МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ
КАФЕДРА УПРАВЛІННЯ КІБЕРБЕЗПЕКОЮ ТА ЗАХИСТОМ ІНФОРМАЦІЇ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: “ПОБУДОВА СИСТЕМИ МОНІТОРИНГУ ТА РЕАГУВАННЯ НА ІНЦИДЕНТИ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ В SOC”

на здобуття освітнього ступеня магістра
зі спеціальності 125 Кібербезпека та захист інформації
освітньо-професійної програми Управління інформаційною та кібернетичною безпекою

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Надія СВЯТСЬКА

(підпис) *Ім’я, ПРІЗВИЩЕ здобувача*

Виконав: Здобувачка вищої освіти гр. УБДМ-61
Надія СВЯТСЬКА

Керівник: д.т.н., проф.
Віталій САВЧЕНКО

Рецензент:

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут кібербезпеки та захисту інформації

Кафедра Управління кібербезпекою та захистом інформації
Ступінь вищої освіти магістр
Спеціальність 125 Кібербезпека та захист інформації
Освітньо-професійна програма Управління інформаційною та кібернетичною безпекою

ЗАТВЕРДЖУЮ
Завідувач кафедрою УКБЗІ
_____ Світлана ЛЕГОМІНОВА
"____" _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Студентка Святській Надії Андріївні
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: "Побудова системи моніторингу та реагування на інциденти інформаційної безпеки в SOC"

керівник кваліфікаційної роботи Віталій САВЧЕНКО, д.т.н., проф.
(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від "30" жовтня 2025 р. № 467.

2. Строк подання кваліфікаційної роботи "11" грудня 2025 р.
3. Вихідні дані до кваліфікаційної роботи:.
4. Перелік питань, які потрібно розробити:
   1.. Теоретичні основи моніторингу та реагування в SOC
   2.. Проєктування та побудова системи моніторингу SOC
   3.. Технічне налаштування системи моніторингу та реагування

5. Перелік ілюстративного матеріалу: *презентація*
6. Дата видачі завдання "02" жовтня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назви етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Визначення об’єкту, предмету, мети та завдань дослідження.	10.10.2025	Виконано
2.	Збір та аналіз літератури.	23.10.2025	Виконано
3.	Аналіз основ моніторингу та реагування в SOC.	27.10.2025	Виконано
4.	Дослідження та побудова системи моніторингу SOC.	10.11.2025	Виконано
5.	Визначення напрямів та методів налаштування системи моніторингу та реагування.	15.11.2025	Виконано
6.	Формулювання висновків за результатами дослідження.	22.11.2025	Виконано
7.	Оформлення роботи.	04.12.2025	Виконано
8.	Оформлення презентації.	14.12.2025	Виконано
9.	Отримання рецензії на роботу.	18.12.2025	Виконано
10.	Захист в ЕК.	20.01.2026	Виконано

Здобувачка вищої освіти _____ Надія СВЯТСЬКА
(підпис) *(Ім’я, ПРІЗВИЩЕ)*

Керівник
кваліфікаційної роботи _____ Віталій САВЧЕНКО
(підпис) *(Ім’я, ПРІЗВИЩЕ)*

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ

ПОДАННЯ
ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ
ЩОДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ
на здобуття освітнього ступеня магістра

Направляється здобувачка Святська Н.А.. до захисту кваліфікаційної роботи
(прізвище та ініціали)
за спеціальністю 125 Кібербезпека та захист інформації
(код, найменування спеціальності)
Освітньо-професійної програми Управління інформаційною та кібернетичною безпекою
(назва)
на тему: "Побудова системи моніторингу та реагування на інциденти інформаційної безпеки в SOC"
Кваліфікаційна робота і рецензія додаються.

Директор ННІКБЗІ _____ Євгенія ІВАНЧЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Висновок керівника кваліфікаційної роботи

Здобувачка **СВЯТСЬКА Надія** у кваліфікаційній роботі проаналізувала основи моніторингу та реагування в SOC, вивчила методи та засоби побудови системи моніторингу SOC, а також дослідила практичне застосування та налаштування системи моніторингу та реагування.

СВЯТСЬКА Надія показала високу теоретичну і практичну підготовку, володіння науково-дослідницькими методами, вміння самостійно знаходити шляхи вирішення проблеми дослідження. Результати дослідження апробовані на конференції "Стратегії кіберстійкості: управління ризиками та безперервність бізнесу" 27 лютого 2025 року.

Все це дозволяє оцінити кваліфікаційну роботу здобувачки **СВЯТСЬКОЇ Надії** на оцінку "_____" та присвоїти їй кваліфікацію "Магістр з кібербезпеки та захисту інформації за освітньо-професійною програмою Управління інформаційною та кібернетичною безпекою".

Керівник кваліфікаційної роботи _____ Віталій САВЧЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

"____"_____2025 року

Висновок кафедри про кваліфікаційну роботу

Кваліфікаційна робота розглянута. Здобувачка Святська Н.А. допускається до захисту даної роботи в Екзаменаційній комісії.

Завідувач кафедрою
Управління кібербезпекою та захистом
інформації _____ Світлана ЛЕГОМІНОВА
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

ВІДГУК РЕЦЕНЗЕНТА
на кваліфікаційну магістерську роботу

здобувачки вищої освіти Святської Надії Андріївни
на тему "Побудова системи моніторингу та реагування на інциденти інформаційної безпеки в SOC"

Актуальність В умовах зростання кількості кібератак і ускладнення інформаційних систем питання ефективного моніторингу та реагування на інциденти інформаційної безпеки є надзвичайно актуальним. Побудова та впровадження SOC як централізованого елемента захисту дозволяє організаціям своєчасно виявляти загрози та мінімізувати їхній вплив. Тому обрана тема кваліфікаційної роботи є своєчасною та має практичне значення.

Позитивні сторони

1. У кваліфікаційній роботі розглянуто теоретичні засади функціонування SOC, розкрито роль SIEM, XDR та SOAR у процесах моніторингу й реагування на інциденти. Матеріал викладено логічно та послідовно.

2. Значною перевагою є практична реалізація системи моніторингу на базі Microsoft Sentinel з описом налаштування джерел даних, аналітичних правил, візуалізації та автоматизованого реагування. Це свідчить про прикладну спрямованість роботи та розуміння реальних процесів SOC.

3. Кваліфікаційна робота оформлена відповідно до встановлених вимог. Матеріал викладено послідовно та структуровано, висновки є логічними й обґрунтованими. Ключові аспекти дослідження проілюстровано схемами, таблицями та прикладами. Автор опрацювала значну кількість наукових і технічних джерел, у тому числі англомовних, що свідчить про достатній рівень теоретичної підготовки та вміння працювати з фаховою літературою.

Недоліки

1. До певних недоліків слід віднести те, що у роботі можна було б більш детально розглянути порівняльний аналіз альтернативних SIEM/XDR-платформ або ширше висвітлити обмеження та ризики автоматизованого реагування в SOC, зокрема з погляду помилкових спрацювань і впливу на бізнес-процеси.

Однак, вищезгадані зауваження не впливають на загальну позитивну оцінку кваліфікаційної роботи.

Висновок: Кваліфікаційна робота виконана на належному науково-методичному рівні і заслуговує позитивної оцінки, а здобувач Святська Надія Андріївна заслуговує присвоєння кваліфікації "Магістр кібербезпеки за освітньо-професійною програмою "Управління інформаційною та кібернетичною безпекою".

Рецензент:

_____
підпис *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 92 стор., 25 рис., 20 табл., 40 джерел.

Метою роботи є дослідження та розробка підходів до побудови системи моніторингу й реагування на інциденти інформаційної безпеки в межах Security Operations Center (SOC).

Об'єктом дослідження є система моніторингу та реагування на інциденти інформаційної безпеки підприємства.

Предмет дослідження – архітектура, методи та засоби функціонування SOC з використанням технологій SIEM, XDR та SOAR.

Методи дослідження. Для досягнення поставленої мети використано методи системного аналізу, моделювання архітектур інформаційних систем, аналізу та кореляції подій безпеки, а також методи управління інцидентами інформаційної безпеки. У практичній частині застосовано аналітичні методи роботи з телеметрією, мову запитів KQL, а також підходи до автоматизації реагування на інциденти. Для оцінювання ефективності роботи SOC використано метрики MTTD та MTTR.

Короткий зміст роботи. У роботі сформовано теоретичні засади функціонування SOC як централізованої організаційно-технічної системи моніторингу та реагування на інциденти інформаційної безпеки. Проаналізовано роль і взаємодію компонентів SIEM, XDR та SOAR у процесі виявлення, аналізу та усунення загроз. Розроблено архітектуру системи моніторингу SOC у вигляді багатошарового конвеєра обробки подій. Практично реалізовано систему моніторингу та реагування на базі Microsoft Sentinel, включно з підключенням джерел телеметрії, створенням аналітичних правил виявлення, візуалізацією подій та автоматизованим реагуванням. Проведено тестування роботи системи на прикладі сценарію атаки типу password guessing та виконано оптимізацію параметрів виявлення для зниження хибних спрацювань.

Галузь застосування. Отримані результати можуть бути використані під час проєктування, впровадження та вдосконалення SOC на підприємствах, а також для побудови систем моніторингу й реагування на інциденти інформаційної безпеки з

використанням сучасних SIEM- та SOAR-платформ.

КЛЮЧОВІ СЛОВА : SECURITY OPERATIONS CENTER, SOC, SIEM, XDR, SOAR, MICROSOFT SENTINEL, МОНІТОРИНГ БЕЗПЕКИ, РЕАГУВАННЯ НА ІНЦИДЕНТИ, ІНФОРМАЦІЙНА БЕЗПЕКА.

ABSTRACT

The text part of the qualification work for obtaining a master's degree: 92 pages, 25 figures, 20 tables, 40 sources.

The purpose of the work is to research and develop approaches to building a system for monitoring and responding to information security incidents within the Security Operations Center (SOC).

Object of research is the system for monitoring and responding to information security incidents at the enterprise.

Subjec*t *of research is the architecture, methods, and means of SOC operation using SIEM, XDR, and SOAR technologies.

Research methods To achieve the set goal, methods of system analysis, modeling of information system architectures, analysis and correlation of security events, as well as methods of information security incident management were used. In the practical part, analytical methods of working with telemetry, the KQL query language, and approaches to automating incident response were used. MTTD and MTTR metrics were used to evaluate the effectiveness of the SOC.

Brief content of research. The work forms the theoretical basis for the functioning of the SOC as a centralized organizational and technical system for monitoring and responding to information security incidents. The role and interaction of SIEM, XDR, and SOAR components in the process of detecting, analyzing, and eliminating threats are analyzed. The architecture of the SOC monitoring system is developed in the form of a multi-layered event processing pipeline. A monitoring and response system based on Microsoft Sentinel is implemented in practice, including the connection of telemetry sources, the creation of analytical detection rules, event visualization, and automated response. The system was tested using a password guessing attack scenario, and detection parameters were optimized to reduce false positives.

Field of research. The results obtained can be used in the design, implementation, and improvement of SOCs at enterprises, as well as for building

systems for monitoring and responding to information security incidents using modern SIEM and SOAR platforms.

KEYWORDS: SECURITY OPERATIONS CENTER, SOC, SIEM, XDR, SOAR, MICROSOFT SENTINEL, SECURITY MONITORING, INCIDENT RESPONSE, INFORMATION SECURITY.

ЗМІСТ

ВСТУП

Сучасні інформаційні системи підприємств характеризуються зростанням складності, широким використанням хмарних сервісів і збільшенням обсягів подій безпеки, що потребують постійного аналізу. В умовах ускладнення кібератак та скорочення часу між проникненням і завданням шкоди особливої актуальності набуває побудова централізованих систем моніторингу та реагування на інциденти інформаційної безпеки. Ефективним підходом до вирішення цього завдання є впровадження Security Operations Center (SOC), який забезпечує безперервний збір телеметрії, аналітичну обробку подій та координоване реагування на інциденти.

Мета роботи полягає у дослідженні підходів до побудови системи моніторингу та реагування на інциденти інформаційної безпеки в межах Security Operations Center та розробці практичної моделі її реалізації з використанням сучасних SIEM-, XDR- та SOAR-технологій.

Для досягнення цієї мети в роботі необхідно виконати наступні **завдання**:

1. Дослідити теоретичні основи функціонування SOC та процесів моніторингу і реагування на інциденти інформаційної безпеки.

2. Розробити архітектуру системи моніторингу SOC з урахуванням інтеграції SIEM та XDR.

3. Реалізувати практичну модель моніторингу та реагування на інциденти з використанням Microsoft Sentinel, включно з аналітичними правилами та автоматизованим реагуванням

Об'єкт дослідження -. процеси моніторингу та реагування на інциденти інформаційної безпеки в організаціях.

Предмет дослідження –. архітектура, методи та засоби функціонування системи SOC на основі технологій SIEM, XDR та SOAR

Методи дослідження. У роботі використано методи системного аналізу, аналізу та кореляції подій безпеки, моделювання архітектур інформаційних систем, а також практичні методи налаштування SIEM і автоматизованого

реагування. Для реалізації аналітичної обробки подій застосовано мову запитів KQL.

Наукова новизна одержаних результатів полягає у формуванні узагальненої архітектури SOC у вигляді багатошарового конвеєра обробки подій та у поєднанні аналітичних механізмів SIEM із автоматизованим реагуванням SOAR у межах єдиної операційної моделі.

Практичне значення одержаних результатів полягає в можливості використання запропонованої архітектури, методики налаштування та тестування SOC для впровадження або вдосконалення систем моніторингу й реагування на інциденти інформаційної безпеки на підприємствах.

Апробація результатів кваліфікаційної роботи відбулася на конференції “Стратегії кіберстійкості: управління ризиками та безперервність бізнесу” 28 лютого 2024 року.

РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ МОНІТОРИНГУ ТА РЕАГУВАННЯ В SOC

1.1. Аналіз концепцій Security Operation Center (SOC)

Центр безпеки (SOC) є нервовим центром стратегії кібербезпеки організації. Це централізована організаційна одиниця, яка об'єднує людей, процеси та технології для постійного моніторингу, виявлення, аналізу та реагування на кіберзагрози. Працюючи безперервно (або відповідно до потреб організації), SOC забезпечує захист у режимі реального часу та підтримує довгострокове поліпшення стану безпеки організації.

Сучасні кіберзагрози, включаючи програми-вимагачі, атаки на ланцюги постачання та складні постійні загрози, вимагають розвитку SOC. Сучасні SOC все більше покладаються на автоматизацію, штучний інтелект (AI) та інформацію про загрози, щоб випереджати зловмисників, прискорювати виявлення та скорочувати час реагування.

Місія SOC значно ширша, ніж просто виявлення загроз. Аналітики та інженери з безпеки постійно відстежують активність на серверах, у мережах, базах даних, додатках, кінцевих точках, веб-сайтах та інших критично важливих системах. Така діяльність дозволяє їм якнайшвидше виявляти загрози безпеці та оперативно реагувати на них, визначати вразливості й потенційні точки компрометації, а також збирати та аналізувати інформацію про загрози, пов’язані як із відомими, так і з новими ризиками. Окрім цього, SOC оцінює та оптимізує ефективність існуючих інструментів і засобів контролю безпеки. По суті, центр операцій безпеки переслідує подвійну мету: забезпечувати виявлення інцидентів та реагування на них у режимі реального часу і водночас постійно вдосконалювати загальний стан кібербезпеки організації [1].

Окрім реагування на інциденти, SOC також розслідує першопричини, повідомляє про вразливості та планує запобіжні заходи, щоб подібні інциденти не повторювалися. SOC працюють у різних секторах, таких як фінанси, уряд,

енергетика та промисловість, і зазвичай використовують багаторівневу ієрархічну структуру, в якій аналітики та інженери отримують ролі відповідно до свого досвіду та знань.

У межах концепції SOC важливим є його визначення як постійно діючої організаційної функції, що забезпечує централізований огляд стану безпеки підприємства та формує єдину точку контролю над усіма джерелами телеметрії. Такий підхід дозволяє створити узгоджену систему спостереження, у якій різнорідні дані перетворюються на структуровану інформацію про стан середовища. SOC у цьому контексті виступає елементом операційної інфраструктури, що забезпечує спроможність підприємства своєчасно оцінювати власний рівень захищеності та виявляти фактори, що можуть негативно вплинути на його безперервність.

Ключовою концепцією SOC є поєднання трьох базових складових: людей, процесів та технологій.

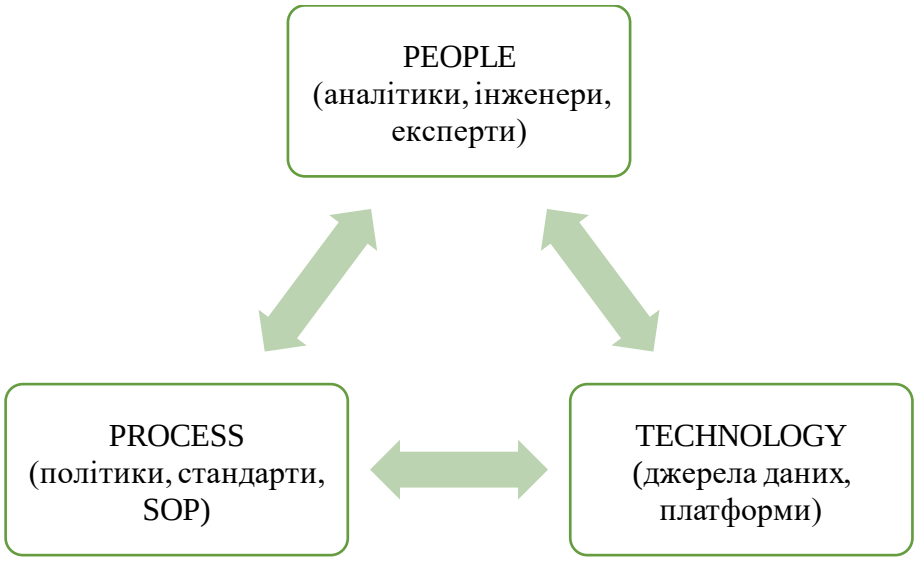

Рис. 1.1. SOC як інтегрована система

Така триєдина структура визначає логіку його функціонування. Технологічна складова забезпечує збирання великих обсягів подій із різних частин інфраструктури та їх подальшу обробку [2]. Людський фактор формує

експертну складову: саме спеціалісти здійснюють інтерпретацію даних, приймають рішення та встановлюють правила, за якими SOC функціонує. Процесуальний компонент забезпечує повторюваність, узгодженість і стандартизацію операцій. Усі ці елементи не існують окремо: концепція SOC передбачає їх взаємозалежність і координацію, що формує його як цілісну систему.

У загальній структурі SOC передбачається наявність організаційних рівнів, але не як елемент внутрішніх процесів реагування, а як характеристика підходу до розподілу завдань і відповідальності всередині центру [3]. Завдяки такому структурному поділу забезпечується ефективна взаємодія між різними категоріями фахівців – як технічними, що відповідають за налаштування й підтримку інструментів моніторингу, так і аналітичними, що зосереджуються на інтерпретації інформації та оцінці впливу на безпеку. Структурне розмежування дає змогу адаптувати SOC до масштабів організації, рівня її цифрової інфраструктури та кількості джерел подій, що потребують контролю.

Суттєвою концепцією SOC є визначення його як постійно діючого центру з повною відповідальністю за спостереження за інфраструктурою. На відміну від інших підрозділів, SOC не виконує функції за потреби – його існування передбачає неперервність операцій. Такий режим є необхідним через те, що події безпеки виникають у будь-який момент, а тому організація повинна зберігати здатність підтримувати високий рівень обізнаності щодо власного середовища незалежно від часу доби чи рівня навантаження. У цьому полягає концептуальна відмінність SOC від інших елементів системи управління безпекою.

Структура SOC окреслюється як поєднання взаємопов’язаних функцій, кожна з яких забезпечує певний аспект операційної спроможності. Одним із ключових елементів є технічна інфраструктура, що включає джерела логів, мережеві компоненти, серверні системи та інші елементи, з яких надходить інформація для подальшого аналізу [4]. Ця інфраструктура формує основу для спостереження, оскільки без доступних, структурованих і достовірних даних центр операцій безпеки не може виконувати свої функції.

Організаційна структура також передбачає окреме інженерне середовище, відповідальне за функціонування механізмів збору, нормалізації та передачі подій. Інженерні функції включають підтримку апаратних і програмних компонентів, налаштування – інструментів, оптимізацію параметрів збору даних та контроль за стабільністю їх надходження. Наявність спеціалізованих технічних фахівців гарантує, що SOC отримує повний і безперервний потік інформації про стан інфраструктури [5].

