КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 3 (31), 2025
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2025.31.1031
УДК 004.8

Волотовський Олександр Владиславович
студент, магістр кафедри безпеки інформаційних технологій
Національний університет «Львівська політехніка», Львів, Україна
ORCID: 0009-0003-3102-3694
oleksandr.volotovskyi.mkbbi.2024@lpnu.ua

Банах Роман Ігорович
доктор філософії, доцент кафедри безпеки інформаційних технологій
Національний університет «Львівська політехніка», Львів, Україна
ORCID: 0000-0001-6897-8206
roman.i.banakh@lpnu.ua

Піскозуб Андріян Збігнєвич
кандидат технічних наук, доцент,
доцент кафедри захисту інформації
Національний університет «Львівська політехніка», Львів, Україна
ORCID: 0000-0002-3582-2835
andriian.z.piskozub@lpnu.ua

РОЗРОБКА АГЕНТА ДЛЯ АНАЛІЗУ СПОВІЩЕНЬ НА ОСНОВІ ШТУЧНОГО ІНТЕЛЕКТУ З ВИКОРИСТАННЯМ ПРОГРАМНИХ ЗАСОБІВ N8N ТА ELASTIC SECURITY

Анотація. У даній статті розглядається питання підвищення ефективності реагування на інциденти у центрах операцій безпеки шляхом впровадження автоматизованої системи аналізу подій, що поєднує платформу оркестрації робочих процесів n8n та великі мовні моделі. Застосування такого підходу враховує актуальні виклики кібербезпеки, зокрема безперервне зростання обсягу сповіщень, ускладнення інфраструктурних середовищ, обмеженість людських ресурсів та зниження ефективності ручного аналізу подій. Запропонована архітектура включає SIEM-систему Elastic Security як основне джерело подій, платформу n8n як оркестратор, LLM-агентів для природномовної інтерпретації інцидентів, а також зовнішні сервіси збагачення даних (AbuseIPDB, VirusTotal). Ключові функції системи охоплюють автоматизоване отримання, нормалізацію та збагачення подій безпеки, визначення рівня ризику, класифікацію інцидентів, формування коротких аналітичних звітів у форматі 5W (Who, What, When, Where, Why) та генерацію релевантних рекомендацій для реагування. Для оцінювання працездатності розробленого підходу проведено комплексне експериментальне дослідження, що включало моделювання реальних сценаріїв SOC, генерування типових інцидентів, виконання автоматизованого конвеєра аналізу та порівняння результатів із роботою аналітиків першого рівня підтримки. Окрему увагу приділено оцінюванню стабільності роботи LLM-агента за умов зміни формату подій, різноманітності джерел системних журналів та варіативності якості вхідних даних. Додатково досліджено поведінку системи у випадках неоднозначних або частково відсутніх атрибутів інцидентів, а також протестовано механізми компенсації впливу неповних або шумових даних. Результати аналізу засвідчили, що інтегрований підхід дозволяє автоматизувати значну частину рутинних операцій, забезпечує високу повторюваність прийнятих рішень, усуває суб'єктивність, притаманну ручному аналізу, та сприяє формуванню уніфікованих вердиктів. Окремо оцінено якість формування аналітичних висновків, відповідність опису події її реальному контексту, здатність LLM-класифікатора визначати ключові сутності, причини події та початкові рекомендації щодо дій реагування. Також перевірено сумісність системи з операційною інфраструктурою SOC та її здатність до масштабування за умов зростання кількості подій без зниження продуктивності. Отримані результати підтверджують, що запропонована система підвищує швидкість і якість реагування на інциденти, створює основу для переходу від реактивної до проактивної моделі

SOC та зменшує когнітивне навантаження на аналітиків. Завдяки модульності оркестратора n8n і гнучкості LLM-агентів система може адаптуватися до нових типів загроз, розширювати функціонал без значних витрат та інтегруватися з широким спектром корпоративних сервісів. Таким чином, розроблене рішення формує технологічне підґрунтя для подальшого впровадження інтелектуальної автоматизації в інцидент-менеджмент і підтримує розвиток SOC наступного покоління.

Ключові слова: автоматизація SOC; n8n; великі мовні моделі; Elastic Security; оркестрація інцидентів; кібербезпека; SIEM; LLM-агент.

ВСТУП

Сучасні Центри операцій безпеки (SOC) функціонують в умовах безперервного зростання кількості інцидентів та подій, що генеруються хмарними сервісами, кінцевими пристроями, платформами управління ідентичностями та мережевими сенсорами. Значна частина таких сповіщень має низький пріоритет або є повторюваною, однак навіть за цих умов потребує перегляду аналітиками першої лінії , відповідальними за первинну класифікацію, збирання контексту та оперативний аналіз. Розширення інфраструктури та збільшення площини атаки призводять до зростання навантаження на фахівців SOC, що спричиняє ефект «втоми від сповіщень», уповільнення обробки інцидентів та підвищення ризику пропуску критичних загроз. Це формує потребу у впровадженні автоматизованих інструментів, здатних прискорити процес первинного аналізу, зменшити когнітивне навантаження на аналітиків і підвищити стійкість операцій безпеки.

Одним із перспективних напрямів розвитку SOC є інтеграція платформ оркестрації робочих процесів з аналітичними компонентами на основі штучного інтелекту. Оркестраційні рушії забезпечують автоматизоване отримання, нормалізацію, збагачення та маршрутизацію даних, тоді як великі мовні моделі (LLM) здатні інтерпретувати контекст подій, оцінювати рівень ризику та формувати структуровані аналітичні висновки. Поєднання цих підходів створює потенціал для часткової автоматизації діяльності аналітиків першої лінії, скорочення часу реагування та зменшення обсягу ручних операцій на ранніх етапах обробки інцидентів.

Попри активний розвиток автоматизації в сфері кібербезпеки, практичні аспекти інтеграції оркестраційних платформ з LLM для автоматизації процесів SOC досліджені недостатньо. Наявні наукові праці переважно зосереджуються на вивченні окремих компонентів — оркестраційних систем, методів побудови інтелектуальних агентів, можливостей SIEM-платформ або застосуванні технологій SOAR. Натомість відсутні комплексні дослідження, що описують інтегровану методологію, яка поєднує SIEM-дані, автоматизацію робочих процесів і LLM-аналітику в єдину операційну модель. Це обґрунтовує необхідність глибшого аналізу систем автоматизованого аналізу на основі реальних даних.

Постановка проблеми. Незважаючи на прогрес у сфері автоматизації, SOC значною мірою покладаються на ручний збір інформації про події, що збільшує час аналізу, сприяє перевантаженню аналітиків і знижує достовірність прийняття рішень на ранніх етапах. На даний момент відсутні інтегровані, експериментально підтверджені рішення, які поєднують оркестрацію робочих процесів, обробку подій у SIEM та аналітику на основі LLM у єдиний конвеєр первинної обробки інцидентів. Усунення цієї прогалини є критично важливим для підвищення ефективності SOC і загальної безпеки організацій.

Аналіз останніх досліджень і публікацій. Наукові та прикладні джерела демонструють, що платформи автоматизації робочих процесів здатні ефективно оркеструвати отримання та обробку даних у середовищах кібербезпеки [1]. Підходи до застосування штучного інтелекту в реагуванні на інциденти підтверджують можливість використання LLM для підтримки або часткової емуляції функцій аналітиків першого рівня [2]. SIEM-системи, такі як Elastic Security, відіграють ключову роль у структурованому зборі, кореляції та аналізі даних про інциденти, що робить їх важливим компонентом автоматизованих конвеєрів обробки [3]. Індустріальні дослідження підкреслюють, що технології SOAR сприяють підвищенню швидкості реагування та зменшенню операційного навантаження на аналітиків [4].

