Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:
В.о. завідувача кафедри
кібербезпеки
та захисту інформації
_____ Іван ПАРХОМЕНКО
«__» _____ 2025 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи

галузь знань *12 Інформаційні технології*
(шифр і назва галузі знань)

спеціальність *125 Кібербезпека та захист інформації*
(код і назва спеціальності)

освітній ступень *магістр*

освітньо-наукова програма *Кібербезпека*
(назва освітньої програми)

на тему: «Метод автоматизації процесів реагування на інциденти»

Виконавець: студента ІІ курсу, групи КБм-21

_____ **Данило ТАБАЧЕНКО**
(підпис) (Ім'я, ПРІЗВИЩЕ)

	Ім'я, ПРІЗВИЩЕ	**Підпис**
Науковий керівник	Лариса МИРУТЕНКО	
Нормоконтроль	Юрій БАБЕНКО	

Київ 2025

Міністерство освіти і науки України

Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
В.о. завідувача кафедри
кібербезпеки
та захисту інформації
_____ Іван ПАРХОМЕНКО
«25» жовтня 2024 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи

спеціальності *125 Кібербезпека та захист інформації*
(код і назва спеціальності)

освітній ступень *магістр*

Здобувача(ки) КБм-21 (група) Табаченка Данила Олексійовича (прізвище ім'я по-батькові)

Тема кваліфікаційної роботи Метод автоматизації процесів реагування на інциденти

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Рішення засідання кафедри кібербезпеки та захисту інформації факультету інформаційних технологій протокол № 4 від 24.10.2024 р.

2. МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень Процес реагування на інциденти.

Предмет досліджень механізми реагування на інциденти ІБ.

Мета розробка методу автоматизації процесів реагування на інциденти інформаційної безпеки на основі штучного інтелекту.

Вихідні дані для проведення роботи Методи та механізми автоматизації реагування на інциденти

3. ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Наукова новизна запропоновано використання методу автоматизованого реагування на інциденти на основі штучного інтелекту.

Практична цінність забезпечення автоматизованого реагування на інциденти за допомогою штучного інтелекту.

4. ЕТАПИ ВИКОНАННЯ РОБОТИ

Найменування етапів робіт	Строки виконання робіт (початок-кінець)
Уточнення постановки задачі	25.10.2024 – 01.12.2024
Аналіз літературних джерел	01.12.2024 –15.01.2025
Використання Threat Intelligence у IR	15.01.2025 – 30.01.2025
Активне виявлення загроз з Threat Hunting	30.01.2025 – 15.02.2025
Аналіз впливу моделювання загроз та оцінки ризиків на процес IR	15.02.2025 – 24.02.2025
Аналіз сучасних інструментів автоматизації реагування на інциденти	24.02.2025 – 03.03.2025
Розробка методу автоматизованого реагування на інциденти	03.03.2025 – 06.03.2025
Реалізація програмного застосунку	06.03.2025 – 01.05.2025
Аналіз переваг застосунку	01.05.2025 – 07.05.2025
Оцінка ефективності програмного застосунку	07.05.2025 – 13.05.2025
Оформлення пояснювальної записки згідно методичних рекомендацій	13.05.2025– 18.05.2025
Подача пакету документів на розгляд ЕК	19.05.2025

Завдання видала _____ Лариса МИРУТЕНКО
(підпис) (Ім’я, ПРІЗВИЩЕ)

Завдання прийняв до виконання _____ Данило ТАБАЧЕНКО
(підпис) (Ім’я, ПРІЗВИЩЕ)

Дата видачі завдання: 25.10.2024 р.

Термін подання кваліфікаційної роботи до ЕК 19.05.2025 р.

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Метод автоматизації процесів реагування на інциденти»: 86 сторінок, 23 рисунки та 3 таблиці. 36 літературних джерел.

Об’єкт дослідження – процес реагування на інциденти.

Мета роботи - розробка методу автоматизації процесів реагування на інциденти інформаційної безпеки на основі штучного інтелекту.

Методи дослідження: аналіз відкритих джерел, аналіз систем захисту центру оперативного захисту підприємства, моделювання процесу автоматизації реагування на інциденти.

У роботі досліджено сучасні засоби підприємств для реагування інциденти. Проведено аналіз міжнародних стандартів з реагування на інциденти та наявні інструменти автоматизації, які пришвидшують вирішення інцидентів. Запропоновано метод для автоматизації реагування на інциденти на основі штучного інтелекту.

Наукова новизна: запропоновано використання методу автоматизованого реагування на інциденти на основі штучного інтелекту.

Актуальність теми: Кібератаки невпинно ускладнюються, ускладнюючи детектування і аналіз, що збільшує час на реагування. Традиційні методи реагування вимагають багато ручної роботи, що в свою чергу збільшує час вирішення інцидентів і несе більші фінансові втрати у випадках серйозних інцидентів. Поєднання технологій штучного інтелекту з наявними методами реагування дозволить організаціям звільнити значну кількість ресурсів, передаючи на аналіз події. Запропонований метод може використовуватися як для повністю автоматичного реагування на загрози, так і для допомоги спеціалістам з кібербезпеки в аналізі кіберзагроз.

Ключові слова: інциденти ІБ, захист від атак, методи реагування на кібератаки, кібератака, кіберзахист, AI, SOAR, SOC.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕННЬ

АС	–	автоматизована система
ЗІ	–	захист інформації
ІБ	–	інформаційна безпека
ООП	–	об'єктно-орієнтоване програмування
ПЗ	–	програмне забезпечення
СУІБ	–	система управління інформаційною безпекою
ШІ	–	штучний інтелект
AI	–	Artificial intelligence
API	–	Application Programming Interface
CISO	–	Chief Information Security Officer
CVE	–	Common Vulnerabilities and Exposures
EDR	–	Endpoint Detection and Response
IDS	–	Intrusion Detection System
IOA	–	Indicator of Attack
IOCs	–	Indicators of compromise
IR	–	Incident Response
IRT	–	Incident Response Team
IPS	–	Intrusion Prevention System
ML	–	Machine Learning
NIST	–	National Institute of Standards and Technology
SIEM	–	Security information and event management
SOAR	–	Security Orchestration, Automation and Response
SOC	–	Security Operations Center
SOCaaS	–	Security Operations Center as a service
TIP	–	Threat Intelligence Platform
TTPs	–	Tactics, Techniques, and Procedures
UEBA	–	User behavior analytics
XDR	–	Extended detection and response

ЗМІСТ

ВСТУП

З розвитком цифровізації сучасні технології проникли у всі сфери діяльності людини, майже будь-яка дія супроводжується використанням IoT приладів, а цінність інформації невпинно зростає. Зі збільшенням залученості технологій у наше життя збільшується кількість інформації, яку збирають про людей. Всі ці фактори призводять до збільшення кіберзагроз, які становлять все більшу небезпеку, особливо для сектору освіти, держави, охорони здоров'я через можливий доступ до персональних даних. За звітом Check Point Research, у 2024 році відбулося 38% зростання глобальних кібератак порівняно з попереднім роком. Кожного тижня організації захищаються від 1564 атак, відповідно до цих цифр проблеми кіберзахисту та розробки новітніх рішень стають все дедалі очевиднішими.

Центр операційної безпеки є одним з ключових відділів безпеки сучасного підприємства. Безперервний процес роботи дозволяє здійснювати моніторинг на реагувати на загрози в найкоротші терміни незалежно від часу атаки, що дозволяє зупинити атаку до того як атакуючий отримає доступ до критичних даних. Основне завдання будь-якої команди є своєчасне реагування, треба зупинити атаку до того, як зловмисник отримає доступ до критичних даних і зможе вплинути на доступність системи. Зростаюча кількість і складність кібератак змушує спеціалістів кібербезпеки висувати нові вимоги до систем безпеки. Таким чином, для ефективної роботи SOC вже недостатньо просто мати інструменти для виявлення кібератак з ручним реагуванням аналітика.

Один з основних напрямків розвитку у сфері кібербезпеки - це автоматизація процесів реагування на інциденти, оскільки вона дозволяє значно прискорити виявлення, аналіз та усунення інцидентів, зменшуючи їхній вплив на підприємства. Швидкість реагування безпосередньо впливає на обсяг збитків для організації; тож кожна хвилина тепер надзвичайно важлива. У ситуаціях, коли зловмисник має доступ до мережі будь-яка затримка, людський чинник або невизначеність можуть призвести до збільшення розміру збитків, оскільки саме у цей час може статися

витік даних або подальше просування по мережі. Саме тому автоматизація я є ключовим елементом покращення ефективності та одним із найсуттєвіших напрямків розвитку SOC. Створення сценаріїв для реагування на типові інциденти разом з використанням аналітичних інструментів допомагають ефективніше виявляти різні атак, скорочують час виявлення загроз та їх ліквідацію. Коректно налаштовані автоматизовані процеси можуть помітно знизити ризики й максимально зменшити фінансові втрати.

Тож актуальність роботи полягає в розробці методу автоматизованого реагування на інциденти ІБ на основі ШІ для прискорення швидкості реагування на інциденти.

Метою роботи є розробка методу автоматизації процесів реагування на інциденти інформаційної безпеки на основі штучного інтелекту.

Для досягнення зазначеної мети кваліфікаційної роботи поставлено наступні завдання:

- Дослідити нормативно правову бази в сфері реагування на інциденти;
- дослідити сучасні підходи реагування на інциденти;
- розробити метод автоматизованого реагування на інциденти інформаційної безпеки на основі штучного інтелекту;
- реалізувати програмний застосунок для автоматизованого реагування на інциденти на основі розробленого методу.

Об’єктом дослідження є процес реагування на інциденти.

Предметом дослідження є механізми реагування на інциденти ІБ.

Методи дослідження:

- аналіз відкритих джерел;
- аналіз систем захисту центру оперативного захисту підприємства;
- моделювання системи автоматизації реагування на інциденти.

Практичною цінністю є забезпечення автоматизованого реагування на інциденти за допомогою штучного інтелекту

РОЗДІЛ 1
МІЖНАРОДНІ СТАНДАРТИ З РЕАГУВАННЯ НА ІНЦИДЕНТИ ІБ

1.1 Міжнародні стандарти ISO 27 серії

Сімейство стандартів ISO 27000 це набір взаємодоповнюючих стандартів інформаційної безпеки, які об'єднані для забезпечення всесвітньо визнаної системи найкращих практик управління інформаційною безпекою. Стандарти мають важливе значення у сфері кібербезпеки. Вони забезпечують узгоджений підхід до побудови ефективних систем захисту інформації, сприяють дотриманню нормативних вимог і підвищують довіру до організацій з боку клієнтів, партнерів та регуляторів. Інцидентам же присвячено декілька пунктів обов’язкових контролів ISO 27001, ISO 27002 та цілий нормативний документ ISO 27035.

ISO/IEC 27001:2022 регламентує створення, впровадження та вдосконалення системи управління інформаційною безпекою (СУІБ), яка допомагає організаціям захищати свої дані від загроз, таких як несанкціонований доступ, викрадення або знищення. Стандарт, що дозволяє структуровано підходити до аналізу ризиків, їхнього зменшення та управління інформаційними активами.

ISO/IEC 27002:2022 є доповненням до 27001 та регламентує впровадження конкретних заходів з безпеки, надаючи деталізовані інструкції щодо їхньої реалізації. Структура стандарту наведена на рисунку 1.

Загалом стандарт містить такі контролі:

- організаційні – 37;
- пов’язані з людиною – 8;
- фізична безпека – 14;
- технологічні – 34

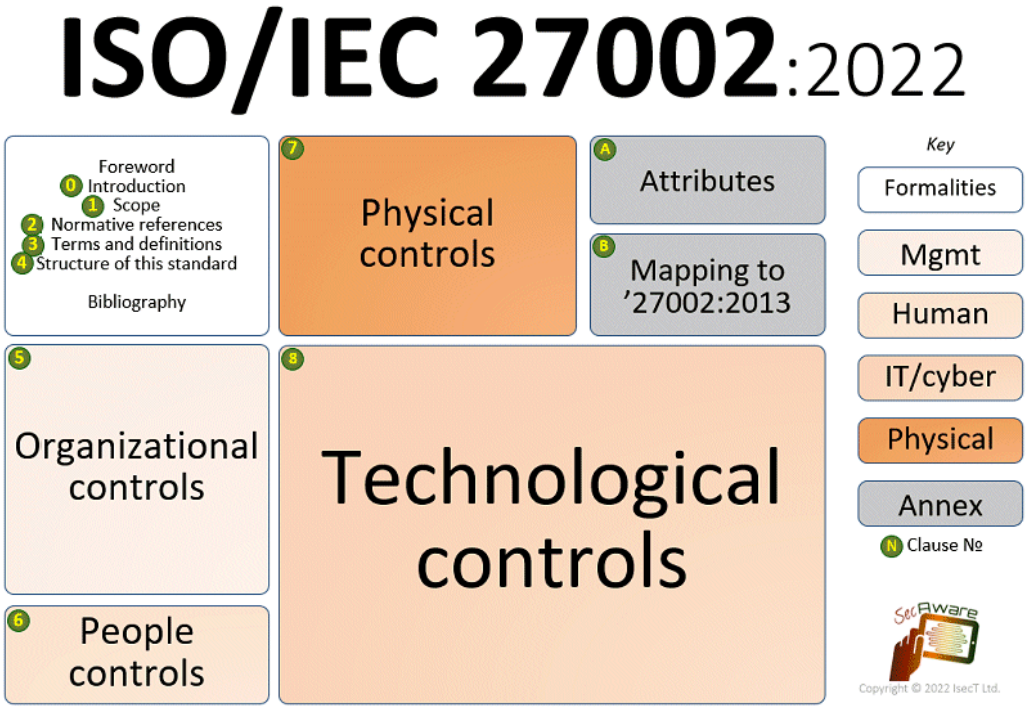

Рисунок 1.1. Структура ISO 27002:2022

Для ефективного процесу реагування на інциденти в стандарті виділено 5 організаційних контролів та один пов'язаний з людьми.

5.24 Керування плануванням та підготовкою до інцидентів інформаційної безпеки.

Управління інцидентами інформаційної безпеки має виконуватися шляхом планування та підготовки. Організація має визначити, і впровадити процеси, ролі та обов'язки з управління інцидентами інформаційної безпеки.

При чому організація має розглянути та впровадити:

- спосіб повідомлення про події інформаційної безпеки;
- процес управління інцидентами інформаційної безпеки, а саме: адміністрування, документування, виявлення, сортування, метод визначення пріоритетів, аналіз, комунікація та координація залучених осіб;
- процес реагування на інциденти та отримання уроків від них;
- допуск лише компетентного персоналу до інцидентів. Персонал має бути забезпечений процедурною документацією та постійно проходити навчання з підвищення кваліфікації;

Контроль також визначає цілі управління інцидентами інформаційної безпеки, вони мають бути узгодженими, а всі залучені особи до процесу вирішення інцидентів мають розуміти пріоритети організації. Має бути створений план управління інцидентами, який покриває різні сценарії реагування, та процедури реагування на інциденти, які визначають:

- логування процесів керування інцидентами;
- процес збору доказів;
- процес оцінки інцидентів на основі отриманих даних;
- моніторинг, детектування, класифікацію, аналіз та повідомлення про інциденти;
- процес реагування і ескалації інцидентів;
- шляхи координації з зацікавленими та залученими до вирішення інцидентів особами;
- аналіз інцидентів та впровадження покращень, змін у системах, політиках, процедурах, які необхідні задля недопущення повторення інциденту;

Процедури повідомлення про інциденти мають включати:

- дії у разі виникнення події інформаційної безпеки. Можуть включати в себе негайне реагування на потенційний інцидент або повідомлення контактних осіб;
- заздалегідь створені форми інцидентів, які користувачі мають використовувати для повідомлення про інцидент, що забезпечить виконання всіх дій при повідомлені про інцидент (див. додаток Б);
- створення процесу зворотного зв’язку з особами, які повідомили про інцидент;
- створення звітів про інцидент.

5.25 Оцінка та прийняття рішень щодо подій у сфері інформаційної безпеки.

Згідно з контролем організація має перевіряти події інформаційної безпеки для ідентифікації потенційних інцидентів. Для цього має бути створена ефективна категоризація та пріоритезацію. Персонал, який відповідає за інциденти має

провести аналіз і зробити свій експертний висновок, а всі результати перевірки мають бути записані.

5.27 Навчання через інциденти

Всі знання отримані протягом інциденту інформаційної безпеки мусять бути використані для отримання більш захищеної інформаційної системи. Знання можуть бути використані задля вдосконалення політик, процедур, а також для виявлення слабких місць в системі, які необхідно виправити або запровадити додатковий контроль задля вчасного реагування на потенційний інцидент.

5.28 Збір доказів

Контроль визначає ефективне управління доказами, пов'язаними з інцидентами інформаційної безпеки для дисциплінарних та юридичних дій. Організації мають розробити внутрішні процедури зі збору доказів. При чому великий акцент зроблений на збір згідно з національними законодавствами задля подальшого використання цих доказів у юридичних установах. Задля цього необхідно впевнитися, шо докази є точними і не були

підроблені, інформаційна система з якої брали докази працювала коректно під час збору доказів.

5.29 Інформаційна безпека під час збоїв

Пункт визначає, що організація має підтримувати свою безпеку під час збоїв інформаційних систем, для цього необхідно створити аналоги контролів. Також необхідно розробити, впровадити, тестувати, переглядати та оцінювати спеціальні плани для підтримки та відновлення критичних бізнес процесів після переривання або збою роботи систем.

Саме тому організація має впровадити та підтримувати:

- плани забезпечення безперервної роботи інформаційної системи;
- процеси для забезпечення функціонування наявних засобів контролю інформаційної безпеки у разі збоїв;
- альтернативні заходи контролю для забезпечення інформаційної безпеки в разі неможливості підтримки основних засобів під час збоїв.

6.8 Звітування про події інформаційної безпеки

Організація має створити механізм за допомогою якого користувачі зможуть своєчасно повідомляти про події інформаційної безпеки, які він спостерігає або підозрює. Механізм повідомлення про інциденти має бути легкий, доступний та зрозумілий. Для цього весь персонал має бути проінструктований про свій обов'язок повідомляти про інциденти, механізми повідомлення про інцидент та події про які необхідно повідомляти:

- неефективні контролі інформаційної безпеки;
- порушення конфіденційності, цілісності та доступності інформації;
- людські помилки;
- недотримання політик інформаційної безпеки;
- порушення фізичної безпеки;
- несправності або аномальна поведінка інформаційних систем;
- порушення доступу;
- вразливості;
- підозри на зараження інформаційної системи шкідливим програмним забезпеченням.

Стандарт ISO/IEC 27035 – управління інцидентами інформаційної безпеки визначає основні концепції та фази управління інцидентами інформаційної безпеки. Стандарт поділений на 4 частини:

- ISO/IEC 27035-1:2023 – принципи і процеси;
- ISO/IEC 27035-2:2023 – настанови щодо планування та підготовки до реагування на інциденти;
- ISO/IEC 27035-3:2020 – настанови щодо операцій з реагування на інциденти;
- ISO/IEC 27035-4:2024 – координація.

Перша частина стандарту представляє основні концепції та етапи управління інцидентами інформаційної безпеки, а також способи покращення управління інцидентами. Ця частина поєднує ці концепції з принципами в структурованому підході до виявлення, звітування, оцінки та реагування на інциденти, а також застосування отриманих уроків.

Також стандарт надає визначення ключовим термінам процесу реагування на інциденти:

Подія інформаційної безпеки – це подія, що вказує на можливе порушення інформаційної безпеки або збій засобів контролю.

Інцидент інформаційної безпеки – це одна або декілька пов'язаних та ідентифікованих подій інформаційної безпеки, які відповідають встановленим критеріям і можуть завдати шкоди активам організації або поставити під загрозу її діяльність.

