРЗ:

- подання індикаторів у табличному вигляді з можливістю фільтрації та пошуку;
- візуалізацію часової активності IoC;
- відображення типів загроз та джерел надходження даних;
- підтримку масштабування при збільшенні обсягів IoC.

3.2.5 Побудова аналітичних візуалізацій

Після формування представлення даних (Data View) наступним етапом є побудова аналітичних візуалізацій, що забезпечують огляд характеристик завантажених IoC та динаміки їх появи. У середовищі Kibana для цього використовується модуль Visualize Library, який містить кілька типів

аналітичних компонентів (рисунок 3.13).

На панелі створення нового віджета користувачеві пропонуються основні інструменти візуалізації:

– Lens – універсальний редактор з механізмом drag-and-drop, рекомендований для більшості сценаріїв побудови діаграм;

– Maps – модуль побудови картографічних уявлень (у даній роботі не використовувався);

– TSVB – інструмент для аналізу часових рядів;

– Custom Visualization – засіб побудови користувацьких графіків на основі Vega.

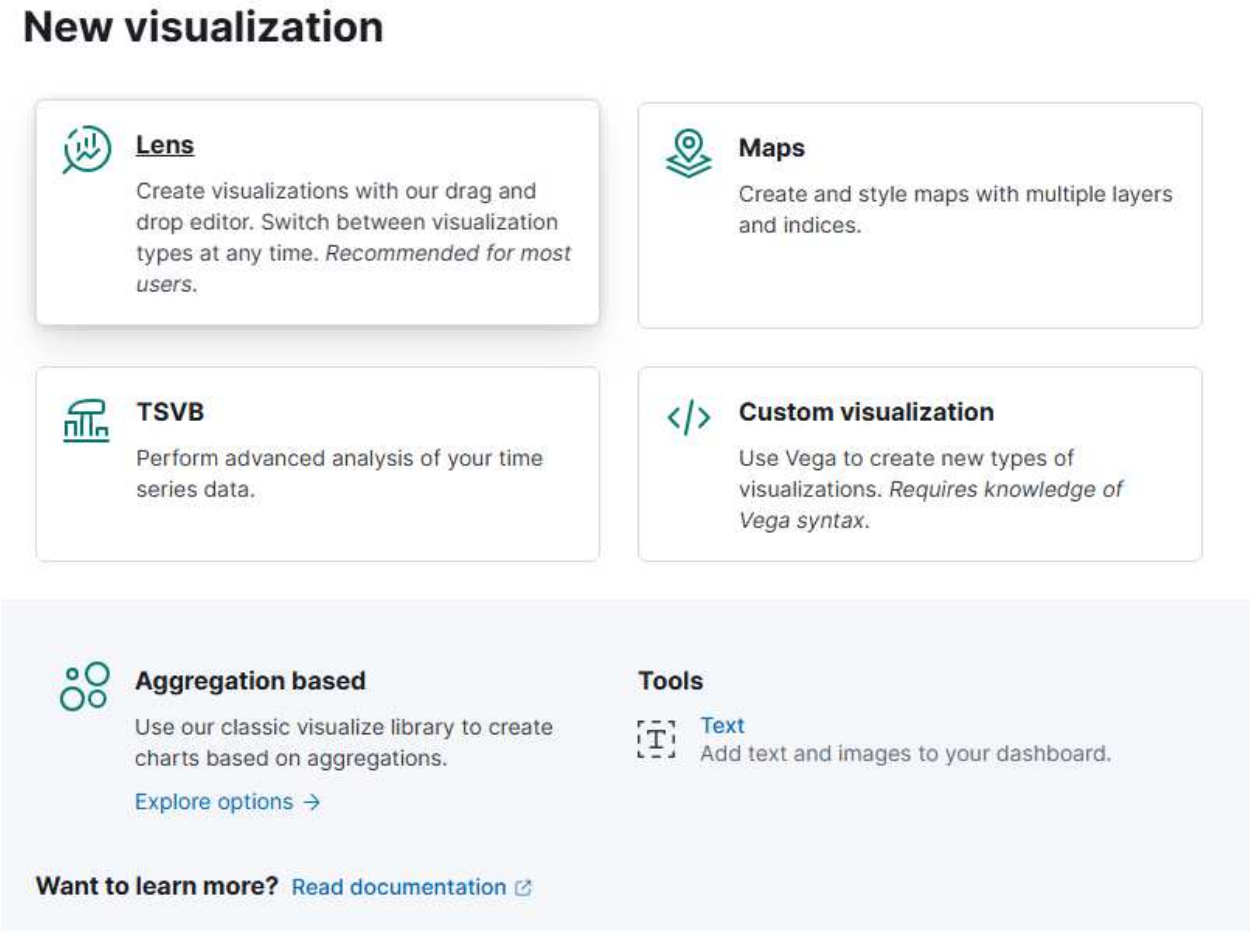


Рисунок 3.13 - Панель створення віджетів

Для побудови візуалізації обрано модуль Lens, оскільки він забезпечує інтуїтивний вибір метрик та вимірів на основі доступних у Data View полів.

На робочій області Lens (рисунок 3.14) відображається перелік

доступних полів Data View TI Indicators: @timestamp, indicator_ip, source, threat_type, description.Ці поля можуть бути використані як групувальні атрибути або як змістовні значення, що відображаються у таблицях, гістограмах чи інших діаграмах.