Разом із тим інтеграція оркестрації, SIEM-даних і LLM-аналітики все ще недостатньо висвітлена, що визначає актуальність даного дослідження.

Мета статті. Метою статті є проєктування, реалізація та оцінка автоматизованої системи первинної обробки інцидентів, що інтегрує оркестрацію робочих процесів, збирання подій із SIEM та модулю аналітики на основі великих мовних моделей. Для досягнення поставленої мети визначено такі завдання:

- Дослідити можливості Elastic Security та проаналізувати роботу з API для побудови конвеєра автоматизації;
- Розробити архітектуру агента на базі n8n;
- Реалізувати автоматизований процес збору, уніфікації та доповнення подій;
- Інтегрувати велику мовну модель для побудови структурованого опису подій;
- Розробити логіку автоматичної ескалації інцидентів
- Провести перевірку системи;
- Оцінити показники ефективності.

ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖЕННЯ

Теоретичним підґрунтям дослідження є положення сучасної кібербезпеки, що визначають роботу центрів операцій безпеки та механізми аналізу інцидентів. Ключовим елементом виступає концепція SOC як багаторівневої структури моніторингу, де першочергове навантаження припадає на аналітиків початкового рівня. Дослідження спирається на життєвий цикл інциденту, що включає виявлення події, її нормалізацію, збагачення, класифікацію, визначення пріоритету та ескалацію.

Основою технічної частини дослідження є принципи роботи SIEM-систем, зокрема моделі агрегування логів, кореляції подій та формування сповіщень. Використання Elastic Security базується на її архітектурі індексів, механізмах генерування алертів, часових вікнах обробки та API-доступі до телеметрії.

Другий блок теоретичних положень складають принципи автоматизації реагування, що використовуються в SOAR-підході: оркестрація, сценарна обробка даних, фільтрація шумових подій, інтеграція з зовнішніми сервісами та логічне маршрутизування інцидентів. Платформа n8n у цьому контексті відображає концепцію модульного конвеєра — з окремими етапами збору, нормалізації.

У дослідженні застосовано також положення сучасної обчислювальної лінгвістики, які описують принципи роботи великих мовних моделей. Їх використання ґрунтується на здатності інтерпретувати напівструктуровані дані, виділяти ключові атрибути події, формувати узагальнення за схемою 5W та здійснювати первинну класифікацію відповідно до поведінкових патернів і тактичних категорій MITRE ATT&CK.

Окрему групу складають методи оцінювання ефективності систем реагування, що включають метрики часу виявлення й часу реагування (MTTI, MTTA, MTTR), а також метрики якості класифікації — точність, повноту та F1-міру. Ці показники використовуються як формальна основа для порівняння ручного та автоматизованого підходів.

МЕТОДИКА ДОСЛІДЖЕННЯ

Практичне впровадження та тестування розробленої системи виконувалося в умовах діючої інфраструктури. Для аналізу застосовувалися інциденти, що надходили до команди SOC у щоденній операційній діяльності, включаючи події з таких джерел, як AWS CloudTrail, Google Workspace, Microsoft Defender, Linux та Windows Security Events. Усі дані використовувалися у тому вигляді, в якому вони надходили до аналітиків, без додаткової обробки чи втручання, відповідно до внутрішніх політик безпеки компанії та з дотриманням вимог конфіденційності.

Об’єктом дослідження є процес обробки інцидентів інформаційної безпеки в операційному центрі безпеки, зокрема на рівні первинного реагування аналітиками першої лінії. Дослідження зосереджене на технологічних, процедурних та аналітичних аспектах цього процесу, включаючи виявлення, класифікацію, збагачення та формування вердиктів щодо інцидентів.

Критеріями оцінки ефективності стали класичні метрики продуктивності SOC: середній час до початку аналізу (MTTA), середній час до реагування (MTTR), рівень хибнопозитивних спрацьовувань, кількість інцидентів, що опрацьовуються одним аналітиком за зміну, а також якісні показники, такі як точність класифікації типу події, відповідність сформованого вердикту реальному рівню ризику, повнота опису та релевантність запропонованих дій. На основі цих параметрів було проведено комплексне зіставлення результатів ручної та автоматизованої обробки.

Інструментально дослідження базувалося на інфраструктурі Elastic Security, де здійснювався моніторинг подій із таких джерел, як AWS CloudTrail, Google Workspace, Microsoft Defender, системи на базі Linux та Windows. Для побудови автоматизованого процесу обробки подій використовувалася платформа n8n, яка реалізувала логіку інтеграції з LLM. Оцінювання вердиктів здійснювалося у порівнянні з ручною експертною оцінкою подій та внутрішніми політиками компанії щодо категоризації ризиків.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Elastic Security є комплексним рішенням для моніторингу, виявлення та аналізу інцидентів, розробленим на основі платформи Elastic Stack [6]. У запропонованій системі API Elastic виступає основним джерелом даних для автоматизованого агента, тоді як платформа n8n здійснює передачу подій безпеки у форматі JSON через відповідні програмні інтерфейси. До таких подій належать атрибути на кшталт назви правила (rule.name), ідентифікатора користувача, IP-адреси джерела, імені хоста, часової позначки (@timestamp) та інших артефактів, необхідних для подальшої аналітики. Elastic API забезпечує доступ до даних у «сирому» форматі, що дозволяє інтеграційним конвеєрам оперувати первинною інформацією та здійснювати її гнучке оброблення в режимі реального часу.

Використання HTTP-вузлів у n8n забезпечує можливість періодичного звернення до потоків даних або окремих індексів Elastic з метою отримання актуальних сповіщень про події безпеки. Після надходження дані включаються в автоматизований конвеєр, де проходять нормалізацію та збагачення. На цьому етапі можуть виконуватися операції з виділення індикаторів компрометації (IOC), зіставлення подій з матрицею MITRE ATT&CK, а також побудова структурованих аналітичних підсумків у форматі 5W (Who, What, When, Where, Why). Така інтеграційна модель істотно скорочує тривалість первинного аналізу та зменшує навантаження на аналітиків, передаючи повторювані дії автоматизованим модулям. Ці модулі можуть працювати як з великими мовними моделями, так і з наявними рішеннями класу SOAR, що підвищує загальний рівень автономності процесів.

Архітектура системи побудована таким чином, щоб забезпечити максимальну прозорість і можливість аудиту кожного етапу. Усі операції — від отримання даних через Elastic API та їх збагачення до побудови аналітичного висновку мовною моделлю — є модульними, відтворюваними та контрольованими засобами керування версіями. Це дає змогу формувати детальні операційні журнали та метадані, які фіксують логіку прийняття рішень і слугують доказовою базою для внутрішнього контролю та зовнішніх перевірок. Наявність такої простежуваності підвищує довіру до автоматизації та забезпечує фахівців SOC, аналітиків реагування й аудиторів чітким описом перебігу процесів.

Завдяки цьому система спрощує доведення відповідності вимогам міжнародних стандартів, включно з ISO/IEC 27001, NIST Cybersecurity Framework та SOC 2 [8, 9].

Оскільки кожне сповіщення проходить передбачувану та повторювану послідовність — від збагачення до класифікації, — система здатна формувати структуровані машиночитні звіти, що фіксують фактичне застосування політик безпеки на кожному етапі обробки. Такий підхід підсилює прозорість та відтворюваність процесів, що є важливим як для внутрішнього контролю, так і для зовнішніх аудитів чи форензічних розслідувань, оскільки забезпечує доказову базу й узгодженість із прийнятним рівнем ризику.