Виникнення події не обов’язково означає виникнення інциденту, оскільки не всі події несуть в собі загрозу конфіденційності, цілісності або доступності, така подія має визначатися як хибне спрацювання.

Команда реагування на інциденти (IRT) – команда кваліфікованих і довірених членів організації, яка займається інцидентами протягом їх життєвого циклу.

Управління інцидентами інформаційної безпеки – застосування послідовного та ефективного підходу до реагування на інциденти інформаційної безпеки.

Обробка інцидентів – дії з виявлення, звітування, оцінювання, реагування, вирішення та навчання на основі інформації про інциденти інформаційної безпеки.

Реагування на інциденти – дії, що вживаються для пом'якшення або вирішення інциденту інформаційної безпеки, у тому числі дії, що вживаються для захисту та відновлення нормальних умов функціонування інформаційної системи та інформації, що в ній зберігається.

Взаємодію об’єктів інциденту інформаційної безпеки: загроз, подій, інцидентів, інформаційних систем та вразливостей наведено на рисунку 2.

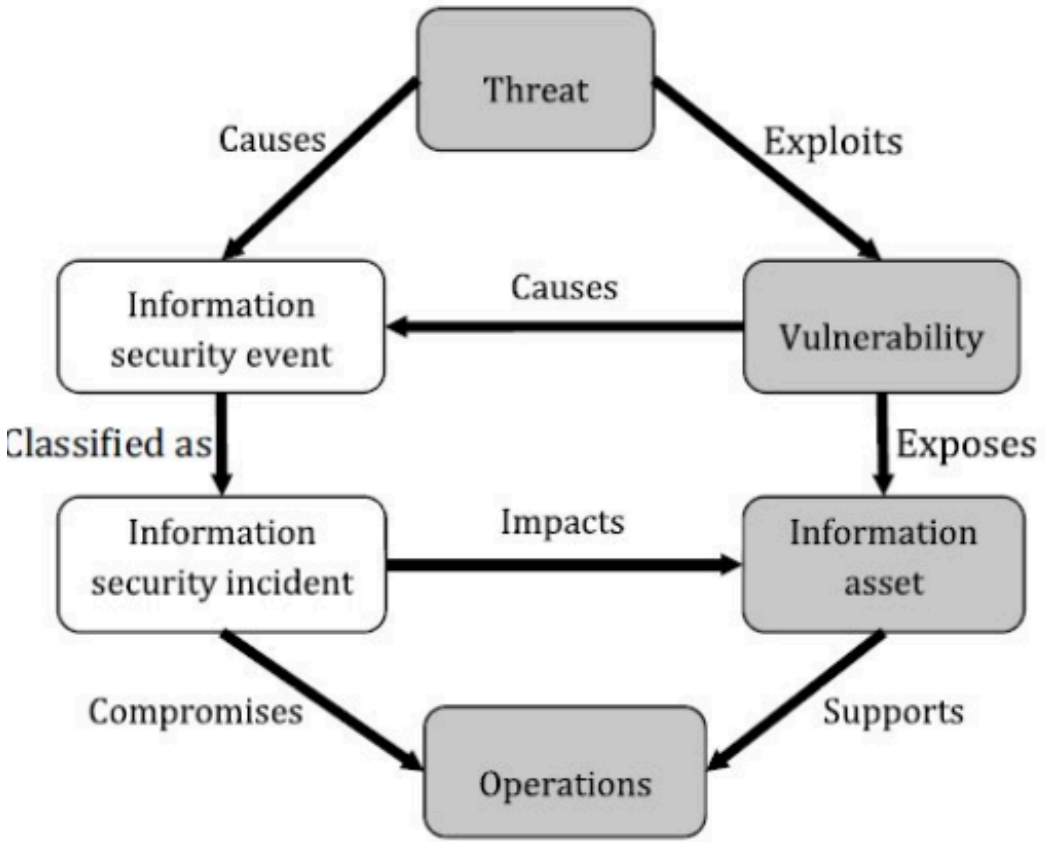

Рисунок 1.2. Взаємодія об’єктів в ISO 27035

Стандарт визначає такі кроки для досягнення ефективного і структурованого управління інцидентами:

- виявлення та ефективне реагування на події інформаційної безпеки;
- інциденти оцінюються та оброблюються у найбільш ефективний спосіб;
- мінімізація негативного впливу інцидентів на бізнес-операції;
- встановлення зв'язку з антикризовим управлінням та управлінням безперервністю бізнесу
- постійне управління вразливостями;
- Засвоєння уроків з інцидентів, покращення використання засобів контролю інформаційної безпеки, а також вдосконалення загального плану управління інцидентами інформаційної безпеки.

Стандарт також приділяє увагу виконанні контролів вказаних у ISO 27001 для побудови СУІБ. Взаємозв’язок з СУІБ зображено на рисунку 3.

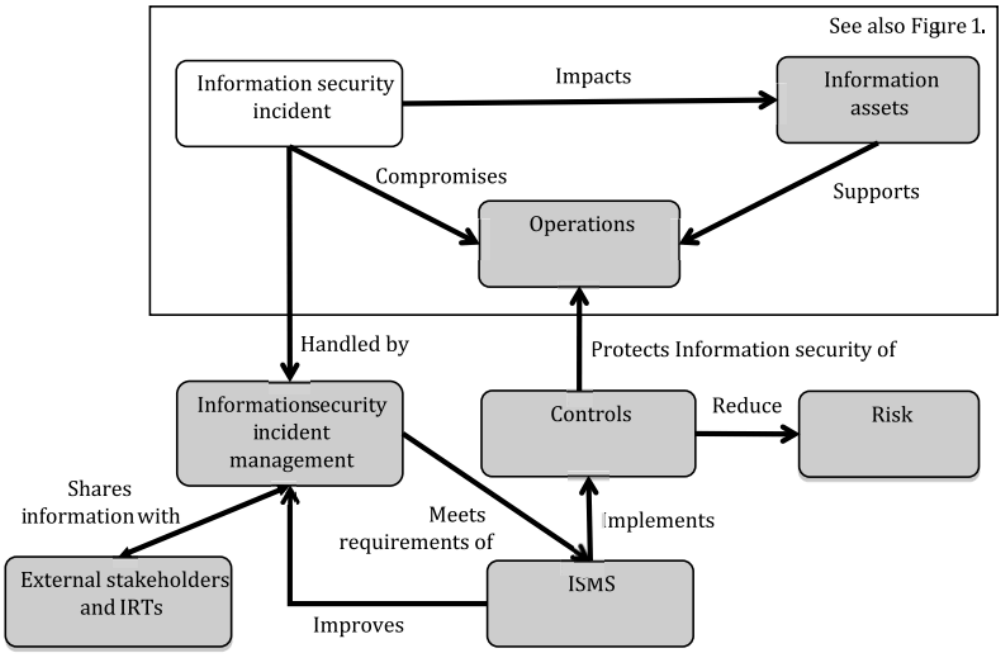

Рисунок 1.3. Взаємодія управління інцидентами інформаційної безпеки з СУІБ.

Використання структурованого підходу до управління інцидентами інформаційної безпеки може принести значні переваги, а саме:

- підвищення інформаційної безпеки;
- зниження впливу інцидентів на бізнес;
- фокус запобігання інцидентів у майбутньому;
- покращення пріоритетності дій;
- підвищення якості збору та розслідування доказів;
- сприяння бюджетному та ресурсному обґрунтуванню;
- поліпшення управління ризиками;
- підвищення обізнаності про безпеку в організації;
- покращення політики та процедур безпеки.

Загалом стандарт визначає 5 етапів в процесі управління інцидентами інформаційної безпеки (див. рис 4).

Рисунок 1.4. Життєвий цикл інциденти згідно з ISO 27035.

1. Планування та підготовка.

Для ефективного управління інцидентами організація має все спланувати та підготуватися, перед введенням у дію плана управління інцидентами необхідно виконати ряд підготовчих заходів, а саме:

- розробити політику управління інцидентами інформаційної безпеки;
- оновити політику інформаційної безпеки;
- задокументувати детальний план управління інцидентами, включаючи теми комунікацій з особами у разі розкриття інформації;
- створити IRT, розробити навчальну програму та надати її персоналу;
- створити та підтримувати відносини та зв'язки з внутрішніми та зовнішніми організаціями, які безпосередньо залучені до управління подіями, інцидентами та вразливостями інформаційної безпеки;
- створити, впроваджувати та експлуатувати технічні, організаційні та операційні механізми для підтримки плану управління інцидентами інформаційної безпеки та роботи IRT;

- створити програму підвищення обізнаності користувачів;
- протестувати створений план управління вразливостями, його процедур та процесів.

2. Детектування та повідомлення

Другий етап передбачає виявлення, збір інформації, пов'язаної з подіями інформаційної безпеки та існуванням вразливостей інформаційної безпеки, та звітування про них за допомогою ручних або автоматичних засобів. На цьому етапі події та вразливості можуть ще не бути класифіковані як інциденти інформаційної безпеки.

Для цього етапу необхідно виконати такі дії:

- створити, та здійснювати моніторинг систем, мережеву активність;
- виявляти та повідомляти про порушення ІБ;
- виявляти події ІБ та наявність вразливостей;
- збирати інформацію про події та вразливості;
- логувати всю активність пов’язану з результатами аналізу та рішень, які пов’язані з подіями;
- все докази мають зберігатися безпечно;
- слідкувати за роботоспроможністю системою внесення змін для підтримання баз даних інформаційної безпеки у актуальному стані;
- ескалювати події за необхідністю.

3. Оцінка та прийняття рішення

Після отримання інформації про подію її необхідно оцінити для правильної та ефективної оцінки необхідно розділити відповідальність з обробки інцидентів за допомогою ієрархії персоналу з оцінкою, прийняттям рішень. Також до процесу прийняття рішення може бути залучений персонал з відділу ІБ та інших відділів.

Персонал має використовувати процедури, які повинні дотримуватися, включаючи перегляд і внесення змін до звітів, оцінку збитків і повідомлення відповідного персоналу. Індивідуальні дії залежать від типу та серйозності інциденту.

Для етапу необхідно виконати такі дії:

- зібрати інформацію про подію;
- провести оцінку події, визначити чи є загроза;
- переконатися, що всі залучені сторони, зокрема, IRT, належним чином реєструють всі дії, результати та пов'язані з ними рішення для подальшого аналізу;
- забезпечити підтримання режиму контролю змін для відстеження інцидентів інформаційної безпеки та оновлення звітів про інциденти та оновлення звітів про інциденти, а також для підтримання бази даних інформаційної безпеки в актуальному стані.

4. Реагування.

Етап передбачає безпосереднє реагування на інцидент, відповідно до інформації визначеної на попередніх етапах. Реагування може відбуватися у режимі реального часу або близько до реального часу.

На етапі реагування організація виконує такі дії:

- перевірку чи знаходиться інцидент під контролем і якщо так, виконати реагування. Якщо інцидент не під контролем необхідно виконати заходи з реагування відповідно з процедурами управління кризисними ситуаціями;
- необхідно визначити та виділити ресурси на реагування;
- здійснити ескалацію у разі необхідності;
- забезпечити реєстрацію у системі документування всіх дій;
- забезпечити зберігання доказів;
- повідомити про інцидент всіх осіб згідно з планами комунікації під час інцидентів. Особливо важливо буде повідомити власників активів про інцидент.
- після відновлення необхідно запустити дії після інциденту;
- створити звіт по інциденту;
- повідомити всіх залучених осіб про закриття інциденту.

Вся інформація зібрана під час етапу має зберігатися у базі даних інформаційної безпеки для допомоги в оцінці, рішеннях під час можливих майбутніх інцидентів.

5. Вивчені уроки.

Заключний етап під час якого досліджується як саме інцидент або вразливість були усунені. Під час цього етапу організація:

- визначає засвоєні уроки під час інциденту;
- робить аналіз інциденту для внесення змін в існуючі контролі ІБ;
- перевіряє ефективність процесів, процедур та механізму повідомлення про інциденти;
- за бажанням повідомляє про результати цього процесу іншим особам;
- аналізує чи необхідно поділитися про вразливість або інцидент з спільнотою для захисту від таких подій;
- робить аналіз ефективності IRT.

Друга частина стандарту ISO 27035-2 присвячена першому етапу - планування та підготовка та другому етапу - вивчені уроки. Документ детально описує створення та вимоги до документації системи управління інцидентами інформаційної безпеки та збору IRT. Також велика частина документа описує створення ефективної системи навчання співробітників базовим речам інформаційної безпеки.

Для створення системи управління інцидентами необхідно розробити політику, план, процедури, процеси та зібрати IRT.

Політика управління інцидентами інформаційної безпеки має включати в себе:

- мету, завдання, сферу застосування;
- власника політики та час перегляду;
- важливість політики та залученість вищого керівництва у процес управління інцидентами;
- типи інцидентів та їх категорії;
- детальні інструкції про механізм повідомлення про інциденти;
- візуалізацію процесу управління інцидентами;
- вимоги до процесу аналізу після інциденту, такі які навчання через інциденти та покращення процесів;
- визначені ролі, обов’язки та осіб відповідальних за рішення на різних етапах інцидентів;

- посилання на документ з подіями та класифікацією інцидентів;
- структуру IRT, її цілі, ролі в команді, обов’язки та повноваження, вимоги до звітності, інформування керівництва;
- посилання на зовнішні організації, які надають підтримку у вирішенні інцидентів;
- стислий виклад законодавчих та регулятивних вимог пов’язаних з управлінням інцидентами;
- перелік та посилання на інші політики, процедури, документи, які пов’язані з процесом управління інцидентами.

План управління інцидентами - це детальний документ, який описує дії, які необхідно виконати в разі виявлення інциденту інформаційної безпеки. План випливає з політики управління інцидентами та ґрунтується на ній і починає діяти щоразу, коли виявляється подія інформаційної безпеки.

Для створення необхідно визначити ключові критерії прийняття рішень, визначити процеси для підтримки етапів управління інцидентами. Детальні дії плану мають бути пов’язані з етапами, які були розглянуті в цій роботі раніше, а саме: планування та підготовка, детектування та повідомлення, оцінка та прийняття рішення, реагування, вивчені уроки. План має забезпечити кроки для негайного реагування та для реагування в довгостроковій перспективі. Всі інциденти інформаційної безпеки повинні проходити ранню оцінку потенційного негативного впливу на бізнес-операції, як короткострокового, так і довгострокового. Крім того, він повинен передбачати деякі заходи, необхідні для реагування на інциденти інформаційної безпеки, які є абсолютно непередбачуваними, коли потрібні спеціальні засоби контролю. Навіть для таких ситуацій мають бути загальні вказівки щодо кроків, які можуть бути необхідними. Також план має містити інформацію про постійний перегляд ефективності процесів, процедур, форматів звітностей та організаційної структури реагування на інциденти.

Перед впровадження плану організація має впевнитися, що задокументовано і перевірено необхідні процедури та процеси. У кожному документі повинні бути вказані групи або особи, відповідальні за його використання та управління ним.

Важливо розуміти, що не всі документи повинні бути легкодоступними як всередині організації, так і для громадськості організації, так і для широкої громадськості. Деякі особи можуть використати знання з цих документів для того щоб приховати свою незаконну діяльність і запобігти викриттю.

Зміст процедур залежить від потенційних подій, інцидентів та вразливостей, а також типів інформаційних системних активів, які можуть бути задіяні, та їхнього оточення. Вони мають відображати кроки, які мають бути виконані для усунення інцидента. Це морже бути досвід зовнішніх постачальників чи інших професійних команд або ж внутрішній досвід з минулих інцидентів. На всі відомі типи подій мають існувати процедури, а також певні кроки для усунення нових до цього невідомих подій.

IRT відіграє критичну роль у ІБ організації і вимагає співпраці всього персоналу організації для виявлення, вирішення та розслідування інцидентів, пов'язаних з інформаційною безпекою інцидентів інформаційної безпеки. В процесі роботи команди ключову роль відіграє довіра. Всі особи, які залучені до процесу мають бути впевнені в професіоналізмі команди, також довіра може бути здобута завдяки прозорим процесам реагування на інциденти. Для цього потрібно пояснювати користувачам, як команда захищає конфіденційність, які дані збираються як управляють подіями безпеки, інцидентами та вразливостями. Користувачі мають не боятися повідомляти про події, саме тому має бути оприлюднена анонімність або її відсутність для осіб або сторін, які повідомляють про підозри щодо інцидентів або вразливостей інформаційної безпеки. IRT має бути здатна ефективно задовольняти потреба управління інцидентами організації, а також мати можливість керувати інформаційною безпекою організації. Команда має проходити аудити для підтвердження ефективності роботи. Окрім того гарним способом перевірити ефективність є призначення окремого менеджера, який буде слідкувати за управлінням інцидентами та вразливостями.

Склад IRT може бути різна в залежності від потреб та розміру організації. Однак має бути чітко визначено групу активів за яку вона відповідає, це може бути група серверів, IP адрес, приналежність до домену або окремий офіс чи країна. Хоча

основна задача команди полягає в реагуванні на інцидент, багато організацій використовує її для превентивної функції, покращуючи стандарти та практику безпеки у межах своїх активів.

IRT можуть бути структуровані у різний спосіб, зокрема, за секторами, активами, організаційною структурою, або за іншими ознаками. Для створення IRT, слід враховувати розмір організації, важливість інформації та сумісність з іншими організаціями. організаціями, важливість інформації та сумісність з іншими організаціями.

Один із способів структуризації - за обсягом моніторингу, і в цьому випадку існує три різні типи (див. рис. 5):

- одиничний - об'єктом моніторингу є одна організація або одна IRT, що здійснює моніторинг декількох організацій або цілей;
- ієрархічний - Одна або кілька IRT перекривають сфери моніторингу. Це може підвищити надійність заходів з реагування на інцидент;
- віддалений тип - збираючи події безпеки з віддалених місць, цей тип зазвичай використовується для аутсорсингових підприємств (спеціалізованих підприємств з інформаційної безпеки) для моніторингу об'єктів.

Основні напрямки діяльності IRT:

- Управління інтегрованими системами безпеки: Моніторинг та управління подіями інформаційної безпеки агентами, встановленими на системах (системи детектування вторгнень, системи виявлення вторгнень, брандмауер);
- впровадження узгодженої політики: Мінімізація ризиків для інформаційної системи шляхом застосування послідовного набору завдань реагування відповідно до визначеної політики;

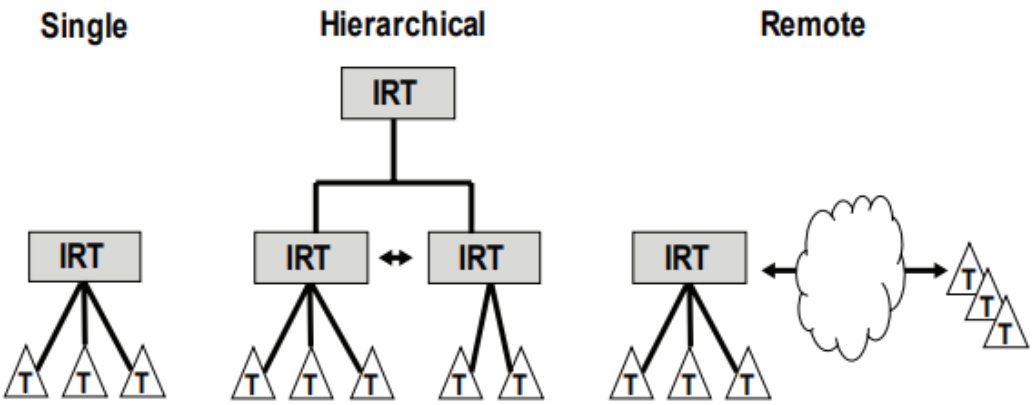

Рисунок 1.5. Типи IRT

- оперативне реагування: Швидке реагування на загрози, порушення та атаки, щоб мінімізувати збитки та зменшення витрат на відновлення.