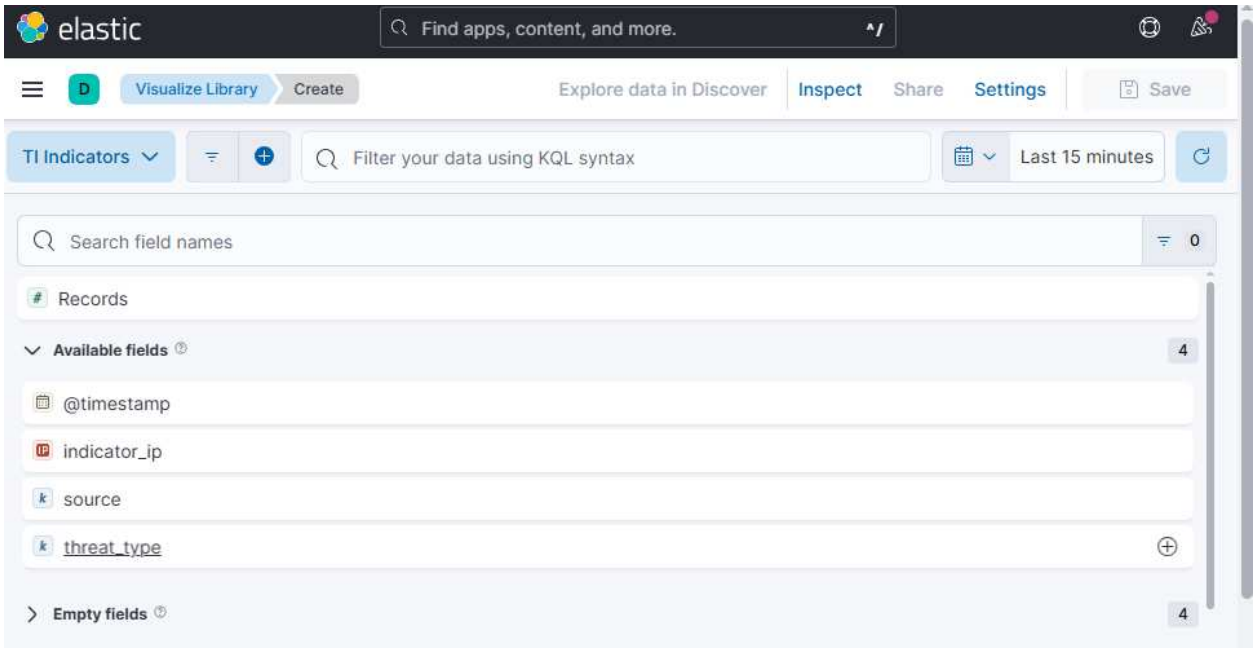


Рисунок 3.14 - Робоча область Kibana Lens

Користувач може обрати тип візуалізації (наприклад, вертикальну гістограму) та перетягнути поле @timestamp у поле Horizontal axis, а будь-який з атрибутів IoC (наприклад, threat_type або source) — у поле Breakdown для групування даних за типами загроз чи джерелами надходження індикаторів.

Після вибору параметрів Lens автоматично будує візуалізацію, яка відображає динаміку появи індикаторів у вибраному часовому діапазоні. Надалі створені графіки можуть бути додані до інтегрованої панелі Threat Intelligence Overview.

Створення аналітичних віджектів у Lens є ключовим етапом побудови інтерактивної панелі моніторингу, що дозволяє оперативно оцінювати кількість IoC, розподіл за типами загроз, джерелами отримання та іншими характеристиками.

Було створено візуалізації наведені на рисунках 3.15-3.19.

На рисунку 3.15 наведено часову гістограму надходження IoC. Візуалізація побудована у форматі Bar vertical stacked, що дозволяє:

- відображати кількість IoC у кожний часовий інтервал (30 секунд);
- розрізняти типи загроз (scanner, botnet, malware, anonymizer, dns, other) за кольорами;
- аналізувати моменти пікової активності, коли до системи надходить найбільша кількість індикаторів;
- оцінювати одночасну появу різних категорій загроз.