Фінальним етапом обробки даних у запропонованій архітектурі є застосування великих мовних моделей (LLM), зокрема системи Grok, яка виконує функції віртуального аналітика першого рівня. Після отримання даних від n8n модель здійснює їх аналіз за схемою 5W (хто, що, коли, де, чому), формуючи узагальнений опис ключових характеристик інциденту: учасників події, характер поведінки, час і місце її виникнення та обґрунтування значущості. LLM також здійснює попередню оцінку серйозності інциденту (наприклад, «потребує розслідування» або «низький пріоритет»).

Модель виконує завдання, які зазвичай вимагали б участі аналітика першої лінії, однак робить це значно швидше та масштабованіше, забезпечуючи прискорення обробки й стандартизацію результатів. Формалізована структура інтерпретації подій дає змогу усувати суб’єктивність, характерну для мануальної обробки, особливо в умовах зміни персоналу, різного рівня кваліфікації або втоми аналітиків. Додаткова інтеграція зовнішніх джерел збагачення — репутаційних баз IP-адрес, контекстних метаданих правил, міток MITRE ATT&CK — розширює обсяг доступної моделі інформації та сприяє формуванню більш точного аналітичного висновку. У процесі тривалої роботи з реальними SOC-даними мовні моделі поступово адаптують свої класифікаційні підходи до внутрішніх пріоритетів організації [10], формуючи контекстно обізнаний, частково автономний компонент первинного аналізу, що зменшує обсяг ручної роботи й підвищує точність сигналів.

Агент функціонує з фіксованою періодичністю — кожні 5 хвилин. Такий інтервал забезпечує баланс між оперативністю отримання нових сповіщень та стабільністю функціонування конвеєра. З технічного погляду, 5-хвилинна частота враховує час, необхідний Elasticsearch для завершення індексації, а використання «зсувного вікна» у логіці запитів мінімізує ризик пропуску подій навіть у разі пікових навантажень. З експлуатаційної точки зору, цей інтервал підтримує прийнятний рівень API-викликів та звернень до LLM, не перевантажуючи обчислювальні ресурси й забезпечуючи стабільний ритм роботи системи. Водночас періодичність у 5 хвилин не створює надмірного потоку оперативних подій для аналітиків, як це могло б відбуватися в інтервалах 1–2 хвилини, і не спричиняє затримок, характерних для довших відрізків.

Таким чином, визначена періодичність забезпечує оптимальне співвідношення між швидкістю виявлення сповіщень, передбачуваністю роботи конвеєра та можливостями аналітичної команди.

Вибраний 5-хвилинний інтервал узгоджується з типовим операційним ритмом роботи SOC і забезпечує оптимальний часовий проміжок для первинного впорядкування сповіщень. Така частота дозволяє своєчасно відокремлювати повторювані або фонові події, корелювати їх із попередніми інцидентами та, за потреби, залишає достатньо часу для ручної перевірки у випадку накладання сповіщень. Це дає змогу аналітикам утримувати стабільну продуктивність без перевантаження низькоцінним шумом або накопичення незакритих завдань.

З позиції проєктування автоматизованих систем фіксовані інтервали спрощують організацію циклів обробки даних. Агенти опрацьовують події послідовними часовими вікнами, що робить роботу контрольних точок передбачуваною та зручною для аудиту. Така модульність формує повторюваний шаблон, придатний для масштабування на інші джерела телеметрії або додаткові середовища без істотних змін до архітектури.

На другому етапі робочого процесу ініціюється запит до REST API Elastic Security з отриманням сповіщень у форматі JSON. Повернені структури містять стандартизовані атрибути — rule.name, ідентифікатори користувачів, IP-адреси джерела та призначення, характеристики хоста, часові позначки. Таким чином система отримує вже формалізовані та машиночитні дані, які надалі підлягають нормалізації та збагаченню.

Надійність цього етапу забезпечується двома ключовими рішеннями. По-перше, запит виконується в межах діапазону від останньої контрольної точки до поточного моменту, що запобігає повторній обробці подій і водночас нівелює ризик пропуску сповіщень, які були проіндексовані із затримкою. По-друге, використання пагінації на основі механізму search_after дає змогу коректно опрацьовувати великі обсяги записів без перевищення обмежень пам’яті або пропускної здатності. У сукупності це створює надійний транспортний шар, який передає дані до всіх наступних модулів аналізу.

Архітектура системи передбачає відмовостійкість. У разі короткочасних мережевих збоїв, нестабільності хмарних сервісів чи інших непередбачених ситуацій конвеєр не припиняє роботу. Завдяки механізму контрольних точок обробка продовжується з місця зупинки, що виключає дублювання або втрату сповіщень і забезпечує цілісність даних навіть за умов затримок індексації у SIEM або тимчасових збоїв в Elasticsearch.

У контексті безпеки реалізовано сувору модель контролю доступу на основі API-ключів та OAuth-токенів, що відповідає принципу найменших привілеїв та забезпечує повну простежуваність операцій. Такі механізми є необхідними в середовищах, де вимоги стандартів ISO/IEC 27001 та SOC 2 передбачають чітку регламентацію процедур та аудитованість процесів [11].

Наступним етапом є робота інтеграційної JavaScript-функції у n8n, яка виконує нормалізацію й структурування вхідних даних. Незважаючи на просту реалізацію, цей модуль забезпечує уніфікацію ключових атрибутів, що є критично важливим для подальшого аналізу. Особлива увага приділяється визначенню вихідної IP-адреси: залежно від джерела телеметрії вона може міститися у полях source.ip, client.ip, host.ip, destination.ip або observer.ip. Скрипт послідовно перевіряє доступні поля та записує перше валідне значення у нормалізоване поле src_ip; за його відсутності фіксується значення N/A. Це забезпечує передбачувану структуру, необхідну для подальшої автоматизованої обробки.

Уніфікація даних помітно підвищує якість аналізу: без стабільної структури модель значну частину ресурсів витрачає на інтерпретацію формату, а не змісту події. Подібні структурні розбіжності є типовими для телеметрії Elastic та honeypot-середовищ [12], що підкреслює важливість нормалізації.

Етап збагачення доповнює подію компактними, аналітично значущими атрибутами замість передавання «сирого» JSON. Зокрема, скрипт може автоматично звертатися до AbuseIPDB або VirusTotal для отримання репутаційних характеристик IP-адреси [13]. Отримані метадані інтегруються у вигляді атрибутів ip_reputation_score, blacklist_hit чи threat_category, формуючи попередньо скорельований контекст. Це знижує навантаження на модель і підвищує точність подальшої класифікації.

На четвертому етапі нормалізоване сповіщення передається великій мовній моделі, налаштованій на виконання функцій аналітика першої лінії. Її завдання — сформувати стислий, структурований виклад події за схемою 5W на основі ключових атрибутів (rule.name, @timestamp, src_ip, ідентифікатори користувачів та контекст ресурсу). Такий підхід забезпечує оперативність обробки й створює однорідні, стандартизовані резюме, придатні для подальшої кореляції.

Модель працює виключно з наданими даними, без припущень або «домислення» змісту [14]. Це гарантує відтворюваність результатів і зберігає цілісність аналітичних висновків.