Обов’язки команди можуть включати в себе:

- управління та моніторинг: 24 × 7 × 365 моніторинг цілей, проактивний моніторинг та реагування на інциденти, ведення журналів;
- управління звітами: Періодичне звітування про безпеку, управління виправленнями безпеки, звітування про інциденти;
- адміністративне управління: Управління політиками для різних системних середовищ, включаючи управління завданнями та операціями IRT;
- технічне управління: Управління безпекою мережі, системи, додатків, вмісту та сервісів;
- експлуатація та управління системою: Пропускна здатність системи, продуктивність, конфігурація безпеки та управління конфігурацією середовища.

Ефективне реагування на інциденти залежить від підготовленості членів команди. Персонал та їх можливості є дуже великим фактором впливу. Навички, необхідні для членів IRT, можуть включати наступне.

- особисті навички: комунікація, вирішення проблем, командна взаємодія, управління часом і проектами;
- технічні навички: принципи ІБ, аналіз ризиків, моделювання загроз, аналіз вразливостей, аналіз журналів, знання мереж та криміналістична експертиза;

- навички реагування на інциденти: знання політик/процедур, комунікація з командою, аналіз інцидентів, реєстрація та відстеження інформації про інциденти.
- спеціалізовані навички: презентація, лідерство, знання предметної області, програмування.

Для ефективного реагування члени команди мають мати достатні технічні навички такі як:

- знання типових атак, загроз, шкідливого програмного забезпечення, вразливостей;
- бути обізнаними в практиках системних адміністраторів таких як: управління оновленнями, безпечна конфігурація систем, системи резервного копіювання і системи аварійного відновлення;
- криптографія: хешування та алгоритми шифрування, протоколи SSL/TLS;
- знання мережевих протоколів (IPv4, IPv6, ICMP, UDP, TCP) ;
- знання протоколів мережевих додатків (DNS, SMTP, HTTP, HTTPS) ;
- методи збору цифрових доказів, методи зворотного інжинірингу;
- знання комп’ютерних наук, концепцій програмування таких як SDLC, функціональний метод програмування та ООП, архітектури різних систем.

Це базові навички, інші навички необхідні в залежності від технологій, які використовує організація. Члени IRT повинні підтримувати свої знання та навички на належному рівні.

Підсумок стандарту наведений на рисунку 6.

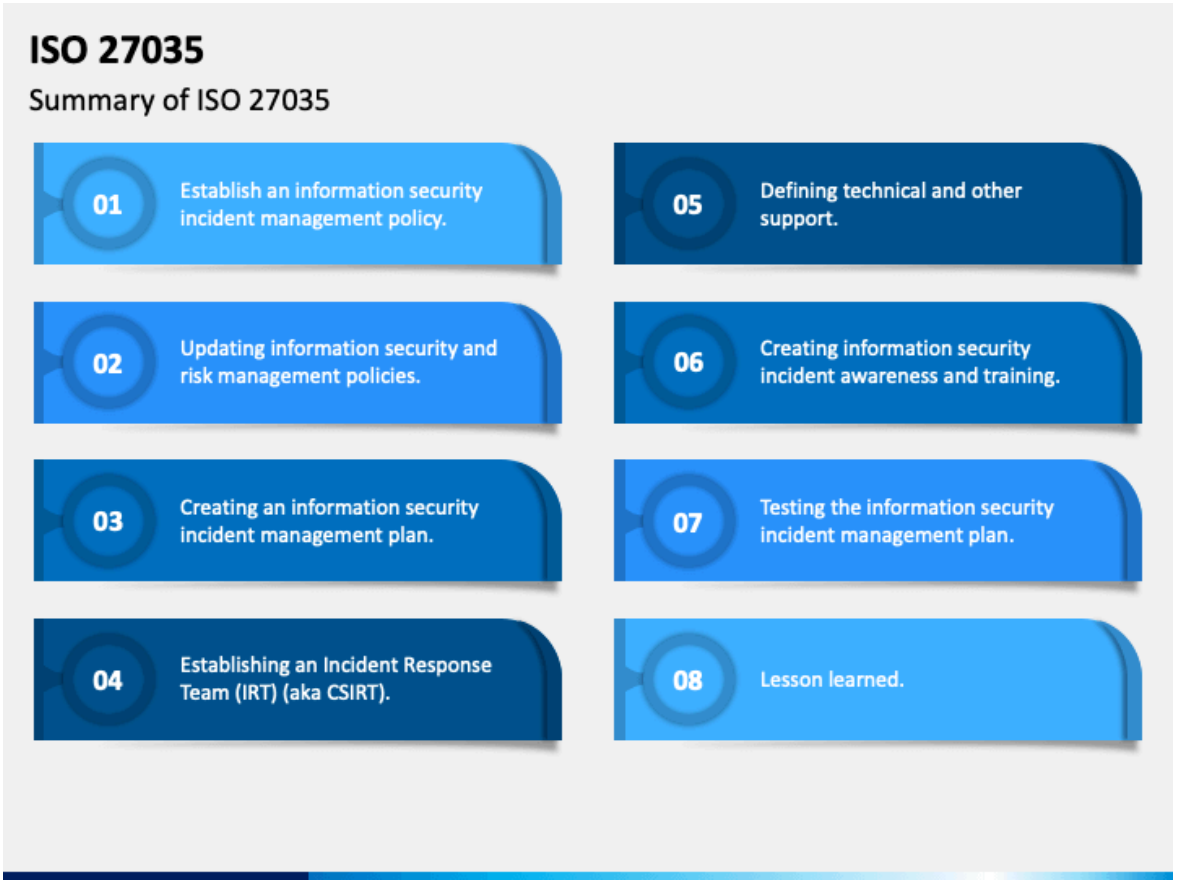

Рисунок 1.6. Підсумок стандарту ISO 27035

1.2 NIST Special Publication 800-61

NIST Special Publication 800-61 Computer Security Incident Handling Guide – це документ Національного інституту стандартів і технологій США, який містить в собі покроковий підхід до розробки та впровадження в організації системи управління інцидентами ІБ. Документ є практичним посібником і містить рекомендаційний характер, але має більш технічний характер, ніж стандарти серії ISO.

Публікація в багатьох моментах схожа з ISO, так наприклад документ починається з опису важливості політики, плану та процедур з реагування на інциденти та містить вимоги до їх створення та підтримки. Загалом вимоги дуже схожі і вимагають залученості вищого керівництва у процеси реагування інцидентів. Те саме стосується структури IRT та можливого складу команди. Однак SP 800-61 звертає більшу увагу на потреби організації у реагуванні на інциденти, організація має продумати для себе:

- чи необхідно їй реагування на інциденти у режимі 24/7? Більшості організацій це необхідно, але треба також проаналізувати чи потрібні організації постійна присутність спеціаліста з реагування на інциденти на робочому місці чи просто достатньо організувати зв’язок з спеціалістом по телефону;
- чи необхідні організації члени команди на повний робочий день? Якщо обмежений бюджет або персонал команду реагування на інциденти можна розглядати як команду термінової допомоги. Коли виникає надзвичайна ситуація, організація зв’язується з IRT;
- моральний стан співробітників. Робота з реагування на інциденти є дуже стресовою, як і чергування. Таке поєднання призводить до того, що члени команди реагування на інциденти легко піддаються надмірному стресу. Розподіл ролей, зокрема зменшення обсягу адміністративної роботи, за виконання якої відповідають члени команди, може суттєво підвищити моральний дух команди;
- витрати. Витрати є важливим фактором, особливо якщо працівники повинні бути на місці 24/7. Необхідно передбачити достатнє фінансування для навчання та підтримання навичок.
- організації можуть не включити в бюджет витрати, пов'язані з реагуванням на інциденти, наприклад, достатнє фінансування для навчання та підтримання навичок.
- кваліфікація персоналу. Обробка інцидентів вимагає спеціальних знань і досвіду в декількох технічних областях; широта і глибина необхідних знань варіюється в залежності від серйозності ризиків організації.

Публікація пропонує інший цикл життя інцидентів в порівнянні з ISO (див. рис. 7).

Етап підготовки є ключовим для забезпечення ефективного реагування на інциденти інформаційної безпеки. У цьому етапі описані основні заходи, які необхідно реалізувати до виникнення інциденту. Основна мета — створення необхідних умов для швидкого виявлення, аналізу та реагування на інциденти.

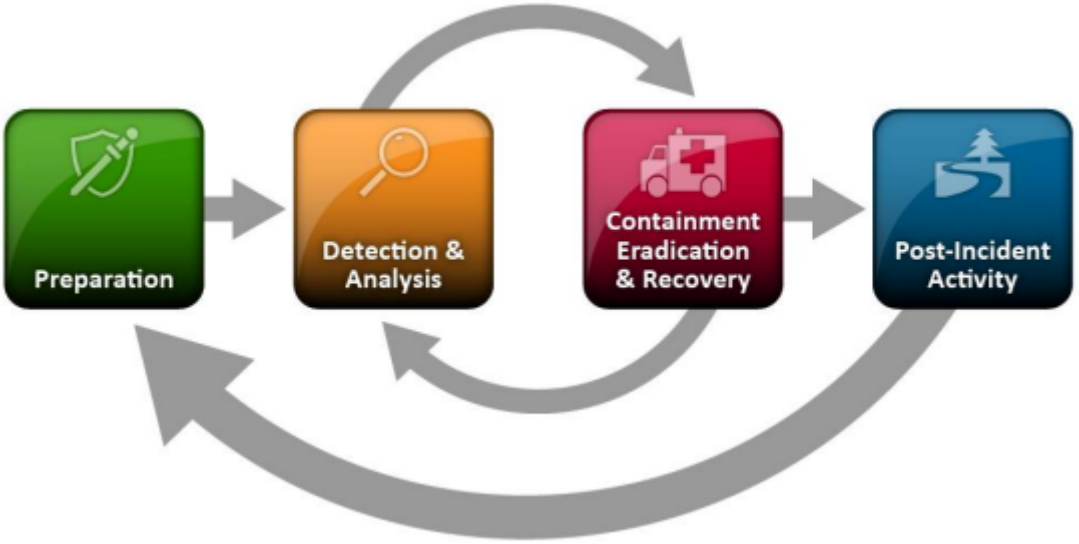

Рисунок 1.7. Життєвий цикл інцидентів згідно з NIST

Для цього необхідно підготувати:

- контактну інформацію з членами команди і іншими залученими з процесом реагування на інциденти;
- механізми повідомлення про інциденти;
- системи відслідковування інцидентів;
- сервери для проведення криміналістичної експертизи та систем резервного копіювання;
- ноутбуки для аналізу даних, аналізу пакетів і написання звітів;
- USB для збору даних з інфікованих систем;
- аксесуари для збору доказів, такі як: камери, аудіо записи;
- списки портів, які використовує компанія і які часто під атакою;
- документація про ОС, додатки, протоколи, системи безпеки;
- діаграми мережі ти списки критичних активів;
- базові показники очікуваної активності мережі, системи та додатків;
- криптографічні хеші критично важливих файлів22 для прискорення аналізу інцидентів, перевірки та усунення

Також для більшої ефективності IRT команди створюють інфраструктуру для криміналістичного аналізу, яка доступна в будь-який момент. Наприклад додатки для аналізу пакетів, дисків.

Детектування та аналіз зосереджується на оперативному виявленні інцидентів, зборі релевантних даних і проведенні аналізу для визначення природи та серйозності загрози. Для ефективного реагування на інциденти необхідно розуміти вектори атак на організацію:

- зовнішні девайси як USB;
- веб атаки;
- електронна пошта;
- імітація атаки типу «людина посередині», несанкціоновані бездротові точки доступу та SQL-ін'єкції атаки, що включають в себе видавання себе за іншу особу;
- неправильне використання ресурсів: Будь-який інцидент, спричинений порушенням політики організації щодо прийнятного використання;
- втрата або крадіжка обладнання.

Для правильного аналізу подій необхідно:

- мати профілі мереж та систем. Тобто характеристики очікуваної активності цих систем.
- розуміння нормальної поведінки систем.
- створити політику збереження журналів
- створити кореляцію подій.
- всі годинники серверів мають бути синхронізовані.
- підтримувати та використовувати інформаційну базу знань.
- фільтрувати дані. Немає сенсу витрачати час на незначні події, необхідно робити пріоритезацію.

Етап локалізації, ліквідації та відновлення спрямований на зупинення інциденту та зменшення збитків від інцидентів. Для цього організація має обрати план локалізації: короткостроковий – миттєві заходи або довгостроковий – перенаправлення трафіку, оновлення правил брандмауера та сегментація мережі. Також необхідно зібрати всі можливі докази про інцидент такі як: знімки дисків, оперативної пам’яті. Необхідно ідентифікувати IP нападника, перевірити його через відкриті джерела, в базах даних інцидентів, спробувати ідентифікувати нападника та

його техніки. Після локалізації необхідно ліквідувати причини інциденту, а саме виявити шкідливі компоненти і видалити їх або ж усунути вразливості. Під час відновлення необхідно відновити системи з резервних копій для повної робото спроможності систем, як і до інциденту.

Етап дії після інциденту є завершальним у циклі реагування на інциденти. Він включає оцінку виконаних дій, документування інциденту, визначення уроків та впровадження змін для покращення процесів реагування. Цей етап вдосконалює загальну стратегію кібербезпеки та запобігає повторенню аналогічних інцидентів у майбутньому.

Загалом на цьому етапі має мати відповіді на такі питання:

- що конкретно сталося і коли?
- наскільки добре з інцидентом впорався персонал та керівництво?
- чи були дотримані процедури та чи були вони ефективними?
- чи були зроблені якісь кроки або дії, які могли б перешкодити відновленню?
- що можна зробити щоб інцидент не повторився?
- що персонал і керівництво зробили б по-іншому, якби подібний інцидент стався наступного разу?
- на які індикатори слід звернути увагу в майбутньому для виявлення подібних інцидентів?
- які додаткові інструменти або ресурси нам необхідні для детектування, аналізу та усунення майбутніх інцидентів?

Після серйозних атак варто проводити зустрічі після інциденту, які виходять за межі команди та організації для забезпечення механізму обміну інформацією. Зустрічі, присвячені вивченим урокам, мають й інші переваги. Звіти з таких зустрічей є хорошим матеріалом для навчання нових членів команди, показуючи їм, як більш досвідчені члени команди реагують на інциденти.

Збір інформації про інциденти дозволяє робити аналіз ефективності та можуть бути використані для виправдання додаткового фінансування відділу. Організація має збирати статистику про інциденти і саме інформативні дані. Необхідно

вирішити, які дані інциденту збирати на основі вимог звітності та очікуваної віддачі. Можливі метрики можуть включати:

Кількість інцидентів. Відносний обсяг роботи IRT, велика кількість інцидентів може свідчити про погані контролі ІБ або через недбалість в реагуванні на інциденти.

Час обробки інциденту. Можна відслідковувати загальний час витрачений на обробку, окремі етапи, як наприклад час реагування та локалізації інциденту, час повідомлення про інцидент відносно часу події.

Об'єктивна оцінка кожного інциденту. Оцінка аналізу логів, повідомлень або іншої документації, можливо її можна покращити, що зменшить час реагування. Оцінка чи міг інцидент бути повторенням минулого інцидента. Розрахунок втрат. Оцінка, які дії могли б запобігти інциденту

1.3 Порівняння стандартів ISO 27035 з NIST SP 800-61

Реагування на інциденти надзвичайно важливий процес у побудові системи захисту будь-якої сучасної організації. Правильний підхід до побудови центру безпеки організації збільшує швидкість реагування на інциденти, обмежує їх вплив на організацію, що тим самим зменшує втрати.

Одними з найбільш відомих документів, що регламентують цей процес, є міжнародний стандарт ISO/IEC 27035 та спеціальна публікація NIST SP 800-61. Обидва документи описують ключові аспекти управління інцидентами, однак мають суттєві відмінності у підходах, структурі та рівні деталізації.

Таким чином можна визначити основні відмінності між документами:

1. Сертифікація.

1.1. Серія документів ISO є стандартом. Саме цим стандартом можна підтвердити, що в компанії побудований процес управління інцидентами.

1.2. NIST SP 800-61 – це рекомендація.

2. Підхід до управління інцидентами.

2.1. ISO/IEC 27035 – документ визначає стратегічний підхід до процесу IR. Він складається з довгострокової інтеграції процесів у систему управління інформаційною безпекою. Документ визначає загальні принципи реагування, ключові етапи обробки інцидентів та способи покращення процесу після виявлених інцидентів.

2.2. NIST SP 800-61 – документ має більш практичний та технічно-орієнтований підхід. Він містить в собі дуже детальні рекомендації, щодо реагування на інциденти, класифікацію загроз, процедури аналізу та конкретні інструменти безпеки, які можуть бути використані.

3. Структурна відмінність етапів реагування.

3.1. ISO виділяє 5 етапів реагування.

3.2. NIST виділяє 4 етапи реагування.

4. Рівні деталізації та практична цінність.

4.1. ISO зосереджений на побудові загальної системи реагування на інциденти. Він надає концептуальну основу для встановлення процесів на стратегічному рівні, яка має бути адаптована до організацій.

4.2. NIST містить детальні інструкції побудови процесів реагування, набору персоналу, потенційних інструментів, які можуть використовуватися.

5. Автоматизація процесів

5.1. ISO акцентує увагу на автоматизації процесів збору логів, моніторингу та аналізу.

5.2. NIST містить конкретні рекомендації щодо впровадження SIEM систем, використання скриптів та систем оркестрації та автоматизації безпеки – SOAR.

Загалом на цій інформації можна створити порівняльну таблицю (див. таб. 1).

Таблиця 1.1.

Порівняння ISO та NIST

Критерій	ISO/IEC 27035	NIST SP 800-61

Аудит	Підходить для зовнішніх аудитів та сертифікації	Не призначений для сертифікації, фокусується на операційній ефективності
Підхід	Стратегічний, інтеграція в СУІБ	Тактичний, детальні рекомендації
Цільова аудиторія	менеджмент, стратегічний рівень	SOC, аналітики безпеки, операційний персонал
Структура	5 етапів	4 етапи
Деталізація	Загальні принципи	Більш практичний документ.
Автоматизація процесів	Загальні вказівки, орієнтований на структуру	Надає приклади інцидентів, сценарії обробки. Рекомендує інструментів автоматизації.
Зв’язок з іншими стандартами	Частина серії ISO/IEC 27000 (інтегрується з 27001, 27002)	Часто використовується разом з NIST 800-53, 800-171

Висновки за розділом 1

У розділі було детально розглянуто одні з найважливіших документів для побудови процесу реагування на інциденти: ISO/IEC 27035 та NIST SP800-61. Обидва є фундаментальними для побудови ефективної системи, однак мають свої підходи та особливості.

ISO 270335 – є міжнародним стандартом та пропонує комплексний, структурований підхід до управління інцидентами, охоплюючи всі його цикли: від планування та підготовки до виявлення, аналізу, реагування та винесення уроків.

NIST SP800-61 - розроблений Національним інститутом стандартів і технологій США був орієнтований на американські організації, проте широко використовується у всьому світі. Документ зосереджується на практичних аспектах реагування на інциденти та має детальні рекомендації побудови процесів. Також велику увагу приділено комунікації та координації дій між зацікавленими сторонами.

Вибір документу для побудови процесу управління інцидентами залежить від потреб організації та її контексту, проте потрібно розуміти, що ISO це стратегія, яка не дає відповіді на те, як побудувати процес на операційному рівні. Стандарт орієнтований на побудові комплексної системи управління інцидентами, тоді як NIST акцентує увагу на швидке виявлення та реагування на загрози. Оптимальною стратегією є використання обох документів, це дозволить забезпечити і стратегічне планування, і операційну частину реагування.