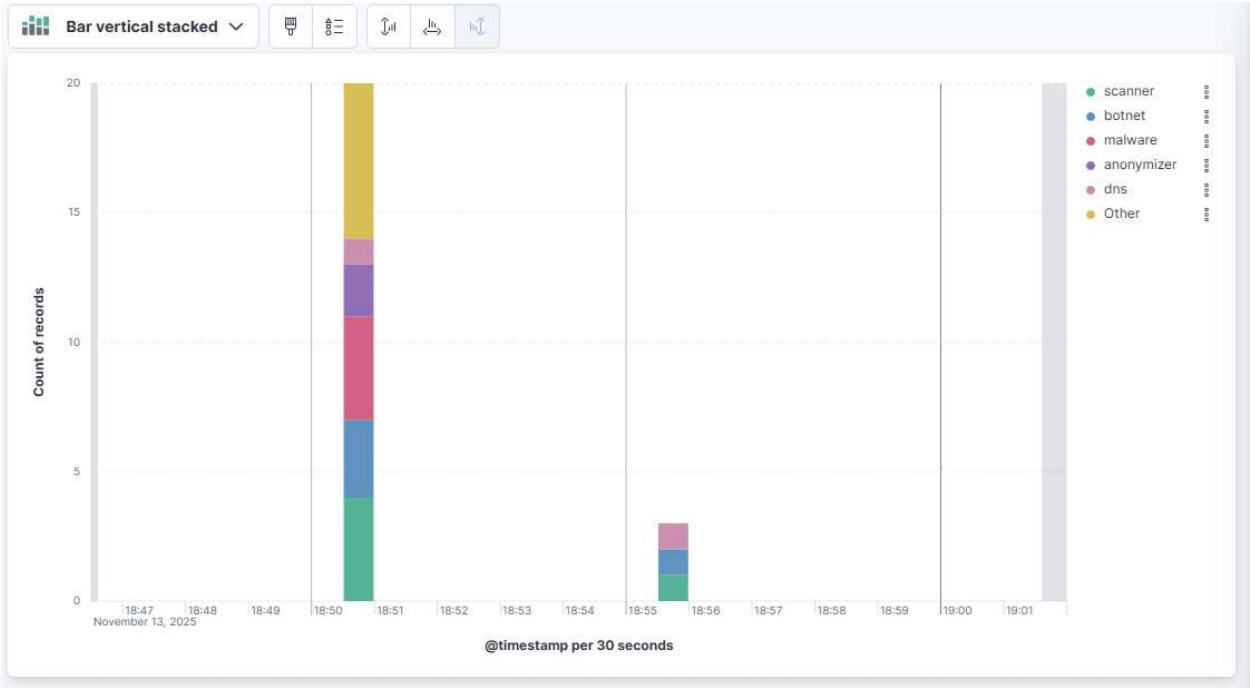


Рисунок 3.15 - Динаміка появи IoC у часі

У наведеному прикладі спостерігається чітко виражений пік активності в момент вставки тестового набору індикаторів - одночасно надходять IoC різних типів, що ілюструє можливості Kibana щодо часової кореляції та агрегування потокових даних.

На рисунок 2.16 наведено кругову діаграму, яка відображає структуру джерел надходження IoC. Вона дозволяє оцінити ефективність та інформативність різних РЗ-фідів, баланс між внутрішніми та зовнішніми джерелами, а ткож домінуючі домінуючі канали надходження IoC (у даному

прикладі найбільшу кількість становить категорія Other, що може свідчити про неоднорідність або різноманітність джерел).

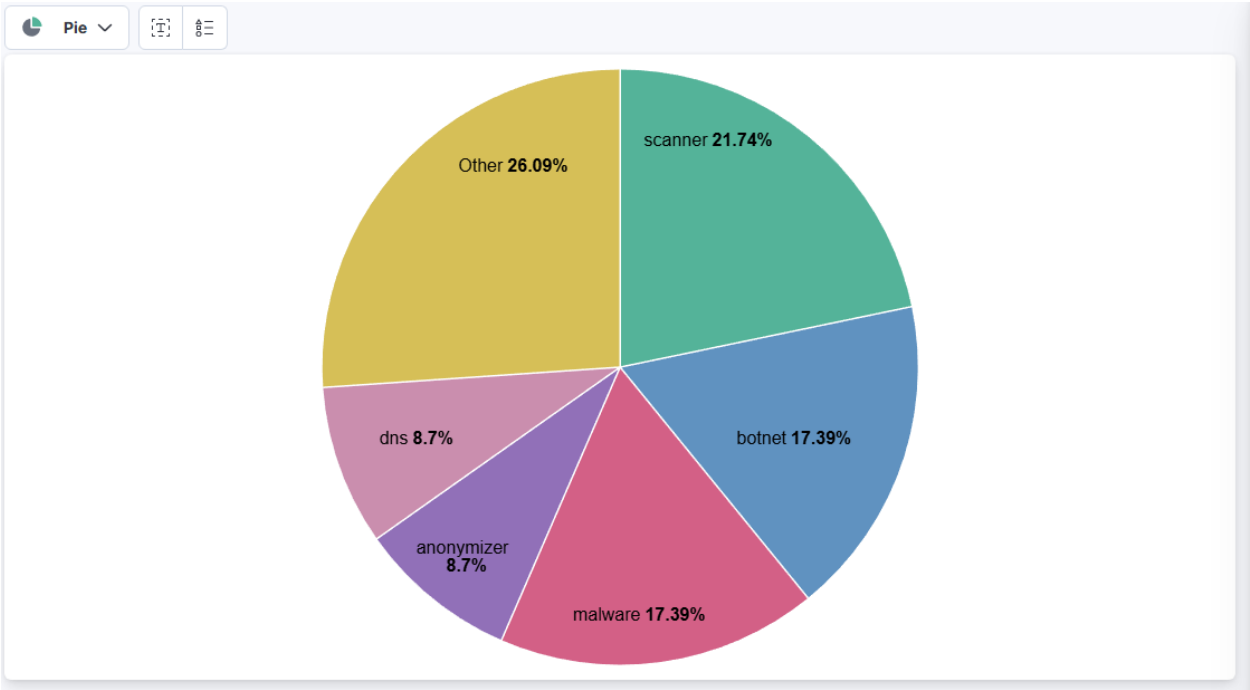


Рисунок 3.16 - Розподіл IoC за джерелами походження

Перевагою такої візуалізації є можливість швидкого розуміння структури каналів отримання IoC та визначення найбільш корисних із них.