Компонент обізнаності щодо загроз інтегровано безпосередньо у логіку промпту. Якщо у сповіщенні наявна вихідна IP-адреса, модель враховує репутаційні дані, отримані з AbuseIPDB або аналогічних джерел. У разі виявлення ознак зловмисності ця інформація додається до секції WHY і впливає на початковий аналітичний висновок. Нейтральні або невідомі індикатори враховуються, але не визначають фінальну оцінку.

На практиці цей етап перетворює технічні, фрагментовані структури на стислий і зрозумілий аналітичний виклад протягом кількох секунд. Порівняно з мануальною обробкою, модель забезпечує значне скорочення часу інтерпретації (MTTI) та повідомлення аналітика (MTTN), зменшуючи ресурсні витрати на розшифрування структури події.

У підсумку LLM виконує роль когнітивного модуля аналітичного конвеєра: визначає рівень серйозності, класифікує інцидент, виділяє ключові індикатори та створює готову до дії аналітичну довідку. Підхід демонструє високу масштабованість і добре узгоджується з типовою роботою аналітика першої лінії. Подібні концепції застосовувалися й у середовищах AWS під час автоматизованих перевірок безпеки відповідно до CIS Benchmarks [15].

Отримана у процесі роботи інформація використовується для поступового вдосконалення моделі та узгодження її рішень з реальними пріоритетами організації. З часом система точніше відображає внутрішній ризик-профіль, особливості активів і типові сценарії реагування [16]. Таким чином, інструмент еволюціонує від статичного

модуля до адаптивного цифрового помічника, здатного враховувати специфіку SOC і зменшувати кількість хибнопозитивних сповіщень. Інтеграція зворотного зв'язку аналітиків додатково покращує фільтрацію та релевантність рекомендацій.

Фінальним етапом є формування та надсилання структурованого сповіщення. Підготовлений LLM висновок подається у вигляді електронного листа або тикета, який містить блок WHO–WHAT–WHEN–WHERE–WHY, а також ключові атрибути події: ідентифікатор користувача, src_ip, часову мітку, назву правила та обґрунтування класифікації. Такий формат забезпечує «високу сигнальність»: інформація подана стисло, без зайвого шуму, та готова до оперативного використання.

У разі потреби поглибленого аналізу лист може містити пряме посилання на відповідну подію у SIEM (наприклад, URL Kibana), що спрощує доступ до повного контексту [17]. Службові метадані — ідентифікатори правил, унікальний ідентифікатор інциденту (UUID) або вкладений JSON — забезпечують простежуваність та автоматизацію подальших дій.

Цей етап завершує повний цикл обробки: машинний аналіз перетворюється на практичну, оперативну інформацію, оформлену у форматі, звичному для SOC-аналітиків. Незважаючи на високий рівень автоматизації, результат залишається надійним, аудитним і придатним для швидкого реагування (рис. 1).

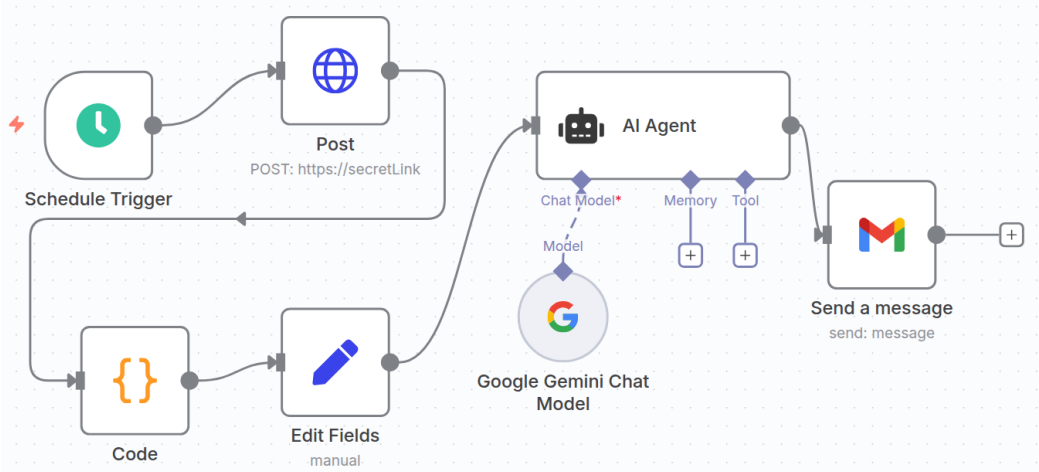

Рис. 1. Сценарій автоматизації обробки даних із використанням Google Gemini Chat Model

Нижче наведено приклад результатів аналізу, отриманих за допомогою цієї технології:

- WHO (хто постраждав?):
  - Користувач/сутність: arn:aws:iam::123456789012:user.devops-admin
  - Обліковий запис: devops-admin
- WHAT (що сталося?):
  - Було зафіксовано ініціацію входу до AWS Console з IP-адреси, яка не є типовою для цього IAM-користувача.
  - Вхід виконано успішно, без застосування багатофакторної автентифікації (MFA).
- WHEN (коли виявлено?):
  - 1970-01-01 00:00:00 (приклад часової мітки у тестовому середовищі).

- WHERE (де зафіксовано?):
  - Source IP: 11.22.33.44
  - AWS Account ID: 123456789012
  - Регіон: us-east-1
- WHY (чому це важливо?):
  - Категорія інциденту: Initial Access / Anomalous Authentication
  - Пояснення: IP-адреса 185.217.69.152 має рейтинг довіри 97% у базі AbuseIPDB та пов'язана з активністю brute-force і несанкціонованим використанням облікових даних. Подія стосується привілейованого IAM-користувача та відбулася без MFA, що суттєво підвищує ймовірність компрометації облікових даних.

Для реалізації адаптивного маршрутизування оновлений робочий процес (рис. 2) включає кілька послідовних логічних розгалужень, які забезпечують автоматизоване прийняття рішень відповідно до контексту події. Після того як AI-агент виконує розбір сповіщення та його збагачення — зокрема, отримує репутаційні дані з AbuseIPDB та інших джерел загрозових відомостей — результати передаються до легкого логічного модуля, відповідального за оцінку серйозності, достовірності та операційного контексту.

Додаткова JavaScript-функція здійснює остаточну перевірку та визначає, чи потребує подія ескалації. Якщо інцидент перевищує встановлені порогові значення — наприклад, пов'язаний із підтвердженою зловмисною IP-адресою або вказує на можливе компрометування облікових даних — система ініціює негайне сповіщення відповідальних фахівців 2 лінії або інших задіяних сторін згідно з процедурами. Події з низьким рівнем впевненості або мінімальною загрозою реєструються у фоні й переходять до відкладеної черги.

Перевага такого підходу полягає у здатності системи відокремлювати малозначущі сигнали від інцидентів, що потребують пріоритетного реагування. Завдяки цьому аналітики не витрачають ресурси на шумові сповіщення, тоді як критичні події автоматично отримують статус високої важливості.

У результаті система забезпечує дотримання встановлених SLA, підтримує стабільний операційний ритм і гарантує доставку релевантних сповіщень саме тим спеціалістам, які відповідають за подальше реагування, без надмірного інформаційного навантаження.