РОЗДІЛ 2
СУЧАСНІ ПІДХОДИ ДО ПРОЦЕСІВ РЕАГУВАННЯ НА ІНЦИДЕНТИ

2.1 Роль Threat Intelligence у реагуванні на інциденти

Threat Intelligence - це детальна, дійсна інформація про загрози кібербезпеки. Розвідка загроз допомагає командам безпеки примінити проактивний підхід до виявлення і пом’якшення кібератак. Інформація може включати в себе:

- Індикатори компрометації;
- механізми атаки;
- вплив атаки;
- способи детектування атаки;
- практичні поради по захисту.

Cyber Threat Intelligence допомагає організаціям бути в курсі нових загроз для захисту. Команди з кібербезпеки аналізують інформацію, яку вони збирають про атаки для навчання та покращення захисту систем безпеки.

Розвідка загроз також допомагає зупинити або пом’якшити атаку, яка вже триває. Чим більше ми знаємо про атаку, тим легше її зупинити. Існують різні види розвідки загроз, від високорівневої та нетехнічної інформації до дуже детальних технічних подробиць про конкретні атаки. Загалом можна визначити такі види розвідки:

- стратегічна - інформація високого рівня про контекст загрози. Містить в собі мотиви для розуміння побудови стратегії безпеки. Це нетехнічна інформація з аналізом ризиків. Інформація призначена для вищого керівництва - CISO, CIO, CTO;
- тактична - фокусується на технічному аналізі шкідливого програмного забезпечення. Містить в собі індикатори компрометації для захисту систем безпеки. Інформація використовується системами безпеки та аналітиками центру безпеки;
- оперативна включає в себе інформацію про те як саме загрози реалізуються та як захиститися від них. Включає в себе вектори атак, інструменти та інфраструктури, які використовують нападники, типи організацій жертв і стратегії

захисту. Така інформація допомагає зрозуміти вірогідність стати ціллю атаки та покращити свої системи безпеки для захисту. Інформація призначена для команди реагування на інциденти, мисливцям на загрози, аналітикам.

Аналітика загроз має вирішальне значення для організацій усіх розмірів, допомагаючи їм розуміти зловмисників, швидше реагувати на інциденти і заздалегідь передбачати загрози. Інструменти аналізу загроз і кіберзагроз допомагають організаціям зрозуміти ризики різних типів атак і як найкраще захиститися від них. Аналіз кіберзагроз також допомагає пом'якшити атаки, які вже відбуваються. IT-відділ організації може збирати власну аналітику загроз або може покладатися на службу аналітики загроз для збору інформації та надання рекомендацій щодо найкращих методів забезпечення безпеки. Організації, що використовують програмно-визначені мережі, можуть використовувати аналітику загроз для швидкого переналаштування своєї мережі для захисту від певних типів кібератак.

Життєвий цикл розвідки загроз - це безперервний процес, який перетворює необроблені дані на дієву розвідку, спрямовуючи команди з безпеки до прийняття обґрунтованих рішень (див. рис. 2.1). Цей цикл складається з шести ключових кроків, кожен з яких створює зворотний зв'язок для постійного поліпшення:

1. Вимоги. На цьому етапі необхідно визначити цілі та методологію програми розвідки відповідно до потреб зацікавлених сторін. Ключові питання включають розуміння мотивації зловмисника, визначення поверхні атаки та окреслення дій для покращення захисту.

2. Збір даних. Необхідно зібрати інформацію з таких джерел, як журнали трафіку, публічні дані, форуми, соціальні мережі та експертів, щоб відповідати визначеним вимогам.

3. Обробка. Організовуємо та очищаємо необроблені дані у формат, придатний для аналізу, що може включати розшифрування файлів, переклад іноземних даних або форматування у вигляді електронних таблиць.

4. Аналіз. Аналізуємо оброблені дані, щоб відповісти на питання, поставлені на етапі визначення вимог, і сформувати дієві висновки та рекомендації.

5. Поширення. Формуємо результати у зручному для сприйняття форматі, пристосованому до аудиторії зацікавлених сторін, у вигляді звітів або слайдів, не перевантажуючи їх технічними деталями.

6. Зворотній зв'язок. Збираємо відгуки від зацікавлених сторін для вдосконалення майбутніх операцій з розвідки загроз, коригування пріоритетів або зміни формату звітності за необхідності.

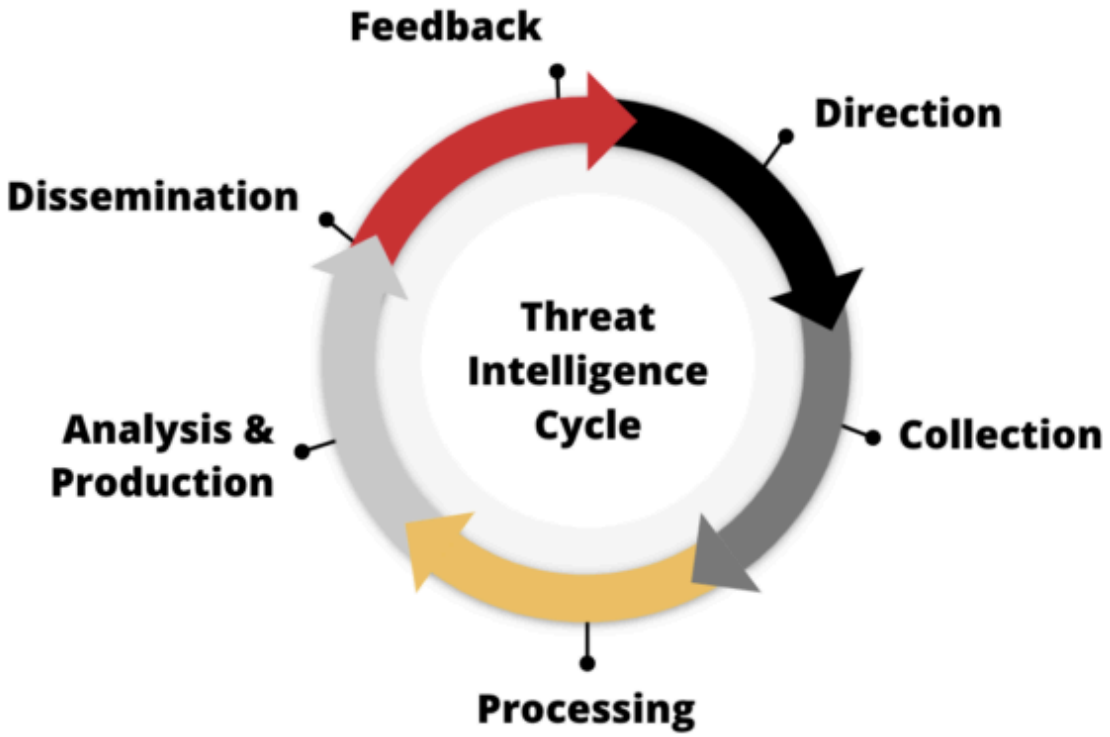

Рисунок 2.1. Процес Threat Intelligence

Threat Intelligence є дуже важливим компонентом захисту організацій. Якщо інтегрувати його у систему захисту організації отримаємо проактивний захист від загроз.

2.2 Активне виявлення загроз за допомогою Threat Hunting

Threat Hunting – це проактивний пошук загроз всередині організації. Полювання на загрози передбачає, що зловмисник знаходиться всередині вашої мережі і спрямоване на їх детектування. Проникнувши всередину, зловмисник може непомітно залишатися в мережі місяцями, непомітно збираючи дані, шукаючи конфіденційні матеріали або отримуючи облікові дані для входу в систему, які дозволять йому переміщатися всередині організації між різними системами.

Після того, як атакуючому вдалося уникнути виявлення і атака проникла в систему захисту організації, багатьом організаціям не вистачає розширених можливостей виявлення, необхідних для того, щоб зупинити просунуті постійні загрози і не дати їм залишитися в мережі. Ось чому полювання на загрози є важливим компонентом будь-якої стратегії захисту.

Мисливці за кіберзагрозами часто дотримуються таких основних кроків при розслідуванні та усуненні загроз і атак:

- створення теорії або гіпотези про потенційну загрозу. Можна почати з визначення типових TTP зловмисника;
- провести дослідження логів. Мисливці за загрозами досліджують дані, системи та діяльність організації, перевіряють логи в SIEM після чого збирають та обробляють відповідну інформацію;
- визначення тригера. Результати досліджень та інші інструменти безпеки можуть допомогти мисливцям за загрозами визначити відправну точку для розслідування;
- дослідження загрози. Мисливці за загрозами використовують свої дослідження та інструменти безпеки, щоб визначити, чи є загроза зловмисною;
- реагування та усунення. Після визначення загрози необхідно вжити заходів для усунення;

Процес полювання на загрози наведений на рисунку 2.1.

Рисунок 2.2. Процес полювання на загрози.

Розслідування зазвичай відбувається в одній з 3 форм:

- структуроване полювання - Формальні рамки, такі як рамки MITRE Adversary Tactics Techniques and Common Knowledge (ATT&CK), спрямовують структуроване полювання. Вони шукають визначені індикатори атаки і тактику, методи і процедури відомих суб'єктів загрози;
- неструктуроване полювання - неструктуроване полювання є більш непередбачуваним, ніж структуроване. Воно часто починається з виявлення індикатора компрометації в системі організації. Потім мисливці шукають, що спричинило IoC і чи він все ще існує в мережі;
- ситуативне полювання – починається після якоїсь унікальної ситуації. Зазвичай воно ґрунтується на результатах внутрішньої оцінки ризиків або аналізу тенденцій і вразливостей IT-середовища;

У проактивному полюванні є три моделі:

- розслідування на основі гіпотез. Часто ініціюється новою загрозою, яка була виявлена за допомогою великого масиву даних про атаки, що дає уявлення про новітні тактики, методи і процедури зловмисників (TTP). Після виявлення нової ТТП мисливці за загрозами намагаються з'ясувати, чи не зустрічається специфічна поведінка зловмисників у їхньому власному середовищі;
- розслідування на основі відомих індикаторів компрометації або ознак атаки. Цей підхід до полювання на загрози передбачає використання тактичної розвідки загроз для каталогізації відомих IOC та IOA, пов'язаних з новими загрозами. Потім вони стають тригерами, які мисливці за загрозами використовують для виявлення потенційних прихованих атак або поточної зловмисної діяльності;
- розширена аналітика і розслідування за допомогою машинного навчання. Третій підхід поєднує в собі потужний аналіз даних і машинне навчання для просіювання величезного обсягу інформації з метою виявлення аномалій, які можуть свідчити про потенційну зловмисну активність. Ці аномалії стають мисливськими зачіпками, які досліджуються кваліфікованими аналітиками для виявлення прихованих загроз.

Всі три підходи - це людські зусилля, які поєднують ресурси розвідки загроз з передовими технологіями безпеки для проактивного захисту систем та інформації організації.

Команди безпеки використовують різні інструменти для полювання на загрози. Деякі з найпоширеніших включають:

- SIEM - це рішення для забезпечення безпеки, яке допомагає організаціям розпізнавати та усувати загрози та вразливості до того, як вони встигнуть порушити бізнес-операції. SIEM допомагають виявляти атаки на більш ранніх стадіях і зменшують кількість помилкових спрацьовувань, які доводиться розслідувати мисливцям за загрозами;
- виявлення та реагування на кінцевих точках - EDR використовує аналітику в режимі реального часу та автоматизацію на основі штучного інтелекту для захисту кінцевих користувачів, кінцевих пристроїв та IT-активів організації від кіберзагроз, які не піддаються традиційним засобам захисту кінцевих точок;

- розширене виявлення та реагування (XDR) - мисливці за загрозами можуть використовувати XDR, яка забезпечує аналіз загроз і автоматизоване запобігання атакам, щоб досягти більшої видимості загроз.

Найкращі практики полювання на загрози включають в себе:

- мисливці за загрозами мають мати повну інформацію про організацію. Мисливці за загрозами досягають найбільшого успіху, коли розуміють загальну картину.
- підтримка інструментів безпеки, таких як: SIEM, XDR та EDR. Мисливці за кіберзагрозами покладаються на автоматизацію та дані, що надаються цими інструментами, щоб швидше виявляти загрози в більш повному контексті для швидшого їх усунення;
- постійне оновлення знань відносно нових загроз і тактик зловмисників. Останні постійно розвиваються і вдосконалюють свої методи. Інформація має бути найновішою;
- навчання співробітників виявляти та повідомляти про підозрілу поведінку. Інформуючи своїх співробітників можливо зменшити ймовірність внутрішніх загроз;
- впровадження управління вразливостями, щоб зменшити загальний ризик для організації;

Полювання на загрози значно доповнює стандартний процес виявлення, реагування та усунення інцидентів. У той час як технології безпеки аналізують вихідні дані для генерування оповіщень, полювання на загрози працює паралельно - за допомогою запитів і автоматизації - для вилучення зачіпок з тих самих даних.

Потім їх аналізують мисливці за загрозами, які вміють виявляти ознаки активності супротивника. Цей процес наведено на рис. 2.2.

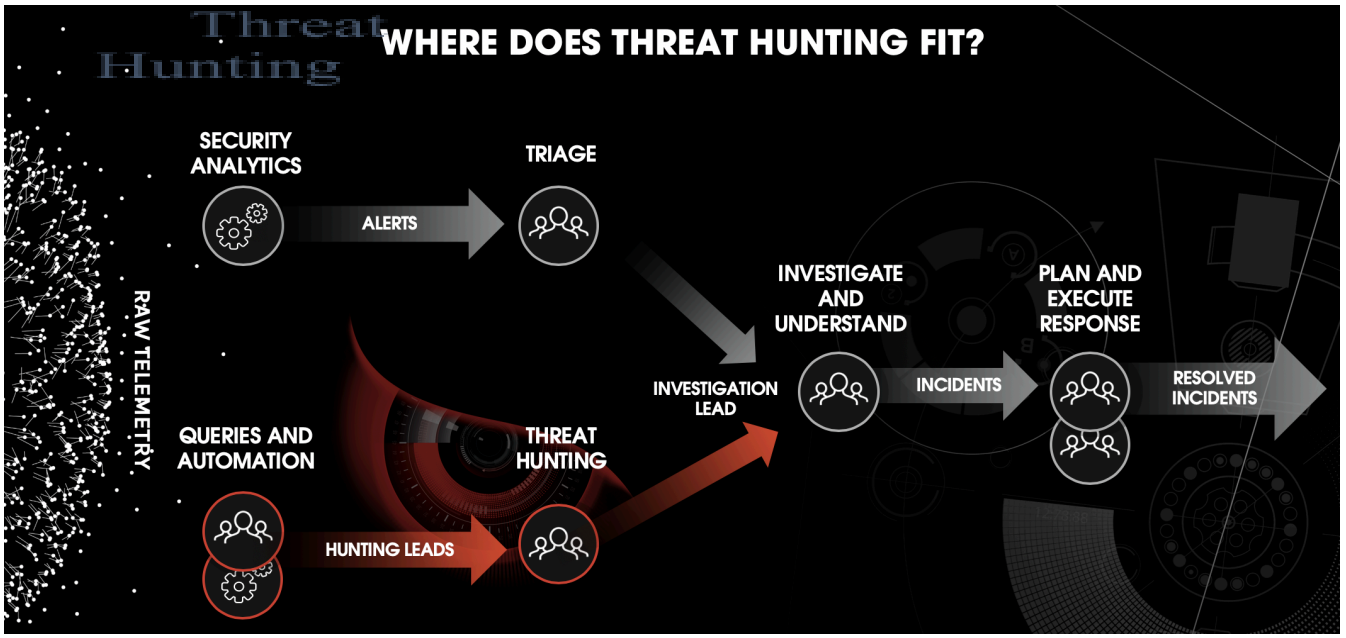

Рисунок 2.3 Threat Hunting в порівнянні з реагуванням на інциденти.

2.3 Моделювання загроз та аналіз ризиків

Операційні центри безпеки щодня мають справу із загрозами на практиці, організації, які вони захищають, піддаються атакам, і ці атаки виявляються та усуваються для запобігання збиткам. Саме тому для кожного члена центра безпеки і команди безпеки в цілому важливо розуміти слабкі місця системи, яку вони захищають та потенційні вектори атак зловмисників. З цим допоможе моделювання загроз, яке забезпечує контекст загроз, вразливості систем та допоможе визначитись з пріоритетом захисту.

Моделювання загроз - це структурований процес виявлення та перерахування потенційних загроз, таких як вразливості або відсутність механізмів захисту, а також визначення пріоритетності заходів зі зниження рівня безпеки. Моделювання загроз має на меті надати захисникам і команді безпеки аналіз того, які засоби контролю безпеки необхідні на основі поточних інформаційних систем і ландшафту загроз, найбільш ймовірних атак, їх методології, мотивів і цільової системи.

Оцінка ризиків - це процес визначення ймовірності та серйозності ризиків, які можуть вплинути на цілі вашої організації, такі як продуктивність, репутація або відповідність вимогам. Оцінка ризиків допомагає організаціям кількісно оцінити та

визначити пріоритетність ризиків на основі їхнього потенційного впливу та ймовірності, а також визначити відповідні стратегії реагування на ризики, такі як пом'якшення, передача, уникнення або прийняття ризиків. Здійснюючи оцінку ризиків, можна розподіляти свої ресурси та зусилля більш ефективно та раціонально, а також зменшити невизначеність та нестабільність вашого середовища.

Оцінка ризиків складається з кількох ключових етапів, таких як визначення контексту та цілей організації або процесу, що підлягають захисту, а також визначення зацікавлених сторін, вимог та очікувань. Крім того, потенційні ризики повинні бути визначені за допомогою таких джерел, як реєстри ризиків, інтерв'ю або опитування. Потім ризики слід проаналізувати на основі їхнього впливу та ймовірності за допомогою таких методів, як якісний або кількісний аналіз, матриці або діаграми. Крім того, ризики повинні бути оцінені на основі їх прийнятності з використанням таких критеріїв, як рівні ризику, порогові значення або бали. Нарешті, до ризиків слід ставитися шляхом вибору та реалізації стратегій реагування на ризики, таких як пом'якшення, передача, уникнення або прийняття ризиків.

Моделювання загроз та оцінка ризиків є взаємодоповнюючими, але різними процесами, які служать різним цілям і мають різну сферу застосування. Моделювання загроз фокусується на виявленні та аналізі загроз, які можуть поставити під загрозу вашу безпеку, в той час як оцінка ризиків фокусується на оцінці та управлінні ризиками, які можуть вплинути на ваші цілі. Моделювання загроз є більш технічним і конкретним, в той час як оцінка ризиків є більш стратегічною і цілісною. Моделювання загроз зазвичай виконується до або під час розробки чи розгортання системи або додатку, тоді як оцінка ризиків зазвичай виконується після або як частина управління чи аудиту організації або процесу.

Загалом ідею в основі моделювання загроз можна описати таким чином: системи, які можуть бути атаковані, що може піти не так, що було зроблено недостатньо для зниження ризику та оцінка успішності виправлень.

Головні переваги від моделювання загроз:

- проактивна ідентифікація ризиків. Моделювання загроз дозволяє організаціям виявляти потенційні вразливості та вектори атак до того, як вони будуть використані. Усуваючи ризики на етапі проектування або на ранніх стадіях розробки, команди можуть проактивно впроваджувати засоби контролю безпеки, зменшуючи ймовірність дорогих порушень або збоїв у роботі;
- ефективний розподіл ресурсів. Визначаючи пріоритетність ризиків на основі їхнього потенційного впливу, моделювання загроз допомагає організаціям ефективніше розподіляти ресурси безпеки. Це гарантує, що час, бюджет і персонал будуть зосереджені на пом'якшенні найбільш критичних загроз, максимізуючи загальну віддачу від інвестицій в програму кібербезпеки;
- зменшення поверхні атаки. Моделювання загроз може виявити вразливості у вашій IT-екосистемі, для того щоб їх можна було швидко та ефективно виправити. Створення діаграм потоків даних і графічних зображень шляхів атак, а також визначення пріоритетів активів і ризиків - допомагають IT-командам отримати більш глибоке розуміння мережевої безпеки та архітектури;
- пріоритезація ризиків. Дані про загрози, отримані в результаті моделювання можуть використовуються для визначення критичних проблем у безпеці системи;
- покращення відповідності вимогам стандартів. Моделювання загроз допомагає компаніям дотримуватися законів і нормативних актів про конфіденційність і безпеку даних, які вимагають від організацій розуміти, як вони можуть наражати на небезпеку конфіденційні дані.