Аналітична складова структури SOC відповідає за інтерпретацію отриманої інформації. Її основою є робота з даними, що відображають активність у ключових системах підприємства. Концепція SOC передбачає, що аналітичні функції виконуються у стандартизованому середовищі, де результати аналізу представляються у вигляді структурованих висновків, доступних для операційних та управлінських рішень. Аналітична частина також забезпечує узгодженість між технічними даними та стратегічними цілями безпеки організації.

Завдяки такому структуруванню SOC отримує здатність об’єднувати технічні та організаційні компоненти у єдиний центр відповідальності за стан безпеки, що є фундаментальною концепцією його існування.

У рамках загальної концепції центру операцій безпеки важливим є поділ SOC на типи залежно від способу їх організації та розміщення у структурі підприємства. Одним із базових різновидів є внутрішній SOC, що повністю розміщується в межах організації. Такий центр характеризується прямим контролем над усіма його елементами, доступністю персоналу та можливістю гнучкої адаптації до специфічних потреб підприємства. Внутрішній SOC зазвичай створюється у великих організаціях із критичною інфраструктурою та значним обсягом даних.

Інший різновид – віртуальний SOC – передбачає використання віддалених фахівців або часткову децентралізацію операцій. Такий підхід дозволяє створити центр без необхідності фізичної присутності команди в одному місці [6]. Він застосовується у випадках, коли підприємство не потребує цілодобового

контролю або прагне оптимізувати витрати на персонал і фізичну інфраструктуру.

Глобальний SOC використовується у структурах із розгалуженою географією. Такий центр координує діяльність регіональних підрозділів, забезпечуючи єдине бачення стану безпеки у масштабах усієї організації. Він дозволяє уникнути дублювання функцій, гармонізувати політики та забезпечити узгодженість дій у глобальному масштабі [7].

Окреме місце займають хмарні та гібридні SOC. Вони використовують інфраструктуру, частково або повністю розміщену у хмарних середовищах. Такі центри можуть об'єднувати локальні системи спостереження з хмарними потоками даних, що є особливо актуальним для підприємств, які використовують багатохмарні архітектури.

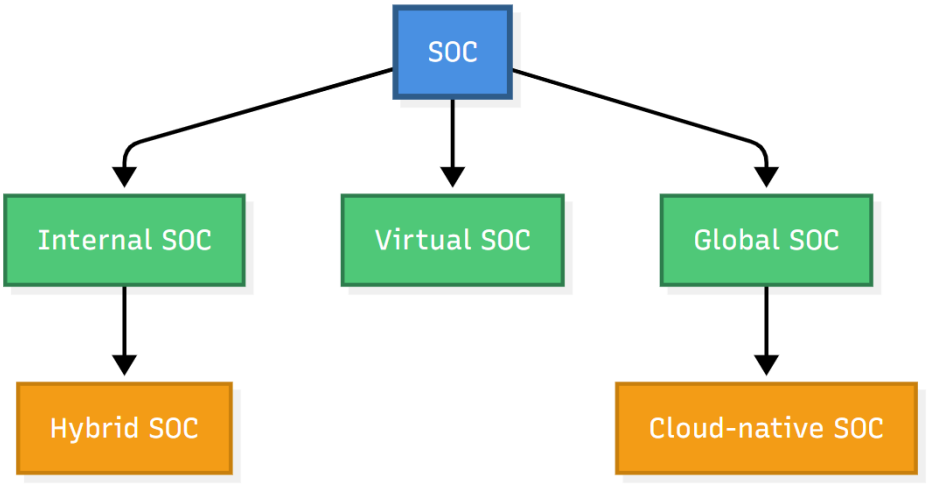

Рис. 1.2. Дерево класифікації SOC за типом розміщення

У межах концепцій SOC визначаються також класифікаційні групи за способом реалізації. SOC може функціонувати на основі програмних рішень, апаратних комплексів або використовувати існуючу інфраструктуру підприємства. Також існує класифікація за моделлю володіння: внутрішній чи повністю аутсорсинговий центр. Кожен із цих варіантів визначає рівень контролю, обсяг відповідальності та глибину інтеграції в операційні процеси організації.

1.2. Реагування на загрози, інциденти та цикл управління інцидентами

У межах функціонування SOC фундаментальне значення має чітке розуміння природи загроз, характеристик інцидентів і пов'язаних із ними механізмів реагування. SOC розглядає загрозу як потенційну дію, подію або процес, здатний порушити конфіденційність, цілісність чи доступність інформаційних ресурсів, спричинити небажану зміну їхнього стану або створити умови для майбутньої компрометації. Загроза у цьому контексті не обмежується самою атакою, а включає намір, можливість її реалізації, присутність вразливості та потенційну шкоду для організації. У відповідних матеріалах підкреслюється, що загрози формуються різними чинниками: діями зловмисників, помилками користувачів, технічними відмовами та природними подіями, що впливають на функціонування інформаційних систем. Ключовим у визначенні загроз є їхній зв'язок із контекстом діяльності підприємства, оскільки одна й та сама подія може мати різний рівень небезпеки залежно від операційної специфіки та критичності активів [8].

Типи загроз у середовищі SOC охоплюють широкий спектр явищ, включаючи зовнішні та внутрішні джерела небезпеки. До зовнішніх загроз належать спроби несанкціонованого доступу з мережі, шкідливі дії з боку організованих груп, автоматизовані масові атаки, а також структуровані кампанії цілеспрямованого впливу. Внутрішні загрози виникають унаслідок некоректних дій співробітників, помилкового поводження з даними або навмисних деструктивних дій інсайдерів. Саме існування таких умов, як неправильні конфігурації, відкриті порти, слабкі засоби контролю доступу або недостатній моніторинг журналів, створює для SOC необхідність постійного аналізу ризиків, пов'язаних із можливими сценаріями розвитку загроз.

Для SOC важливо враховувти, що загрози не існують у статичному вигляді. Вони еволюціонують разом зі зміною технологічного середовища, появою нових технік проникнення та розвитком кіберзлочинних інструментів. Це визначає необхідність їхньої класифікації за характеристиками поведінки, рівнем

складності, методами проникнення та впливом на інфраструктуру. Саме так формується поняття інциденту, яке SOC використовує для опису фактично реалізованої загрози або події, що має ознаки порушення безпеки. Інцидент визначається як підтверджена або ймовірна невідповідність нормальному стану системи, яка призводить або може призвести до порушення її функціонування. Інцидент включає всі випадки, коли подія безпеки набуває ознак негативного впливу, а отже потребує реагування, фіксації та подальшого аналізу [9].

У практиці SOC інцидент не обмежується конкретною атакою. Він охоплює широкий спектр станів, починаючи від підозрілої активності та завершуючи повномасштабною компрометацією системи. Критерії визначення інциденту формуються на основі ознак, що свідчать про відхилення від нормативної роботи системи. Такі ознаки можуть мати вербальний характер, зокрема повідомлення про підозрілі операції, несанкціоновані дії користувачів або відомості про відмови, які не мають очевидних технічних причин. Вербальні індикатори використовуються персоналом SOC як додатковий контекст для класифікації події, особливо коли технічні джерела ще не надали достатню кількість даних. Таким чином інцидент-менеджмент у SOC поєднує технічні та операційні джерела інформації, що створює цілісне уявлення про характер події.

Класифікація інцидентів є ключовим елементом реагування. Існує декілька видів ознак, які дозволяють SOC визначати тип інциденту. До таких ознак належать джерело походження інциденту, вплив на системи, характер відхилень у роботі мережевих компонентів і масштаби потенційної шкоди. Інциденти можуть бути пов’язані зі спробами проникнення, використанням вразливостей, некоректним використанням привілеїв, шкідливим програмним забезпеченням або аномальною поведінкою систем. Характеристика кожного типу формується на підставі технічних параметрів, таких як зміна конфігурації системних служб, виконання нетипових команд, відхилення у мережевій поведінці, зміна доступності сервісів та інші фактори, що документуються засобами спостереження.

Особливу увагу приділяють розмежуванню понять атаки, вторгнення та порушення. Атака є наміром або спробою завдати шкоди, вторгнення означає отримання несанкціонованого доступу, тоді як порушення передбачає зміну стану системи внаслідок деструктивних дій. Для SOC таке розмежування має принципове значення, оскільки різні стани інцидентів потребують різної глибини реагування, різного горизонтального залучення фахівців і різної терміновості опрацювання. Чітка термінологічна база дозволяє формалізувати процеси реагування та створює умови для субординації дій у межах різних рівнів SOC [10].

В основі реагування на інцидент лежить концепція циклічного управління, яка діє як поєднання аналітичних і технічних процедур, що здійснюються у визначеній послідовності. Цей цикл починається з ідентифікації події. На цьому етапі SOC отримує початкові дані з моніторингових систем, журналів чи оперативних повідомлень, після чого визначає, чи має подія ознаки можливої загрози. Наступна фаза полягає в аналізі, що включає уточнення першопричин, встановлення характеру активності та перевірку наявності відомих ознак компрометації. Такий аналіз реалізується у межах операційних процедур, забезпечуючи об’єктивність та узгодженість рішень.

Після аналізу SOC переходить до етапу реагування, який включає контроль за поширенням інциденту, обмеження деструктивних дій та відновлення нормального функціонування систем. Також можна запровадити принцип ізоляції у разі підозри на шкідливі дії, коли окремі компоненти можуть бути відключені від мережі для запобігання розвитку атаки. На основі цмх дії можна визначити напрямки для подальшого удосконалення процедур SOC, оскільки ця діяльність дозволяє фіксувати закономірності в роботі систем та адаптувати правила виявлення до нових загроз.

Рис. 1.3 Процес роботи SOC над інцидентом

У межах повного реагування важливу роль відіграє розподіл завдань між рівнями SOC, що здійснюється через організацію роботи рівнів L1, L2 і L3. На рівні L1 здійснюється первинна оцінка подій [11]. Цей рівень відповідає за фіксацію основних параметрів інциденту, ідентифікацію його видимих ознак і прийняття рішення щодо подальшого опрацювання. L1 є точкою входу інцидентів у SOC і забезпечує швидку реакцію на різноманітні типи подій. Рівень L2 зосереджується на поглибленому аналізі. L2 уточнює технічні деталі інциденту, співставляє події з іншими джерелами й встановлює можливий контекст дії зловмисника. Цей рівень оцінює вплив інциденту на інфраструктуру та визначає необхідність залучення додаткових процедур реагування. Рівень L3 відповідає за розбір найскладніших інцидентів. L3 проводить всебічне технічне дослідження, формує точну реконструкцію подій, визначає ступінь проникнення та забезпечує підготовку технічних рекомендацій щодо запобігання аналогічним ситуаціям у майбутньому.

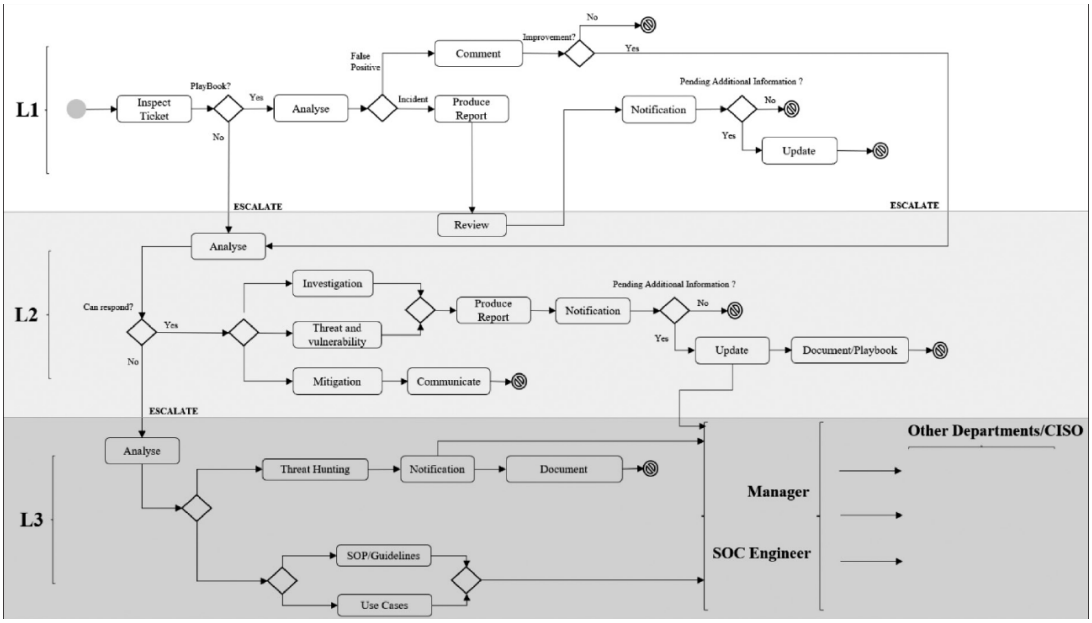

Рис. 1.4 Робочий процес реагування на інциденти SOC

Індикатори компрометації визначаються як інформаційні маркери, що дають змогу SOC підтвердити факт вторгнення або встановити технічні характеристики дії зловмисника. Такими маркерами можуть бути сліди виконання команд, сторонні процеси, характерні зміни конфігурацій, невідомі мережеві з'єднання, підозрілі файли або ознаки використання облікових записів у нетиповий спосіб. Індикатори, які можна використовувати як IoC, і всі вони спрямовані на фіксацію відхилень у поведінці системи від нормального стану, можуть бути статичними, тобто пов'язаними з конкретними значеннями, такими як IP-адреси, хеші файлів або доменні імена, або поведінковими, що визначають зміну звичної роботи системи [12]. Для SOC IoC мають ключову роль, оскільки дозволяють швидко класифікувати інцидент, оцінити його серйозність і підтвердити наявність шкідливої активності.

Вербальні індикатори інцидентів відображають повідомлення, що можуть надходити від персоналу або користувачів, які помітили підозрілі дії, не пов'язані з їхніми звичайними функціями. Такі повідомлення мають важливе значення у тих випадках, коли технічні засоби ще не опрацювали достатнього обсягу даних відповідно до прописаних в системі правил. SOC використовує їх як частину первинного контексту, зіставляючи з технічними ознаками [13].

Цикл управління інцидентами потребує ретельної взаємодії всіх елементів SOC, зокрема процедур фіксації даних, аналізу поведінкових аномалій та використання практик кореляції подій. Логи вважаються чи не найважливішим джерелом інформації. Журнали роботи систем, мережеві лог-файли, записи виконання команд, дані автентифікації та інші матеріали слугують підґрунтям для аналізу інцидентів. Кореляція логів дозволяє SOC реконструювати ланцюги подій, узгоджувати час і характер дій та визначати, чи був інцидент частиною ширшої атаки або компроментації.

1.3. Дослідження технологічної бази SOC: SIEM, XDR, SOAR

Технологічна база центру операцій безпеки складається з взаємопов'язаних платформ, що забезпечують повний цикл моніторингу, виявлення та реагування на інциденти. Вона інтегрує засоби обробки великих обсягів журналів подій, системи поведінкового аналізу та інструменти автоматизованої оркестрації. Центральне місце в цьому середовищі займає архітектура SIEM, яка формує єдину точку консолідації інформації, необхідної для подальшої аналітичної обробки. На її основі створюється розширений контекст за допомогою XDR, що забезпечує збалансоване поєднання даних з різних технологічних доменів та застосування моделей виявлення аномалій. Завершальним елементом є SOAR, який формалізує реакції на інциденти, переводячи їх із площини ручного виконання у сферу автоматизованих і відтворюваних процесів. [14]

У межах роботи SOC SIEM виконує функції центру обробки подій. Архітектурно він побудований навколо механізмів збору та аналітичної інтерпретації журналів подій (логи), що надходять від різних компонентів інфраструктури. Агрегація журналів є важливим етапом, оскільки телеметрія формується у різних форматах, з різною глибиною деталізації та в різні часові моменти. Уніфікація даних через нормалізацію дає змогу перетворювати ці різнорідні записи на узгоджену модель подій. Після цього система може

застосовувати правила кореляції, які визначають причинно-наслідкові або часові зв’язки між подіями, що походять з різних джерел. Такі зв’язки дозволяють SOC переходити від аналізу одиничних журналів до оцінки складних багатоступеневих сценаріїв шкідливої активності.

У сучасних архітектурах SIEM також виступає платформою, що підвищує рівень ситуаційної обізнаності SOC за рахунок довготривалого зберігання даних та можливості виконувати ретроспективний аналіз. Журнали подій формують багаторівневу історичну базу інцидентів, у якій можна простежити розвиток окремих інцидентів і знайти приховані послідовності дій, що не були помітними під час первинного моніторингу. SIEM використовується для оперативного аналізу подій, реконструкції минулих інцидентів, визначення вразливих ділянок інфраструктури та оцінювання ефективності засобів захисту [15].

Суттєвою характеристикою SIEM є здатність обробляти великі обсяги подій у реальному часі. Це необхідно, оскільки сучасні інфраструктури генерують величезну кількість логів із серверів, мережевих пристроїв, сервісів автентифікації, застосунків і систем контролю доступу. Аналіз таких потоків вручну є складним і довготривалим процесом, тому SIEM використовує високопродуктивні механізми індексування та фільтрації, які забезпечують швидкий доступ до подій, що відповідають певним критеріям. Ці механізми дозволяють SOC виявляти відхилення у поведінці системи в режимі наближеному до реального часу, зменшувати кількість хибнопозитивних сигналів та концентрувати увагу на подіях, які потребують поглибленого розслідування [16].

У технологічній системі SOC SIEM виконує функцію базового шару, на якому ґрунтується робота систем розширеного виявлення XDR. XDR об’єднує дані з різних доменів безпеки – кінцевих точок, мережевих сегментів, систем ідентифікації, хмарних середовищ та поштової інфраструктури. Це дозволяє створювати багатовимірний погляд на інцидент, у якому різні фрагменти активності користувачів і систем формують єдиний контекст. На відміну від SIEM, який значною мірою покладається на правила кореляції та визначені

шаблони подій, XDR застосовує поведінкові моделі, що аналізують характер властивостей і взаємозв’язків між діями системи.

Поведінковий підхід у XDR реалізується через моделювання нормальної активності користувачів і пристроїв. Коли виявляється відхилення від цих моделей, формується сигнал про потенційну загрозу. Таке виявлення аномалій дає змогу SOC виявляти інциденти, які традиційні сигнатурні системи пропустили б через відсутність відповідних правил або ознак. Важлива особливість XDR полягає у здатності аналізувати послідовності дій, а не окремі події, що створює умови для розпізнавання складних багатокрокових атак. Це дозволяє ідентифікувати ранні ознаки проникнення, коли шкідлива активність ще не набула руйнівного характеру [17].

XDR також забезпечує автоматичне формування контексту інциденту. Дані, що надходять з кінцевих точок, мережі та інших систем, інтегруються в єдине аналітичне уявлення. SOC отримує змогу переглядати події не у вигляді окремих логів, а як логічно пов'язаний набір фактів, що описують розвиток взаємодії атаки з інфраструктурою. Це особливо важливо для SOC з великим обсягом подій, оскільки дає можливість скоротити час ручного аналізу, зменшити навантаження на аналітиків і підвищити точність визначення пріоритетів.

Поглиблена аналітика XDR формує передумови для автоматизації реагування, яку забезпечує SOAR. У технологічному середовищі SOC SOAR виступає системою, що узгоджує дії різних платформ, створює стандартизовані процедури реакції та автоматично виконує визначені операції, коли подія відповідає певним критеріям. Основою SOAR є механізм плейбуків – формалізованих сценаріїв реагування, у яких визначено послідовність дій, необхідних для усунення інциденту або обмеження його впливу [18].

Плейбуки дозволяють SOC відтворювати реакції однаково якісно незалежно від часу доби, навантаження на аналітиків або складності інциденту. Вони забезпечують систематичну перевірку індикаторів, виконання дій щодо блокування шкідливих об’єктів, ізоляцію сегментів мережі, взаємодію з

сервісами управління доступом та іншими компонентами інфраструктури. Виконання плейбуків зменшує кількість рутинних завдань, що покладаються на аналітиків L1, і дає змогу їм зосередитись на інцидентах, які потребують людського аналізу.

SOAR також забезпечує можливість інтеграції з різними засобами захисту, включно з EDR, IAM, мережевими фільтрами та хмарними сервісами. Це дає SOC змогу реагувати на інциденти на всіх рівнях інфраструктури [19].