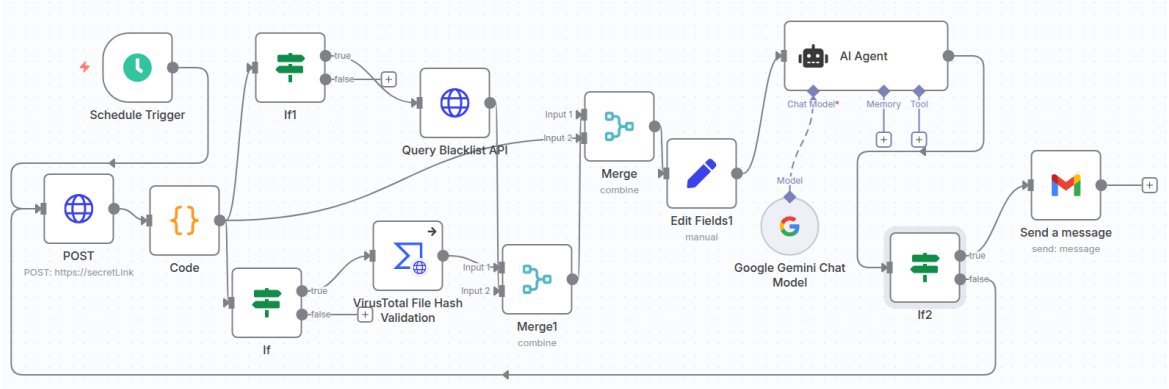

Рис. 2. Удосконалений робочий процес обробки сповіщень із LLM-збагаченням та логікою ескалації

На рисунку 2 подано оновлену версію робочого процесу, у якій реалізовано низку архітектурних змін, спрямованих на динамічне маршрутизування подій та автоматизовану ескалацію інцидентів. На відміну від традиційних лінійних схем, запропонований підхід передбачає умовну логіку, що реагує на результати попереднього аналізу подій, виконаного мовною моделлю.

Ключовим елементом є перевірка у вузлі If, який порівнює сформовані LLM характеристики події з попередньо визначеними порогами — рівнем серйозності, ступенем впевненості, типом виявленої загрози та іншими контекстними атрибутами. Завдяки цьому система одразу розмежовує другорядні сповіщення та інциденти, що потенційно становлять значну небезпеку, забезпечуючи пріоритезацію без участі аналітика.

Подальше уточнення виконує додатковий JavaScript-модуль. Він здійснює перевірки, які складно формалізувати одним правилом: звіряє події зі списками дозволених або заблокованих об’єктів, оцінює відповідність дій користувача його ролі, а також порівнює поточну активність із характерною поведінкою за попередній період. Отримані результати повертаються у основну логіку, формуючи додатковий рівень валідації перед ескалацією.

Після завершення перевірок процес розгалужується. Події, що мають високий рівень критичності або підтверджені індикатори компрометації, автоматично передаються до модуля сповіщення, який інформує відповідальних аналітиків чи контактні групи відповідно до чинних playbook-процедур. Натомість події з низьким рівнем впевненості реєструються для подальшого перегляду або обробляються у фоновому режимі.

Такий механізм забезпечує оптимальний баланс між швидкістю реагування та ефективністю використання людських ресурсів. Інциденти, що потребують негайної уваги, автоматично потрапляють до пріоритетного каналу, тоді як малозначущі сигнали не створюють додаткового навантаження на команду. У практичному вимірі це дозволяє стабільно виконувати вимоги SLA, скорочувати час реагування та запобігати перевантаженню аналітиків другого й третього рівнів рутинними або повторюваними сповіщеннями.

Інтеграція систем автоматизації з великими мовними моделями у діяльності Центру операцій безпеки (SOC) є практичним кроком до підвищення продуктивності та зниження навантаження на аналітиків. Поєднання n8n та хмарної LLM-моделі забезпечує можливість обробляти великий потік сповіщень значно швидше, ніж традиційні мануальні процеси аналітиків першої лінії.

Ключовою перевагою системи є різке скорочення часу інтерпретації інцидентів: замість кількох хвилин, необхідних аналітику для первинного огляду, контекстуалізації та формування короткого резюме, LLM виконує ці задачі за секунди. Це безпосередньо зменшує показники MTTA та MTTR, що є критичними у сучасних SOC із великим обсягом подій.

Автоматизована обробка не лише прискорює роботу, а й зменшує кількість рутинних операцій. Система самостійно аналізує сповіщення, додає зовнішній контекст і формує структурований опис із вмістом 5W [18]. Це дозволяє аналітикам зосередитися на складніших інцидентах — розслідуваннях, кореляції подій та стратегічних рішеннях. Важливою особливістю є стабільність результатів: на відміну від людського фактора, продуктивність LLM не залежить від навантаження або втоми, а логіка формування висновків залишається послідовною [19].

Додатковою перевагою є низька вартість упровадження. Використання відкритих інструментів та безоплатних API дозволяє уникнути значних фінансових витрат, характерних для комерційних SOAR-платформ, та зберігати гнучкість. Архітектура залишається масштабованою: при необхідності її можна розширити, додавши нові джерела даних, аналітичні модулі або інтерфейси взаємодії [20].

Система також демонструє високу відмовостійкість: контрольні точки, дедуплікація та механізми відновлення роботи забезпечують стабільну обробку подій за умов мережевих затримок чи технічних перебоїв. Гнучка модульна структура дозволяє змінювати або оновлювати окремі елементи без порушення цілісності конвеєра.

У підсумку запропонована архітектура забезпечує швидкість, стабільність і масштабованість без складності, притаманної традиційним SOAR-рішенням. Вона є ефективним інструментом як для великих SOC, так і для невеликих команд, які прагнуть підвищити рівень автоматизації, знизити когнітивне навантаження та забезпечити більш послідовний і якісний аналіз інцидентів.

Одним із ключових показників ефективності запропонованої системи є порівняння часу обробки інцидентів мануально та в автоматизованому режимі. У традиційній моделі SOC основні етапи аналізу виконуються аналітиками першої лінії мануально: перевірка технічних атрибутів події, пошук додаткових даних у зовнішніх сервісах, оцінювання ризику та формування короткого висновку. За результатами вимірювань, отриманих під час досліджень, проведених у компанії XXX, середній час повного аналізу одного інциденту становив 5–20 хвилин, що відповідає типовим галузевим орієнтирам. Фіксація тривалості виконання здійснювалась за допомогою системи Jira, де автоматично реєструвалися часові позначки для кожного етапу.

За умови опрацювання 10–27 подій на добу аналітик витрачав у середньому понад 1 годину 23 хвилини лише на первинне опрацювання інцидентів. Водночас мануальна обробка характеризується ризиком пропуску важливих атрибутів та хибної оцінки серйозності події, що підтверджується наявними дослідженнями [21].

Автоматизована система усуває більшість зазначених недоліків. Після активації тригера n8n самостійно отримує подію з Elastic Security або Paperless API, виконує її попередній аналіз, здійснює звернення до зовнішніх репутаційних сервісів (AbuseIPDB, VirusTotal), передає дані LLM-моделі Google Gemini, формуючи рекомендації та фінальний вердикт. За експериментальними спостереженнями середній час такого циклу становив 30–125 секунд. Журнали виконання n8n підтвердили стабільність цих значень незалежно від черговості подій та навантаження.

Людське втручання залишалося необхідним лише для складних або неоднозначних випадків (10–25 % сповіщень), що узгоджується з сучасними підходами до використання автоматизації в SOC. Аналіз змін аналітиків у відповідних case-повідомленнях показав, що переважна більшість автоматичних вердиктів не потребувала корекцій.

Узагальнені результати свідчать, що середній час мануальної обробки одного інциденту становив 12,8 хвилини, тоді як автоматизований процес виконує аналогічний набір операцій за 1,7 хвилини. Це забезпечує прискорення в 8–15 разів залежно від джерела подій та кількості репутаційних запитів. Порівняння наведено на рис. 3, а детальні значення — у таблиці 1.