Існує п’ять основних етапів моделювання загроз (див. рис. 2.3):

1. Визначення вимог безпеки.
2. Створення схеми додатку.
3. Виявлення загроз.
4. Пом’якшення загроз.
5. Підтвердження того, що загрози були мінімізовані.

Ключовим кроком в моделюванні загроз є декомпозиція інфраструктури або конкретного додатку, яке може бути атаковане. Для моделювання ми повинні чітко та

в деталях розуміти, як працює додаток, як він взаємодіє з об’єктами в своїй системі. Тому ми маємо розуміти поведінку додатку в контексті різних ситуацій. Необхідно визначити потенційні точки входу та вразливості, а також те, як вони змінюються під час взаємодій. Для того, щоб нічого не пропустити рекомендовано використовувати діаграми для опису потоку даних. Візуальна представлення допомагає краще зрозуміти шляхи входу даних у систему, їх обробку та вихід.

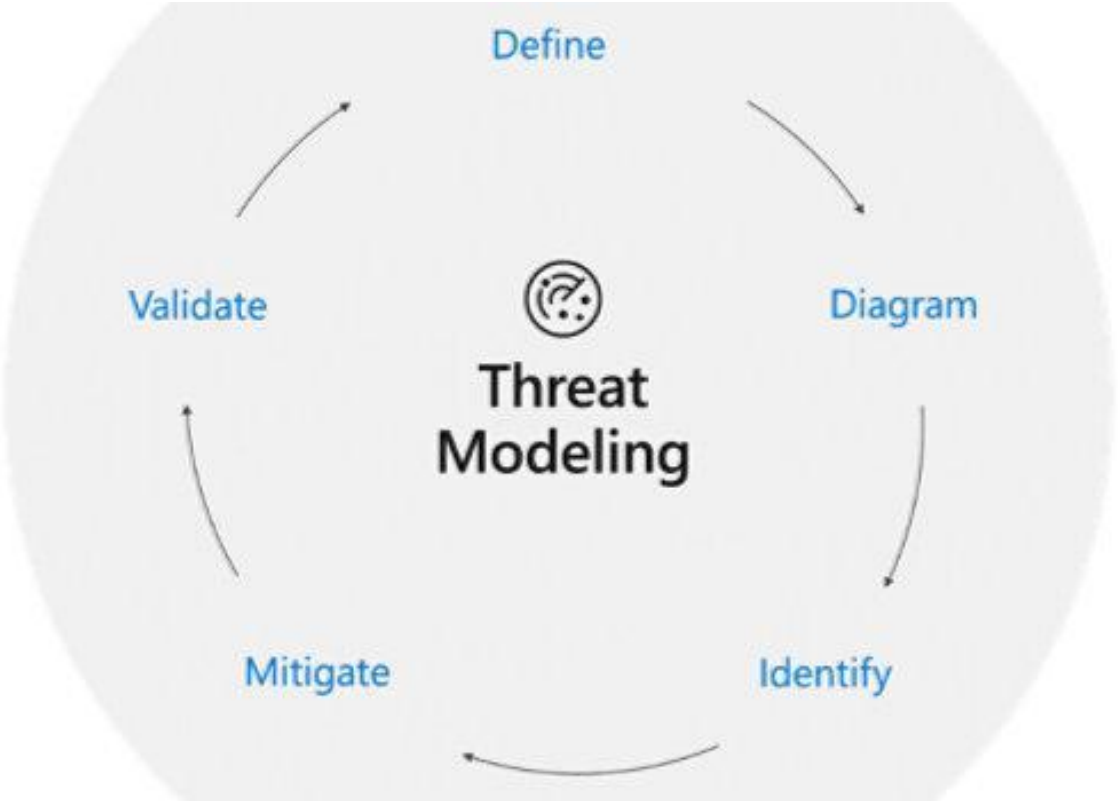

Рисунок 2.4. Етапи моделювання загроз

Існує декілька найбільш популярних методологій моделювання загроз:

1. STRIDE

STRIDE розшифровується як спуфінг, фальсифікація, відречення, розкриття інформації, відмова в обслуговуванні (DoS) та зловживання привілеями.

- Спуфінг - коли комп'ютер або людина видає себе за того, ким вона не є;
- фальсифікація – порушення цілісності даних;
- відречення втручається в процес встановлення зв'язку між дією та особою, яка її вчинила;

- розкриття інформації передбачає розголошення конфіденційної інформації;
- DoS унеможливлює використання ресурсу;
- підвищення привілеїв дозволяє доступ до ресурсу без правильної авторизації.

	Type of Threat	What Was Violated	How Was It Violated?
S	Spoofing	Authentication	Impersonating something or someone known and trusted.
T	Tampering	Integrity	Modifying data on disk, memory, network, etc.,
R	Repudiation	Non-repudiation	Claim to not be responsible for an action
I	Information Disclosure	Confidentiality	Providing information to someone who is not authorized
D	Denial of Service (DoS)	Availability	Denying or obstructing access to resources required to provide service
E	Elevation of Privilege	Authorization	Allowing access to someone without proper authorization

Рисунок 2.5. Методологія STRIDE.

STRIDE проста методологія, яка добре підходить для аналізу оцінки потенційних загроз для додатків, мережі та активів даних організації, однак методологія не враховує ймовірність загроз та не визначає пріоритети.

2. DREAD

DREAD розшифровується як потенціал шкоди, відтворюваність, можливість використання, постраждалі користувачі та можливість виявлення.

- Потенційний збиток показує, скільки шкоди може завдати негативна подія;
- відтворюваність визначає, наскільки легко відтворити атаку;
- можливість використання - це легкість, з якою зловмисник може запустити атаку;
- постраждалі користувачі включають деталізацію відсотка користувачів, на яких вплинула подія;

- виявлення визначає, наскільки легко виявити вразливість.

3. PASTA

PASTA - процес моделювання атак та аналізу загроз. Цей процес складається з семи кроків:

- Визначення цілей організації;
- визначення технічного обсягу проекту;
- декомпозиція;
- аналіз загроз;
- аналіз слабких місць і вразливостей;
- моделювання атак;
- аналіз ризику та впливу на бізнес.

PASTA це ризик-орієнтований підхід.

4. VAST

VAST - це візуальне, гнучке та просте моделювання загроз. VAST є основним елементом платформи моделювання загроз під назвою ThreatModeler. VAST інтегрується в робочі процеси, розроблені з використанням принципів DevOps.

Модель вимагає використання спеціальних інструментів, однак надає велику масштабованість та після правильної інтеграції є автоматичною.

5. Trike

Trike - це модель з відкритим вихідним кодом, який спрямований на захист системи, а не на відтворення того, як актор може її атакувати. За допомогою фреймворку Trike користувачі створюють модель програми або системи, яку вони захищають. Абревіатура CRUD використовується, щоб побачити, хто може це зробити:

- Створення даних;
- читання даних;
- оновлення даних;
- видалення даних.

Це вивчається за допомогою діаграми потоку даних. Загрози, що розглядаються, включають або підвищення привілеїв, або відмову в обслуговуванні. Trike поєднує моделювання загроз і контроль ризиків.

6. OCTAVE

OCTAVE - оперативно-критична оцінка загроз та вразливостей. Модель була розроблена університетом Карнегі-Меллона. OCTAVE складається з трьох різних етапів:

- Створення профілів загроз на основі конкретних активів;
- виявлення вразливостей в інфраструктурі;
- розробка стратегій і планів безпеки.

OCTAVE допомагає визначити критичні активи організації та їхні вразливості. Модель фокусується на бізнес цілях та на управлінню ризиками.

7. NIST

NIST складається з чотирьох етапів:

- ідентифікація системи та опис того, як вона працює, в тому числі як вона керує даними, що знаходяться в ній або залежать від неї;
- визначення відповідних векторів атак, на які буде спрямована модель;
- визначення необхідних засобів контролю безпеки для пом'якшення наслідків атак;
- аналіз створеної моделі для оцінки її ефективності.

Для ефективного моделювання загроз центр безпеки має керуватися принципами:

- Використовуйте всю доступну інформацію. Каталог користувачів, списки активів. SOC має доступ до такої інформації і вона має використовуватися.
- Автоматичне оновлення профілів. Профілі вразливостей і профілі захисту можуть оновлюватися в основному автоматично за допомогою безперервного сканування.
- Оцінка кількості вхідних даних. Більша кількість вхідних даних додає складності, а отже, вимагає більше ресурсів. Необхідно переконатися, що складність процесу відповідає вимогам та рівню загроз організації.

- Безперервне вдосконалення процесів. Регулярна оцінка ефективності та якості процесів. SOC може працювати над оптимізацією процесу.

Через використання структурованих методологій команди безпеки можуть прогнозувати загрози, визначати пріоритети захисту та зменшувати ризики. Для оптимального функціонування центру операційно безпеки моделювання потенційних загроз має бути неперервним процесом який постійно покращується завдяки збору інформації про загрози, симуляцій атак та аналізу реальних інцидентів. Шляхом інтеграції моделі загроз у повсякденне функціонування SOC організації можуть бути в курсі новітніх атак і випереджаючи зловмисників та підсилюючи загальну безпеку.

Моделювання загроз є критично важливим компонентом проективної стратегії захисту. Використовуючи структуровані методології команди безпеки можуть передбачати загрози, визначати пріоритети захисту та зменшувати ризики. Для ефективного центру операційної безпеки моделювання загроз має бути безперервним процесом, який постійно вдосконалюється за допомогою розвідки загроз, симуляцій атак та аналізу реальних інцидентів. Інтегруючи моделювання загроз у щоденні операції SOC, організації можуть випереджати зловмисників і посилити загальну стійкість своєї системи безпеки.

2.4 Інструменти автоматизації реагування на інциденти

Зростання кількості та складності кіберзагроз змушує організації шукати нові способи ефективного реагування на інциденти. Традиційні методи аналізу загроз, засновані на статичних правилах, часто не справляються з новітніми атаками, що швидко еволюціонують. В цьому контексті машинне навчання та штучний інтелект стають потужними інструментами для автоматизації та оптимізації процесу реагування на інциденти ІБ. Вони автоматизують і впорядковують завдання безпеки, підвищуючи ефективність і точність процесів безпеки. Автоматизувавши повторювані завдання, аналітики SOC можуть зосередитися на більш важливих видах діяльності, роблячи процес більш ефективним і результативним. Завдяки

використанню методів глибокого навчання та керованого машинного навчання, ШІ та МН посилюють систему безпеки організації та допомагають адаптуватися до новітніх загроз. Ці технології стають незамінними в кібербезпеці, пропонуючи більш надійний, цілісний підхід до захисту від кібератак і підвищення загальної безпеки підприємства.

Головні плюси від автоматизації процесів центру безпеки:

- покращене виявлення загроз. Автоматизація використовує штучний інтелект і аналітику даних для обробки великих обсягів даних про загрози та відсіювання хибних спрацьовувань;
- зменшення обсягу сповіщень і концентрація уваги аналітиків на справжніх загрозах, а не на хибних спрацюваннях, це прискорює ідентифікацію та розслідування справжніх загроз;
- швидше вирішення інцидентів. На додаток до автоматизації функцій виявлення загроз, автоматизація SOC також може прискорити реагування на інциденти. Заздалегідь визначені сценарії дозволяють автоматично обробляти певні загрози, скорочуючи середній час усунення;
- підвищення продуктивності SOC. Автоматизація SOC усуває ручні, повторювані завдання для персоналу служби безпеки. Це підвищує продуктивність SOC, використовуючи час і зусилля аналітиків там, де вони найбільш необхідні;
- сценарії реагування на загрози. Автоматизовані сценарії не лише підвищують швидкість, але й забезпечують узгодженість реакцій. Це допомагає зменшити кількість інцидентів безпеки, спричинених помилками, допущеними при виконанні ручних повторюваних завдань;
- більша масштабованість SOC. Автоматизація SOC підвищує масштабованість SOC, передаючи певні завдання від людей-аналітиків до автоматизованих систем. Автоматизовані сценарії набагато більш масштабовані, ніж ручні процеси;
- зменшення операційних витрат. Автоматизація SOC зменшує час, що витрачається на виконання ручних, повторюваних завдань в SOC. Як результат, організація платить менше для досягнення того ж рівня безпеки;

- підвищення задоволеності роботою. Вигорання є поширеним явищем у сфері безпеки. Зменшення кількості ручних завдань і навантаження на SOC може допомогти підвищити рівень задоволеності співробітників служби безпеки своєю роботою.

Автоматизація складний і необхідний процес, який можна інтегрувати задля вирішення таких задач:

- реагування на інциденти. SIEM та SOAR автоматизують реагування на загрози, використовуючи заздалегідь визначені сценарії для автоматичного запуску відповідних дій з усунення або стримування загроз до того, як інцидент буде проаналізований аналітиком - наприклад, тимчасове блокування облікового запису адміністратора для запобігання зловмисному доступу до чутливих активів;
- автоматизація процесу збору даних про загрози. Автоматизовані платформи розвідки загроз збирають дані з декількох каналів, фільтрують їх на предмет релевантності для організації та надають дієві рекомендації щодо ефективного виявлення та пом'якшення загроз. Індикатори загроз та індикатори атак можна відправляти у системи безпеки для детектування та захисту від загроз;
- полювання на загрози. Інструменти автоматизації розвідки загроз можуть шукати конкретні IoC, включаючи IP-адреси, пов'язані з відомими шкідливими серверами, і незвичайні шаблони входу в певні часові рамки;
- оцінка та зменшення ризиків. Автоматизація SOC допомагає захисникам знаходити і оцінювати серйозність прогалин і слабких місць в хмарних конфігураціях, API, мережах, управлінні доступом до ідентифікаційних даних і багато іншого, знімаючи важку ношу з перевірки і усунення неправильних конфігурацій в численних IT-компонентах;
- розслідування. Розслідувати минулі та поточні інциденти, виявляти першопричини та вдосконалювати майбутні плани захисту легко з автоматизованими інструментами судово-медичного аналізу. Автоматизація дозволяє легко збирати та аналізувати історичні дані, активність користувачів, мережевий трафік і зміни файлів, які допомагають відтворити послідовність подій.

Для автоматизації можуть використовуватися такі рішення:

- SIEM системи. Критично важлива технологія для центрів безпеки, яка забезпечує збір, кореляцію та аналіз подій безпеки з різних джерел. Автоматизований збір подій забезпечує організацію подіями з усіх систем інфраструктури, а автоматизовані механізми кореляції дозволяють аналізувати величезний обсяг даних та виявляти підозрілі події. Використання AI та ML дозволяє SIEM самостійно навчатися на основі попередніх інцидентів та підозрілої активності;
- SOAR - це програмне рішення, яке дозволяє командам безпеки інтегрувати та координувати окремі інструменти безпеки, автоматизувати повторювані завдання та оптимізувати робочі процеси реагування на інциденти та загрози. SOAR-системи об'єднують SIEM, Threat Intelligence, EDR, NDR та інші інструменти безпеки, створюючи централізовану платформу для управління кіберзагрозами. Використання сценаріїв автоматичного реагування дозволяє швидко та ефективно усувати загрози;
- XDR (Extended Detection and Response) – це розширена платформа для виявлення та реагування. XDR аналізує загрози на рівні кінцевих точок, мережі, хмарних сервісів та електронної пошти, створюючи єдину систему захисту. Завдяки вбудованому штучному інтелекту та машинному навчанню XDR може автоматично пріоретизувати загрози, усувати помилкові спрацьовування та блокувати атаки у реальному часі.

Порівняння рішень наведено на таблиці 2.1.

Таблиця 2.1.

Порівняння програмних засобів для автоматизації процесів IR

Категорія	XDR	SIEM	SOAR
Функції	Виявлення загроз на кількох рівнях, автоматизоване реагування	Збір логів, кореляція та моніторинг для виявлення загроз	Оркестрація та автоматизація процедур реагування на інциденти
Видимість	Кінцеві точки, мережа, хмара	Логи мережі/хостів	Інструменти безпеки та робочі процеси

	повну картину атаки		обробці після виявлення
Підхід до реагування	Вбудовані інструменти для реагування	Алерти, ручне масштабування (ескалація)	Автоматизація типових дій через playbook
Інтеграції	Нативна інтеграція з продуктами безпеки	Різноманітні джерела логів	Широка підтримка сторонніх інструментів і систем тікетів
Можливості детекції	Відображає багатоступеневі атаки, бачить	Обмежено окремими подіями або логами	Мінімальне виявлення, зосереджене на

продовження таблиці 2.1

Впровадження автоматизації в операційних центрах безпеки має низку переваг, але водночас створює низку викликів. Інтеграція автоматизації з існуючими системами і технологіями може бути складною, що вимагає ретельного планування і налаштування, щоб гарантувати сумісність і оптимізувати ефективність.

Критично важливим завданням є досягнення балансу між автоматизацією та людським наглядом. Хоча автоматизація чудово справляється з рутинними завданнями, людське розуміння залишається важливим для інтерпретації складних загроз і прийняття обґрунтованих стратегічних рішень. Досягнення належного балансу гарантує, що автоматизація SOC доповнює, а не замінює участь людини, тим самим зберігаючи надійний рівень аналізу та реагування на загрози безпеці.

Для того, щоб автоматизація SOC досягла свого повного потенціалу, необхідно, щоб SOC мав чітко визначені процеси, які піддаються автоматизації. Це вимагає ретельного визначення робочих процесів SOC та визначення повторюваних завдань, що виконуються вручну, які можна автоматизувати. Провівши такий аналіз, організації можуть сприяти більш плавній інтеграції, тим самим підвищуючи операційну ефективність і зміцнюючи спроможність SOC швидко і точно реагувати на загрози.

Сучасні системи кібербезпеки більше не можуть ефективно функціонувати без використання машинного навчання та штучного інтелекту. Обсяг даних, що обробляється SOC, постійно зростає, а традиційні підходи до виявлення загроз на

основі сигнатур та правил не здатні забезпечити швидку та точну реакцію на складні атаки. Сьогодні майже кожна сучасна система безпеки використовує машинне навчання та штучний інтелект для підвищення ефективності.

Інтеграція штучного інтелекту в операційні центри безпеки змінює ландшафт передового виявлення загроз. Здатність штучного інтелекту швидко обробляти великі масиви даних і виявляти закономірності робить його необхідним для виявлення кіберзагроз, які звичайні протоколи безпеки можуть не помітити. Серед важливих компонентів можна виділити наступні:

- виявлення аномалій. Алгоритми штучного інтелекту вміють розпізнавати нетипову поведінку, яка може свідчити про потенційні порушення безпеки;
- індикатори компрометації (IoC). Системи штучного інтелекту можуть ефективно виявляти IoC, тим самим зміцнюючи загальну систему безпеки організацій;
- розвідка загроз. ШІ відіграє вирішальну роль у створенні динамічної та прогностичної інформації про загрози, яка необхідна для проактивних заходів безпеки.

Методології машинного навчання (ML) у реагуванні на інциденти допомагають командам безпеки діяти швидше та ефективніше. Внесок ML охоплює:

- прогностичний аналіз. Використовуючи історичні дані, моделі ML можуть передбачати можливі інциденти безпеки, полегшуючи вжиття превентивних заходів;
- автоматизоване реагування. ML може оптимізувати певні операції з безпеки, мінімізуючи інтервал між виявленням загрози та реагуванням на неї;
- безперервне навчання. Алгоритми ML розвиваються з часом, підвищуючи швидкість реагування системи безпеки.