З погляду операційної ефективності SOAR формує рівень стандартизації, який забезпечує відтворюваність дій SOC. Кожен етап реагування може бути перевірений, документований і повторно використаний у подальших сценаріях. Це дозволяє зменшити ризики неправильного реагування, усунути людські помилки та оптимізувати взаємодію між різними підрозділами, що залучаються до усунення інцидентів [20].

Таблиця 1.1.

Функціональні характеристики технологічної бази SOC

№	Компонент	Основна роль	Тип обробки даних	Ключові механізми	Результат роботи
1	SIEM	Консолідація журналів та подій	Нормалізація, кореляція	Інтерпретація логів, кореляційні правила	Виявлення подій, історичний аналіз
2	XDR	Розширене виявлення загроз	Об’єднання телеметрії з різних доменів	Поведінкові моделі, UEBA	Контекстуалізація інциденту, виявлення аномалій
3	SOAR	Автоматизація реагування	Виконання сценаріїв	Плейбуки, оркестрація, інтеграція систем	Автоматизовані дії та усунення загроз

SIEM, XDR і SOAR формують єдину технологічну екосистему SOC, у якій дані та аналітика рухаються через кілька рівнів опрацювання. Журнали подій та телеметрія спочатку збираються і систематизуються, потім аналізуються з використанням поведінкових моделей, а результати цього аналізу трансформуються в автоматизовані реакції. Таке поєднання створює замкнений контур реагування, в якому SOC отримує змогу працювати з великими даними,

розпізнавати складні загрози та усувати їх у мінімальний проміжок часу. Ефективність SOC залежить від здатності цих технологій взаємодіяти, забезпечуючи високу точність аналізу та оперативність реагування.

Висновки до розділу 1

У першому розділі було сформовано теоретичну основу побудови та функціонування Security Operations Center як ключового елемента системи управління інформаційною безпекою організації. Проведений аналіз концепцій SOC дозволяє визначити його не лише як сукупність технічних засобів, а як постійно діючу організаційну функцію, що поєднує людей, процеси та технології в єдину керовану систему. Саме така інтеграція забезпечує централізований контроль за станом безпеки, безперервне спостереження за інфраструктурою та здатність своєчасно реагувати на загрози.

Розгляд моделей організації SOC показав, що вибір внутрішнього, віртуального, глобального, хмарного або гібридного підходу визначається масштабом підприємства, критичністю активів та рівнем зрілості процесів безпеки. Незалежно від моделі розміщення, фундаментальним залишається принцип безперервності операцій та чіткого розподілу відповідальності між рівнями SOC, що створює передумови для ефективного реагування на інциденти.

У межах дослідження реагування на загрози та інциденти було обґрунтовано важливість чіткого термінологічного розмежування понять загрози, події та інциденту. Запропонований цикл управління інцидентами демонструє, що реагування в SOC має циклічний характер і поєднує ідентифікацію, аналіз, стримування, усунення наслідків і подальше вдосконалення процедур. Значну роль у цьому процесі відіграють індикатори компрометації, кореляція подій та аналіз логів, які формують технічну основу для прийняття обґрунтованих рішень.

Дослідження технологічної бази SOC показало, що ефективне функціонування центру неможливе без поєднання SIEM, XDR та SOAR. SIEM

забезпечує централізований збір і аналіз подій, XDR формує розширений контекст і дозволяє виявляти складні поведінкові аномалії, а SOAR переводить реагування в площину автоматизованих і стандартизованих процесів. Сукупно ці технології формують замкнений контур моніторингу та реагування, який є основою сучасного SOC і створює підґрунтя для подальшого практичного проєктування системи моніторингу та реагування на інциденти інформаційної безпеки

РОЗДІЛ 2 ПРОЄКТУВАННЯ ТА ПОБУДОВА СИСТЕМИ МОНІТОРИНГУ SOC

2.1 Архітектура системи моніторингу та вимоги до інфраструктури

Проєктування SOC доцільно починати не з вибору окремих інструментів, а з визначення цільової операційної моделі, яка задає стратегічний напрямок розвитку, очікувані результати та потрібний рівень зрілості, причому сама «зрілість» не має бути самоціллю – її рівень виводиться з амбіцій і цілей SOC. Це важливо для архітектури, тому що саме амбіції (наприклад, 24/7 моніторинг, швидкість реагування, глибина аналітики, вимоги комплаєнсу) прямо впливають на обсяги телеметрії, моделі зберігання журналів, потрібну пропускну здатність каналів та обчислювальні ресурси аналітичного рівня. У підході SOC-CMM пропонується фіксувати поточний і цільовий стани, виконувати розрив-аналіз і далі реалізовувати зміни через керований беклог та безперервне вдосконалення з регулярними оцінюваннями «виміряти й адаптувати» [21].

У загальному вигляді SOC є центральним елементом архітектури кібербезпеки організації, що забезпечує моніторинг подій і попереджень у реальному часі та дозволяє швидко реагувати на загрози. Модель побудови SOC може бути внутрішньою, спільно керованою або повністю керованою постачальником, і вибір моделі залежить від масштабу організації, складності IT-ландшафту та рівня ризику; водночас у всіх моделях SOC відповідає за виявлення та реагування на загрози й підтримку конфіденційності, цілісності та доступності даних.

Архітектуру слід описувати так, щоб вона була незалежною від кадрової/організаційної моделі: навіть якщо частина функцій виконується зовнішнім провайдером, телеметрія, логування, нормалізація, кореляція та керування інцидентами мають бути спроєктовані як цілісний конвеєр, де кожен етап має визначені входи, виходи, SLA та контроль доступу.

Архітектуру системи моніторингу SOC доречно фіксувати у трьох взаємопов’язаних поданнях: логічному, функціональному та мережному.

Логічна архітектура описує «що саме» є у системі моніторингу та як рухаються дані: джерела телеметрії, транспорт і колектори, рівень зберігання журналів, аналітичний рівень (кореляція, правило виявлення, пошук), рівень керування інцидентами, а також рівень автоматизації реакцій. Функціональна архітектура відповідає на питання «які функції» виконує SOC у конвеєрі: збір, очищення/нормалізація, агрегація, збагачення контекстом, настроювані правила виявлення, розслідування, ескалація та реагування. Мережева архітектура конкретизує «де і як» фізично/логічно проходить трафік телеметрії: сегментація, точки збору, шлюзи/проксі, канали до хмарних сервісів, ізоляція керувальних площин та вимоги до пропускної здатності й затримок [22].

У технічному сенсі SOC-архітектура майже завжди зводиться до багатошарового конвеєра подій, де критичною є модульність і можливість нарощування, оскільки підключення нових джерел і нових сценаріїв виявлення неминуче змінює навантаження. Дослідження про ефективну архітектуру SIEM підкреслює, що незалежно від вибору реалізації ключові параметри (моніторинг у реальному часі, аналітика, керування журналами, звітність, оновлення, профілювання поведінки тощо) мають бути враховані як вимоги на рівні системи, а сама архітектура має бути модульною, щоб дозволяти додавати компоненти в майбутньому відповідно до потреб.

У практиці це означає, що логічна архітектура SOC має бути незалежною від конкретного «двигуна» кореляції чи конкретного агента збору: зміни інструментів не повинні руйнувати конвеєр даних, схему нормалізації та підхід до керування життєвим циклом інцидентів [23].

Нижче на Рисунку 2.1 наведено узагальнену схему (як «каркас» логічної архітектури), де показано основні вузли та напрямки потоків телеметрії; конкретні продукти (SIEM, XDR, SOAR) підставляються у відповідні рівні без зміни загальної логіки.

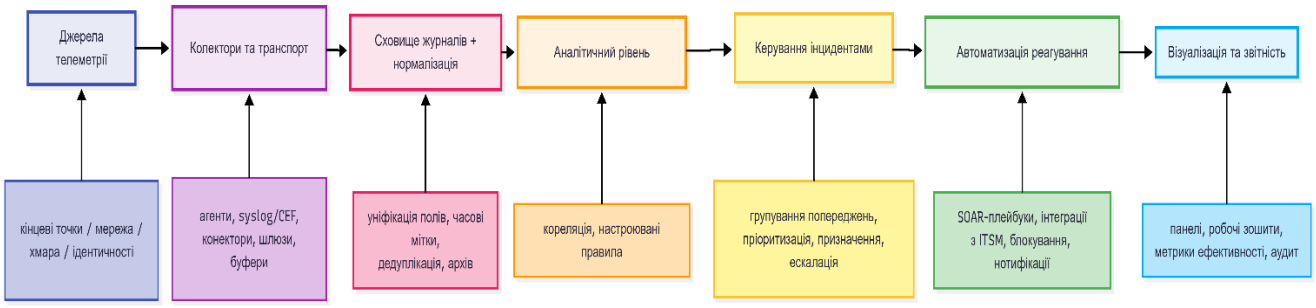

Рис. 2.1 Узагальнена схема архітектури SOC

З погляду вимог до інфраструктури найчастіше обмежувальним фактором стає не окремий сервер, а баланс між швидкістю прийому подій, можливістю їх оперативно індексувати/запитувати, та продуктивністю кореляції й правил виявлення. Окреме порівняльне дослідження рушіїв кореляції показує, що кореляційні інструменти різняться за пропускною здатністю та масштабованістю, а також за споживанням пам'яті; зокрема спостерігалось, що Java-орієнтовані рішення можуть вимагати більше пам'яті, а окремі рушії демонструють суттєві проблеми масштабування у складних сценаріях, тоді як інші показують вищу пропускну здатність у найвимогливіших тестах [24].

Для архітектури SOC це транслюється у вимогу проєктувати аналітичний рівень так, щоб він витримував пікові навантаження не лише за кількістю подій, а й за складністю правил. Крім того, що сам по собі рушій кореляції не утворює повноцінний SIEM: для завершеної системи потрібні вхідні/вихідні компоненти, модулі зберігання та механізми динамічного налаштування чи API для інтеграції, тобто інфраструктурні вимоги виникають у всьому ланцюжку, а не в одному модулі.

Практична вимога до збору телеметрії в межах SOC полягає у забезпеченні повноти спостережуваності для ключових площин: кінцевих точок, мережі, хмари та ідентичностей. Кінцеві точки мають генерувати події ОС і засобів захисту, мережа – журнали міжмережевих екранів, проксі, IDS/IPS, VPN, балансувальників, а хмарні платформи – журнали адміністративних дій, події безпеки, сигнали з хмарних сервісів і контролів доступу. Облікові записи критичні для виявлення атак на автентифікацію та несанкціонованих доступів,

тому журнали служб керування користувачами мають бути першокласним джерелом, так само як і журнали доступу до критичних застосунків. В гібридному середовищі потрібно мати достатню обчислювальну потужність, сховище та пропускну здатність мережі для обробки трафіку й даних, які генерує організація, а набір інструментів зазвичай охоплює SIEM та суміжні класи на кшталт XDR і SOAR [25].

Питання логування в SOC виходить за межі «увімкнути журнали» і на практиці складається з трьох технічних завдань: стандартизації форматів, нормалізації полів та управління строками зберігання. Стандартизація означає, що різнорідні джерела (наприклад, журнали Windows, syslog з Linux, журнали мережевих пристроїв) мають бути приведені до узгодженого формату транспорту та парсингу, щоб подальша аналітика не залежала від «особливостей» конкретного вендора. Нормалізація означає уніфікацію ключових атрибутів подій (хто, що, де, коли, з якого вузла/адреси, над чим виконано дію), забезпечення коректних часових міток і можливість збагачення контекстом. Управління строками зберігання (retention policies) – це компроміс між комплаєнсом/форензикою та вартістю зберігання й продуктивністю запитів: занадто коротке зберігання позбавляє можливості побачити «повільні» атаки, а надмірно довге без поділу на «гаряче» і «холодне» сховище деградує швидкодію пошуку і збільшує витрати. У сучасних SOC ці політики зазвичай реалізуються на рівні центрального сховища журналів і дублюються механізмами архівації для довгострокового зберігання, коли оперативні запити виконуються по «гарячому» шарові, а старі дані виносяться до архіву з рідшим доступом [26].

Як приклад практичної реалізації SIEM-рівня у хмарній моделі можна розглянути Microsoft Sentinel, де центральним елементом даних виступає Log Analytics Workspace, у який надходять журнали з під’єднаних джерел через конектори. Типова послідовність розгортання передбачає створення робочого простору, увімкнення Sentinel на цьому просторі, під’єднання джерел подій (зокрема журнали Windows, журнали хмарних сервісів, мережеві пристрої через типові конектори), а далі використання вбудованих або власних правил

виявлення, панелей візуалізації та автоматизації реакцій через плейбуки на базі Azure Logic Apps.

Архітектурно це означає, що вимоги до інфраструктури «переїжджають» у площину правильної організації потоків даних та керування доступом до робочого простору, а також у проєктування схеми даних і політик зберігання всередині платформи.

З точки зору функцій SOC важливо, що в Sentinel аналітика та пошук базуються на запитах мовою KQL, які застосовуються як для пошуку загроз, так і для правил виявлення, що запускаються за розкладом або в потоці [27].

Автоматизація реагування у SOC є продовженням аналітики: коли правило виявлення породжує попередження, система керування інцидентами групує пов'язані попередження в інцидент, після чого можуть запускатися плейбуки реагування. Плейбуки можуть виконувати дії на кшталт блокування IP-адреси на рівні Azure NSG та надсилання сповіщень електронною поштою, а також інтегруватися з ITSM-системами (наприклад, створення інциденту в ServiceNow) і прив'язуватися до інцидентів через правила автоматизації в Sentinel.

Це важливо тим, що інфраструктура SOC включає збір і аналітику та виконання керованих дій у середовищі: отже, мережна архітектура та модель доступу повинні дозволяти безпечний виклик керувальних API (для блокування, ізоляції, створення заявок), а також передбачати аудит цих дій і збереження слідів автоматизації (наприклад, у Log Analytics або у окремому сховищі) для відстеження й контролю.

Проєктуючи мережну архітектуру SOC у гібридному середовищі, слід розділяти площину збору телеметрії та площину керування. Потоки журналів мають бути односторонніми за логікою доступу (від джерела до колектора/платформи), з мінімально потрібними відкритими портами, а керувальні інтеграції (плейбуки, конектори до ITSM, дії з політиками мережі) мають працювати через окремо контрольовані облікові записи та з принципом найменших привілеїв.

Важливим елементом є також «придатність даних до виявлення», тобто можливість будувати правила, які опираються на часові вікна, частоти та зв’язки між подіями. Типовий сценарій виявлення грубого підбору паролів у SIEM-системах формалізується як умова перевищення заданого порогового значення кількості невдалих спроб автентифікації протягом визначеного часового інтервалу («понад N невдалих входів за T секунд») із подальшим формуванням попередження та ініціюванням керованих дій реагування, зокрема блокування джерельної IP-адреси [28]. У межах такого правила саме параметри частоти подій і часових рамок визначають обчислювальне навантаження на аналітичний рівень системи моніторингу, а також зумовлюють критичність точності та узгодженості часових міток у журналах подій.

Аналогічно, у багатокрокових атаках використовується поняття частота/часове вікно та прив’язка до MITRE ATT&CK через ідентифікатор техніки, що вимагає, аби платформа зберігала й обробляла дані так, щоб кореляція між подіями з різних джерел була технічно можливою (спільні поля, узгоджені часові мітки, стабільні ідентифікатори сутностей) [29].

З огляду на це, Таблиця 2.1 нижче узагальнює технічні вимоги до інфраструктури для кожного рівня SOC-архітектури у термінах «яку функцію виконує рівень» та «які ризики виникають без відповідної інфраструктури».

Таблиця 2.1

Технічні вимоги до інфраструктури для кожного рівня SOC-архітектури

Рівень SOC-архітектури	**Технічний зміст рівня**	**Ключові інфраструктурні вимоги та ризики**
Джерела телеметрії	Події ОС, застосунків, EDR/XDR-сигнали, мережеві журнали, журнали хмари та ідентичностей	Повнота логування і стабільні часові мітки; без цього правила виявлення стають «сліпими» або дають хибні спрацювання.
Збір і транспорт	Агенти, syslog/CEF, конектори, шлюзи, буферизація	Пропускна здатність і стійкість до піків; без буферів можливі втрати подій і «дірки» у розслідуванні.
Сховище та нормалізація	Індексування, уніфікація полів, дедуплікація, архів/retention	Баланс швидкого пошуку й довгого зберігання; без політик зберігання деградує продуктивність і зростає вартість.
Рівень SOC-архітектури	Технічний зміст рівня	Ключові інфраструктурні вимоги та ризики
Керування інцидентами	Групування попереджень, пріоритизація, призначення, ескалація	Надійність і аудитованість процесу; без цього втрачається контроль над життєвим циклом інцидентів.
Автоматизація реагування	SOAR-плейбуки, інтеграції, керовані дії (блокування/заявки)	Безпечні облікові записи, контроль доступу до API, журналювання дій; приклади включають блокування IP і створення заявок в ITSM

Вимоги до збору телеметрії та організації логування безпосередньо корелюють із моделями тактик і технік, описаними в MITRE ATT&CK. У межах цієї моделі тактики відображають тактичні цілі противника, тобто мотивацію виконання певних дій, тоді як техніки характеризують способи досягнення цих цілей. Такий підхід створює методологічну основу для формування правил виявлення, оцінювання повноти покриття сценаріїв атак, а також ідентифікації прогалин у спостережуваності та захисних механізмах [30].

Окреме значення має використання MITRE ATT&CK для оцінювання прогалин захисту та рівня зрілості SOC. У контексті архітектури системи моніторингу це означає, що вимоги до збору телеметрії мають визначатися не

наявними джерелами подій, а цільовими тактиками і техніками, які SOC повинен виявляти та аналізувати. Відповідно, склад і структура телеметричних даних, а також методи їх аналітичної обробки мають проєктуватися з урахуванням конкретних сценаріїв виявлення, а не формуватися як несистематизований набір журналів.

2.2 Інтеграція SIEM, XDR та інших компонентів системи моніторингу

Інтеграція компонентів у сучасному SOC будується навколо централізованого збирання подій безпеки, їх уніфікації та подальшої аналітичної обробки, метою якої є перетворення великого потоку різнорідних сигналів у невелику кількість змістовних сповіщень та інцидентів, придатних для оперативного реагування. Базову роль у цьому процесі виконує SIEM, оскільки її архітектурним ядром виступає механізм кореляції, який нормалізує, скорочує, фільтрує та агрегує події з гетерогенних джерел, а також забезпечує автоматизоване застосування правил і політик з формуванням релевантних повідомлень для аналітика. Такий механізм не лише зменшує шум за рахунок дедуплікації та фільтрації, але й підтримує встановлення причинно-наслідкових зв’язків і кореневий аналіз (root cause analysis) шляхом співставлення кількох подій у спільному контексті.

На практиці інтеграційна модель SOC передбачає розгортання ланцюга «збирання → агрегація → нормалізація → зберігання → кореляція та аналітика → сповіщення/інциденти → візуалізація та реагування» [31]. На етапі агрегації потоки журналів і подій надходять від різних доменів інфраструктури; далі модуль нормалізації приводить їх до уніфікованого подання (зокрема через перетворення різних форматів у JSON для подальшої обробки), що знижує залежність аналітики від специфіки джерела. На рівні кореляції застосовується набір правил, який трансформує «кучу подій» у менший набір більш значущих подій, зокрема шляхом об’єднання кількох попереджень в одне або шляхом формування сповіщення навіть тоді, коли окремі засоби не створили сповіщення,

але у сукупності журнали демонструють підозрілу активність (наприклад, спроби введення неправильного пароля).

Нижче на Рисунку 2.2 наведено узагальнену схему потоку подій у SIEM-центрованій архітектурі SOC, яка відображає ключові вузли інтеграції та місця, де відбувається зменшення шуму й підвищення контекстності подій [32].