ISSN 2663 - 4023

Таблиця. 1

Результати порівняння середніх показників часу аналізу аналітика та N8N

Джерело подій	Середній час 1 інциденту, хв (аналітик)	Середній час 1 інциденту, хв (N8N)	Час 25 інцидентів, хв (аналітик)	Час 25 інцидентів, хв (N8N)	Зміна — N8N швидше, разів
AWS CloudTrail	15	1,8	375	45	↓ у 8,3 рази
Google Workspace	10	1,3	250	32,5	↓ у 7,7 рази
Linux	12	1,6	300	40	↓ у 7,5 разів
Windows	13	1,7	325	42,5	↓ у 7,6 рази
Microsoft Defender	14	2,1	350	52,5	↓ у 6,7 рази

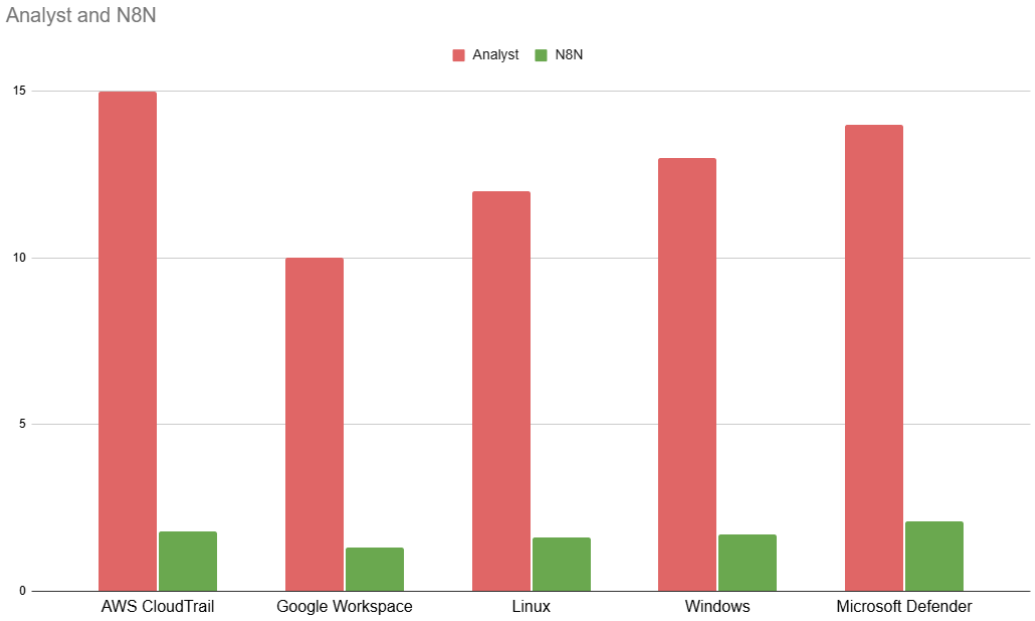

Рис. 3. Порівняння сумарного часу обробки спрацювання

Отримані дані демонструють, що автоматизація суттєво підвищує продуктивність SOC, зменшує навантаження на аналітиків та забезпечує стабільнішу якість класифікації. Додатково фіксується економічний ефект: зниження MTTA та MTTR підвищує пропускну здатність команди або дозволяє підтримувати існуючий рівень навантаження меншою кількістю працівників. Це відповідає сучасним тенденціям розвитку SOC, де мануальні операції поступово заміщаються інтелектуальними автоматизованими компонентами.

Оцінювання роботи системи проведено на основі 25 сповіщень, отриманих із Elastic Security. Для кожного спарацювання порівнювали два результати: експертний висновок аналітика та автоматично сформований вердикт системою. Оцінювання охоплювало визначення типу інциденту, адекватність оцінки рівня ризику, повноту відтворення контексту події та релевантність запропонованих дій.

Для кількісної оцінки застосовано метрики Precision, Recall та F1-міру. За результатами порівняння 22 з 25 вердиктів моделі збіглися з оцінками SOC-аналітика, що засвідчує стабільну роботу системи на реальних даних. Узагальнені результати наведено в таблиці 2 та на рисунку 4.

Таблиця 2

Середні показники якості сповіщень проаналізованих N8N

Критерій оцінки	Результат LLM	Пояснення
Точність класифікації типу інциденту	91 %	Коректно ідентифіковано тип події (мережева, файлова, користувацька тощо)
Відповідність рівня ризику (Severity)	87 %	У більшості випадків рівень серйозності збігався з експертною оцінкою
Повнота аналітичного опису (Context Coverage)	97 %	Модель ураховувала всі ключові атрибути: IP, час, джерело, причину
Релевантність рекомендацій (Action Quality)	86 %	Пропоновані дії відповідали політикам реагування SOC
Середній F1-score	0,89	Загальний показник узгодженості аналітичних висновків

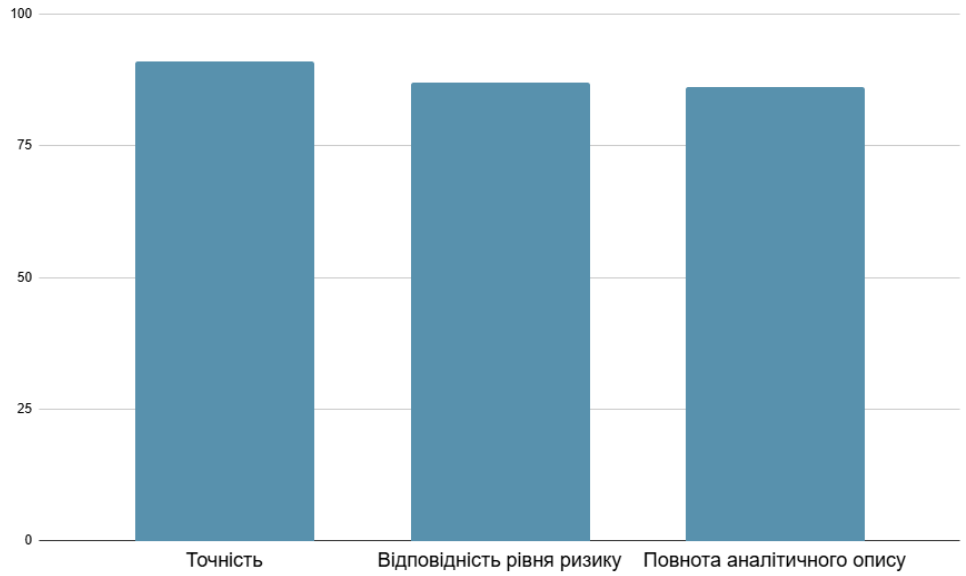

Рис. 4. Узагальнені результати обробки інцидентів автоматизованою системою

Основні розбіжності спостерігалися у випадках, коли події мали неповні або суперечливі атрибути (зокрема, нестачу даних про автентифікацію або неоднозначні IP-адреси). Для таких ситуацій застосовано механізм контролю впевненості: якщо модель формувала оцінку нижче 70%, інцидент автоматично передавався на ручний перегляд. Загалом отримані результати підтверджують здатність системи формувати коректні аналітичні висновки на основі даних Elastic Security та демонструють практичну придатність рішення як інструмента автоматизації первинного аналізу у SOC.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У дослідженні проведено комплексний аналіз можливостей автоматизації первинної обробки інцидентів у Центрах операцій безпеки (SOC) шляхом інтеграції оркестраційної платформи n8n, SIEM-системи Elastic Security та великих мовних моделей (LLM). На основі запропонованої архітектури створено прототип інтелектуального агента, який автоматично отримує події через Elastic API, виконує їх нормалізацію, збагачення, інтерпретацію у форматі 5W та формує початковий аналітичний висновок для подальшого реагування.