Інтеграція штучного інтелекту та машинного навчання в центри безпеки докорінно змінює їхній функціонал, що призводить до підвищення ефективності та покращення аналітики. Ця трансформація реалізується за допомогою декількох ключових механізмів:

- оркестрування безпеки. ШІ сприяє інтеграції різноманітних інструментів і процесів безпеки, тим самим оптимізуючи операційні робочі процеси;
- аналіз у реальному часі. Технології ШІ здатні миттєво обробляти та аналізувати дані про безпеку, що підвищує якість пропонованих послуг безпеки;
- підвищення ефективності управління інцидентами безпеки. ШІ значно зменшує навантаження на аналітиків з безпеки, дозволяючи їм сконцентруватися на більш складних завданнях;
- постійне вдосконалення завдяки механізмам зворотного зв'язку ШІ та ML.

Сила ШІ та ML полягає в їх здатності до безперервного навчання та адаптації, що є критично важливою вимогою у сфері кібербезпеки, яка постійно розвивається. Це постійне вдосконалення характеризується наступним:

- адаптивні алгоритми. Алгоритми і моделі ML вдосконалюються, асимілюючи нові дані про безпеку, що призводить до підвищення точності з часом;
- покращення на основі зворотного зв'язку. Системи штучного інтелекту та машинного навчання вдосконалюють свої функції на основі зворотного зв'язку з користувачами, що призводить до створення більш ефективних рішень у сфері безпеки;
- співпраця з експертами з кібербезпеки. Інформація, отримана за допомогою ШІ та ML, допомагають дослідникам і фахівцям у галузі безпеки розробляти кращі стратегії захисту.

Виявлення аномалій служить важливим елементом у сфері кібербезпеки, а алгоритми штучного інтелекту (ШІ) відіграють важливу роль у цьому процесі. Ці алгоритми систематично вивчають великі набори даних, щоб виявити відхилення від заздалегідь визначених моделей, які часто вказують на потенційні кіберзагрози. Системи штучного інтелекту встановлюють орієнтир для типової мережевої поведінки, дозволяючи їм розпізнавати нетипові дії, які можуть залишитися непоміченими аналітиками людини. Ця функціональність має життєво важливе значення для оперативного виявлення можливих порушень безпеки, в тому числі тих, які створюють невідомі загрози. Ефективність штучного інтелекту при

виявленні аномалій пояснюється його здатністю до безперервного навчання та адаптації, що підвищує його здатність виявляти навіть самі нюанси нерівномірності мережевого трафіку, дій користувача або продуктивності системи.

Аналітика поведінки користувачів та організацій (UEBA) - це рішення для кібербезпеки, яке використовує алгоритми та машинне навчання для виявлення аномалій у поведінці не лише користувачів корпоративної мережі, але й маршрутизаторів, серверів та кінцевих точок у цій мережі. UEBA прагне розпізнати будь-яку особливу або підозрілу поведінку - випадки, коли є відхилення від звичайних повсякденних шаблонів. UEBA значно покращує виявлення складних і витончених кіберзагроз, особливо тих, які не піддаються традиційним заходам безпеки. Його підхід до поведінкового аналізу є особливо практичним у боротьбі з новими загрозами та угрупованнями, що робить його все більш важливим для бізнесу.

Машинне навчання значно підвищує захищеність, сприяючи негайному реагуванню на виявлені загрози. На відміну від звичайних систем, які часто вимагають втручання людини, алгоритми ML можуть швидко реагувати на потенційні небезпеки. Ця швидка реакція є життєво важливою для зменшення наслідків кібератак. Системи ML здатні автономно виконувати такі заходи, як ізоляція скомпрометованих мереж, обмеження доступу для підозрілих користувачів або розгортання додаткових протоколів безпеки для усунення загрози. Можливості ML в реальному часі не тільки забезпечують швидку реакцію на встановлені загрози, але й дозволяють адаптуватися до нових і розвиваються ризиків, що робить його важливим компонентом сучасних стратегій кібербезпеки.

Тим не менш, їх впровадження піднімає різні питання безпеки, особливо щодо управління та захисту конфіденційних організаційних даних. Основною проблемою, пов'язаною з використанням рішень AI та ML від зовнішніх постачальників, є необхідність доступу цих систем до великої кількості організаційних даних для навчання та оптимізації. Ці дані часто охоплюють чутливу та критичну інформацію щодо систем, інфраструктури та операцій організації. Будь-яке розкриття цієї інформації стороннім особам може представляти значні ризики.

Головним питанням безпеки є вірогідність несанкціонованого використання цієї інформації. Коли дані надсилаються на зовнішні сервери для навчання моделі, вони можуть бути перехоплені або доступ до даних можуть отримати зловмисники після отримання несанкціонованого доступу до систем на яких зберігаються дані. Крім того, існує ризик того, що постачальник може неправильно використовувати дані, будь то ненавмисно або зі зловмисними намірами. Ця проблема особливо виражена, коли моделі ШІ використовуються для завдань кібербезпеки, оскільки зловмисники потенційно можуть використовувати слабкі місця в самій моделі ШІ.

З точки зору інформаційної безпеки, найбільш розумною стратегією є використання локальних моделей ШІ, які навчаються на даних, що зберігаються в інфраструктурі організації. Зберігаючи дані всередині, ризик витоку даних значно знижується, тим самим значно зменшуючи ризики.

Однак локальні моделі мають певні обмеження. Істотним недоліком є залежність від даних, які доступні виключно в рамках організації. Отже, навчання цих моделей обмежується обмеженим набором даних, який не може охоплювати різноманітні та складні моделі, знайдені у великих наборах даних. Це обмеження може призвести до того, що місцеві моделі будуть менш точними, менш гнучкими та менш оснащеними для управління широким спектром сценаріїв порівняно з моделями, навченими на більш широких та різноманітних наборах даних.

З іншого боку, моделі штучного інтелекту та машинного навчання, пропоновані сторонніми постачальниками, зазвичай отримують вигоду від навчання великим обсягам даних, які часто надходять з декількох галузей, середовищ та додатків. Цей доступ до комплексного набору даних дозволяє цим моделям бути більш стійкими та адаптованими, оскільки вони можуть навчатися з більш широкого спектру моделей, поведінки та ситуацій. Моделі, що надаються постачальниками, як правило, більш просунуті і здатні узагальнювати в більш широкому діапазоні контекстів, підвищуючи їх ефективність у практичному застосуванні.

Таким чином, в той час як місцеві моделі визначають пріоритетність безпеки даних, зберігаючи конфіденційну інформацію всередині організації, вони часто не відповідають глибині та широті своїх можливостей навчання. На відміну від них,

хмарні моделі, які використовують великі набори даних, часто перевершують локальні моделі за точністю, передбачуваною потужністю та адаптивністю. Цей баланс між безпекою та продуктивністю є важливим фактором, який слід враховувати при виборі між локальними моделями та хмарними рішеннями.

Впроваджуючи подібні системи з глибокою інтеграцією ШІ та високим рівнем автоматизації, критично важливо пам'ятати про потенційний вплив на людський фактор у довгостроковій перспективі. Хоча автоматизація значно підвищує ефективність та швидкість, існує ризик поступової втрати практичних навичок та відчуття реальних загроз у аналітиків, якщо їхня робота зводиться лише до перегляду результатів роботи автоматизованих систем. Без постійної практики аналізу даних, розслідування складних інцидентів та вивчення нових технік атак, кваліфікація персоналу може знижуватися. Тому вкрай необхідно забезпечувати програми безперервного навчання та тренування для аналітиків, проводити періодичний аудит рішень, прийнятих ШІ, та підтримувати середовище, де глибокі знання та експертиза людини залишаються найвищою цінністю. Автоматизація та ШІ повинні розглядатися як інструменти, що підсилюють можливості людини, а не повністю їх замінюють.

2.5 Метрики ефективності процесів реагування на інциденти

Метрики оцінки ефективності SOC допомагають організації зрозуміти наскільки добре він використовує ресурси, а також оцінює ефективність дій з реагування на інциденти та відновлення, що проводяться командами SOC:

- середній час виявлення (MTTD) - це критична метрика, яка визначає середню тривалість, необхідну для команди SOC для виявлення інциденту. Більш низький показник MTTD означає високу продуктивність, що відображає оперативність команди при своєчасному виявленні та усуненні інцидентів, що зменшує вплив загрози;
- середній час вирішення інциденту (MTTR) доповнює MTTD, оцінюючи ефективність та доцільність зусиль SOC з реагування на інциденти. Більш низький

MTTR означає швидкі та високоефективні процеси вирішення інцидентів. MTTR, як правило, охоплює такі завдання, як дослідження першопричини, впровадження засобів правового захисту та виконання процедур відновлення;

- вартість інциденту дозволяє організаціям кількісно оцінити як прямі, так і непрямі витрати, пов'язані з інцидентом. Прямі витрати охоплюють витрати, такі як час і ресурси, виділені для виявлення та реагування, а також судові витрати. Непрямі витрати пов'язані зі збитками доходів, пов'язаними з оборотом клієнтів, регуляторними штрафами, репутаційною шкодою та іншими пов'язаними факторами. Крім того, можуть бути додаткові витрати, такі як витрати, пов'язані з модернізацією програмного забезпечення та профілактичними заходами проти майбутніх інцидентів.

Висновки за розділом 2

Розвиток інформаційних технологій призводить до збільшення складності систем, що несе з собою збільшення поверхності атак та потенційних вразливостей у системах. Це вимагає від спеціалістів з безпеки створювати нові підходи до захисту інфраструктури, які зможуть відповідати на сучасні виклики. Реагування на інциденти це вже довгий час не просто процес очікування атаки, а потім її ліквідація. Це обширний процес покращення систем безпеки, проактивний пошук загроз, обмін інформацією про загрози та постійне вдосконалення систем безпеки.

Розвідка загроз є одним з основних інструментів сучасного реагування. Організації діляться інформацією про загрози: нове шкідливе забезпечення, вектори атак, вразливості в системах та особливо способи покращення безпеки. Знаючи техніки та тактики зловмисників організації можуть підготуватися до атак та ліквідувати їх в найкоротші терміни. Використання Threat Intelligence дозволяє детектувати, як нові атаки так і атаки, які вже пройшли повз системи безпеки і залишаються в тіні. Проактивний пошук загроз дозволяє знайти аномальну активність в мережі не покладаючись на наявні механізми безпеки. Підхід дозволяє змінити погляд на систему, ніби вона вже скомпроментована. Використання

аналітичних методів, гіпотез, інформації про атаки дозволяє знаходити загрози всередині організацій та усувати їх.

Моделювання загроз та оцінка ризиків дуже важливі процеси в безпеці кожної організації, саме вони дозволяють ідентифікувати активи, їх призначення та слабкі сторони. Використовуючи такі методології як STRIDE, PASTA, DREAD організації можуть визначити поверхню атаки, спрогнозувати методи атак, потенційні цілі, що вкаже на необхідні дії з покращення систем та дозволить розробити нові сценарії з реагування на інциденти. Структурований підхід до моделювання загроз та аналізу ризиків сприяє зниженню кількості інцидентів завдяки виявленню проблем у системах захисту, що дозволяє визначити пріоритети та ефективно спрямувати ресурси для підвищення безпеки систем.

Організації спрямовують величезну кількість ресурсів на автоматизацію задач, що дозволяє зменшити кількість роботи в майбутньому, збільшити її ефективність і відсіяти рутинну роботу. Тому автоматизація процесів відділу безпеки є логічним розвитком систем безпеки. Створення таких систем як SOAR, XDR, SIEM значно покращує безпеку організації, пришвидшує реагування на загрози, дозволяє інтегрувати різні засоби безпеки та дозволяє виконувати реагування на всі загрози через одну систему.

Не менш важливу роль в сучасних системах безпеки відіграють AI та ML. Вони дозволяють значно покращити виявлення аномалій, аналізують величезну кількість даних та знаходять складні атаки. Ручне виконання цих завдань є непосильною задачею, тому все більше ці технології інтегруються у системи безпеки. На сьогоднішній день компанії активно інтегрують їх, що безпосередньо збільшує якість детектування загроз, допомагає аналітикам швидше приймати рішення, автоматично реагують на прості загрози. Поведінковий аналіз користувачів та інші технології на основі ШІ дозволяють прогнозувати атаки, що значно підвищує рівень кіберзахисту організацій.

Використання всіх цих процесів та технологій вимагає проведення чіткого аналізу ефективності процесів реагування на інциденти. Саме тому в стандартах визначено використання спеціальних метрик, які дають можливість визначити

швидкість реагування, ефективність роботи та загальні тенденції атак. Основними метриками є: MTTD, MTTR, MTTA&A, FPR, FNR, вартість інциденту, кількість ескалацій та кількість інцидентів.

РОЗДІЛ 3
АВТОМАТИЗІЯ ПРОЦЕСУ РЕАГУВАННЯ НА ІНЦИДЕНТИ НА ОСНОВІ ШТУЧНОГО ІНТЕЛЕКТУ

3.1 Розробка методу автоматизованого реагування

В умовах постійного розвитку інформаційно-телекомунікаційних систем, зростання кількості та складності кіберзагроз, ефективне та своєчасне реагування на інциденти інформаційної безпеки стає критично важливим завданням для будь-якої організації. У цих умовах ефективність роботи SOC стає ключовим фактором захисту організацій. Однак, традиційні підходи до моніторингу та реагування, що значною мірою покладаються на людський аналіз, стикаються з серйозними викликами. Саме тут критично проявляється проблематика людського фактору. По-перше, безперервний потік однотипних сповіщень, значна частина яких є хибними спрацюваннями, неминуче призводить до втоми від сповіщень. Аналітики стають менш уважними, ризик пропустити дійсно критичну подію значно зростає. По-друге, час реакції при ручній обробці є суттєвим недоліком. Процес, що включає отримання сповіщення, пошук та аналіз релевантних логів, кореляцію даних, прийняття рішення та ручне виконання реагування, може займати від кількох хвилин до годин. Цей час є критичним, оскільки зловмисник може встигнути перейти до наступних, більш небезпечних фаз атаки. По-третє, неуважність та потенційні помилки, викликані втомою або високим навантаженням, можуть призвести до неправильної класифікації інциденту або помилок при конфігурації засобів захисту. Нарешті, забезпечення цілодобового моніторингу кваліфікованими аналітиками є складним та дорогим завданням.

Таким чином, надмірна залежність від ручної праці при обробці масових, низькорівневих сповіщень стає слабкою ланкою в системі захисту. Для подолання цих викликів необхідні інструменти, здатні автоматизувати рутинні процеси,

забезпечити швидку та консистентну реакцію, і, що найважливіше, зменшити негативний вплив людського фактору.

Впровадження та реалізація автоматизованого реагування на інциденти інформаційної безпеки на основі штучного інтелекту вирішує всі вище наведені проблеми. Такий механізм буде здатний виконувати первинний аналіз логів, виявляти підозрілу активність та відсіювати явні хибні спрацювання. Підхід дозволить не замінити аналітика, а доповнити та підсилити його можливості, звільняючи людські ресурси від монотонної роботи та дозволяючи їм сфокусуватися на складних інцидентах, проактивному пошуку загроз за допомогою threat hunting та вдосконаленні загальної стратегії захисту.

Рішення безпеки такі як IDS, IPS, SIEM вже давно можуть детектувати сканування сканерами і генерувати величезну кількість сповіщень на цю активність. Після кожного такого сповіщення аналітик безпеки змушений перевіряти величезну кількість подій в SIEM для визначення цілей сканування та іншої підозрілої активності. Всі ці дії є дуже часозатратними та ресурсоємними. Значна частина таких сповіщень є хибними, викликані легітимною активністю або неправильною логікою сповіщень. Водночас завантаженість аналітиків впливає на швидкість прийняття рішень та на їх правильність.

Все це створює нагальну потребу в автоматизації процесів реагування на інциденти. Метод автоматизації реагування на інциденти на основі штучного інтелекту описує послідовний підхід до створення програм, що автоматизують реагування на інциденти інформаційної безпеки з використанням верифікації або прийняття рішень за допомогою штучного інтелекту. Етапи реалізації методу (див. рис. 3.1):

1. Інтеграція програмного застосунку у систему управління інцидентами ІБ.
2. СУІБ звертається до програмного застосунку для аналізу.
3. Застосунок відправляє події у ШІ.
4. Застосунок отримує аналіз від ШІ.

5. На основі аналізу виконується повернення до пункту один або запуск модулю реагування

6. СУУІБ логує всі виконані дії програмним застосунком.

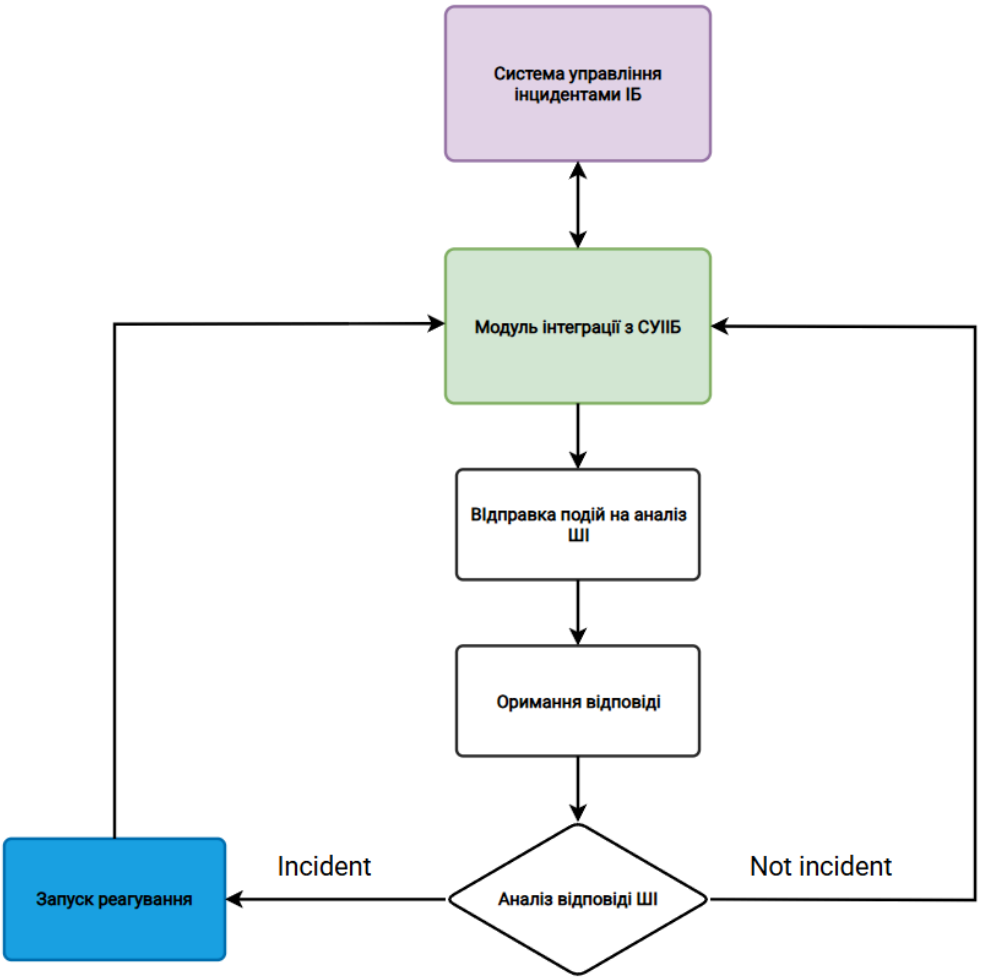

Рисунок 3.1. Сукупність кроків для реалізації методу

Першим етапом будь-якої кібератаки є розвідка. Збір інформації є критично важливим етапом, не знаючи особливості системи та її слабкості атака не зможе бути успішною. Одним з видів розвідки є сканування портів інформаційних систем цілі, за допомогою якого можна визначити працюючі сервіси, версії та їх вразливості за допомогою CVE. І хоча саме по собі сканування портів не несе загрози для організації воно є чітким індикатором зловмисної активності та підготовки до більш серйозних дій.