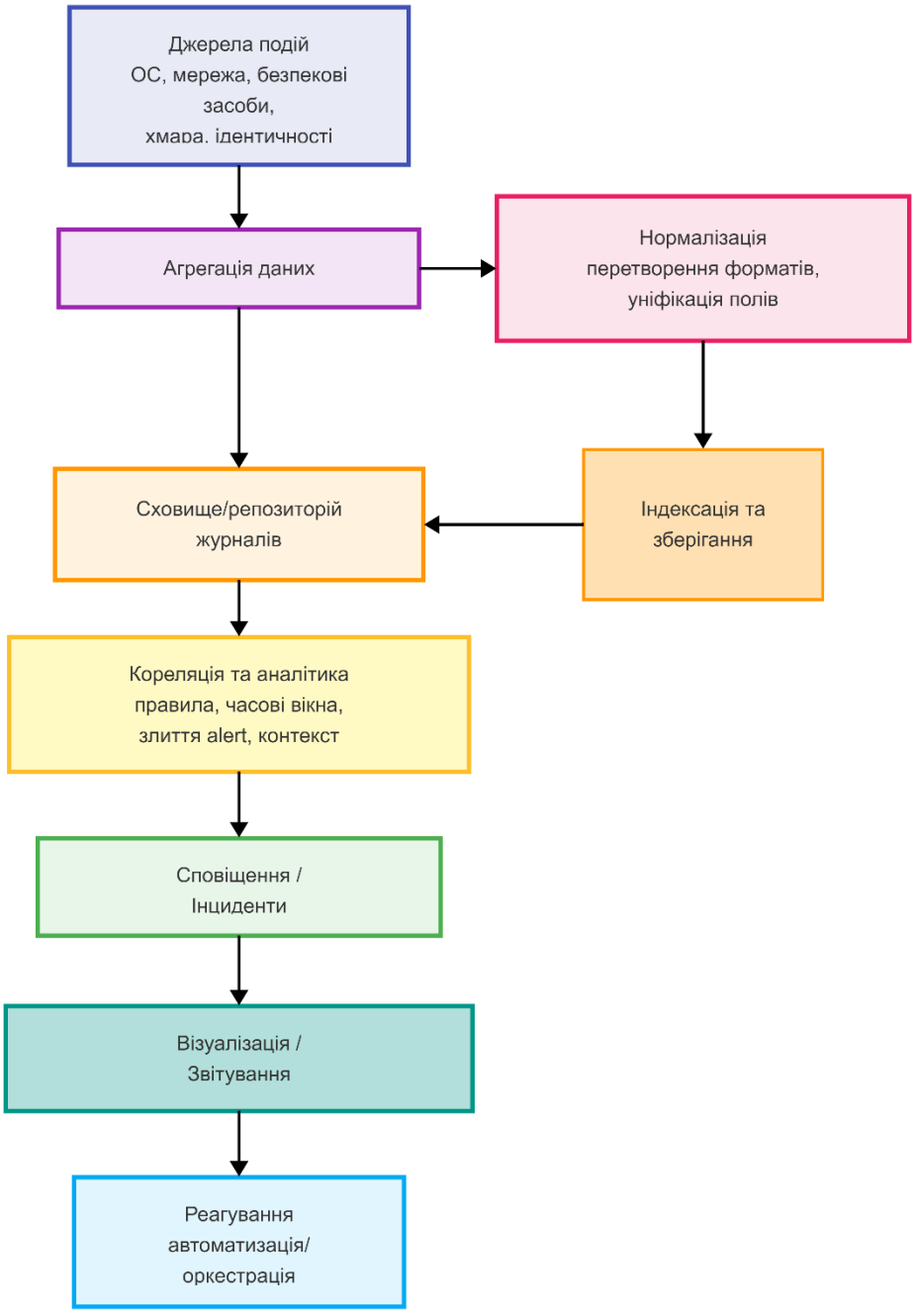

Рис.2.2 Узагальнена схема потоку подій у SIEM-центрованій архітектурі SOC

Критично важливим для інтеграції є те, що збільшення кількості джерел і зростання швидкості надходження подій підвищує навантаження на механізм кореляції, особливо коли використовуються складні правила (включно з правилами другого порядку, що корелюють результати інших правил), збільшуються часові контекстні вікна й виникає потреба обробляти правила в обмеженому часовому інтервалі. У таких умовах кореляційний механізм стає визначальним фактором продуктивності SIEM: за недостатньої ефективності зростають затримки обробки або з'являється ризик втрати подій, а вимоги до масштабування інфраструктури підвищуються.

Щоб окреслити комплементарність SIEM і XDR у складі SOC, доцільно розглядати XDR як інтегрований набір можливостей виявлення, аналітики та реагування, що поєднує сигнали принаймні з кількох доменів (наприклад, мережі та системного моніторингу), забезпечуючи міждоменний контекст. У сучасних комерційних рішеннях XDR-стек може бути вбудованим елементом SIEM-платформи як набір модулів «виявлення–аналітика–реагування», а також як розширення для аналізу мережевого трафіку й системного моніторингу, що підсилює функцію SOC з точки зору повноти спостережуваності та швидкості локалізації активності противника [33].

Узагальнюючи функціональні ролі, SIEM зосереджується на централізованому збиранні, нормалізації, довготривалому зберіганні журналів і кореляції подій з різних джерел; XDR доповнює це більш щільним міждоменним зв'язуванням сигналів та орієнтацією на швидке виявлення та реакцію в межах конкретних доменів (зокрема кінцевих точок і мережі), після чого результати збагачують загальну картину в SIEM.

Таблиця 2.2 нижче подає порівняльне узагальнення саме в термінах інтеграції SOC, з акцентом на те, які артефакти й вихідні дані формуються кожним компонентом та як вони зливаються в єдину модель подій.

Таблиця 2.2

Порівняльне узагальнення SIEM і XDR

Ознака інтеграції в SOC	**SIEM (централізована кореляція)**	**XDR (міждоменний контекст виявлення/реагування)**
Основний тип даних	Журнали/події з багатьох джерел, уніфіковані та збережені	Сигнали й сповіщення, зібрані та пов'язані між доменами
Ключовий механізм	Нормалізація, зменшення шуму, правила кореляції, злиття alert	Зв'язування телеметрії різних доменів у сценарії та сутності
Типовий результат	Сповіщення та інциденти на основі кореляції подій	Збагачені сповіщення/ланцюжки подій з міждоменними прив'язками
Точка інтеграції	Центральне сховище та кореляційний механізм	Постачальник сигналів і контексту для централізованої аналітики

З технічного погляду інтеграція SOC починається з підключення та агрегації джерел подій, де домени «операційні системи», «мережа та засоби безпеки», «хмарні сервіси» й «ідентичності» мають бути представлені на рівні даних у форматах, придатних до нормалізації та кореляції. У межах SIEM-архітектури джерела надсилають журнали, події та контекстні дані до модуля агрегації, після чого дані проходять нормалізацію (зокрема у формат JSON) і зберігаються, а кореляційний модуль порівнює нормалізовані записи з набором правил і формує alert у випадку збігу [34].

У системах, де використовується Microsoft Sentinel як SIEM/SOAR, інтеграція реалізується через збирання даних у Log Analytics Workspace (Azure Monitor Logs), застосування аналітики на основі KQL, формування сповіщень і групування їх в інциденти, а також подальшу автоматизацію реагування через плейбуки (Azure Logic Apps). Sentinel позиціонується як хмарно-орієнтоване рішення SIEM і SOAR, яке збирає дані з різних джерел у реальному часі, застосовує аналітику та машинне навчання для виявлення загроз і підтримує автоматизацію реакцій.

Окремий клас інтеграційних задач пов’язаний з форматами журналів і каналами доставки. Для мережевих пристроїв і частини засобів безпеки типовим механізмом є Syslog, який може забезпечувати надійну доставку з використанням TCP та підтримувати шифрування повідомлень (наприклад, у реалізаціях syslog-ng). Паралельно у середовищах SOC застосовуються поширені формати обміну подіями, такі як CEF і JSON. Вибір формату важливий для подальшої нормалізації, оскільки в архітектурі SIEM модуль нормалізації має приводити події з різних форматів до спільного представлення, щоб правила кореляції були незалежними від конкретного джерела.

У контексті Microsoft Sentinel підключення джерел подій здійснюється через вбудовані з’єднувачі, зокрема для журналів керування ідентичностями (Azure AD), журналів безпеки Windows, Syslog/CEF для мережевих пристроїв, а також для інтеграції з продуктами Microsoft Defender. При цьому Sentinel акумулює журнали, події та сповіщення від різних засобів, а аналітика на основі KQL використовується як для пошуку (hunting), так і для правил виявлення, включно з налаштовуваними правилами виявлення [35].

Для узгодженості інтеграції в SOC корисно розглядати джерела подій не лише за типом (Windows/Linux/мережа/хмара/ідентичності), а й за тим, які саме об’єкти та атрибути вони забезпечують для кореляції. Це визначає можливість міждоменного зв’язування за сутностями «користувач», «кінцева точка», «IP-адреса», «процес», «служба» та «часова ознака». У термінах кореляційного модуля, який «складає пазл» із фрагментів подій, повнота ідентифікаційних атрибутів безпосередньо впливає на здатність зливати кілька сповіщень в одне та відновлювати послідовність дій у межах інциденту.

У цьому контексті доцільно відобразити типову матрицю джерел і очікуваних атрибутів, Таблиця 2.3, яка використовується як практичний орієнтир для інтеграції й подальшого проєктування кореляційних правил

Таблиця 2.3

Типова матриця джерел і очікуваних атрибутів

Домен джерела	**Типові події**	**Ключові атрибути для кореляції**	**Очікуваний транспорт/формат**
Windows	події безпеки, невдалі входи	користувач, кінцева точка, час, IP-джерело, код події	агент/збір подій; уніфікація в сховище
Linux	автентифікація, системні повідомлення	користувач, хост, час, джерело з'єднання	Syslog (TCP), нормалізація/JSON
Мережеві пристрої та засоби безпеки	мережеві з'єднання, блокування, IDS/IPS	джерело/призначення, порт, протокол, час, дія	Syslog/CEF
Хмара	журнали сервісів, аудит	обліковий запис, ресурс, дія, час	з'єднувачі/журнали сервісів
Ідентичності	входи, аудит, зміни	користувач, результат входу, гео/час, зміни об'єктів	з'єднувачі журналів керування ідентичностями

Далі інтеграція переходить на рівень аналітики та кореляції, де ключовим артефактом є правило виявлення. На рівні SIEM правило виявлення задає логіку перетворення подій у сповіщення, використовуючи часові вікна, пороги, умови на поля та контекстні залежності. У rule-based кореляції кожна вхідна подія, після можливої нормалізації, співставляється з набором правил; за збігу вона може бути повідомлена, приглушена або може ініціювати визначену дію [36].

Ефективна кореляція у SOC досягається через послідовність стадій, що зменшують шум і підвищують змістовність результату: відкидання нерелевантних подій, агрегація та дедуплікація, маскування подій, що є наслідком відмов нижчих рівнів, і кореневий аналіз із використанням залежностей між подіями. Такий поділ на стадії формує основу для оптимізації правил: замість «важких» умов на сирих потоках спочатку застосовується фільтрація й агрегування, а вже потім більш складні контекстні конструкції.

Таблиця 2.4

Стадії кореляційної обробки подій безпеки та їх функціональне призначення

Стадія кореляції	**Функція**	**Практичний ефект для SOC**
Фільтрація подій	відкидання нерелевантного	зменшення обсягу для подальшої обробки
Агрегація та дедуплікація	об'єднання близьких/ідентичних	зменшення дублювань і «шуму»
Маскування	ігнорування похідних подій	уникнення хибних причинно-наслідкових ланцюжків
Кореневий аналіз	пошук залежностей між подіями	пояснення інциденту через сукупність сигналів

У Microsoft Sentinel правила виявлення реалізуються як вбудовані правила (rule templates) та налаштовувані правила виявлення, а аналітичні запити KQL виступають основою як для виявлення, так і для пошуку підозрілої активності. KQL використовується для ефективного запитування великих масивів даних і формування умов, що спрацьовують за певного патерна. У межах одного правила можуть бути задані пороги, часові інтервали (bin) та агрегації, які безпосередньо визначають як чутливість виявлення, так і навантаження на аналітичний рівень [37].

Нижче наведено приклад мінімалістичного налаштовуваного правила виявлення у вигляді KQL-запиту, де умова формалізує перевищення порогу невдалих входів для користувача протягом заданого інтервалу часу

```
SecurityEvent
| where EventID == 4625
| summarize FailedCount = count() by TargetUserName, bin(TimeGenerated, 1h)
| where FailedCount > 5
```

Ця логіка показує типову конструкцію «лічильник подій у часовому вікні» та демонструє, що параметри порогу і часової бінаризації одночасно задають і критерій виявлення, і профіль обчислювального навантаження, оскільки

аналітика повинна виконувати агрегування по сутностях у межах кожного вікна [38].

Для узгодження підходів у SOC важливо, що правила виявлення можуть бути описані у формалізованому вигляді незалежно від конкретної платформи. Наприклад, у Wazuh SIEM типове правило грубого підбору паролів задається через параметри частоти (frequency) та часових рамок (timeframe), а дія включає формування сповіщення та блокування джерельної IP-адреси. Такий опис не є «платформною специфікою», а відображає універсальну структуру правила: умова + часові параметри + дія.

```
<rule id="multiple-failed-logins-detection" name="Detect Multiple Failed Logins">
  <description>Identifies brute-force attacks by detecting multiple failed login attempts.</description>
  <condition>
    <failed_login_attempts>
      <count>2</count>
      <timeframe>30</timeframe>
    </failed_login_attempts>
  </condition>
  <action>
    <alert>Raise an alert and block the source IP.</alert>
  </action>
</rule>
```

Власне параметри count/timeframe у цій конструкції є сутністю налаштовуваного правила виявлення, оскільки вони визначають, коли послідовність подій переходить із «фонового шуму» в «підозрілу активність», а також задають вимоги до точності часових міток журналів і до пропускної здатності кореляційного механізму, який повинен обробити потік подій у межах заданого вікна.

Практична інтеграція SIEM і XDR у SOC у середовищі Microsoft додатково посилюється тим, що Sentinel інтегрується з продуктами Microsoft Defender (зокрема для кінцевих точок і ідентичностей), отримуючи збагачені сигнали, які можна корелювати з журналами мережі, хмари та автентифікації. Це забезпечує єдину точку огляду для аналітика: сповіщення з доменних засобів (де «перший удар» часто фіксує XDR/EDR) поєднуються з ширшим контекстом SIEM, що полегшує злиття сповіщень в інциденти та пріоритизацію.

Узгоджена модель кореляції для SOC передбачає не лише «виявити подію», а й сформувати інцидент як агреговане представлення загрози. Sentinel групує кілька сповіщень в один інцидент, який містить сутності (IP, користувач, машина) та часову лінію, що дозволяє розгортати розслідування від рівня окремих подій до рівня сценарію. Автоматизація реагування реалізується через плейбуки, які можуть ініціювати дії на кшталт блокування IP-адреси або ізоляції кінцевої точки у відповідь на спрацювання правила [39].

Окрему увагу в інтеграції SOC займає оптимізація правил виявлення з метою зменшення кількості хибних спрацювань і стабілізації навантаження. З боку кореляційного механізму «дорогими» є правила з великими часовими вікнами, високою кількістю залежностей і багаторівневим вкладенням, оскільки вони потребують зберігання контексту в пам’яті та обробки великих потоків подій. Системно це означає, що налаштування правил має враховувати компроміс між чутливістю та продуктивністю: збільшення кількості джерел і правил, зростання швидкості подій і ускладнення логіки кореляції прямо впливають на затримки обробки.

З позиції логіки правил, оптимізація базується на трьох групах важелів: по-перше, коректне попереднє зменшення шуму (фільтрація, дедуплікація), по-друге, точне визначення часових рамок та порогів, по-третє, коректне групування за сутностями (користувач, IP-адреса, кінцева точка) й використання послідовностей подій замість одиничних індикаторів. Показовим прикладом такого підходу є правило крос-кореляції подій автентифікації, де підозра формується не лише від «невдалих входів», а від патерна «невдалий вхід, після

якого протягом короткого часу відбувся успішний», що переводить подію з категорії потенційної помилки користувача у категорію підозрілої активності, підвищуючи точність виявлення.

Для узгодження міжплатформних підходів у SOC корисно формалізувати «паспорт правила», який лишається інваріантним незалежно від того, чи реалізується правило в Sentinel (KQL + scheduled rule), чи в іншій SIEM (XML/правила кореляції). У такому паспорті фіксуються параметри, які одночасно керують чутливістю, точністю та навантаженням.

Таблиця 2.5

Паспорт правила

Поле «паспорта правила»	**Семантика**	**Приклад реалізації**
Мета правила	який сценарій виявляється	грубий підбір паролів
Джерела даних	з яких журналів/таблиць	SecurityEvent / журнали входів
Умова	логічний предикат на поля	EventID/ознака failure
Поріг і часове вікно	count у timeframe/bin	>5 за 1h або >2 за 30s
Групування за сутністю	ключ агрегування	TargetUserName / SourceIP
Дія	що робити після спрацювання	сповіщення, блокування IP, ескалація

У підсумку інтеграція SIEM, XDR та суміжних компонентів SOC є не «підключенням журналів як таких», а побудовою керованого конвеєра даних, де формати й канали доставки забезпечують нормалізацію, механізм кореляції перетворює потоки подій у змістовні сповіщення, а налаштовувані правила виявлення задають формалізовані критерії виявлення та реагування з урахуванням продуктивності й ризику хибних спрацювань. У середовищі Microsoft Sentinel ця модель реалізується через централізоване збирання у Log Analytics Workspace, аналітику KQL, правила виявлення та механізми інцидент-менеджменту й автоматизації через плейбуки, а інтеграція з Microsoft Defender забезпечує додатковий доменний контекст для міждоменного зв’язування подій та інцидентів.

2.3. Практична реалізація моделі виявлення загроз у системі SOC

Практична модель виявлення загроз у SOC реалізується як сукупність керованих правил виявлення, аналітичних запитів та механізмів збагачення контексту, які разом формують цілісний процес переходу від телеметрії до інцидентів безпеки. На відміну від архітектурного та інтеграційного рівнів, розглянутих у попередніх підрозділах, цей рівень зосереджений на безпосередній роботі з даними: формуванні сценаріїв атак, налаштуванні умов спрацювання та забезпеченні того, щоб результати аналізу були операційно придатними для аналітиків SOC.

Ключовим принципом практичної реалізації є відокремлення трьох різних, але взаємопов'язаних елементів: правил виявлення, аналітичних запитів та контекстних запитів. Правила виявлення у Microsoft Sentinel реалізуються у вигляді аналітичних правил (Analytics rules) і визначають, коли саме система повинна створювати попередження або інцидент. Аналітичні запити на мові KQL слугують логічним ядром таких правил або інструментом для перевірки гіпотез і дослідження поведінки. Контекстні запити не призначені для виявлення загроз безпосередньо, але використовуються для збагачення інцидентів додатковими атрибутами, що підвищують точність оцінки ризику та прискорюють тріаж [40].

Практична модель виявлення починається з визначення сценаріїв загроз, які SOC прагне виявляти. Такі сценарії формуються не на рівні окремих подій, а на рівні поведінки сутностей у часі. Наприклад, окрема подія доступу до API Microsoft Graph або окреме членство користувача в групі не є інцидентом. Інцидент виникає тоді, коли спостерігається нетипова або надмірна активність, яка виходить за межі очікуваного профілю поведінки

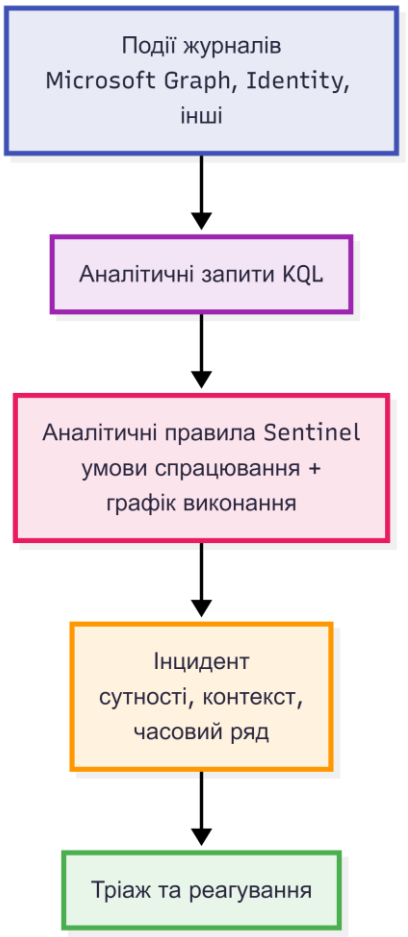

Рис. 2.3 Перехід від телеметрії до інциденту у практичній моделі SOC

Одним із практично значущих сценаріїв є виявлення аномальної активності доступу до Microsoft Graph API, що може свідчити про компрометацію облікового запису або сервісного принципала. Для цього використовується телеметрія MicrosoftGraphActivityLogs, яка дозволяє аналізувати виклики API, IP-адреси та типи об’єктів, від імені яких виконуються запити.