Отримані результати підтвердили, що поєднання n8n та LLM дає змогу суттєво прискорити аналітичні процеси, зменшує когнітивне навантаження на аналітиків першого рівня, підвищує повторюваність і якість рішень та сприяє зростанню оперативності реагування. Розроблена система характеризується прозорістю, адаптивністю та відповідністю сучасним стандартам інформаційної безпеки, а також придатністю для впровадження у середовищах з обмеженими ресурсами без необхідності використання повноцінних SOAR-платформ.

Разом з тим виявлено низку обмежень, зокрема залежність якості аналітичних висновків від повноти та узгодженості SIEM-даних, чутливість до конструкції промптів та відсутність механізмів автоматичного об’єднання множинних подій у складні інциденти. Це вказує на потребу подальшої оптимізації попередньої обробки даних, вдосконалення методів контролю достовірності результатів та розширення інтеграції з операційними процесами SOC.

Перспективними напрямами розвитку системи є:

- розроблення механізмів автоматичного формування інцидентів на основі кореляції пов’язаних подій;
- інтеграція LLM-агентів із сценаріями напівавтоматичного реагування (ізоляція хостів, оновлення IAM-політик, завершення сесій);
- застосування ризик-орієнтованих моделей, що враховують критичність активів та бізнес-контекст;
- підвищення стійкості LLM до шумових та неповних даних шляхом удосконалення нормалізації;

Отримані результати підтверджують практичну доцільність інтеграції інтелектуальних агентів у процеси первинної обробки інцидентів та демонструють потенціал таких систем до підвищення оперативної зрілості SOC.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. n8n.io. (n.d.). GitHub - n8n-io/n8n: Fair-code workflow automation platform with native AI capabilities. GitHub. Retrieved from https://github.com/n8n-io/n8n
2. Reddy, A. R. P. (2025). Zero Trust Architecture: An AI-driven framework for modern cybersecurity challenges. FMDB Transactions on Sustainable Intelligent Networks, 2(1), 10–21. https://doi.org/10.69888/ftsin.2025.000366
3. Elastic. (n.d.). Elastic Security overview. Retrieved from https://www.elastic.co/guide/en/security/current/index.html
4. IBM. (n.d.). What is SOAR (security orchestration, automation and response)? Retrieved from https://www.ibm.com/think/topics/security-orchestration-automation-response
5. Марценюк, Я., Партика, А. (2024). The concept of automated compliance verification as the foundation of a fundamental cloud security model. Computer Systems and Network, 6(1), 108–123. https://doi.org/10.23939/csn2024.01.108

6. Cloud Security Alliance. (2025). Agentic AI Threat Modeling Framework: MAESTRO. Retrieved from https://cloudsecurityalliance.org/blog/2025/02/06/agentic-ai-threat-modeling-framework-maestro
7. Хома, В., Абібуллаєв, А., Піскозуб, А., & Крет, Т. (2024). Comprehensive approach for developing an enterprise cloud infrastructure. In CEUR Workshop Proceedings (Vol. 3654, pp. 201–215). Retrieved from https://ceur-ws.org/Vol-3654/paper17.pdf
8. Вахула, О., Курій, Ю., Опірський, І., & Віталій, С. (2024). Security-as-code concept for fulfilling ISO/IEC 27001:2022 requirements. In CEUR Workshop Proceedings (Vol. 3654, pp. 59–72). Retrieved from https://ceur-ws.org/Vol-3654/paper6.pdf
9. International Organization for Standardization. (2022). ISO/IEC 27001:2022. Retrieved from https://www.iso.org/standard/27001
10. Дейнека, О., Гарасимчук, О., Партика, А., Общта, А., & Коршун, Н. (2024). Designing data classification and secure store policy according to SOC 2 Type II. In CEUR Workshop Proceedings (Vol. 3654). Retrieved from https://ceur-ws.org/Vol-3654/paper7.pdf
11. Волотовський, О., Банах, Р., Піскозуб, А., & Бржевська, З. (2024). Automated security assessment of Amazon Web Services accounts using CIS benchmark and Python 3. In CEUR Workshop Proceedings (Vol. 3826). Retrieved from https://ceur-ws.org/Vol-3826/short29.pdf
12. Cloud Security Alliance. (n.d.). Security Guidance for Critical Areas of Focus in Cloud Computing. Retrieved from https://cloudsecurityalliance.org/artifacts/security-guidance-v4
13. Банах, Р., Піскозуб, А., & Стефінко, Ю. (2016). External elements of honeypot for wireless network. In Modern Problems of Radio Engineering, Telecommunications and Computer Science (pp. 480–482). https://doi.org/10.1109/TCSET.2016.7452093
14. Siam, A., Alazab, M., Awajan, A., Hasan, M. R., Obeidat, A., & Faruqui, N. (2025). IP Safeguard — An AI-driven malicious IP detection framework. IEEE Access, 1–13. https://doi.org/10.1109/ACCESS.2025.3569289
15. Тихолаз, Д., Банах, Р., Мичуда, Л., Піскозуб, А., & Киричок, Р. (2024). Incident response with AWS detective controls. In CEUR Workshop Proceedings (Vol. 3826, pp. 190–197). Retrieved from https://ceur-ws.org/Vol-3826/short6.pdf
16. Center for Internet Security. (n.d.). CIS Amazon Web Services Benchmarks. Retrieved from https://www.cisecurity.org/benchmark/amazon_web_services
17. Стефінко, Ю., Піскозуб, А., & Банах, Р. (2016). Manual and automated penetration testing: Benefits and drawbacks; modern tendency. In Modern Problems of Radio Engineering, Telecommunications and Computer Science (pp. 488–491). https://doi.org/10.1109/TCSET.2016.7452095
18. Microsoft. (n.d.). Microsoft Sentinel Documentation. Retrieved from https://learn.microsoft.com/en-us/azure/sentinel/
19. Dai, R., Lv, P., Gui, Y., Lv, Q., Qiao, Y., Wang, Y., Sun, D., Huang, W., Li, Y., & Wang, X. (2025). An automated attack investigation approach leveraging threat-knowledge-augmented large language models. arXiv. https://arxiv.org/abs/2509.01271
20. OWASP Foundation. (n.d.). OWASP Top 10 Security Risks. Retrieved from https://owasp.org/www-project-top-ten/
21. TNO. (2017). Human Factors in Cyber Incident Response: Needs, collaboration and The Reporter. Retrieved from https://publications.tno.nl/publication/34626339/5qnMkv/TNO-2017-R11575.pdf

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 3 (31), 2025
ISSN 2663 - 4023

Oleksandr V. Volotovskyi
Master student, Department of Information Technology Security
Lviv Polytechnic National University, Lviv, Ukraine
ORCID: 0009-0003-3102-3694
oleksandr.volotovskyi.mkbbi.2024@lpnu.ua

Roman I. Banakh
Doctor of Philosophy, Associate Professor of the Department of Information Technology Security
Lviv Polytechnic National University, Lviv, Ukraine
ORCID: 0000-0001-6897-8206
roman.i.banakh@lpnu.ua

Andrian Z. Piskozub
Doctor of Philosophy, Associate Professor of the Department of Information Technology Security
Lviv Polytechnic National University, Lviv, Ukraine
ORCID: 0000-0001-6897-8206
andriian.z.piskozub@lpnu.ua