Візуалізації типу top-5 одним із базових інструментів аналітики у ПРЗ. Вони дозволяють швидко визначити найбільш значущі або найчастіше повторювані елементи в наборі даних, фокусуючи увагу аналітика на критично важливих категоріях. Наприклад вони дозволяють вділити найбільш критичні чи поширені загрози, виявити аномалій і піків активності, підвищити ефективність моніторингу тощо.

На рисунку 3.17 показано діаграму, яка демонструє кількісний розподіл IoC за типами загроз у межах сформованого набору даних. Візуалізація відображає топ-значення поля threat_type, згруповані за кількістю записів у data stream logs-ti. Дані демонструють нерівномірний розподіл активності, характерний для тестових або реальних РЗ-фідів, де різні типи загроз можуть з’являтися з різною інтенсивністю.

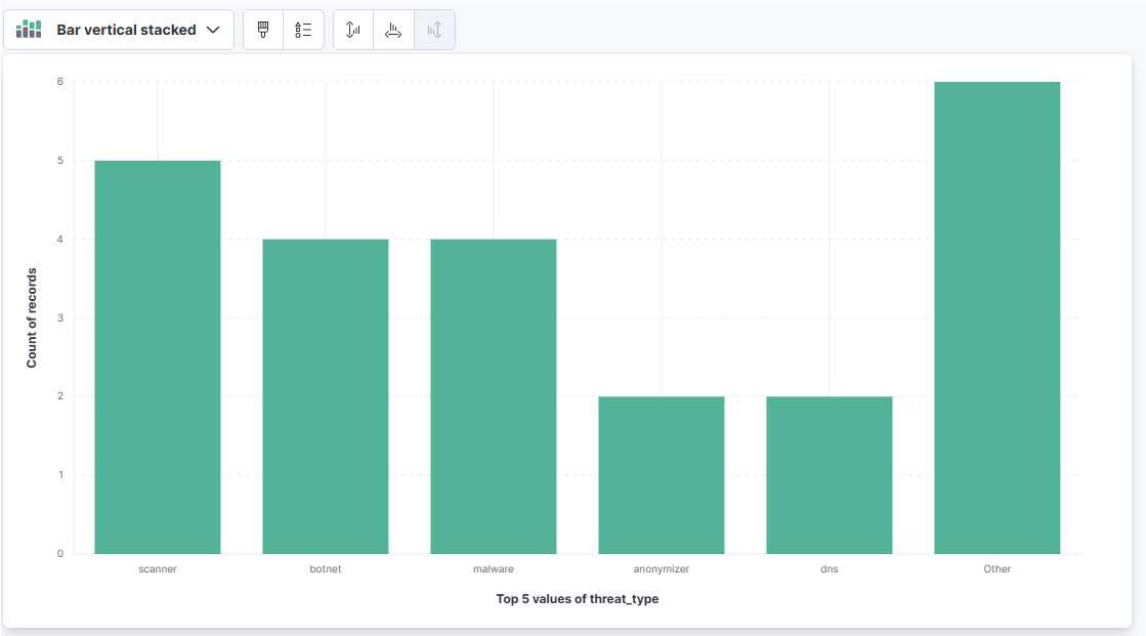


Рисунок 3.17 - Розподіл IoC за джерелами загроз

Візуалізація на рисунку 3.18 демонструє 5 найчастіше повторюваних адрес, що дозволяє визначити найбільш активні або найбільш репрезентативні джерела потенційних загроз. Категорія інші агрегує всі IP-адреси, частота появи яких є меншою, ніж у топ-5, що дає змогу зменшити інформаційний шум і зосередити увагу на ключових елементах.