У межах Microsoft Sentinel цей сценарій реалізується не як довільний запит, а як аналітичне правило, що має чітко визначені параметри виконання, умови спрацювання та результат

Правило Sentinel SR-01. Аномальна інтенсивність викликів Microsoft Graph API

Тип: Analytics rule (Scheduled)

Мета: виявлення нетипової активності доступу до Microsoft Graph API з боку користувача або сервісного принципала

Джерело даних: MicrosoftGraphActivityLogs

Графік виконання: запуск кожні 10 хвилин, аналіз за останні 35 хвилин

Умова спрацювання: наявність результатів, що перевищують заданий поріг різноманітності або кількості викликів

Сутності інциденту:

- Account або Service principal (ObjectId)
- IP address (IPAddress)

Результат: створення інциденту з групуванням за об’єктом доступу

Rule logic (KQL):

```
MicrosoftGraphActivityLogs
| where ingestion_time() > ago(35m)
| extend ObjectId = coalesce(ServicePrincipalId, UserId)
| extend ObjectType = iff(isempty(UserId), "ServicePrincipal", "User")
| where RequestUri !has "microsoft.graph.delta"
| extend NormalizedRequestUri = tostring(parse_url(RequestUri).Path)
| extend NormalizedRequestUri = replace_regex(NormalizedRequestUri, @'[0-9a-fA-F\-]{36}', '<UUID>')
| extend NormalizedRequestUri = replace_regex(NormalizedRequestUri, @'\?.*$', '')
| summarize Endpoints = make_set(NormalizedRequestUri, 1000),
            EndpointCount = dcount(NormalizedRequestUri),
            IPs = make_set(IPAddress)
            by ObjectId, ObjectType
| where EndpointCount > 20
```

Рис. 2.4 Результат запиту до MicrosoftGraphActivityLogs

Для того щоб інциденти, створені цим правилом, були інтерпретованими, модель виявлення передбачає використання контекстних запитів, які не створюють інцидентів самі по собі, але збагачують їх інформацією про сутність.

Одним із таких контекстних запитів є отримання інформації про членство користувача в групах, що дозволяє швидко визначити, чи має обліковий запис підвищені привілеї або доступ до критичних ресурсів.

Контекстний запит CTX-01. Членство користувача в групах

```
IdentityInfo
| summarize arg_max(TimeGenerated, *) by AccountObjectId
| mv-expand GroupMembership
| summarize TotalMemberships = dcount(tostring(GroupMembership)),
        MemberOf = make_set(tostring(GroupMembership), 1000)
        by AccountObjectId, AccountDisplayName, AccountUPN
```

Рис. 2.4 Результат запиту щодо членства користувачів в групах

Аналогічним чином використовується контекстний запит для отримання інформації про призначені ролі, що особливо важливо для пріоритизації інцидентів, пов'язаних із доступом до адміністративних функцій.

Контекстний запит CTX-02 – Призначені ролі користувача

```
IdentityInfo
| summarize arg_max(TimeGenerated, *) by AccountObjectId
| mv-expand AssignedRoles
| where isnotempty(AssignedRoles)
| summarize TotalRoles = dcount(tostring(AssignedRoles)),
            Roles = make_set(tostring(AssignedRoles), 100)
            by AccountObjectId, AccountDisplayName, AccountUPN
```

Рис. 2.5 Результат запишу щодо призначених ролей користувача

У практичній моделі SOC ці контекстні запити використовуються під час тріажу або автоматичного збагачення інциденту, але не розглядаються як правила виявлення, оскільки не містять умови, яка визначає факт загрози.

Таблиця 2.6

Роль правил і запитів у практичній моделі виявлення

Елемент	Призначення	Результат
Аналітичне правило Sentinel	Визначає, коли створюється інцидент	Інцидент SOC
KQL-запит (rule logic)	Формує логіку виявлення	Набір результатів
Контекстний запит	Збагачує інцидент даними	Пріоритизація та швидший тріаж

Отже, практична модель виявлення загроз у SOC реалізується не через окремі запити, а через узгоджену систему правил і контексту. Microsoft Sentinel у цій моделі виступає не просто як сховище журналів, а як платформа, що дозволяє формалізувати поведінкові сценарії у вигляді керованих правил, пов’язаних із сутностями та процесами реагування. Саме така організація дозволяє уникнути плутанини між «запитом» і «правилом», зменшити шум та забезпечити стабільну операційну роботу SOC.

Висновки до розділу 2

У другому розділі роботи розроблено архітектуру системи моніторингу SOC та практичну модель виявлення загроз з використанням інтеграції SIEM, XDR та суміжних компонентів.

Обґрунтовано необхідність визначення цільової операційної моделі SOC перед вибором технічних інструментів, оскільки саме амбіції організації щодо моніторингу, швидкості реагування та глибини аналітики визначають вимоги до телеметрії, пропускної здатності та обчислювальних ресурсів. Запропоновано описувати архітектуру SOC у трьох взаємопов'язаних поданнях: логічному (структура даних та їх рух), функціональному (виконувані функції конвеєра) та мережевому (фізична реалізація та сегментація).

Розроблено узагальнену схему архітектури SOC як багатошарового конвеєра подій, що включає рівні джерел телеметрії, збору та транспорту, сховища і нормалізації, аналітики та кореляції, керування інцидентами та автоматизації реагування. Формалізовано технічні вимоги до інфраструктури для кожного рівня з акцентом на балансі між швидкістю прийому подій, можливістю індексування та продуктивністю кореляційних механізмів.

Визначено принципи інтеграції SIEM та XDR у складі SOC, де SIEM забезпечує централізоване збирання, нормалізацію та кореляцію подій з різних джерел, а XDR доповнює це міждоменним зв'язуванням сигналів та швидким реагуванням у межах конкретних доменів. Розроблено модель кореляційної обробки подій через стадії фільтрації, агрегації, дедуплікації, маскування та кореневого аналізу, що забезпечує зменшення шуму та підвищення змістовності результатів.

На практичному рівні реалізовано модель виявлення загроз у Microsoft Sentinel з використанням аналітичних правил на основі KQL. Розроблено правило виявлення аномальної активності доступу до Microsoft Graph API та контекстні запити для збагачення інцидентів інформацією про членство користувачів у групах та призначені ролі. Формалізовано розмежування між

правилами виявлення, аналітичними запитами та контекстними запитами, що забезпечує операційну придатність моделі для роботи аналітиків SOC.

Запропонована архітектура та практична модель забезпечують модульність, масштабованість та незалежність від конкретних технічних рішень, що дозволяє адаптувати систему до змін у ландшафті загроз без руйнування загального конвеєра обробки подій.

РОЗДІЛ 3 ТЕХНІЧНЕ НАЛАШТУВАННЯ СИСТЕМИ МОНІТОРИНГУ ТА РЕАГУВАННЯ

3.1 Налаштування SIEM-платформи (на прикладі Microsoft Sentinel)

Практичне розгортання Microsoft Sentinel доцільно розглядати як послідовність налаштувань, у яких робочий простір Log Analytics виступає єдиним сховищем і обчислювальним середовищем для телеметрії, а Sentinel додає поверх нього функції аналітики, інцидент-менеджменту, візуалізації та автоматизації. Тому стартовою одиницею конфігурації є саме робочий простір Log Analytics, до якого прив'язується Sentinel і в який згодом підключаються з'єднувачі даних (data connectors), а вже після цього налаштовуються аналітичні правила виявлення та робочі книги (workbooks) для моніторингу. Така послідовність виправдана тим, що Sentinel використовує Log Analytics як репозиторій даних: усі подальші перевірки коректності надходження подій, побудова правил і візуалізацій спираються на наявність даних у таблицях Log Analytics і можливість їх аналізу через KQL.

Практична підготовка починається в Azure Portal зі створення робочого простору Log Analytics. У порталі в полі пошуку обирається "Log Analytics workspaces", далі натискається "+ Create", після чого задаються підписка (subscription), група ресурсів (resource group), назва робочого простору та регіон розміщення. Регіон доцільно обирати узгоджено з тим, де розгорнуті або плануються основні джерела телеметрії (віртуальні машини, служби в Azure), оскільки це спрощує експлуатацію та зменшує ризики затримок/розривів у доставці подій. Після заповнення параметрів використовується "Review + Create" і "Create", що фіксує створення робочого простору як базової точки збору даних для SOC.

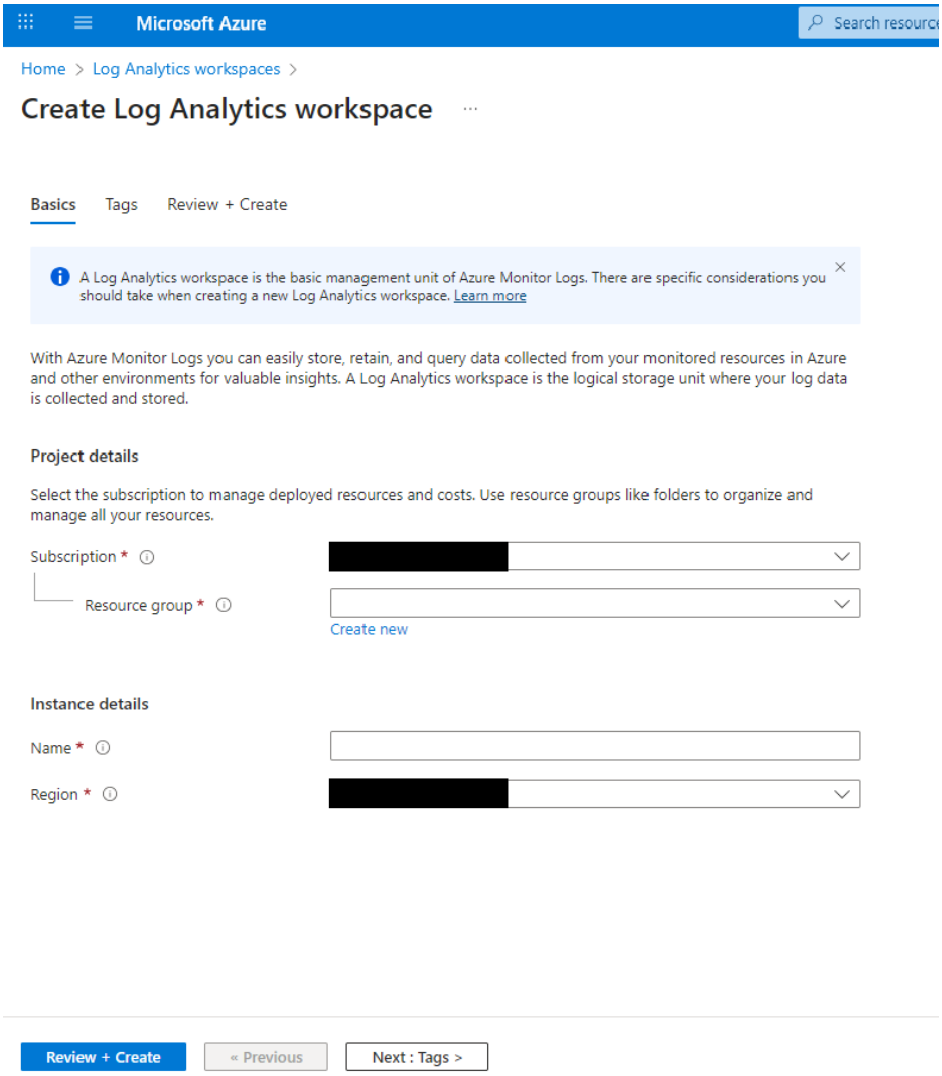

Рис. 3.1 Створення робочого простору Log Analytics

Після створення робочого простору виконується увімкнення Microsoft Sentinel у цьому просторі. У Azure Portal в пошуку обирається “Microsoft Sentinel”, натискається “+ Add”, у майстрі вибирається щойно створений Log Analytics workspace і підтверджується “Add”. На практиці це є ключовою прив’язкою: від цього моменту всі з’єднувачі даних, аналітичні правила та робочі книги будуть налаштовані в контексті конкретного workspace.

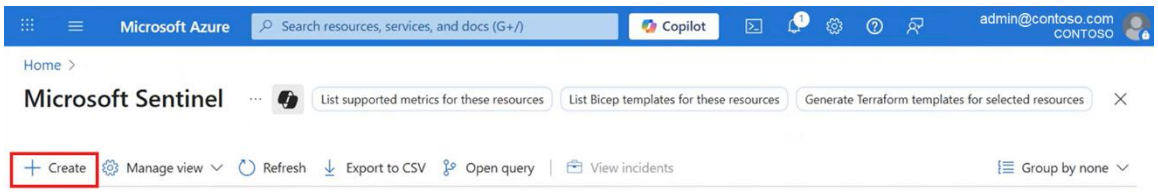

Рис. 3.2 Ілюстративне додавання Microsoft Sentinel до робочого простору

Питання доступів у практичному налаштуванні зводиться до того, що користувач, який розгортає Sentinel, повинен мати можливість створювати й керувати ресурсами в Azure tenant та розгортати сам Sentinel у вибраному робочому просторі. Оообов'язкові передумови це: активна Azure subscription, наявність Log Analytics workspace і коректні права на розгортання та використання Sentinel. Це важливо саме практично: без прав на рівні Azure tenant/ресурсів неможливо коректно завершити жоден із наступних кроків (з'єднувачі даних, аналітика, інциденти).

Після того як Sentinel прив'язано до робочого простору, починається підключення телеметрії. Практично вся робота концентрується на сторінці "Data connectors" у Microsoft Sentinel. Для кожного джерела даних обирається відповідний з'єднувач і виконується підключення за інструкцією з майстра. У базовому сценарії для SOC першими підключаються журнали ідентичностей та автентифікації (Azure Active Directory / Entra ID), журнали подій Windows (SecurityEvent) і мережеві журнали (Syslog/CEF) – це формує мінімально життєздатний набір сигналів для первинного тріажу.

Налаштування Azure Active Directory (Entra ID) у практичному вигляді зводиться до відкриття "Data connectors", пошуку "Azure Active Directory", переходу на сторінку з'єднувача ("Open connector page") і натискання "Connect" для підключення принаймні Sign-in logs та Audit logs. У результаті Sentinel починає отримувати журнали входів і аудит-дані, які потім використовуються і в аналітичних правилах, і у робочих книгах для моніторингу підозрілих входів/аномалій.

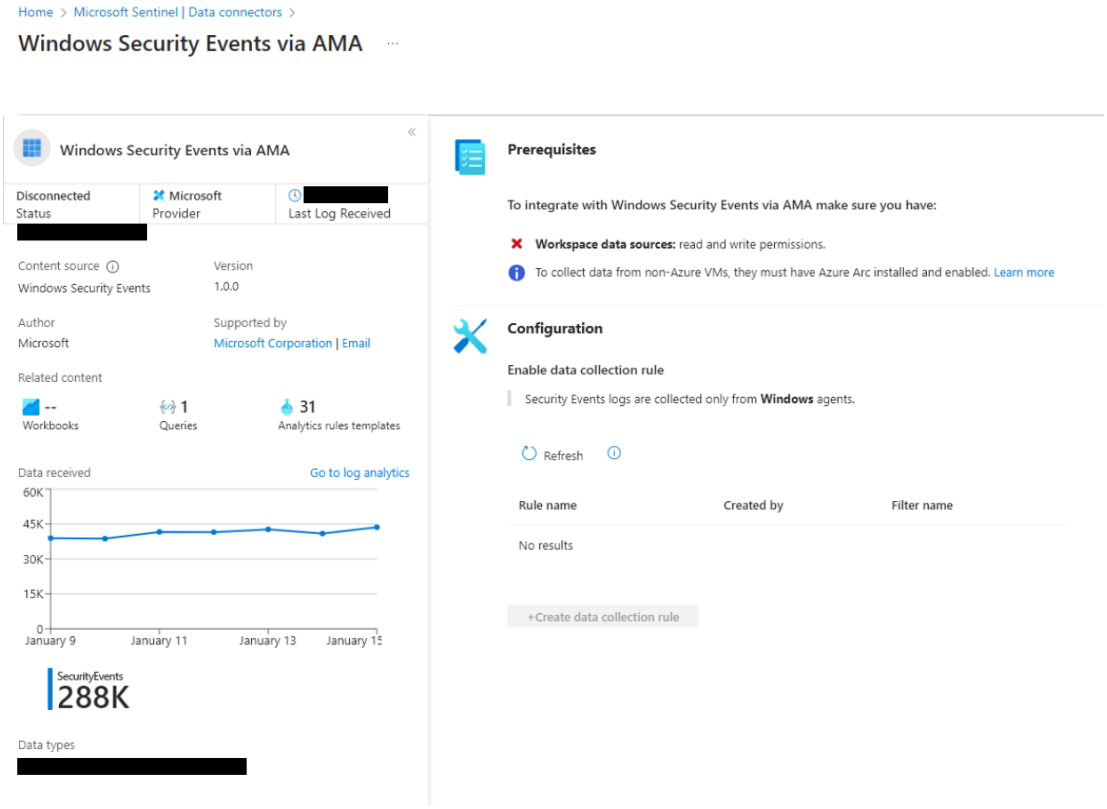

Рис. 3.3 Підключення з’єднувача даних

Для Windows-подій практичний сценарій передбачає інсталяцію агента на Windows-сервер/VM і налаштування відправки подій у Log Analytics workspace. Як мінімум важливо забезпечити надходження таблиці SecurityEvent, оскільки саме вона часто використовується як базова для перевірочних правил (наприклад, події невдалих входів з EventID 4625). У прикладному розгортанні це реалізується через інсталяцію Microsoft Monitoring Agent (MMA) на Windows-хості та прив’язку його до потрібного workspace, після чого вмикається збір SecurityEvent під Windows Event Collection

Окремо на практиці важливо розуміти, що збір даних у Sentinel може виконуватися різними методами інгесту, і вибір методу визначається типом джерела. Для on-premises / IaaS середовищ є різні набори типових шляхів, наприклад: Syslog/CEF, WEF, Logstash, агентний збір для Windows Server (через MMA або AMA) та збір для Linux (через AMA або MMA). Для хмарних сервісів Azure – переважають нативні з’єднувачі даних. Таке розділення практичне, бо

дозволяє одразу планувати “чим доставляємо дані” ще до того, як починається налаштування правил виявлення.

У сценаріях, де організація використовує Windows Event Forwarding (WEF), потік подій будується через WEF Collector (WEC): клієнти отримують налаштування через групові політики (GPO), запитують параметри підписки у WEF Collector, далі пересилають копії подій на сервер-колектор, а вже колектор пересилає події у Sentinel через агент (у схемі – MMA). Така конструкція на практиці дозволяє централізувати збір подій з великої кількості хостів і контролювати профіль збираних подій (які саме журнали і які події надходять у SIEM).

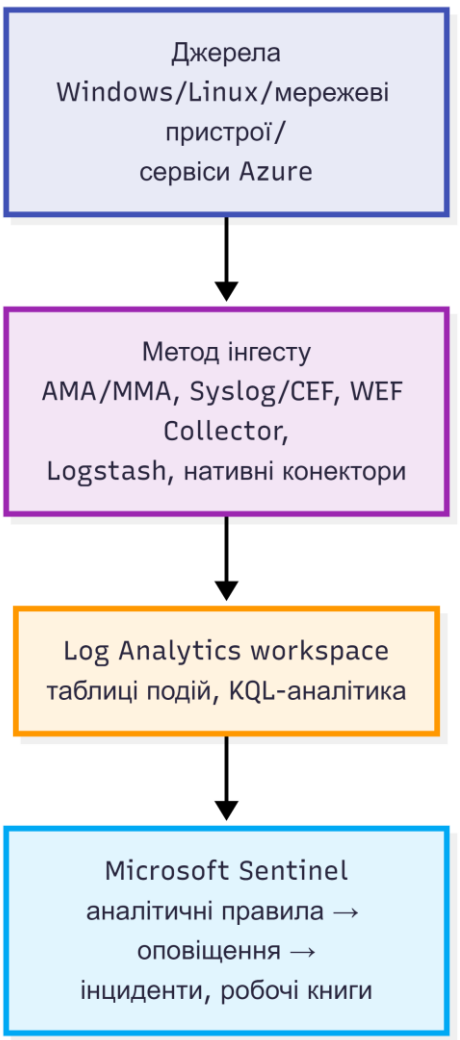

Рис. 3.4 Типовий ланцюг інгесту подій у Microsoft Sentinel (агентний, Syslog/CEF, WEF)

Для практичного оформлення доцільно фіксувати підключення джерел у вигляді “джерело → тип даних → призначення”, щоб показати, що збір не є випадковим, а прив’язаний до задач SOC і майбутніх правил виявлення.