Тому на основі розробленого методу в даній роботі було розроблено механізм для автоматизації реагування на сканування портів за допомогою штучного інтелекту. Було визначено основні вимоги і цілі програми.

Програма має виконувати такі функції:

- аналізувати лог файли;
- шукати підозрілу активність на основі патернів;
- верифікувати потенційні загрози за допомогою штучного інтелекту;
- виконувати автоматичне реагування шляхом блокування IP адрес;
- вести логування всіх підтверджених інцидентів та хибних спрацювань для подальшого аналізу та коректування;
- мати модульну структуру для легкого масштабування.

У наступних підрозділах буде детально розглянуто архітектуру, програмну реалізацію та результати роботи запропонованої системи, що демонструє практичні кроки до автоматизації процесу реагування на інциденти, мінімізуючи людський фактор та знижуючи час на реагування.

3.2 Програмна реалізація механізму автоматизованого реагування на інциденти

Відповідно до розробленого методу наступним логічним кроком розвитку методу є реалізація програмного застосунку на його базі. Розроблена програма побудована на модульному принципі за допомогою мови програмування Python, що забезпечує гнучкість, легкість інтеграції з системними утилітами та доступ до сучасних бібліотек для роботи з API.

Штучний інтелект є верифікатором шкідливої активності. Він має проводити аналіз інформації, визначати чи подія інцидентом на основі заданої інформації в запиті та запускати автоматичне реагування. Для цієї задачі було обрано Gemini 1.5 flash на основі швидкості відповіді, легкої інтеграції, точності відповідей та через безкоштовність. Програма буде мати окремий модуль з API ключем, який буде відповідати за взаємодію з штучним інтелектом.

Для реалізації програми було використано:

- Python 3: Основна мова програмування, обрана через її простоту, велику кількість стандартних та сторонніх бібліотек, зручність роботи з текстом та системними процесами;
- Бібліотеки Python: subprocess (для запуску tail та iptables), re (для регулярних виразів), datetime, os, collections (defaultdict, deque);
- google-generativeai: Офіційна бібліотека Python від Google для взаємодії з API моделей Gemini;
- python-dotenv: Бібліотека для зручного керування конфігураційними параметрами (зокрема, API-ключем) через файл .env;
- tail: Стандартна утиліта Linux/Unix для моніторингу змін у файлах;
- iptables: Стандартний інструмент командного рядка для керування фаєрволом у ядрі Linux. Використовується для динамічного блокування IP-адрес;
- Системні логи Linux: /var/log/kern.log як джерело подій про мережеві з'єднання.

Опис всіх файлів програми:

1. main.py - серце програми, це модуль моніторингу та оркестрації (див. додаток В). Головний скрипт, який відповідає за запуск та координацію роботу всіх модулів. Ініціює моніторинг лог файлів за допомогою команди tail, передає події з заздалегідь визначений лог файлів на аналіз штучного інтелекту, отримує результати аналізу від штучного інтелекту і на основі результату запускає автоматичне реагування (див. рис. 3.2). Також в модулі реалізована фіксація часу витраченого на запуск і відпрацювання кожного модуля, для подальшого аналізу ефективності програми.

```
if is_attack:
    # Actions for confirmed attack
    target_ip = extract_dst_ip(logs)
    alert_title = f"🚨 ALERT: Potential Port Scan CONFIRMED by AI ({detection_time_log}) 🚨"
    # Print alert to console
    print("\n" + "="*40); print(alert_title);
    print(f"Source IP: {src_ip}");
    print(f"Target IP: {target_ip}");
    print(f"Trigger Logs Count: {len(logs)}");
    print("-" * 20);
    print("AI Analysis:"); print(verdict_line);
    [print(expl) for expl in explanation_lines];
    print("="*40 + "\n");
    # CALL BLOCKING PLAYBOOK
    print(f"[{datetime.now()}] Attack confirmed. Initiating response playbook...")
    # --- TIMING: Block Action ---
    block_successful = block_ip(src_ip)
    if block_successful:
        t4_block_end = time.perf_counter()
        total_response_time = t4_block_end - t1_detect_trigger
        print(f"    [TIMING] Total Response Time (Detection to Block): {total_response_time:.4f} seconds")
    else:
        print(f"    [TIMING] IP Blocking failed for {src_ip}. Cannot calculate full response time.")
    # --------------------------
    # Log to file
    log_blocked_ip(alert_title, verdict_line, explanation_lines, src_ip, logs)
```

Рисунок 3.2 Код запуску реагування

2. port-scanning.py - модуль для виявлення сканування портів (див. додаток Д). В ньому реалізована логіка виявлення сканування портів на основі підключень до більше ніж 10 портів за 10 хвилин (див. рис. 3.3). Також має механізм захисту від великої кількості сповіщень, на одну IP адресу приходить тільки одне сповіщення за 10 хвилин.

```
if current_connection_count >= threshold and current_distinct_ports >= min_distinct_ports:
    last_alert_time = alert_timestamps.get(ip)

    # Check throttling for this specific IP
    if not last_alert_time or now - last_alert_time > timedelta(seconds=alert_interval_seconds):
        # Alert triggered for IP 'ip'
        print(f"[Port Scanner] Potential scan detected from Source IP {ip}.")
        print(f"  Conditions met: Connections={current_connection_count} (>= {threshold}), Distinct Ports={current_distinct_ports} (>= {min_distinct_ports})")

        logs_to_send = list(potential_attack_logs[ip]) # Get logs to send
        alert_timestamps[ip] = now # Update last alert time for this IP

        # Clear state for this specific IP after alert
        ip_connections[ip].clear()
        potential_attack_logs[ip].clear()
        ip_ports[ip].clear()

        return (ip, logs_to_send) # Return detected IP and logs
    else:
        # Throttled
        return None
return None
```

Рисунок 3.3 Логіка виявлення сканування портів

3. gemini-integration.py - модуль інтеграції ШІ у програму (див. додаток Г). Забезпечує взаємодію з штучним інтелектом у хмарі – Gemini. Надсилає запит (див. рис 3.4), що містить лог файли підозрілої активності, через API, отримує та повертає текстову відповідь моделі з вердиктом атака це чи ні з поясненням.

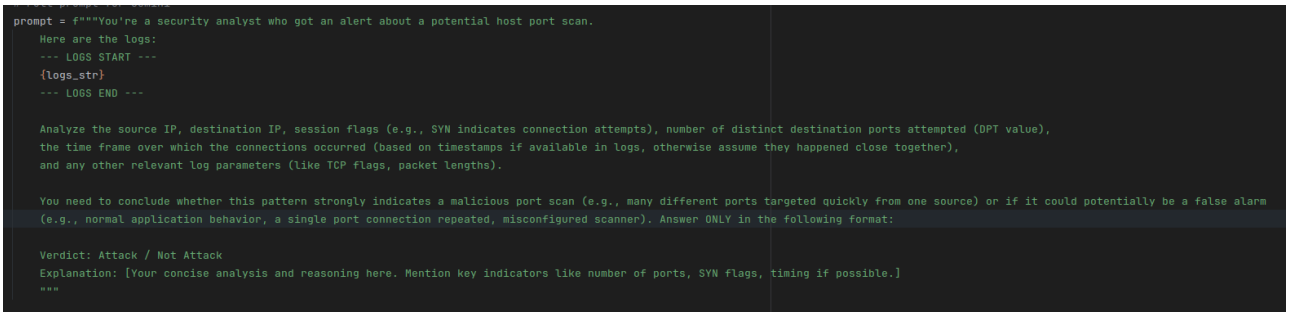

Рисунок 3.4 Запит до штучного інтелекту

4. Scanning_response_playbook.py - сценарій для реагування на інцидент (див. додаток Е). Виконує блокування IP адреси через iptables (див. рис. 3.5) та записує детальну інформацію про реагування у файл scans-ip-block.txt. При детектування хибного спрацювання записує інформацію у файл port_scanning_falsepositives.txt.

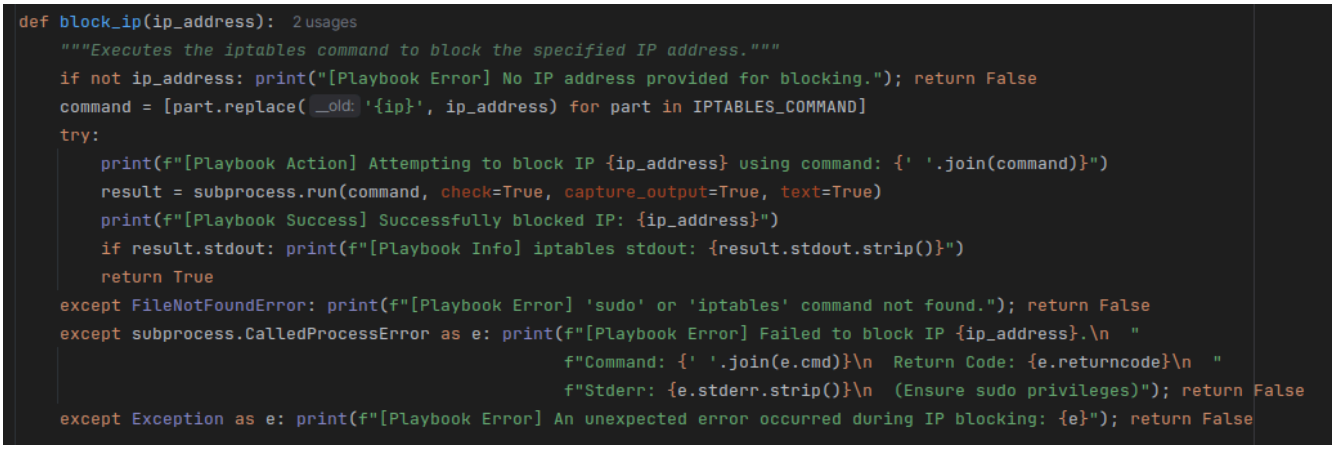

Рисунок 3.5 Блокування IP адрес через iptables

Загалом реалізований програмний застосунок містить 4 модулі і текстові файли для API ключа Gemini та логування всіх дій програми. Структура програми наведена на рисунку 3.6

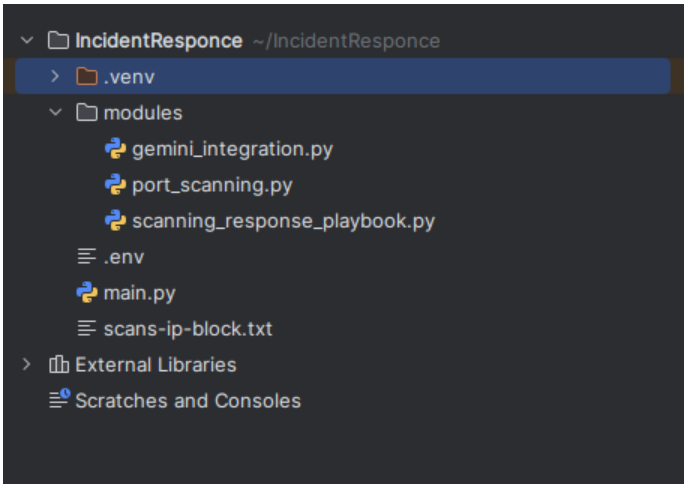

Рисунок 3.6 Структура програми

Результат роботи програми наведено на рисунках 3.7 та 3.8.

```
/home/dtabachenko/IncidentResponce/.venv/bin/python /home/dtabachenko/IncidentResponce/main.py
[Info] Gemini API configured successfully.
Starting Incident Response Automation (Real-time Mode)...
Monitoring log file: /var/log/kern.log using 'tail -f'
Detection Threshold: >=10 connections AND >=10 distinct ports within 600 seconds
Alert Interval per IP: 600 seconds
------------------------------
[Info] Read access to log file verified.
[Info] Checking for sudo/iptables access...
[Info] Basic sudo/iptables access check successful.
------------------------------
[2025-04-19 20:26:19.245357] Starting 'tail -f -n 0 /var/log/kern.log'...
[2025-04-19 20:26:19.246906] 'tail' process started successfully (PID: 23704).
```

Рисунок 3.7 Запуск програми

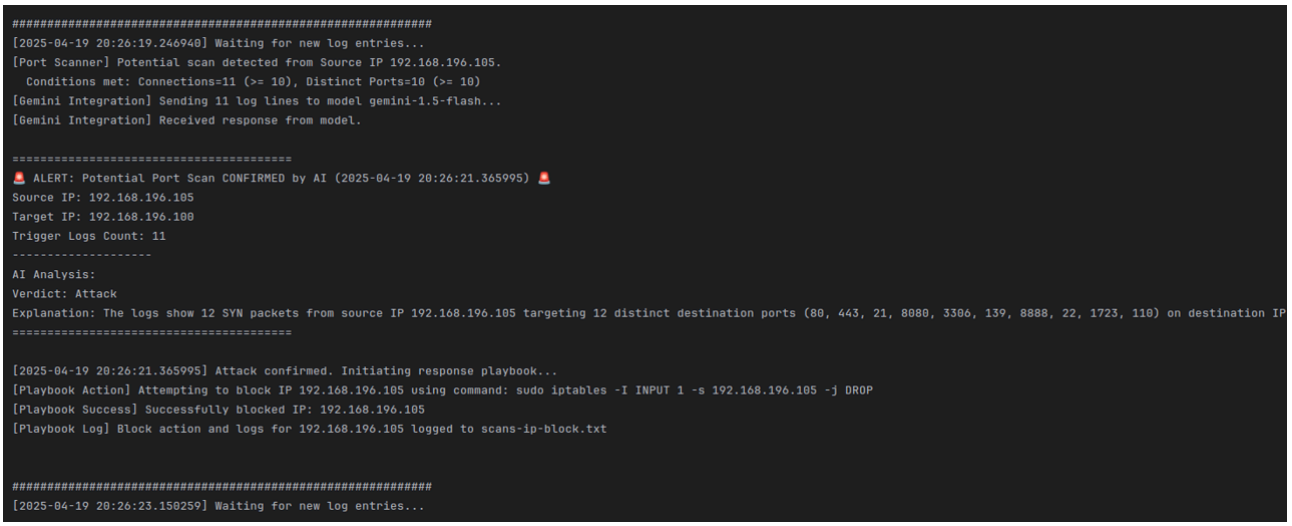

Рисунок 3.8 Детектування потенційної атаки та запуск сценарію реагування.

Після запуску сценарію з реагування інформація про прийняті дії записується в файл scans-ip-block.txt (див. рис. 3.9).

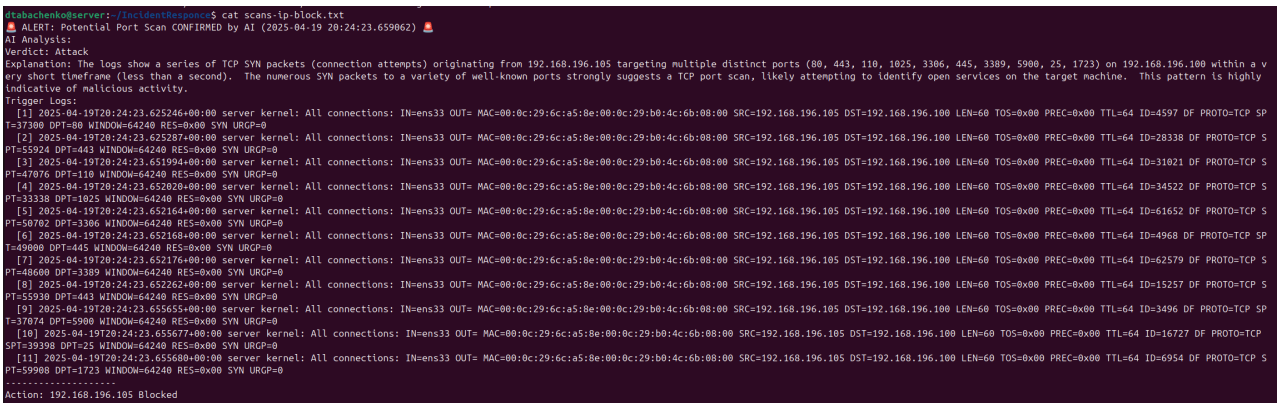

Рисунок 3.9 Вміст файлу scans-ip-block.txt.

Також перевіряємо iptables і бачимо правило на блокування IP адреси 192.168.196.105 (див. рис. 3.10).

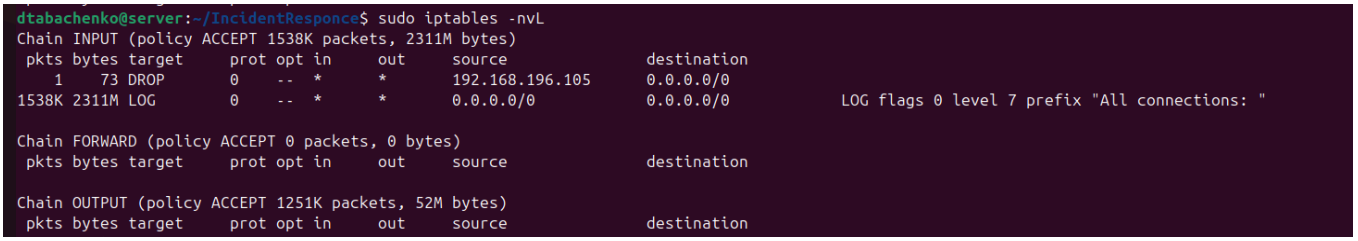

Рисунок 3.10 Перевірка блокування IP адреси через iptables.

3.3 Переваги та недоліки програми

Розроблена програма має ряд переваг в порівнянні з ручним аналізом та ручним реагуванням:

- мінімальний час на прийняття рішення;
- зменшення навантаження на аналітиків;
- підвищення точності реагування. Виключаємо людський фактор: втома, неуважність, різна кількість досвіду;
- можливість масштабування. В метод можна додавати безліч нових модулів для виявлення різних типів кібератак та сценаріїв для реагування;

Недоліки програми:

- залежність від формату логів. Система жорстко прив'язана до формату логів, що генеруються через iptables у директорії /var/log/kern.log;
- залежність від ШІ. Ефективність верифікації напряму залежить від доступності та точності моделі Gemini. Можливі як хибно негативні, так і хибні позитивні (неправильне підтвердження) вердикти ШІ. Також слід враховувати можливу затримку відповіді від API та потенційні витрати, пов'язані з використанням комерційних моделей ШІ;
- продуктивність. Використання tail -f є ефективним, але на системах з екстремально високою інтенсивністю генерації логів (тисячі подій на секунду) читання та обробка кожного рядка в Python може стати слабким місцем;
- обмеження виявлення. На даний момент програма детектує тільки один патерн: багато з'єднань до різних портів. Вона не виявляє інші типи сканувань (Stealth, FIN) або інші види атак;
- ризик конфіденційності. Передача корпоративних даних у відкриту ШІ несе з собою ризики витоку цих даних у майбутньому. Тим паче дані потенційно можуть бути використані компанією, яка володіє ШІ.

Разом з недоліками є багато шляхів вдосконалення програми для майбутнього використання:

- використання локального ШІ. Наприклад модель Llama. Такі моделі достатньо розумні для аналізу даних і не передають інформацію стороннім компаніям;
- інтеграція з SIEM системою для централізованого управління інцидентами.
- розширення логіки виявлення для розпізнавання різних типів сканувань та, потенційно, інших видів атак;
- реалізація механізму тимчасового блокування або автоматичного розблокування IP-адрес через заданий період;
- додавання перевірки репутації IP-адреси (за чорними списками) перед блокуванням як додаткового фактора верифікації;
- додавання можливості конфігурування "білих списків" IP-адрес, які ніколи не повинні блокуватися.