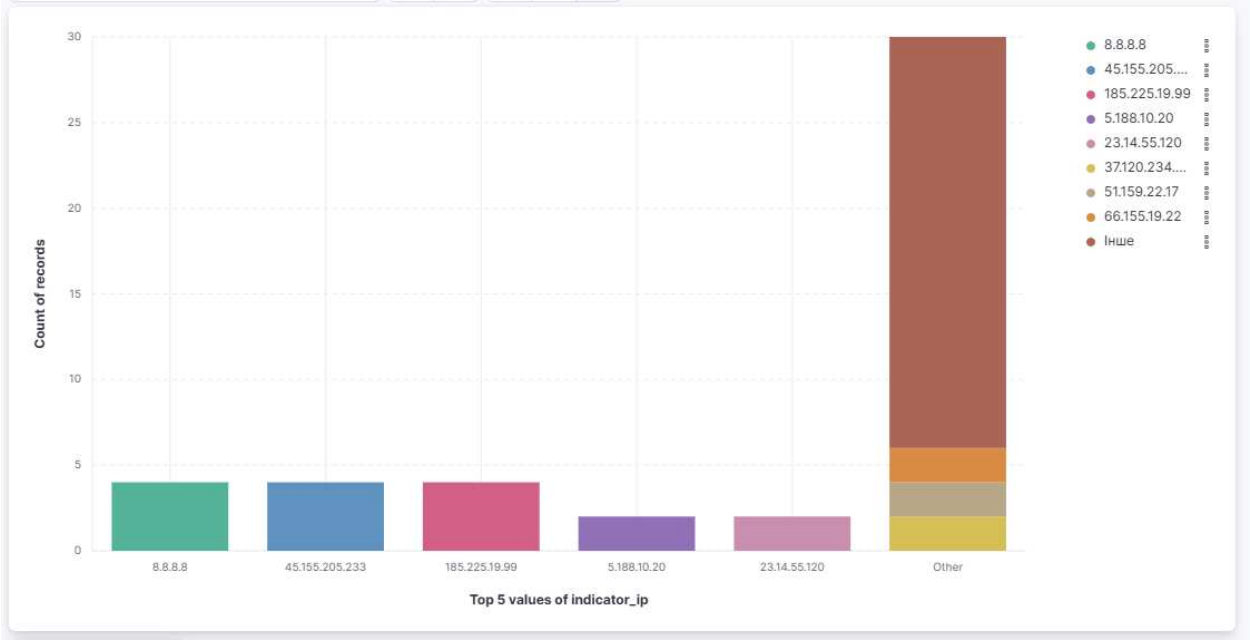


Рисунок 3.18 – Розподіл за індикаторами IP-адрес

Такий тип графіків використовується для виявлення часто представлених показників у потоці РЗ та формування пріоритетів реагування в SOC.

На рисунку 3.19 наведеноо часову динаміку появи IoC, агреговану з інтервалом 30 хвилин. Графік відображає, як змінювалася кількість IoC у часі після їх завантаження до Elasticsearch. Така візуалізація дозволяє оцінити інтенсивність надходження IoC, виявити пікові моменти активності та контролювати динаміку оновлення потоку даних РЗ.

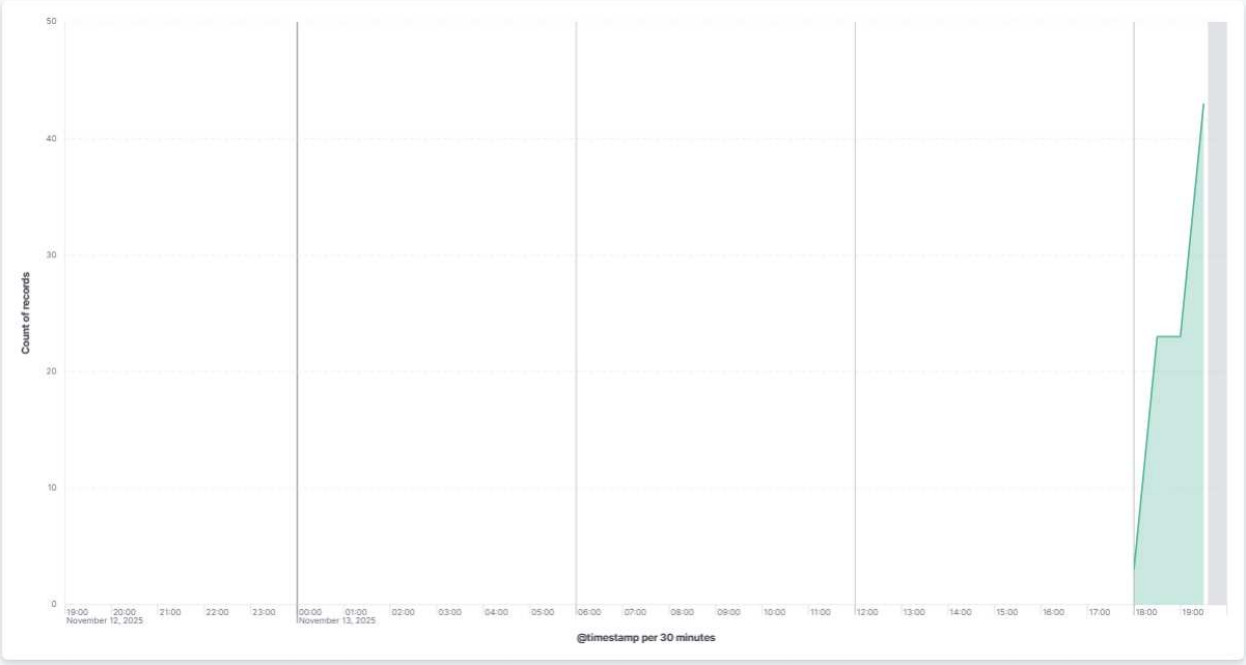


Рисунок 3.19 - Загальна динаміка кількості IoC

На рисунку 3.19 видно, що впродовж більшої частини чсау значення залишаються нульовими, оскільки імпорт IoC не здійснювався. Різкий різкий стрибок кількості записів відповідає моменту пакетного завантаження індикаторів через Python-модуль.

3.2.6 Формування інтегрованої платформи

Усі створені візуалізації були об'єднані в єдину інтерактивну панель (рисунок 3.20), яка забезпечує комплексне подання даних РЗ і дає можливість виконувати багатовимірний аналіз IoC у межах однієї робочої

області Kibana. Створений Dashboard включає такі ключові елементи:

– таблицю індикаторів, що відображає усі отримані IoC із зазначенням їх характеристик;

– часову гістограму появи загроз показує, коли саме надходили індикатори та як змінювалась їх кількість у часовій шкалі;

– розподіл за джерелами індикаторів дозволяє визначити, які джерела формують найбільше IoC;

– графік частоти появи IP-адрес, що дозволяє визначити найбільш активні або критичні індикатори;

– загальна динаміка кількості IoC показує сумарний ріст кількості IoC у часі та дає змогу виявити пікові моменти їх надходження.