Таблиця 3.1

Підключення джерел даних у Microsoft Sentinel: джерело → тип даних → призначення

Джерело	**Метод інгесту / з’єднувач**	**Тип даних (приклад)**	**Практичне призначення в SOC**
Azure Active Directory (Entra ID)	Нативний з’єднувач даних	Sign-in logs, Audit logs	Моніторинг входів, аудит змін, кореляція по користувачах
Windows Security Events	З’єднувач Windows + агент (MMA/AMA)	SecurityEvent	Детекція невдалих входів, процесів, подій ОС
Мережеві пристрої/фаєрволи	Syslog або CEF	Syslog/CEF події	Виявлення мережевих аномалій, блокування джерел
Microsoft Defender (наприклад, Defender for Endpoint)	Нативний з’єднувач даних	Сигнали захисту кінцевих точок	Підсилення детекції та інцидентів endpoint-рівня
Інші рішення/вендори	API / Syslog/CEF / REST API	Кастомні журнали	Розширення покриття під специфіку організації

Практична перевірка коректності надходження телеметрії повинна виконуватися через два паралельні контури. Перший – візуальний: на сторінці конкретного data connector відображається статус підключення. Другий – аналітичний: у Log Analytics виконується простий KQL-запит до очікуваної таблиці (наприклад, SecurityEvent) або використовується мінімальний тестовий запит, який повертає хоча б один рядок, щоб переконатися, що інгест працює. Також існує варіант лабораторної симуляції даних, коли реальної телеметрії ще немає, – це дозволяє тестувати правила та візуалізації без очікування “живих” подій.

Після того як у робочому просторі з’явилися дані, налаштовуються аналітичні правила виявлення (Analytics rules). У практичній реалізації Sentinel є

два базові підходи: увімкнення вбудованих правил із шаблонів (rule templates) і створення настроюваних правил виявлення на основі KQL. Вбудовані правила активуються у вкладці Analytics через “Rule templates”, де можна швидко знайти типові кейси (наприклад, “Multiple failed logins”, “Impossible travel”) і створити правило з попередньо заданими параметрами, перевіривши налаштування перед увімкненням. Це дозволяє швидко отримати первинну детекцію без розробки KQL “з нуля”.

Настроюване правило виявлення в Sentinel складається з чотирьох практичних частин: логіка (KQL), планування виконання (частота запуску та вікно аналізу), збагачення/прив’язка сутностей (entity mapping та, за потреби, enrichment) і налаштування інцидентів (створення інциденту та групування оповіщень). Навіть простий приклад правила для детекції грубого підбору паролів демонструє цю структуру: KQL відбирає події невдалих входів (EventID 4625), агрегує їх за користувачем у часових бін-інтервалах і застосовує поріг (Count > 5). На практиці важливо, що поріг і часовий інтервал – це параметри, які треба підбирати під середовище: вони визначають і чутливість детекції, і обсяг шуму.

```
SecurityEvent
| where EventID == 4625
| summarize FailedCount = count() by TargetUserName, bin(TimeGenerated, 1h)
| where FailedCount > 5
```

Рис. 3.5 Результат запишу щодо подій невдалих входів в систему

Підхід “near-real-time” у межах практичного налаштування означає, що правило має запускатися з малою періодичністю та аналізувати вузьке вікно часу, щоб скорочувати затримку між появою події та створенням оповіщення/інциденту. У термінах конфігурації це реалізується параметрами планування правила: частота запуску (run frequency) і період запиту (lookup/query period). Для таких правил критично важливо, щоб інгест подій у workspace був стабільним і передбачуваним, оскільки правило працює на “свіжому” інтервалі і будь-яка затримка доставки подій може прямо вплинути на момент детекції.

Налаштування створення інцидентів у Sentinel прив’язується до оповіщень, які генерує правило. У практичному майстрі конфігурації правило може створювати інциденти з оповіщень, а також застосовувати групування (наприклад, за назвою оповіщення або сутністю), щоб не множити інциденти без потреби. Це особливо важливо для правил на кшталт підбору паролів: якщо групування відсутнє, SOC може отримати надмірну кількість інцидентів, тоді як правильне групування концентрує сигнал у керовану кількість кейсів для тріажу

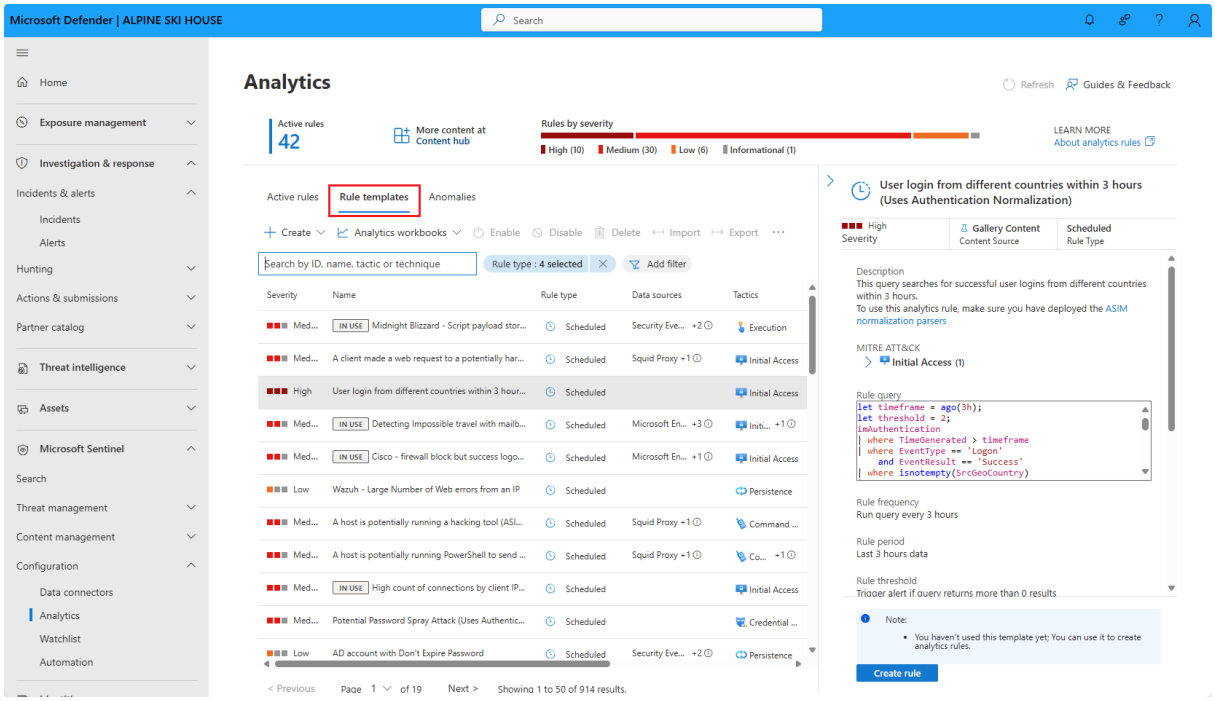

Рис. 3.6 Ілюстративне ввімкнення вбудованих правил із Rule templates

Після налаштування інгесту і правил виявлення SOC потребує інтерфейсу для щоденного моніторингу, і в Sentinel цю роль виконують робочі книги (workbooks) як налаштовувані панелі моніторингу (dashboards). Практична робота з workbooks починається у Sentinel з переходу в розділ Workbooks і створення нової робочої книги (“+ New”) або використання шаблону. Далі обираються джерела, які потрібно візуалізувати (наприклад, Windows Security та Azure AD), після чого додаються візуалізації у вигляді таблиць, графіків і карт, що підтримують тріаж: кількість невдалих входів, топ IP-адрес, активні оповіщення, тренди по часу

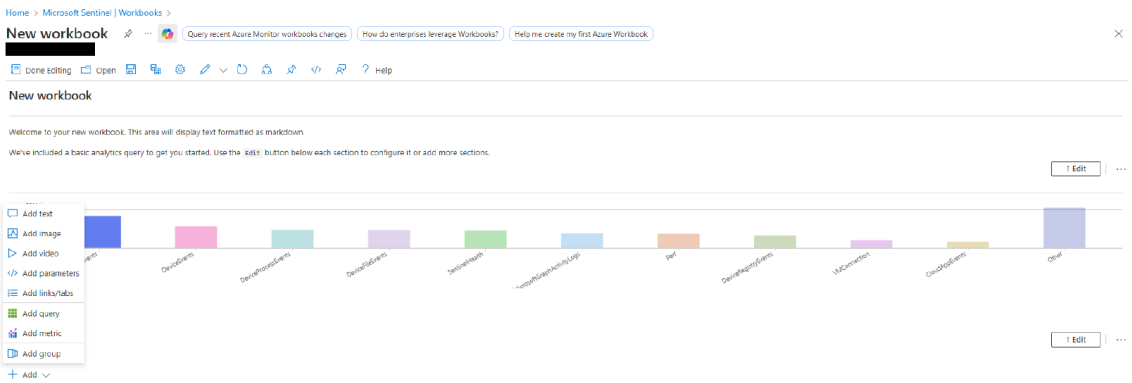

Рис. 3.7 Створення робочої книги (workbook) для моніторингу SOC

У практичному сенсі робоча книга повинна відображати не всю доступну інформацію, а операційні індикатори, які допомагають черговому аналітику за хвилини оцінити стан середовища: чи зростає хвиля невдалих входів, чи з'явилися нетипові джерела, чи є інциденти високої критичності. Оскільки Sentinel інтегрує в одному середовищі KQL-аналітику і візуалізацію, робочі книги фактично стають інтерфейсом для швидкого тріажу: графіки та таблиці мають бути напряму пов'язані з тими ж таблицями/подіями, на яких працюють правила виявлення

Таблиця 3.2

Практична структура SOC-dashboard у Sentinel (workbook)

Блок у workbook	**Дані/фокус**	**Практична роль для тріажу**
"Невдалі входи у часі" (time chart)	Sign-in logs / SecurityEvent	Раннє виявлення хвилі підбору паролів, визначення піків
"Топ IP-адрес" (table)	Sign-in logs / мережеві події	Виявлення домінуючих джерел атак/аномалій
"Активні оповіщення/інциденти" (table)	Incidents/Alerts	Операційна черга SOC для розбору
"Карта/географія входів" (map)	Azure AD sign-ins	Виявлення нетипової географії ("impossible travel")

Підсумовуючи, робочий простір Log Analytics створює технологічний фундамент збору та аналізу через KQL, Microsoft Sentinel додає керування інцидентами й аналітичними правилами, з'єднувачі даних забезпечують керований інгест з ключових доменів (ідентичності, ОС, мережа, хмара), а workbooks формують операційний інтерфейс SOC. Важливо, що всі ці компоненти налаштовуються так, щоб їх можна було перевірити: підключення – через статуси з'єднувачів і наявність рядків у таблицях, правила – через контрольоване спрацювання і появу інцидентів, візуалізації – через коректне відображення трендів і черги інцидентів для тріажу

3.2. Реалізація автоматизованого реагування (SOAR)

Автоматизоване реагування в SOC доцільно трактувати як формалізований механізм виконання повторюваних дій після того, як SIEM вже сформував інцидент. На практиці SIEM-рівень відповідає за виявлення та консолідацію сигналів (alerts) у керовані кейси (incidents), а SOAR-рівень перетворює ці кейси на послідовності дій реагування (workflows), що знижує навантаження на аналітиків та скорочує час реагування. У Microsoft Sentinel це реалізується через Automation (правила автоматизації) і playbooks на базі Logic Apps, де інцидент виступає вхідним об'єктом, а workflow – інструментом обробки та виконання операційних кроків: повідомлення, збагачення (enrichment), створення тікетів, дії стримування (containment) та ведення аудиту. Цінність такої архітектури полягає в тому, що повторювані дії виконуються не "вручну за чек-листом", а стандартизовано, з контрольними умовами запуску, журналюванням результатів і можливістю відтворення процесу для кожного інциденту

Рис. 3.8 Розподіл ролей між SIEM і SOAR у SOC

У практичному сенсі SOAR не "замінює аналітика", а зменшує частку ручної рутини: первинний тріаж, комунікації, створення тікетів, типові

стримувальні дії. Це дозволяє аналітику фокусуватися на розслідуванні та ухваленні рішень у нестандартних ситуаціях.

SOAR-частина життєвого циклу інциденту стартує з моменту, коли інцидент вже створено і потрібно виконати дії, які мають бути однаковими для певного класу сценаріїв: швидко зібрати контекст, оцінити ризик, повідомити відповідальних, виконати стримування або підготувати дані для подальшого розслідування. У Microsoft Sentinel ця логіка реалізується через розділення відповідальності на три шари: аналітичний (Analytics rules), шар запуску (Automation rules) та шар виконання (Logic Apps).

Аналітичний шар формує інцидент із атрибутами, які далі використовуються для автоматизації: критичність (Severity), назва/тип, час, статус, сутності (entities) – наприклад IP-адреса, обліковий запис, хост. Шар запуску описує політику: за яких умов інцидент повинен спричиняти запуск workflow. Це дозволяє уникнути хаотичного запуску автоматизації “на всі події” і задає керовані правила, наприклад запуск тільки для High severity або лише для інцидентів певного типу. Шар виконання – це безпосередній playbook, який отримує дані інциденту і виконує дії у зовнішніх системах (M365, Teams, Azure Resource Manager, ServiceNow тощо), повертаючи результат у вигляді повідомлень, створених записів або змін у політиках доступу

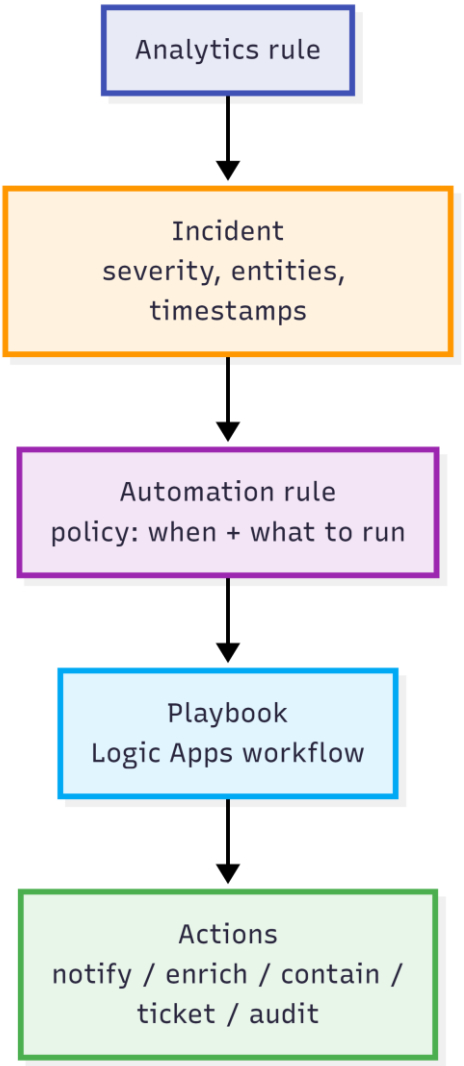

Рис. 3.9 Логічний ланцюг “Incident → Automation → Playbook → Actions”

Ключове практичне питання в цій архітектурі – безпечність автоматизації. Автоматизація не повинна створювати більший ризик, ніж сама подія, тому в SOC зазвичай застосовується принцип “graduated response”: чим ризиковіша дія (наприклад блокування або ізоляція), тим більше контрольних умов перед її виконанням. Це може бути обмеження за Severity, вимога додаткових ознак, узгодження (approvals) або перевірка на allowlist.

Таблиця 3.3

Практичні запобіжники безпечної автоматизації

Ризик	**Запобіжник у SOC**	**Реалізація у workflow / політиці**
Блокування легітимного трафіку	Обмеження запуску на High severity і/або конкретні типи інцидентів	Умова запуску Automation rule + додаткові умови у playbook
Неповні сутності (немає IP/Account)	Валідація вхідних даних перед діями	“Перевірити, що entities містять потрібний тип” → інакше лише повідомлення
Неможливість відкату дії	Журналювання і контроль змін	Логування виконаних змін + збереження параметрів (rule name/priority/target)
Надмірні сповіщення	Групування інцидентів і обмеження повторного запуску	Suppression/групування + “не запускати повторно протягом N хвилин”

Практична реалізація playbook має бути описана не як “набір кліків”, а як чіткий алгоритм: які вхідні дані отримує workflow, які перевірки виконує і які дії запускає. Основою є подія (trigger), що передає інцидент або оповіщення, після чого workflow обробляє поля інциденту і сутності (entities). У типовому SOC-варіанті першим рівнем автоматизації виступає низькоризиковий playbook, який не змінює інфраструктуру, а виконує комунікацію та фіксацію інциденту: надсилає повідомлення черговому та створює запис у системі обліку (якщо використовується ITSM)

Алгоритм базового playbook для High severity інцидентів:

Вхід: Incident (Severity, Title, Status, CreatedTime, Entities[])

1) Перевірити Severity.

2) Якщо Severity != High → завершити без дій або виконати лише логування.

3) Якщо Severity == High:

3.1) Зібрати ключові поля (Title, Time, Link, Entities).

3.2) Виконати повідомлення (Email/Teams).

3.3) (Опційно) Створити тікет у ITSM.

3.4) Записати в аудит результат виконання.

Вихід: повідомлення/тікети/аудит

Технічно важливим практичним кроком є нормалізація сутностей інциденту. У Sentinel сутності можуть містити різні типи (ip, account, host тощо), тому перед виконанням containment-дій потрібна фільтрація: наприклад, блокування IP можливе лише коли є сутності типу IP. Цей принцип варто фіксувати як обов'язкову умову в playbook-логіці, щоб workflow не переходив до ризикових кроків без коректних даних.

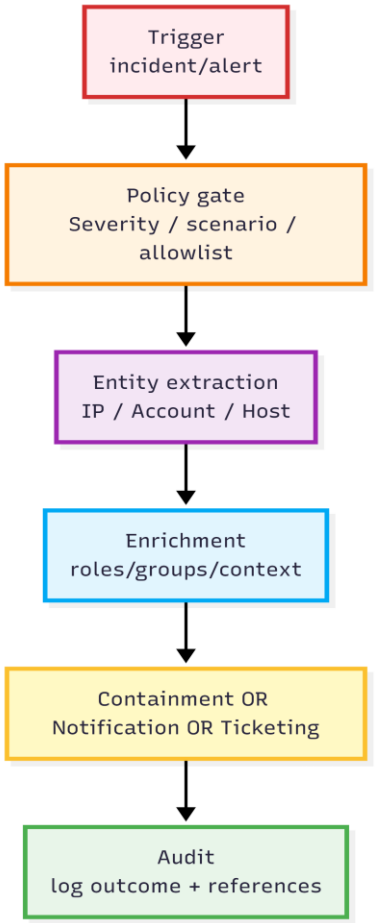

Рис. 3.10 Універсальна структура playbook

Таблиця 3.4

Вхідні поля інциденту та їх практичне використання у playbook

Поле інциденту	Використання	Навіщо потрібне
Severity	“policy gate” (дозвіл/заборона ризикових дій)	Безпечний запуск автоматизації
Title/Name	Тема повідомлення/тікета	Швидка ідентифікація кейсу
CreatedTime/Time	Таймлайн та контроль SLA	Оцінка оперативності реагування
Entities (IP/Account/Host)	Мішені дій реагування	Визначає, що саме блокувати/перевіряти
Link/IncidentId	Посилання для аналітика	Швидкий перехід у SIEM для тріажу

Дії реагування в SOC практично поділяються на чотири групи: сповіщення, збагачення, стримування та ескалація/реєстрація. Такий поділ дозволяє будувати реагування сходинками: спочатку виконувати дії, що не впливають на доступність, і лише потім переходити до containment за наявності достатніх підстав. У рамках Microsoft Sentinel/Logic Apps це означає, що playbook має мати чіткі блоки: повідомлення, enrichment, containment і тикетинг, які можуть комбінуватися залежно від сценарію.

Таблиця 3.5

Практичні дії реагування в SOAR workflow

Категорія	Типова дія	Вхідні дані	Очікуваний результат
Notification	Повідомлення Email/Teams	Title, Severity, Link	Оперативне залучення чергового
Enrichment	Отримання ролей/груп користувача	Account / AccountId	Пріоритизація, зниження false positives
Containment	Блокування IP через NSG	IP	Зупинка/обмеження атакуючого джерела
Ticketing	Створення тікета в ServiceNow	Title, Severity, Entities, Link	Формалізація процесу, аудит, SLA

Для containment-сценарію “block IP” практична логіка будується так, щоб workflow був детермінований і відтворюваний: витягти IP із entities, сформувати правило блокування з унікальними параметрами (назва, priority), застосувати його до NSG, зафіксувати дію і повідомити команду.