DEVELOPMENT OF AN AGENT FOR NOTIFICATION ANALYSIS BASED ON ARTIFICIAL INTELLIGENCE USING N8N AND ELASTIC SECURITY SOFTWARE

Abstract. This article presents a research-driven approach to enhancing the efficiency of incident response within Security Operations Centers by implementing an automated event analysis system that integrates the n8n platform and large language models. This approach addresses modern cybersecurity challenges, including the continuous increase in the number of security alerts, the growing complexity of IT infrastructures, limited human resources, and the declining effectiveness of manual analysis. The developed architecture incorporates the Elastic Security SIEM system as the primary event source, n8n as the orchestration platform for workflow management, LLM-based agents for natural-language interpretation of incidents, and external enrichment services such as AbuseIPDB and VirusTotal. The core system functions include automated collection, normalization, and enrichment of security events, risk scoring, incident type classification, generation of concise analytical summaries in the 5W format (Who, What, When, Where, Why), and automatic creation of relevant response actions. To validate the effectiveness of the proposed approach, a comprehensive experimental study was conducted, encompassing several stages: simulation of real SOC scenarios, generation of representative security incidents, execution of the automated analysis workflow, and comparison of results with the performance of Tier-1 SOC analysts. Particular attention was paid to assessing the stability of the LLM agent under varying event formats, different log sources, and inconsistent data quality. Furthermore, the system's behavior was examined in scenarios involving ambiguous or partially missing incident attributes, as well as its resilience to noisy or incomplete data. The analysis demonstrated that the integrated approach not only automates routine tasks but also ensures high consistency of decisions, minimizes subjective variability inherent to manual analysis, and supports the production of unified and repeatable incident verdicts. Additional evaluation focused on the quality of the generated analytical summaries, the accuracy of contextual interpretation, the model's ability to identify key entities and causal factors, and the relevance of recommended response actions. The study also verified the system's compatibility with existing SOC infrastructure and its ability to scale under increased alert volumes without significant performance degradation. The obtained results confirm that the proposed system not only enhances the speed and quality of incident response but also lays the foundation for transitioning from a reactive to a proactive SOC model, where automated tools take over a substantial portion of the cognitive workload. Due to the modularity of n8n and the flexibility of LLM agents, the system can adapt to new threat types, expand functionality without significant cost, and integrate with various corporate services. Thus, it provides the technological basis for further adoption of intelligent automation in cybersecurity practice and supports the development of next-generation SOC environments.

Keywords: SOC automation; n8n; large language models; Elastic Security; incident orchestration; cybersecurity; SIEM; LLM agent.

REFERENCES

1. n8n.io. (n.d.). GitHub - n8n-io/n8n: Fair-code workflow automation platform with native AI capabilities. GitHub. Retrieved from https://github.com/n8n-io/n8n
2. Reddy, A. R. P. (2025). Zero Trust Architecture: An AI-driven framework for modern cybersecurity challenges. FMDB Transactions on Sustainable Intelligent Networks, 2(1), 10–21. https://doi.org/10.69888/ftsin.2025.000366
3. Elastic. (n.d.). Elastic Security overview. Retrieved from https://www.elastic.co/guide/en/security/current/index.html
4. IBM. (n.d.). What is SOAR (security orchestration, automation and response)? Retrieved from https://www.ibm.com/think/topics/security-orchestration-automation-response
5. Matseniuk, Y., & Partyka, A. (2024). The concept of automated compliance verification as the foundation of a fundamental cloud security model. Computer Systems and Network, 6(1), 108–123. https://doi.org/10.23939/csn2024.01.108
6. Cloud Security Alliance. (2025). Agentic AI Threat Modeling Framework: MAESTRO. Retrieved from https://cloudsecurityalliance.org/blog/2025/02/06/agentic-ai-threat-modeling-framework-maestro
7. Khoma, V., Abibulaiev, A., Piskozub, A., & Kret, T. (2024). Comprehensive approach for developing an enterprise cloud infrastructure. In CEUR Workshop Proceedings (Vol. 3654, pp. 201–215). Retrieved from https://ceur-ws.org/Vol-3654/paper17.pdf
8. Vakhula, O., Kurii, Y., Opirskyi, I., & Vitalii, S. (2024). Security-as-code concept for fulfilling ISO/IEC 27001:2022 requirements. In CEUR Workshop Proceedings (Vol. 3654, pp. 59–72). Retrieved from https://ceur-ws.org/Vol-3654/paper6.pdf
9. International Organization for Standardization. (2022). ISO/IEC 27001:2022. Retrieved from https://www.iso.org/standard/27001
10. Deineka, O., Harasymchuk, O., Partyka, A., Obshta, A., & Korshun, N. (2024). Designing data classification and secure store policy according to SOC 2 Type II. In 11CEUR Workshop Proceedings (Vol. 3654). Retrieved from https://ceur-ws.org/Vol-3654/paper7.pdf
11. Volotovskyi, O., Banakh, R., Piskozub, A., & Brzhevska, Z. (2024). Automated security assessment of Amazon Web Services accounts using CIS benchmark and Python 3. In CEUR Workshop Proceedings (Vol. 3826). Retrieved from https://ceur-ws.org/Vol-3826/short29.pdf
12. Cloud Security Alliance. (n.d.). Security Guidance for Critical Areas of Focus in Cloud Computing. Retrieved from https://cloudsecurityalliance.org/artifacts/security-guidance-v4
13. Banakh, R., Piskozub, A., & Stefinko, Y. (2016). External elements of honeypot for wireless network. In Modern Problems of Radio Engineering, Telecommunications and Computer Science (pp. 480–482). https://doi.org/10.1109/TCSET.2016.7452093
14. Siam, A., Alazab, M., Awajan, A., Hasan, M. R., Obeidat, A., & Faruqui, N. (2025). IP Safeguard — An AI-driven malicious IP detection framework. IEEE Access, 1–13. https://doi.org/10.1109/ACCESS.2025.3569289
15. Tykholaz, D., Banakh, R., Mychuda, L., Piskozub, A., & Kyrychok, R. (2024). Incident response with AWS detective controls. In CEUR Workshop Proceedings (Vol. 3826, pp. 190–197).
16. Center for Internet Security. (n.d.). CIS Amazon Web Services Benchmarks. Retrieved from https://www.cisecurity.org/benchmark/amazon_web_services
17. Stefinko, Y., Piskozub, A., & Banakh, R. (2016). Manual and automated penetration testing: Benefits and drawbacks; modern tendency. In Modern Problems of Radio Engineering, Telecommunications and Computer Science (pp. 488–491). https://doi.org/10.1109/TCSET.2016.7452095
18. Microsoft. (n.d.). Microsoft Sentinel Documentation. Retrieved from https://learn.microsoft.com/en-us/azure/sentinel/
19. Dai, R., Lv, P., Gui, Y., Lv, Q., Qiao, Y., Wang, Y., Sun, D., Huang, W., Li, Y., & Wang, X. (2025). An automated attack investigation approach leveraging threat-knowledge-augmented large language models. arXiv. https://arxiv.org/abs/2509.01271
20. OWASP Foundation. (n.d.). OWASP Top 10 Security Risks. Retrieved from https://owasp.org/www-project-top-ten/
21. TNO. (2017). Human Factors in Cyber Incident Response: Needs, collaboration and The Reporter. Retrieved from https://publications.tno.nl/publication/34626339/5qnMkv/TNO-2017-R11575.pdf