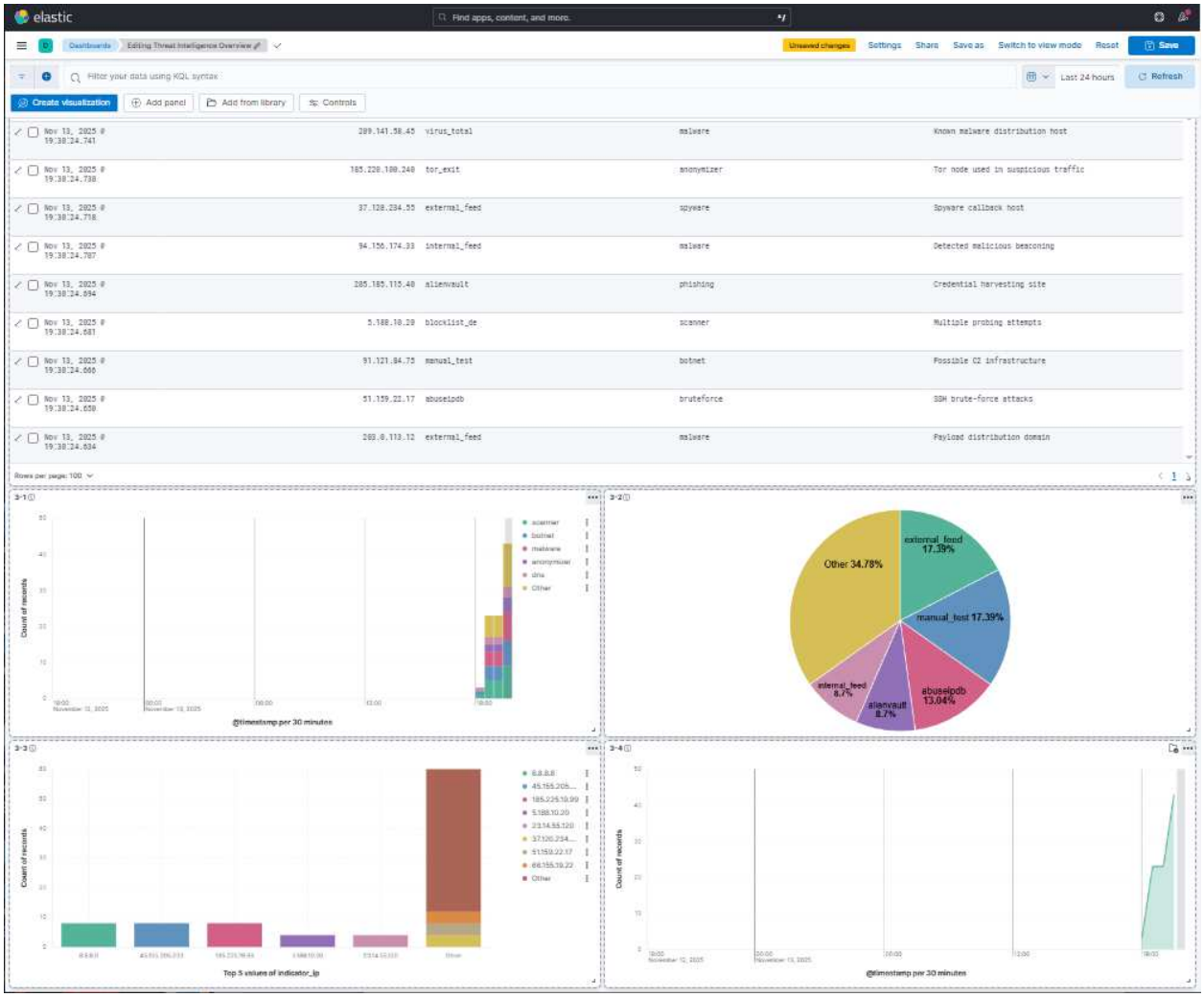


Рисунок 3.20 - Інтерактивна панель ПРЗ

Функціональні можливості Dashboard:

- швидка фільтрація за будь-яким полем (наприклад, тип загрози, джерело або IP-адреса);
- деталізація (drill-down) через клік по елементу графіка з автоматичним оновленням таблиці IoC;
- інтерактивне оновлення всіх візуалізацій відповідно до вибраного фільтра;
- експорт окремих графіків або всієї панелі для звітів чи аналітичних матеріалів;
- масштабування та розширення за рахунок додавання нових типів візуалізацій або додаткових джерел IoC.

Створений Dashboard дозволяє комплексно аналізувати IoC як у часовому вимірі, так і за категоріями та джерелами.

Панель реалізує функціонал плагіна РЗ на основі Kibana та забезпечує повноцінний інструмент для аналітики, моніторингу та прийняття рішень, без необхідності ручного опрацювання IoC. Завдяки інтерактивності Dashboard виступає центральним елементом аналізу для побудованої ПРЗ.

3.3 Оцінка ефективності інтегрованої платформи розвідки загроз

Після наповнення Data Stream тестовими IoC:

- усі записи коректно відображалися в Discover;
- кожне поле автоматично визначено Kibana;
- графіки Lens будувалися без додаткових налаштувань;
- Dashboard працював інтерактивно;
- оновлення даних відбувалося в реальному часі під час повторного запуску Python-модуля.