Алгоритм блокування IP на рівні NSG (containment)

Вхід: Entities[]

1) Витягти всі IP-сутності.

2) Для кожної IP:

2.1) Перевірити allowlist/внутрішні діапазони.

2.2) Згенерувати параметри NSG-rule (name, priority, deny inbound, source IP).

2.3) Створити/оновити правило в NSG.

2.4) Записати результат (успіх/помилка, rule-id).

3) Повідомити SOC-канал про виконану дію.

Вихід: NSG update + повідомлення + аудит

Оркестрація відрізняється від одиничної автоматичної дії тим, що workflow охоплює повний маршрут інциденту: від запуску до фіксації результатів і контролю якості виконання. На практиці є два типи організації: або кілька playbook-ів запускаються за однією умовою (розділення на “notify”, “contain”, “ticket”), або один playbook містить кілька гілок (branching) з умовами, які визначають, які блоки виконуються. Перший підхід простіше супроводжувати, другий – простіше контролювати як один процес

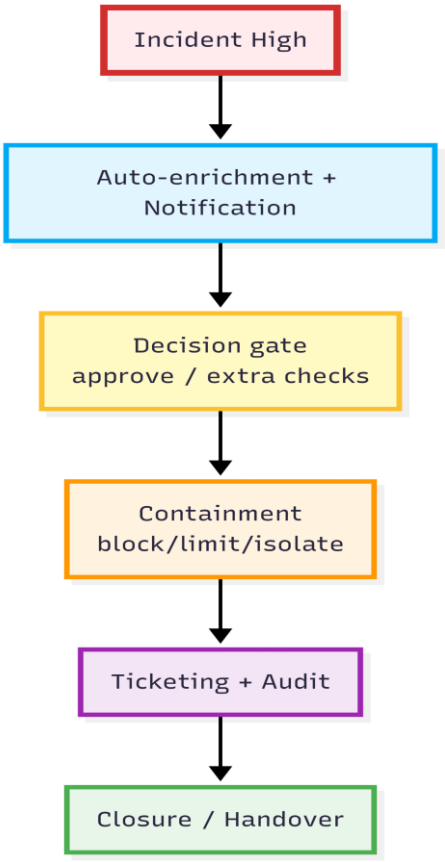

Рис. 3.11 Оркестрація реагування з керованою точкою рішення

Поєднання автоматичних і ручних дій у SOC практично реалізується через “керовані точки рішення”: автоматизація готує дані та виконує безпечні кроки, а ризикові дії виконуються або після підтвердження аналітика, або за сукупністю умов (наприклад High severity + наявність IP + відсутність у allowlist). Це зменшує ризики, але не вбиває швидкість, бо аналітик отримує готовий контекст і лише приймає рішення.

Таблиця 3.6

Розподіл ручних і автоматичних кроків у SOC

Крок	**Режим**	**Чому так**
Збагачення контексту	Автоматично	Не впливає на доступність, зменшує час тріажу
Повідомлення/ескалація	Автоматично	Мінімізує затримку залучення відповідальних
Containment (блокування/ізоляція)	Напівавтоматично	Високий ризик помилкових дій → потрібен контроль
Закриття інциденту	Ручний	Потребує контекстного рішення і підтвердження

Контроль виконання та аудит реагування забезпечуються через фіксацію виконаних дій і їх результатів (успішно/помилка), а також через журналювання параметрів змін (що саме блокувалось, коли, яким правилом, ким/чим ініційовано). Практично це означає, що кожен workflow має завершуватися не лише "дією", а й записом про неї: або в централізованому журналі, або у вигляді коментаря/нотатки до інциденту, або через тікет у ITSM. Саме аудит робить SOAR придатним для експлуатації: без нього автоматизація стає "чорною скринькою", яку складно розслідувати у випадку помилок.

3.3. Тестування та оптимізація роботи SOC

Практичне налаштування SOC не завершується на етапі "джерела підключені" та "правила виявлення створені". Робочий рівень починається там, де SOC здатен керовано перевіряти власні механізми виявлення й реагування, вимірювати їхню якість, а потім системно зменшувати хибні спрацювання, підвищувати покриття сценаріїв і контролювати продуктивність аналітичного рівня. Для цього потрібен замкнений контур: симуляція технік → формування телеметрії → спрацювання правила → інцидент → перевірка реакції (включно з SOAR) → аналіз результатів → корекція правил/джерел/процедур. "Детекція" і "підтвердження детекції" – різні задачі: перша створює оповіщення, друга доводить, що оповіщення виникає стабільно**,** вчасно і з коректними сутностями, а не випадково.

Базова вимога до симуляції – контрольованість. Тест має відтворювати поведінковий фрагмент противника, але в межах середовища, де ризик впливу керований: ізольована інфраструктура, відомі точки входу/виходу, узгоджені правила безпеки, а також можливість швидко розгортати й повертати початковий стан (на практиці це досягається через віртуалізацію та стандартизовані шаблони середовища). Такий підхід відповідає загальній логіці adversary emulation: сценарій готується, емулюється, результати розслідуються, після чого

оцінюється стан захисту та вноситься корекція – це не одноразова перевірка, а ітераційний процес якості захисту.

Керований тест краще будувати не “від інструмента”, а від сценарію. Практично зручно описувати сценарій як ланцюг коротких фаз, де кожна фаза дає очікувані артефакти в телеметрії. Наприклад, тестова послідовність може містити початкове сканування, першу атаку, встановлення з’єднання та спробу витоку даних; окремо як приклади сценаріїв у матеріалах розглядаються password guessing (як процес повторюваних спроб входу) та експлуатація сервісу з подальшим керуванням через C2.

Щоб перевірка була відтворюваною, сценарій одразу прив’язують до тактик/технік (MITRE ATT&CK) як до “каркаса покриття”: SOC визначає, які тактики/техніки він прагне ловити, а симуляція показує – чи реально зібрана телеметрія дозволяє їх бачити і чи правила виявлення дають спрацювання в очікуваній точці ланцюга. На практиці це перетворюється в конкретний артефакт: паспорт тест-кейсу (що запускаємо, які журнали мають з’явитися, яке правило має спрацювати, які сутності повинні бути витягнуті). Далі цей паспорт використовується в розслідуванні та оцінюванні ефективності захисту в межах циклу “симуляція → розслідування → оцінка”.

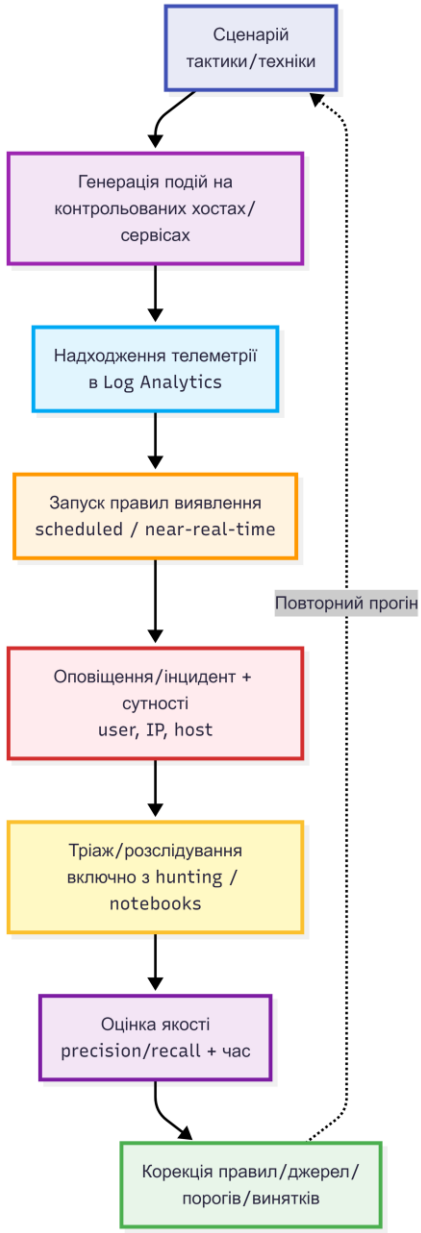

Рис. 3.12 Контур перевірки виявлення в SOC (симуляція → детекція → оцінка → покращення)

Для практичної перевірки важливо мати можливість побачити дані “в потоці” і одразу перевірити гіпотези запитами. У Microsoft Sentinel це робиться через hunting-підхід і використання notebooks для розширеного аналізу. У робочому процесі це виглядає так: одразу після запуску симуляції аналітик контролює, чи з’явились очікувані записи в потрібних таблицях, чи не “загубились” сутності (IP/користувач/хост), і чи можна побудувати стабільний запит, який виділить подію. Це принципово, бо правила виявлення – це вже

“упакований” запит із параметрами виконання, а hunting дозволяє швидко перевіряти логіку до того, як її винесуть у production.

Окремий практичний аспект у симуляції – перевірка інтеграційного контуру “SIEM → SOAR”. У матеріалах розглядається типовий сценарій реагування, де інцидент високої критичності приводить до автоматизованих дій: блокування IP на мережевому рівні (NSG), повідомлення команди, створення запису в ITSM; для цього playbook витягує IP із сутностей інциденту, створює або оновлює security rule та фіксує контекст у повідомленні/тикеті.

В Таблиці 3.7 зображено практичний варіант “паспортів” тестів, який зручно підставляти під будь-який інструмент симуляції (Atomic Red Team / Azure Attack Simulation / інші), не змінюючи структуру перевірки

Таблиця 3.7

Паспорт тест-кейсу для перевірки виявлення та реагування

Сценарій (фаза)	**Очікувана телеметрія (тип)**	**Точка перевірки в Sentinel**	**Очікуваний результат**
Password guessing (повторні невдалі входи)	Події невдалих входів (Windows Security Events / identity)	KQL у Log Analytics + scheduled-правило на поріг	Оповіщення/інцидент, сутність користувача, прив’язка часу вікна
Командна активність + мережевий виклик	Запуски процесів + мережеві підключення	Hunting-запит (процес + мережа)	Спрацювання правила або підтвердження, що даних достатньо для правила
Інцидент високої критичності (тест SOAR)	Будь-який запис, що формує інцидент	Scheduled query rule + увімкнена automation	Запуск playbook, блокування IP/сповіщення/реєстрація дії

Для швидкого тесту “в середовищі без даних” доцільно мати спеціальний тестовий запит, який гарантовано повертає один запис і тим самим примусово створює інцидент для перевірки ланцюга автоматизації. Це підхід “Alternative Quick Test”, де запит замінюється на datatable(...) [...] з фіктивними значеннями Account, IPAddress, TimeGenerated, після чого правило запускають знову й отримують інцидент із потрібними сутностями

```
datatable(Account:string, IPAddress:string)
[
```

```
  "testuser", "192.168.1.100"
]
| extend TimeGenerated = now()
```

Тестування правил виявлення в SOC – це не лише “чи є спрацювання”, а й чи правильно налаштовані параметри виконання, чи правило формує придатний до тріажу інцидент, і чи не створює воно зайвого шуму. Практичний підхід починається з контрольного правила, яке легко пояснити й перевірити. Наприклад, правило для виявлення грубого підбору паролів на базі Windows подій реалізується як scheduled-аналітичне правило з KQL, що відбирає EventID == 4625, агрегує кількість невдалих входів у часовому біні та порівнює з порогом

Критичний момент tuning – це узгодження трьох параметрів: частота запуску, вікно аналізу і поріг. Якщо частота запуску занадто висока, а вікно велике, SOC отримує дублікати й навантаження на аналітичний рівень; якщо вікно мале, а джерела мають затримки інжесту, події можуть “випасти” між запусками. Тому на практиці tuning роблять не приблизно, а через цикл: симуляція → вимір → зміна параметрів → повтор. У цьому циклі корисно оцінювати детекцію не лише “спрацювало/ні”, а через метрики якості. У матеріалах прямо використовується поділ на precision та recall (точність і повнота): precision показує частку коректних спрацювань серед усіх спрацювань, recall – частку виявлених подій серед усіх подій, які мали бути виявлені.

У Sentinel правильне тестування правила включає не лише запит, а й налаштування інцидентів та сутностей. Існує практичний варіант, коли на етапі налаштування правила додається “alert enrichment” із явним описом сутностей у JSON, щоб інцидент гарантовано містив IP-адресу як entity; це особливо корисно для тестового стенду або для перевірки playbook-ів, які очікують entity типу ip.

Після того як правило стабільно спрацьовує, tuning переходить у фазу “зменшення шуму”. У практичних SOC-сценаріях шум часто приходить як “фонові” події середовища: легітимні сканування, сервісні акаунти, регламентні завдання, адміністративні активності. У матеріалах це описується як проблема

background noise, через яку велика частина часу витрачається на false positives; як практичний приклад наведено підхід зменшення шуму через вимикання/переналаштування правил, що дають “порожні” або нецінні попередження (зокрема через фільтрацію за префіксами повідомлень), що якісно змінює роботу SOC.

У Sentinel аналогічна логіка реалізується через винятки в запитах, suppression, або через правила групування й маршрутизації (коли не все повинно ставати інцидентом однакового рівня).

Таблиця 3.8

Практичні важелі tuning для зниження хибних спрацювань

Джерело шуму	**Як проявляється в інцидентах**	**Практичний прийом tuning**
Сервісні/технічні акаунти	Порогові правила (brute force) спрацьовують на автоматизації	Виняток за Account / AccountUPN, окреме правило для сервісних акаунтів
Сканування/моніторинг	Підозрілі мережеві події “завжди”	Allowlist IP/підмереж, окремі пороги для internal/external
Надто загальна агрегація	Інциденти без придатних сутностей	Додати entity mapping/enrichment, уточнити поля агрегації
Дублювання спрацювань	Багато однакових інцидентів	Grouping / suppression, синхронізація частоти запуску і вікна

Доцільно визначити також SOC-правило експлуатації (policy), яке визначає, коли правило виявлення вважається “придатним”: наприклад, якщо правило протягом визначеного періоду має низький precision (переважно false positives), воно переводиться у режим спостереження (без створення інцидентів) або отримує обов’язкові винятки/уточнення; якщо правило має низький recall (пропускає контрольні тести), переглядається телеметрія та логіка агрегації. Саме так tuning стає керованим процесом, а не ручним “підкручуванням”.

Продуктивність SOC вимірюється не лише швидкістю людини, а й затримками системи: як швидко дані доходять, як швидко виконується кореляція, як швидко формується інцидент, і скільки часу минає до стримування/усунення. Операційна ефективність SOC оцінюється через пару

метрик MTTD (mean time to detect) та MTTR (mean time to respond), які фіксують середній час до виявлення і середній час до реагування. Ці показники є зручними для прив'язки до результатів тестів та об'єктивного вимірювання продуктивності системи моніторингу. У практичному сенсі MTTD “розкладається” на: затримку надходження телеметрії + затримку виконання правил (scheduled / near-real-time) + затримку тріажу. MTTR залежить від того, чи є SOAR-дії, наскільки вони автоматизовані, і чи інцидент містить потрібні сутності для виконання playbook-ів.

На рівні SIEM важливо пам’ятати, що кореляція – це обчислювально інтенсивний процес: аналітичний рівень повинен обробляти потоки подій у (квазі)реальному часі, а сам механізм кореляції часто описують як послідовність фаз (збирання/нормалізація → агрегація/кореляція → створення результату), де найбільше навантаження припадає на етапи агрегації та кореляції. Питання продуктивності прямо пов’язане з тим, скільки правил, які саме вікна аналізу, які об’єми даних і яка складність запитів

Щоб оптимізація була практичною, варто вести “паспорт навантаження” для аналітичного рівня: які правила найчастіше запускаються, які дають найбільше інцидентів, які запити найдорожчі, де виникає затримка. Це добре поєднується з контуром тестування: контрольні сценарії запускаються регулярно, а SOC фіксує не лише факт спрацювання, а й часові характеристики. Для цього зручно використовувати ідею вимірювання продуктивності через часові інтервали та метрики якості, де час – окремий ключовий індикатор ефективності.

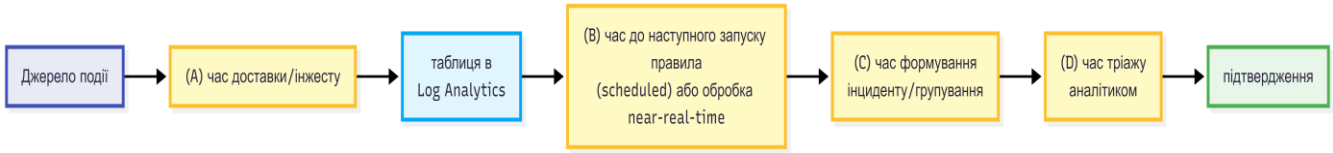

Рис. 3.13 Декомпозиція затримки виявлення (MTTD) у практичному SOC

Оптимізація працює тоді, коли SOC може зменшити A, B або D (автоматизацією та якістю інциденту), а не лише “підкручувати поріг”.

Розвиток SOC логічно оформлювати як безперервний цикл удосконалення. На практиці це означає: після кожного тесту або реального інциденту SOC уточнює правила виявлення, коригує винятки, додає потрібні джерела, а також розширює автоматизацію реагування там, де інцидент вже стабільно точний. У матеріалах наведений приклад “інтегрованого” реагування, коли Sentinel і автоматизація використовуються для блокування підозрілої IP-адреси на мережевому рівні та подальших дій обробки/збагачення; такий кейс добре застосовувати як контрольний для перевірки повного ланцюга “виявлення → стримування → фіксація”.

Таблиця 3.9

Мінімальний набір контрольних показників для “здоров’я” SOC у тестах

Показник	**Як отримується практично**	**Навіщо потрібен**
MTTD	Час від події сценарію до створення інциденту	Виявляє проблеми інжесту/частоти правил/кореляції
MTTR	Час від інциденту до виконання дії (SOAR/ручної)	Показує ефект автоматизації та якості інциденту
Precision / Recall	Розмітка результатів контрольних прогонів	Вимірює баланс “шум/покриття” і напрям tuning
Частка шумних правил	Частота інцидентів без підтвердження	Дозволяє керовано вимикати/уточнювати джерела шуму

Окремо для практики дуже корисно мати “механізм тестування без реальних даних”. Тестування функціональності SOC у Microsoft Sentinel реалізується через створення тестових інцидентів за допомогою scheduled query rule. За відсутності реальних подій безпеки використовується конструкція datatable(...), яка примусово генерує інцидент і запускає відповідні playbook-и. Такий підхід забезпечує регулярну перевірку працездатності всього ланцюга автоматизації без очікування реальних атак

Саме так оптимізація перестає бути “раз на пів року”, а переходить у режим постійної перевірки: сценарій прогнали → інцидент створився → playbook відпрацював → результати зафіксовані → параметри скориговані → сценарій повторили.

Розглянемо сценарій інциденту SOC відповідно до складеної моделі.

1. Паспорт тест-кейсу (SOC Test Case)

Назва сценарію: Password Guessing / Brute-force Authentication

Мета: перевірка здатності SOC виявляти повторювані невдалі спроби автентифікації в обмеженому часовому вікні та коректно ініціювати реагування

Рівень: Identity / Endpoint

Тип перевірки: керована симуляція + аналітичне тестування

Таблиця 3.10

Паспорт тест-кейсу

Поле	Значення
Тактика (MITRE ATT&CK)	Credential Access
Техніка	Brute Force
Джерело телеметрії	Windows Security Events / Identity logs
Ключові події	Невдалі спроби входу
Очікуваний результат	Інцидент високої/середньої критичності
Задіяні сутності	Account, IP address
Тип реагування	Сповіщення → (опційно) блокування

2. Модель подій сценарію

Сценарій password guessing у SOC **не виявляється однією подією**, а лише через **кореляцію в часі**.

Очікувана поведінка в телеметрії

- кілька невдалих спроб входу;
- однаковий обліковий запис або одна IP-адреса;
- короткий часовий інтервал;
- відсутність успішної автентифікації між спробами.