Для підтвердження працездатності та ефективності розробленої інтегрованої ПРЗ було проведено оцінювання її продуктивності за ключовими технічними показниками (таблиця 3.1). Аналіз охоплював

процеси імпорту IoC, їх індексацію, оброблення кореляційними механізмами та швидкодію при виконанні запитів у Kibana.

Таблиця 3.1 - Основні показники продуктивності

Показник	Опис	Значення	
		отримане	рекомендоване
Latency (затримка індексації)	Час від моменту надходження IoC до появи в data stream `logs-ti`.	~1,2–1,5 с	≤ 5 с
Query time (час виконання запитів)	Середній час формування візуалізацій у Dashboard.	~0,4–0,7 с	≤ 3 с
Resource usage (навантаження CPU/RAM)	Завантаження при одночасному імпорті 50+ IoC.	CPU: 28–35 %, RAM: 1,1–1,3 ГБ	≤ 70 %
Match rate	Частка подій, що відповідають IoC (у тестовому наборі).	~0,8 %	< 1 %
False positive rate	Спрацьовування Indicator Match, що не підтвердили загрозу.	~6–7 %	≤ 10 %
Indexing throughput	Швидкість завантаження індикаторів Python-модулем.	~150–180 документів/с	>100 док./с

Проведена оцінка показала, що спроєктована інтегрована ПРЗ на основі Elasticsearch та плагіну Kibana працює стабільно, має низькі затримки

індексації, забезпечує швидке відображення даних на панелі Threat Intelligence Overview та демонструє прийнятні показники навантаження. Це підтверджує її готовність до подальшого розширення, інтеграції зовнішніх фідів РЗ і використання у реальних середовищах.

ВИСНОВКИ

Проведено дослідження програмних засобів збору, оброблення та візуалізації даних у системах РЗ, зокрема функціональних можливостей Elastic Stack і плагіну Kibana, що дозволило визначити місце Kibana як аналітичного компонента в архітектурі ПРЗ.

Проведено аналіз інструментів та методології побудови ПРЗ, який дозволив сформувати модель платформи, що дозволяє забезпечити всі етапи процесу РЗ - отримання IoC, їх аналітичної інтерпретації, фільтрації та відображення на інтерактивних панелях.

Розроблено архітектуру інтегрованої ПРЗ на основі принципів модульності, контейнеризації та централізованого зберігання даних. Запропонована архітектура забезпечує спеціалізоване сховище IoC, підтримку інтеграції даних з різних джерел, автоматизовані механізми їх оброблення, а також сучасні аналітичні інтерфейси для побудови інтерактивних панелей візуалізації IoC. Реалізація такої структури забезпечує масштабованість, гнучкість та можливість подальшого розширення функціональних можливостей TI-платформи.

Розгорнуто архітектуру проектованої ПРЗ, яка включає Elasticsearch як ядро зберігання й індексації, Kibana як інтерфейс аналітики. Створена інфраструктура дозволяє виконувати кореляцію даних, здійснювати моніторинг показників IoC і формувати базові інструменти для ухвалення рішень у процесах кіберзахисту.

Реалізовано модуль імпорту IoC на Python, налаштовано шаблон даних, механізми оброблення та представлення інформації в Kibana. Створене рішення забезпечує узгоджену взаємодію між компонентами та відтворює повний цикл роботи ПРЗ.

Проведене тестування працездатності та створеної ПРЗ результати якого підтвердили коректність роботи усіх реалізованих компонентів і продемонстрували можливість оперативного отримання, агрегації та

аналізу IoC.

Оцінка ефективності запропонованого рішення показала, що ПРЗ забезпечує виявлення активності різних типів загроз, аналіз їх динаміки, класифікацію за джерелами та формування інформативних візуалізацій, що підвищує ефективність процесів моніторингу безпеки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. CERT-UA офіційні звіти. [Ектронний ресурс].- Режим доступу: https://cert.gov.ua/article/38

2. FireEye/Mandiant Threat Intelligence Reports. [Ектронний ресурс].- Режим доступу: https://www.mandiant.com/resources

3. CrowdStrike Global Threat Report. [Ектронний ресурс].- Режим доступу: https://www.crowdstrike.com/resources/reports/

4. IBM X-Force Threat Intelligence Index. [Ектронний ресурс].- Режим доступу: https://www.ibm.com/reports/threat-intelligence

5. Incident Analysis: Steps to Effective Incident Response. [Ектронний ресурс].- Режим доступу: https://searchinform.com/articles/cybersecurity/measures/incident-response/incident-analysis/

6. Incident Analysis. [Ектронний ресурс].- Режим доступу: https://cybersec.pk/incident-analysis/

7. Poetiray I., Salman M. Information security incident management using iso 27035 standard. Gema Wiralodra. 2023. 14(3). 168-178. 10.31943/gw.v14i3.487.