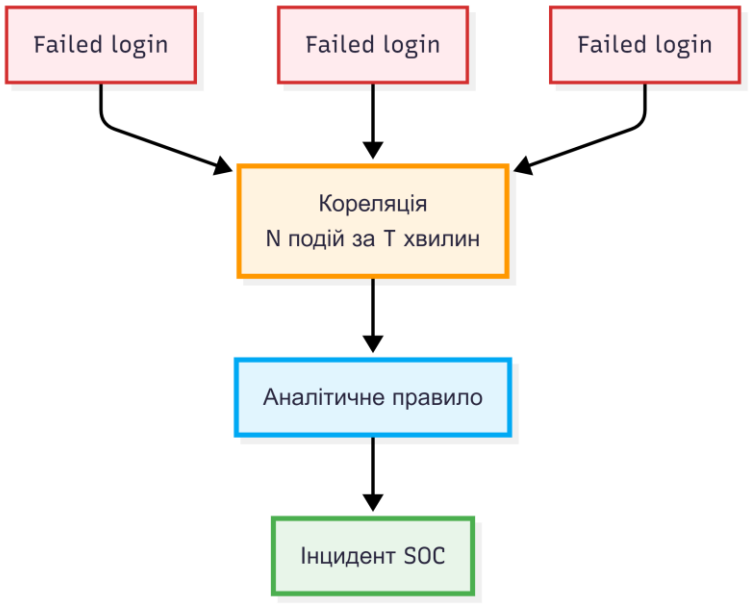

Рис. 3.14 Логіка сценарію

3. Аналітичне правило Microsoft Sentinel (повноцінне правило, не просто KQL)

Тип правила

- **Scheduled analytics rule**
- Причина: кореляція за часовим вікном

Таблиця 3.11

Основні параметри правила

Параметр	Значення
Частота запуску	кожні 5 хвилин
Вікно аналізу	15 хвилин
Поріг	> 5 невдалих входів
Групування	за Account + IP
Створення інциденту	увімкнено

KQL-запит правила

```
SecurityEvent
| where EventID == 4625
| where TimeGenerated > ago(15m)
| summarize FailedAttempts = count()
        by TargetUserName, IpAddress, bin(TimeGenerated, 5m)
```

```
| where FailedAttempts > 5
```

Рис. 3.15 Візуалізація запиту до правила

Entity mapping

Account → TargetUserName

IP → IpAddress

4. Тестування правила (без реальної атаки)

Контрольний тест через тестовий запит

Для перевірки всього ланцюга (правило → інцидент → SOAR) використовується примусовий тестовий інцидент.

```
datatable(TargetUserName:string, IpAddress:string)
[
  "testuser", "192.168.1.100"
]
| extend TimeGenerated = now()
| summarize FailedAttempts = count()
        by TargetUserName, IpAddress, bin(TimeGenerated, 5m)
| where FailedAttempts > 0
```

5. Критерії успіху тесту (SOC Acceptance Criteria)

Таблиця 3.12

Критерії коректної роботи

Критерій	Очікувана поведінка
Створення інциденту	≤ 1 цикл виконання правила
Критичність	Medium або High
Наявність сутностей	Account + IP присутні
Дублювання	Інциденти групуються
Запуск SOAR	Playbook запускається

6. Матриця tuning (робоча SOC-практика)

Таблиця 3.13

Матриця tuning password guessing

Проблема	Як проявляється	Причина	Рішення
Забагато false positives	Інциденти на сервісні акаунти	Легітимні автоматичні входи	Виняток для service accounts
Інциденти без IP	Неможливо блокувати	Відсутня IP у логах	Уточнити джерело телеметрії
Запізніле спрацювання	Інцидент через 30–40 хв	Надто велике вікно	Зменшити window до 10–15 хв
Дублікати інцидентів	10+ однакових кейсів	Немає grouping	Увімкнути grouping за Account+IP
Не спрацьовує	Тест проходить, інциденту немає	Затримка ingestion	Збільшити lookback або частоту

7. Зв’язок з SOAR (реальна експлуатація)

Після стабілізації правила можливі 3 рівні реагування:

1. Notification only: email / Teams для SOC;

2. Semi-automated containment: playbook → блокування IP після підтвердження;

3. Full automation (обмежено): лише для зовнішніх IP + High severity.

Висновки до розділу 3

У третьому розділі роботи виконано практичну реалізацію системи моніторингу та реагування SOC на базі Microsoft Sentinel з деталізацією всіх етапів технічного налаштування, автоматизації та тестування.

Розроблено послідовність розгортання SIEM-платформи, де робочий простір Log Analytics виступає центральним сховищем телеметрії, а Microsoft Sentinel додає функції аналітики, керування інцидентами та візуалізації. Реалізовано підключення ключових джерел даних через data connectors для доменів ідентичностей (Azure AD), подій операційних систем (SecurityEvent), мережевих пристроїв (Syslog/CEF) та хмарних сервісів, що формує мінімально життєздатний набір сигналів для операційної роботи SOC.

Створено аналітичні правила виявлення на основі KQL з налаштуванням параметрів планування (частота запуску, вікно аналізу), порогів спрацювання та прив'язки сутностей (entity mapping). Розроблено робочі книги (workbooks) для операційного моніторингу з візуалізацією невдалих входів у часі, топ IP-адрес джерел атак, активних інцидентів та географічного розподілу автентифікацій.

Реалізовано автоматизоване реагування через інтеграцію Automation rules та Logic Apps playbooks, що забезпечує формалізований механізм виконання повторюваних дій: повідомлення команди, збагачення контексту, стримування загроз (блокування IP через NSG) та створення тікетів у ITSM. Визначено розподіл відповідальності між SIEM-рівнем (виявлення та консолідація сигналів) та SOAR-рівнем (виконання workflows реагування) з акцентом на безпечність автоматизації через контрольні умови запуску та валідацію вхідних даних.

Розроблено методику тестування та оптимізації SOC через замкнений контур: симуляція технік → формування телеметрії → спрацювання правил → створення інцидентів → перевірка реагування → аналіз результатів → корекція параметрів. Формалізовано паспорти тест-кейсів з прив'язкою до MITRE ATT&CK та критеріями успіху перевірки.

Практично реалізовано повний сценарій виявлення та реагування на атаку типу password guessing, що включає scheduled analytics rule з кореляцією подій за часовим вікном, entity mapping для витягування облікових записів та IP-адрес, а також playbook для автоматизованого блокування джерел атак. Визначено метрики оцінювання операційної ефективності SOC через MTTD (mean time to detect) та MTTR (mean time to respond).

Розроблено практичні важелі tuning для зниження хибних спрацювань через винятки для сервісних акаунтів, allowlist IP-адрес, групування інцидентів та узгодження параметрів частоти запуску правил із затримками надходження телеметрії.

Запропонована методика забезпечує керований та відтворюваний процес налаштування, тестування та безперервного вдосконалення SOC з можливістю регулярної перевірки працездатності системи без очікування реальних атак.

ВИСНОВКИ

У розділі 1 було розглянуто теоретичні основи моніторингу та реагування в SOC як постійно діючої організаційної функції кіберзахисту. SOC визначено як інтегровану систему, що поєднує людей, процеси та технології для безперервного спостереження за інфраструктурою, виявлення загроз і координації реагування. Окреслено підходи до організації SOC (внутрішній, віртуальний, глобальний, хмарний і гібридний) та показано, що модель розміщення визначається масштабом підприємства, критичністю активів і зрілістю процесів безпеки. Також розглянуто понятійний апарат реагування, зокрема розмежування загроз, подій та інцидентів, а цикл управління інцидентами описано як послідовність етапів ідентифікації, аналізу, стримування, усунення наслідків та удосконалення процедур. Додатково обґрунтовано роль IoC, кореляції та аналізу логів як бази для прийняття рішень. Завершальним елементом розділу стало дослідження технологічної бази SOC, де показано взаємодоповнюваність SIEM, XDR та SOAR у формуванні замкненого контуру моніторингу й реагування.

У розділі 2 було розроблено архітектуру системи моніторингу SOC та практичну модель виявлення загроз на основі інтеграції SIEM, XDR і суміжних компонентів. Обґрунтовано необхідність формування цільової операційної моделі SOC перед вибором інструментів і визначено доцільність опису архітектури у трьох взаємопов’язаних поданнях: логічному, функціональному та мережевому. Побудовано узагальнений багатошаровий конвеєр подій, що охоплює джерела телеметрії, збір і транспорт, нормалізацію та сховище, аналітику й кореляцію, інцидент-менеджмент та автоматизацію реагування, а також сформульовано вимоги до кожного рівня з урахуванням продуктивності індексування та кореляційних механізмів. Визначено принципи інтеграції SIEM і XDR, де SIEM відповідає за централізацію, нормалізацію та кореляцію, а XDR – за міждоменне зв’язування сигналів і прискорення реагування в окремих доменах. На практичному рівні реалізовано модель виявлення в Microsoft

Sentinel із застосуванням KQL: розроблено правило для виявлення аномальної активності доступу до Microsoft Graph API та контекстні запити для збагачення інцидентів даними про ролі та групи користувачів. У результаті отримано модульну й масштабовану архітектуру, придатну до адаптації під зміни ландшафту загроз.

У розділі 3 було виконано практичну реалізацію системи моніторингу та реагування SOC на базі Microsoft Sentinel із деталізацією налаштувань, автоматизації та тестування. Реалізовано розгортання SIEM-рівня через Log Analytics як сховище телеметрії та Sentinel як надбудову аналітики й керування інцидентами, а також підключено ключові джерела даних через data connectors (Azure AD, SecurityEvent, Syslog/CEF, хмарні сервіси). Налаштовано аналітичні правила на основі KQL із параметрами планування, порогами спрацювання та entity mapping, і створено workbooks для операційного моніторингу. Реалізовано реагування через Automation rules та Logic Apps playbooks із формалізованими діями збагачення, оповіщення, стримування (зокрема блокування IP) та інтеграції з ITSM, із визначенням безпечних умов запуску та валідації. Розроблено методику тестування та оптимізації на основі замкненого контуру «симуляція → телеметрія → спрацювання → інцидент → реагування → аналіз → корекція» та паспорти тест-кейсів із прив'язкою до MITRE ATT&CK. Практично відпрацьовано сценарій password guessing, визначено критерії успіху та підхід до оцінювання ефективності через MTTD і MTTR, а також наведено механізми tuning для зниження хибних спрацювань і підвищення стабільності роботи правил.

Отже, у кваліфікаційній роботі послідовно обґрунтовано теоретичні засади SOC, розроблено цілісну архітектурну модель системи моніторингу та реалізовано практичний прототип на базі Microsoft Sentinel із автоматизованим реагуванням і методикою відтворюваного тестування та вдосконалення, що забезпечує керовану побудову SOC і практичну придатність результатів для операційного використання.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Security Operations Center: A Systematic Study and Open Challenges / M. Vielberth et al. *IEEE Access*. 2020. Vol. 8. P. 227756–227779. URL: https://doi.org/10.1109/access.2020.3045514

2. Muhammad R., Ismail S. A., Hassan N. H. Botnet Detection and Incident Response in Security Operation Center (SOC): A Proposed Framework. *International Journal of Advanced Computer Science and Applications*. 2024. Vol. 15, no. 3. URL: https://doi.org/10.14569/ijacsa.2024.0150389

3. Empowering Security Operation Center with Artificial Intelligence and Machine Learning – A Systematic Literature Review / M. Khayat et al. *IEEE Access*. 2025. P. 1. URL: https://doi.org/10.1109/access.2025.3532951

4. Alert Prioritisation in Security Operations Centres: A Systematic Survey on Criteria and Methods / F. Jalalvand et al. *ACM Computing Surveys*. 2024. URL: https://doi.org/10.1145/3695462

5. Toward Robust Security Orchestration and Automated Response in Security Operations Centers with a Hyper-Automation Approach Using Agentic Artificial Intelligence / Ismail et al. *Information*. 2025. Vol. 16, no. 5. P. 365. URL: https://doi.org/10.3390/info16050365

6. Enhancing Security Operations Center: Wazuh Security Event Response with Retrieval-Augmented-Generation-Driven Copilot / Ismail et al. *Sensors*. 2025. Vol. 25, no. 3. P. 870. URL: https://doi.org/10.3390/s25030870

7. Ofte H. J. The awareness of operators: a goal-directed task analysis in SOCs for critical infrastructure. *International Journal of Information Security*. 2024. URL: https://doi.org/10.1007/s10207-024-00872-6

8. Alert Fatigue in Security Operations Centres: Research Challenges and Opportunities / S. Tariq et al. *ACM Computing Surveys*. 2025. URL: https://doi.org/10.1145/3723158

9. The Next Generation Cognitive Security Operations Center: Adaptive Analytic Lambda Architecture for Efficient Defense against Adversarial Attacks /

K. Demertzis et al. *Big Data and Cognitive Computing*. 2019. Vol. 3, no. 1. P. 6. URL: https://doi.org/10.3390/bdcc3010006

10. Towards Human-AI Teaming to Mitigate Alert Fatigue in Security Operations Centres / M. Baruwal Chhetri et al. *ACM Transactions on Internet Technology*. 2024. URL: https://doi.org/10.1145/3670009

11. Chamkar S. A., Maleh Y., Gherabi N. Security Operations Centers: Use Case Best Practices, Coverage, and Gap Analysis Based on MITRE Adversarial Tactics, Techniques, and Common Knowledge. *Journal of Cybersecurity and Privacy*. 2024. Vol. 4, no. 4. P. 777–793. URL: https://doi.org/10.3390/jcp4040036

12. A comprehensive security operation center based on big data analytics and threat intelligence / J. Wang et al. *International Symposium on Grids & Clouds 2021*, Academia Sinica Computing Centre (ASGC), Taipei, Taiwan Website: https://indico4.twgrid.org/indico/event/14/overview, 22 March 2021. Trieste, Italy, 2021. URL: https://doi.org/10.22323/1.378.0028

13. Building a Better SOC: Towards the Ontology for Security Operations Center Assistance and Replication (OSCAR) / J. M. Novak et al. *Digital Threats: Research and Practice*. 2025. URL: https://doi.org/10.1145/3722233

14. Danquah P. Security Operations Center: A Framework for Automated Triage, Containment and Escalation. *Journal of Information Security*. 2020. Vol. 11, no. 04. P. 225–240. URL: https://doi.org/10.4236/jis.2020.114015

15. Autonomous Agentic AI Architectures for Optimizing Security Operations Centers (SOC) KPIS: Methodology, Impact on Detection, Response, and Recovery / M. Stefanov et al. *Land Forces Academy Review*. 2025. Vol. 30, no. 3. P. 479–493. URL: https://doi.org/10.2478/raft-2025-0046

16. Bassey C., Chinda E. T., Idowu S. Building a Scalable Security Operations Center: A Focus on Open-source Tools. *Journal of Engineering Research and Reports*. 2024. Vol. 26, no. 7. P. 196–209. URL: https://doi.org/10.9734/jerr/2024/v26i71203

17. Python-Based Security Operations Center (SOC) and Forensics Analysis for Incident Cyber Threats / D. P. Muthusamy et al. *International Journal for Research*

in Applied Science and Engineering Technology. 2024. Vol. 12, no. 4. P. 2592–2596. URL: https://doi.org/10.22214/ijraset.2024.60403

18. Lysetsky Y. M., Bobrov S. I. Security Operation System. *Mathematical machines and systems*. 2020. Vol. 2. P. 51–59. URL: https://doi.org/10.34121/1028-9763-2020-2-51-59

19. Building a Better SOC: Towards the Ontology for Security Operations Center Assistance and Replication (OSCAR) / J. M. Novak et al. *Digital Threats: Research and Practice*. 2025. URL: https://doi.org/10.1145/3722233

20. Danquah P. Security Operations Center: A Framework for Automated Triage, Containment and Escalation. *Journal of Information Security*. 2020. Vol. 11, no. 04. P. 225–240. URL: https://doi.org/10.4236/jis.2020.114015

21. Autonomous Agentic AI Architectures for Optimizing Security Operations Centers (SOC) KPIS: Methodology, Impact on Detection, Response, and Recovery / M. Stefanov et al. *Land Forces Academy Review*. 2025. Vol. 30, no. 3. P. 479–493. URL: https://doi.org/10.2478/raft-2025-0046

22. Bassey C., Chinda E. T., Idowu S. Building a Scalable Security Operations Center: A Focus on Open-source Tools. *Journal of Engineering Research and Reports*. 2024. Vol. 26, no. 7. P. 196–209. URL: https://doi.org/10.9734/jerr/2024/v26i71203

23. Sujan Kumar K. Next-Generation Security Operations: Leveraging Automation for Proactive Threat Mitigation. *INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*. 2025. Vol. 09, no. 04. P. 1–9. URL: https://doi.org/10.55041/ijsrem43432

24. Kilincdemir E. C., Celiktas B. Analyst-Aware Incident Assignment in Security Operations Centers: A Multi-Factor Prioritization and Optimization Framework. *Black Sea Journal of Engineering and Science*. 2025. Vol. 8, no. 4. P. 45–46. URL: https://doi.org/10.34248/bsengineering.1693042

25. The Next Generation Cognitive Security Operations Center: Network Flow Forensics Using Cybersecurity Intelligence / K. Demertzis et al. *Big Data and*

Cognitive Computing. 2018. Vol. 2, no. 4. P. 35. URL: https://doi.org/10.3390/bdcc2040035

26. Taqafi I., Maleh Y., Ouazzane K. A MATURITY CAPABILITY FRAMEWORK FOR SECURITY OPERATION CENTER. *EDPACS*. 2022. P. 1–18. URL: https://doi.org/10.1080/07366981.2023.2159047

27. Shahjee D., Ware N. Integrated Network and Security Operation Center: A Systematic Analysis. *IEEE Access*. 2022. Vol. 10. P. 27881–27898. URL: https://doi.org/10.1109/access.2022.3157738

28. Abd Majid M., Zainol Ariffin K. A. Model for successful development and implementation of Cyber Security Operations Centre (SOC). *PLOS ONE*. 2021. Vol. 16, no. 11. P. e0260157. URL: https://doi.org/10.1371/journal.pone.0260157

29. Security Operation Center Methodology for 5G Networks / M. Orsós et al. *Acta Polytechnica Hungarica*. 2025. Vol. 22, no. 2. P. 99–121. URL: https://doi.org/10.12700/aph.22.2.2025.2.6

30. Tilbury J., Flowerday S. Humans and Automation: Augmenting Security Operation Centers. *Journal of Cybersecurity and Privacy*. 2024. Vol. 4, no. 3. P. 388–409. URL: https://doi.org/10.3390/jcp4030020

31. Akshai Sankar N., Fasila K. A. Implementation of SOC using ELK with Integration of Wazuh and Dedicated File Integrity Monitoring. *2023 9th International Conference on Smart Computing and Communications (ICSCC)*, Kochi, Kerala, India, 17–19 August 2023. 2023. URL: https://doi.org/10.1109/icscc59169.2023.10334992

32. Lytvynov V. A., Oksanych I. M., Hrybkov S. V. Regarding the possibilities of implementing the joint functioning of LMS and SIEM manage-ment based on free open-source tools. *Mathematical machines and systems*. 2025. Vol. 1. P. 55–63. URL: https://doi.org/10.34121/1028-9763-2025-1-55-63

33. Aquisição da prestação de serviços de Security Operation Center (SOC) : Anúncio de procedimento no. 16172/2025. *Diário da República II Série*. 2025. 17 June. URL: https://files.diariodarepublica.pt/cp_hora/2025/06/115/419193789.pdf

34. Sujan Kumar K. Next-Generation Security Operations: Leveraging Automation for Proactive Threat Mitigation. *INTERNATIONAL JOURNAL OF*

SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT. 2025. Vol. 09, no. 04. P. 1–9. URL: https://doi.org/10.55041/ijsrem43432

35. “Configuracao e aplicacao de um soc (security operation center) gratuito: um estudo pratico para aplicacao do siem wazuh,” Feb. 2023, doi: 10.5748/19contecsi/pse/sec/7103

36. A. I. Hajamydeen, M. D. Hasni, and M. I. Abdullah, “Integrating Wazuh for Efficient Real-Time Threat Monitoring and Vulnerability Assessment in a SOC Environment”, doi: 10.4018/979-8-3693-2814-9.ch013

37. M. Tashfeen, “Building blocks of incident response: Security operation centers”, doi: 10.1063/5.0148860

38. R. R. Bhavsar and V. Thakar, “Design and Implementation of an Open-Source Security Operations Center for Effective Cyber Threat Detection and Response,” Jan. 2025, doi: 10.21203/rs.3.rs-5795888/v1

39. P. Najafi, F. Cheng, and C. Meinel, “SIEMA: Bringing Advanced Analytics to Legacy Security Information and Event Management,” pp. 25–43, Sept. 2021, doi: 10.1007/978-3-030-90019-9_2

40. N. Peesara, “The Strategic Fusion of SIEM and SOAR in Modern SOCs”. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5664250