8. Cybersecurity Incident Managemen. thttps://www.fortinet.com/it/resources/articles/cybersecurity-incident-management

9. Stallings, W., Brown, L. Computer Security: Principles and Practice. – Pearson, 2021. – 864 p. [Ектронний ресурс].- Режим доступу: https://lib.zu.edu.pk/ebookdata/Law/Computer%20Security%20Principles%20and%20Practice-by%20William%20Stallings-3rd%20ed..pdf

10. Scarfone, K., Mell, P. Guide to Intrusion Detection and Prevention Systems (IDPS). – NIST SP 800-94. [Ектронний ресурс].- Режим доступу: https://csrc.nist.gov/publications/detail/sp/800-94/final

11. Wheeler, D. The Threat Intelligence Handbook. – Sixth Edition. – Recorded Future, 2021. [Ектронний ресурс].- Режим доступу:

https://sprotyvg7.com.ua/wp-content/uploads/2023/08/intelligence-handbook-fourth-edition.pdf

12. Shackleford, D. Implementing and Operationalizing Threat Intelligence. – SANS Institute Whitepaper. [Ектронний ресурс].- Режим доступу: https://www.sans.org/white-papers/implementing-operationalizing-threat-intelligence/

13. MITRE ATT&CK® Framework. – MITRE Corporation. [Ектронний ресурс].- Режим доступу: https://attack.mitre.org/

14. The World's First Truly Open Threat Intelligence Community. [Ектронний ресурс].- Режим доступу: https://otx.alienvault.com/

15. AbuseIPDB – IP address threat intelligence database. [Ектронний ресурс].- Режим доступу: https://www.abuseipdb.com/

16. Recorded Future – Threat Intelligence Overview. [Ектронний ресурс].- Режим доступу: https://www.recordedfuture.com/resources

17. Google Security Operations documentation. [Ектронний ресурс].- Режим доступу: https://cloud.google.com/chronicle/docs

18. MISP – Open Source Threat Intelligence Platform. [Ектронний ресурс].- Режим доступу: https://www.misp-project.org/

19. Cisco Talos Intelligence Group. – Detecting hidden threats: AI meets DNS. [Ектронний ресурс].- Режим доступу: https://talosintelligence.com/

20. TheHive Project – Incident Response Platform. [Ектронний ресурс].- Режим доступу: https://thehive-project.org/

21. Primary threat research from Elastic Security Labs. [Ектронний ресурс].- Режим доступу: https://www.elastic.co/security-labs

22. The Elastic Stack. – Офіційна документація. [Ектронний ресурс].- Режим доступу: https://www.elastic.co/guide/en/kibana/current/index.html

23. Kibana plugins. [Ектронний ресурс].- Режим доступу: https://www.elastic.co/docs/reference/kibana/kibana-plugins

24. What is Kibana? [Ектронний ресурс].- Режим доступу: https://logit.io/blog/post/what-is-kibana/

25. Enable threat intelligence integrations. [Ектронний ресурс].- Режим доступу: https://www.elastic.co/docs/solutions/security/get-started/enable-threat-intelligence-integrations

26. Building Effective Dashboards for Threat Intelligence with Kibana and Grafana. [Ектронний ресурс].- Режим доступу: https://thinkcloudly.com/blog/building-effective-dashboards-for-threat-intelligence-with-kibana-and-grafana

27. Home Lab: Enabling and Configuring Threat Intelligence and Detections. [Ектронний ресурс].- Режим доступу: https://www.leveleffect.com/blog/home-lab-enabling-and-configuring-threat-intelligence-and-detections

28. Empowering Threat Intelligence: Integrating OpenCTI with Elasticsearch for Custom Kibana Dashboards. [Ектронний ресурс].- Режим доступу: https://medium.com/%40melaririgodspower04/empowering-threat-intelligence-integrating-opencti-with-elasticsearch-for-custom-kibana-dashboards-9a279b28a443

29. Panels and visualizations Visualize Library. [Ектронний ресурс].- Режим доступу: https://www.elastic.co/guide/en/kibana/current/lens.html

30. Kibana dashboards. [Ектронний ресурс].- Режим доступу: https://www.elastic.co/guide/en/kibana/current/lens.htmlhttps://www.elastic.co/docs/explore-analyze/dashboards

31. Discover. [Ектронний ресурс].- Режим доступу: https://www.elastic.co/docs/explore-analyze/discover

32. Elastic Doc. [Ектронний ресурс].- Режим доступу: shttps://www.elastic.co/guide/index.html

33. Docker Documentation. Docker Engine and Compose. [Ектронний ресурс].- Режим доступу: https://docs.docker.com/

ДОДАТОК А

Скрипт модуля імпорту індиуаторів компрометації

```
import csv
import requests
import json
from datetime import datetime

ELASTIC_URL = "http://localhost:9200"
DATASTREAM = "logs-ti/_doc"

def send_indicator(indicator_ip, source, threat_type, description):
    url = f"{ELASTIC_URL}/{DATASTREAM}"

    payload = {
        "@timestamp": datetime.utcnow().isoformat() + "Z",
        "indicator_ip": indicator_ip,
        "source": source,
        "threat_type": threat_type,
        "description": description
    }

    r = requests.post(url, json=payload)
    return r.status_code, r.text

def import_from_csv(filename):
    with open(filename, newline='', encoding='utf-8') as csvfile:
        reader = csv.DictReader(csvfile)

        for row in reader:
```

```
            status, message = send_indicator(
                row["indicator_ip"],
                row["source"],
                row["threat_type"],
                row["description"]
            )

            print(f"[{status}] {row['indicator_ip']} -> {message}")

if __name__ == "__main__":
    print("Importing IOC from indicators.csv ...")
    import_from_csv("indicators.csv")
    print("Finished!")
```

ДОДАТОК Б

Копія публікацій