Висновки за розділом 3

У розділі було розглянуто проблематику сучасного реагування на інциденти, а саме величезну залежність від людського фактору у вигляді: втоми, великої кількості сповіщень, неуважність, недостатню кваліфікацію. Реалізація механізму реагування на інциденти на основі штучного інтелекту може вирішити ці проблеми і принести довгоочікувані зміни у процес IR.

В розділі була реалізована модульна програма для реагування на конкретний тип інцидентів - сканування портів. На основі різних модулів програма виконує сканування лог файлів, які на основі попередніх налаштувань містять в собі інформацію про мережеві з'єднання та детектує підозрілу активність на основі кількості з'єднань за визначений проміжок часу та кількості портів. Підозрілі лог файли відправляються на аналіз штучного інтелекту, інтеграція з яким реалізована за допомогою окремого модуля. Завдяки верифікації ШІ програма отримує вердикт і запускає модуль реагування, який в залежності від отриманої інформації запускає модуль реагування, який блокує шкідливу IP адресу та записує інформацію про

інцидент у файл або пропускає сценарій реагування у разі, якщо події не підпадають під інцидент. Ключовими перевагами методу є швидкість, автономність, можливість автоматизувати реагування на рутинні сповіщення, підвищення точності реагування завдяки верифікації ШІ. Програма має обмеження у вигляді залежності від формату лог файлів, швидкість роботи з великими лог файлами, обробка лише одного тип інциденту, використання хмарного ШІ та відсутній механізм автоматичного розблокування IP адрес. Все це вказує на можливі покращення програми особливо збільшення кількості сценаріїв реагування, а її інтеграція з SIEM системами може значно спростити життя аналітиків та фахівців з безпеки.

РОЗДІЛ 4
ОЦІНКА ЕФЕКТИВНОСТІ ЗАПРОПОНОВАНОГО РІШЕННЯ

4.1 Методологія оцінки ефективності

Оцінка ефективності запропонованого рішення буде базуватися на основі стандартів індустрії і рекомендацій NIST. Метрики мають допомогти кількісно оцінити продуктивність розробленої програми визначаючи її головні переваги та недоліки. На основі аналізу метрик ефективності в пункті можна визначити, що для даного рішення найкращими метриками будуть: швидкість детектування інциденту, швидкість реагування, точність прийнятих рішень.

Саме тому у даному підрозділі визначаються три основні критерії оцінки:

- Середній час виявлення (MTTD);
- середній час усунення (MTTR) ;
- рівень хибних спрацювань (FPR).

Середній час виявлення - у контексті розробленого механізму, MTTD визначається як середній час, що проходить від моменту появи першого релевантного запису в лог-файлі, що належить до сканування серверу, що згодом буде класифіковано як інцидент, до моменту, коли внутрішній механізм детектування

системи спрацьовує та сигналізує про потенційний інцидент, відправляючи логи на аналіз штучного інтелекту.

Середній час усунення - метрика вимірює середній час, необхідний програмі для усунення виявленої загрози або завершення процесу реагування на інцидент. В контексті розробленої програми дією усунення є блокування IP-адреси зловмисника та запис даних про блокування у файл.

Рівень хибних спрацювань – ключова метрика для оцінки точності системи та її впливу на легітимний трафік. Визначає частку подій або сесій легітимної активності, які були помилково класифіковані системою як шкідливі та призвели до блокування легітимного трафіку. В рамках розробленої програми FPR наближається до нуля. В програмі вже є чітко визначені критерії для пошуку сканування портів через що хибні спрацювання можуть бути тільки відносно легітимних хостів, які не були додані у білий список. В рамках тестування використовувалися тільки 2 сервера для атаки і захисту, відповідно FPR в цих ідеальних умовах 0%.

Для аналізу метрик було використано версію програми з розширеним виводом інформації у консоль, яка фіксує час кожної дії програми (див. рис. 4.1):

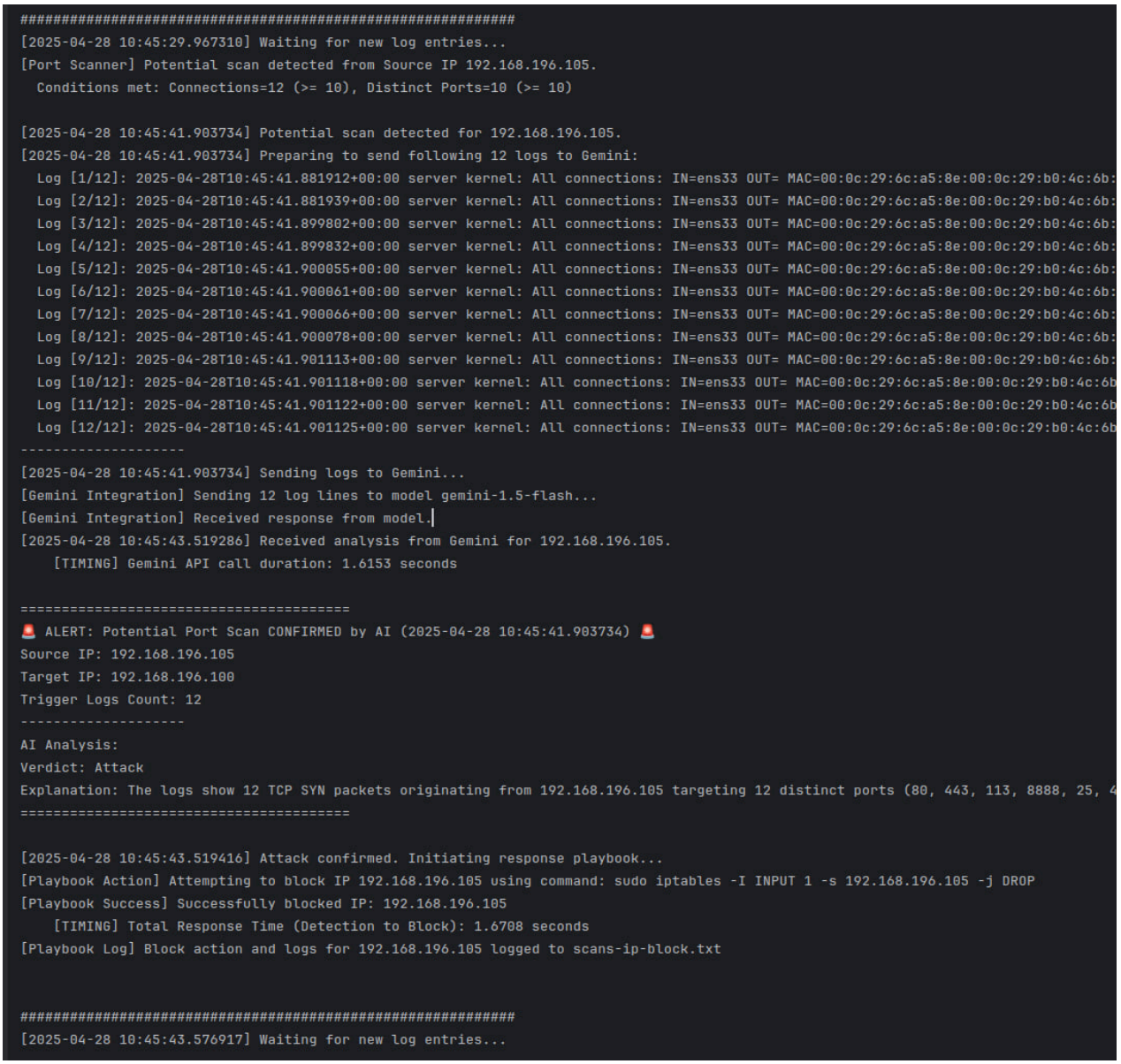

Рисунок 4.1 Робота програми з метриками

Для статистики було проаналізовано такі дані:

1 запуск:

MTTD = 0.8218 секунд

MTTR = 1.6708 секунд

2 запуск:

MTTD = 0.1036 секунд

MTTR = 1.3178 секунд

3 запуск:

MTTD = 0.8218 секунд

MTTR = 1.9540 секунд

Таблиця 4.1

Метрики ефективності застосунку

Метрика	Запуск 1	Запуск 2	Запуск 3	Середнє
MTTD Час виявлення	0.8218	0.1036	0.8218	0.5824
Час аналізу	1.6153	1.2623	1.6589	1.51
MTTR Час Усунення	1.6708	1.3178	1.9540	1.5677

4.2 **Аналіз результатів оцінки та порівняння з альтернативами**

Отримані результати MTTR < 2 секунд є ключовим досягненням, оскільки вони демонструють здатність системи виконати усунення практично миттєво після підтвердження атаки. Оцінка MTTD також свідчить про ефективність первинного детектування.

Порівняння з ручним реагуванням:

- Швидкість: Різниця є колосальною. Ручна обробка аналогічного інциденту отримання сповіщення від SIEM/IDS, перевірка логів, прийняття рішення, створення правила фаєрвола зазвичай займає від кількох хвилин у найкращому випадку при наявності вільного аналітика до годин. Автоматизована система виконує весь цикл (від детектування до блокування) за ~1.5-2 секунди. Це кардинально скорочує "вікно можливостей" для зловмисника.
- Навантаження: Система повністю автоматизує аналіз та реагування на даний тип подій, вивільняючи час аналітиків. Зменшення кількості алертів, що потребують ручного перегляду, є прямою перевагою.
- Повторюваність: Автоматизація гарантує однакове виконання процедури для кожного інциденту, виключаючи помилки через втому чи неуважність людини.

Стандартні галузеві звіти наприклад: IBM Cost of a Data Breach Report, зазвичай вимірюють MTTD та MTTR для всього життєвого циклу успішної кібератаки. Згідно зі звітом IBM за 2024 рік, середній час на ідентифікацію зламу

складає близько 194 днів, а на його стримування – близько 64 днів. Для фінансового сектору ці показники становлять 168 та 51 день відповідно. Розраховані показники MTTR та оціночний MTTD відносяться до дуже специфічної, ранньої стадії потенційної атаки. Вони не є прямо порівнянними з багатоденними показниками ідентифікації та стримування повноцінного зламу. Через що показники отримані при тестуванні програми важко порівняти з реальними даними.

Штучний інтелект та машинне навчання сьогодні є невід'ємною частиною більшості передових рішень у сфері кібербезпеки. Однак роль та глибина інтеграції ШІ можуть суттєво відрізнятися. У багатьох сучасних платформах – SIEM, SOAR, XDR, NGFW/IPS – ШІ переважно використовується як потужний допоміжний інструмент для аналітика:

- Кореляція та пріоритезація: ШІ аналізує величезні обсяги даних з різних джерел, виявляє складні зв'язки між подіями та допомагає пріоритезувати найбільш критичні сповіщення, зменшуючи навантаження на аналітиків;
- виявлення аномалій: ML моделі вивчають нормальну поведінку систем та користувачів і сигналізують про підозрілі відхилення, які можуть вказувати на загрозу, не описувану чіткими правилами;
- збагачення контекстом: Системи автоматично збирають додаткову інформацію про інциденти (репутація IP, дані Threat Intelligence, інформація про вразливості), надаючи аналітику повнішу картину;
- рекомендації та автоматизація рутини: SOAR-платформи можуть використовувати ШІ для пропозиції релевантних сценаріїв реагування або автоматизації окремих кроків аналізу;

Сучасні системи пропонують інтеграцію штучного інтелекту для допомоги в прийняті рішень, інтегруючи ШІ як додаток до програми, однак розвиток цієї технології вже на даний момент дозволяє використовувати її для повноцінного реагування на інциденти. В механізмі ШІ виконує роль аналітика і є повноцінний членом процесу реагування на інциденти, фактично це дозволяє замінити аналітиків 1 рівня, підвищуючи ефективність процесу реагування на інциденти. Так як саме величезна кількість хибних сповіщень займає більшу частину часу аналітиків,

спричиняє перевтому, яка тягне за собою неуважність і помилки. Також хибні сповіщення з'їдають величезну частину бюджетів SOC команд, бо аналіз хибних сповіщень це витрата часу, який можна спрямувати на реальні інциденти.

Висновки за розділом 4

У розділі було проведено оцінку ефективності механізму реагування на інциденти інформаційної безпеки на основі штучного інтелекту. Було визначено 3 основні метрики для оцінки: середній час виявлення - час від першого підозрілого лог файлу до класифікації його як інциденти, середній час усунення - час необхідний програмі для блокування шкідливої IP адреси, рівень хибних спрацювань - частка помилково ідентифікованих подій, як інциденти. Аналіз показав дуже високу продуктивність програми, а саме такі результати: MTTD - 1.56 секунди, MTTR - 0.31 секунди, FPR ~ 0%.

Високу ефективність показала метрика MTTR в той час як аналітику необхідно вручну перевіряти лог файли, підтвердити що це інцидент і виконати реагування, програма виконує це все в неймовірно маленький проміжок часу. FPR наближається до нуля, через логіку програми та визначення інцидентів через чіткі критерії.

Рішення є повністю автоматизованим і самостійно оброблює інциденти сканування портів без потреби втручання людини. На відміну від багатьох існуючих рішень безпеки, де ШІ це просто допоміжний інструмент безпеки, який шукає інформацію по запиту спеціаліста або аналізує конкретну інформацію, в реалізованому рішення ШІ це повноцінний учасник процесу від якого залежить ідентифікація загроз та подальше реагування. В програмі штучний інтелект є аналітиком, що може в майбутньому дати змогу замінити аналітиків першого рівня для реагування на прості інциденти, що дозволить зберегти бюджет відділу на більш кваліфікований фахівців або на більш просунуті системи захисту.

Розроблений механізм показав ряд переваг автоматизації над ручним аналізом та реагуванням. Проте впровадження таких рішень несе за собою ризики неправильної ідентифікації загроз та неправильного реагування. Саме тому для

таких рішень необхідно підтримувати високо кваліфікацію кадрів для перевірки рішень ШІ. Такі інтеграції мають супроводжуватися постійним розвитком кадрів, щоб у разі необхідності спеціалісти могли взяти ситуацію під контроль без допомоги штучного інтелекту.

ВИСНОВКИ

У роботі було досліджено та проаналізовано сучасні підходи до реагування на інциденти. На основі чого було виявлено способи підвищення ефективності центру операційної безпеки. Швидкість та точність реагування одні з основних критеріїв, які можливо суттєво підвищити за допомогою інтеграції штучного інтелекту у процеси IR. Саме тому було розроблено метод автоматизованого реагування на інциденти інформаційної безпеки на основі штучного інтелекту.

У першому розділі було проаналізовано нормативно правову базу пов’язаних з процесом IR, а саме серію міжнародних стандартів ISO 27000, а саме 27001, 27002 та 27035. Перші два стандарта мають багато контролів, які безпосередньо впливають на ефективне реагування на загрози, такі як: керування інцидентами, збір доказів, плани роботи заходів безпеки під час збоїв та інші. В той час як 27001 та 27002 підтримують належний рівень СУІБ в цілому. ISO 27035 визначає фази реагування, принципи та процеси, різні настанови і правильну координацію під час процесу IR. В той час як стандарти ISO містять більш загальний зміст і орієнтований на стратегічне планування процесів. NIST SP 800-61 орієнтований на технічних спеціалістів і містить конкретні рекомендації щодо побудови процесів, детальні рекомендації по реагуванню, системи, які мають бути впровадженні. Загалом ISO це концепція, а NIST це її практична реалізація, саме тому для побудови ефективного процесу необхідно брати найкраще з цих документів.

У другому розділі було проаналізовано сучасні підходи до реагування на інциденти. Адже реагування це не тільки очікування атаки і її нейтралізація, це активне виявлення загроз за допомогою Threat Hunting, коли за допомогою гіпотез та наявної інформації про атаки, спеціалісти з кібербезпеки шукають загрози всередині організації. Це обмін інформацією між організаціями - Threat Intelligence, що допомагає організаціям підготуватися до кібератак та закрити вразливі місця. Моделювання загроз для виявлення слабких місць у системі та цілей потенційних зловмисників за допомогою таких методологій, як: STRIDE, DREAD, PASTA, VAST.

Та одне з найголовніших - автоматизація, адже швидкість реагування напряму впливає на збитки організацій під час кібератак.

Однак всі ці підходи та інструменти можна значно покращити шляхом інтеграції у них штучного інтелекту. Це вирішить такі проблеми як: втома від великої кількості сповіщень, неуважність, час реакції - людський фактор. Моніторинг системи аналітиками є складним та дорогим завданням. Натомість передача реагування на ШІ може вирішити ці проблеми. На основі цього в 3 розділі було розроблено метод автоматизованого реагування на інциденти на основі штучного інтелекту. На основі методу було реалізовано програмний застосунок для автоматизованого реагування на події сканування інфраструктури. Програма реалізована на мові програмування Python з модульною архітектурою та забезпечує моніторинг системних лог файлів в реальному часі, детектування підозрілої активності за заздалегідь визначеними критеріям, верифікацію атаки за допомогою передачі інформації в ШІ та запуск автоматизованого реагування на основі аналізу штучного інтелекту, а саме блокування шкідливих IP адрес через iptables та детальне збереження інформації про всі інциденти та хибні спрацювання. Загалом програма містить 4 модулі, текстові файли для API ключа, логування інформації про заблоковані загрози та інформації про хибні спрацювання.

В 4 розділі було визначено основні критерії ефективності розробленої програми за такими метриками: середній час виявлення, середній час усунення, рівень хибних спрацювань. На основі декількох інцидентів середній час виявлення склав 1.56 секунди, середній час усунення 0.3 секунди, а рівень хибних спрацювань наближається до нуля через чітко визначені критерії пошуку потенційних інцидентів. Результати є ключовою перевагою програми. Неймовірно швидкий час усунення інцидентів, в той час як ручна обробка сповіщень з аналізом лог файлів в SIEM, перевірка інформації, час на прийняття рішення зазвичай займають від хвилини до годин. Така швидкість реагування значно скоротить вікно можливостей зловмисників при інтеграції ШІ в реагування на інші типи інцидентів, що легко реалізувати завдяки модульній структурі програми. Багато сучасних інструментів вже використовує машинне навчання та штучних інтелект у реагуванні на інциденти,

однак перевагою програми є використання ШІ, як повноцінного учасника процесу IR.

Однак, незалежно від результатів продуктивності програми, кваліфіковані кадри все ще пріоритет при побудові процесу реагування на інциденти. Висока продуктивність не забезпечує на 100% правильну роботу програми. Спеціаліст з кібербезпеки мусить перевіряти всі рішення прийняті штучним інтелектом та всю інформацію на основі, якої було прийняте рішення. Навіть при дуже підозрілій активності штучний інтелект може неправильно ідентифікувати загрозу і не відреагувати вчасно або навпаки навіть при невпевненості заблокувати загрозу. Що несе в собі величезні ризики для безперервної роботи організацій.

Ще одним ризиком є конфіденційність при використанні ШІ від сторонніх постачальників. Такі моделі більш просунуті та краще аналізують контекст, проте їх використання несе потенційні ризики витоку даних, так як інтеграція такого рішення у системи безпеки відкриває повне бачення на систему безпеки організації. Саме тому з точки зору ІБ найбільш розумною стратегією є використання локальних моделей ШІ, які навчаються на даних, що зберігаються в інфраструктурі організації. Зберігаючи дані всередині, ризик витоку даних значно знижується, тим самим значно зменшуючи ризики.

Таким чином, всі завдання, які були поставлені відповідно до мети магістерської роботи були виконані в повному обсязі, а саме:

- аналіз нормативно правової бази в сфері реагування на інциденти;
- аналіз сучасних підходів реагування на інциденти;
- розробка методу автоматизованого реагування на інциденти інформаційної безпеки на основі штучного інтелекту;
- реалізація програмного застосунку для автоматизованого реагування на інциденти на базі розробленого методу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Report | Cyber Security Report 2024 [Електронний ресурс]. – Режим доступу: https://www.checkpoint.com/resources/report-3854/report--cyber-security-report-2024

2. Microsoft Digital Defense Report 2024 [Електронний ресурс]. – Режим доступу: https://cdn-dynmedia-1.microsoft.com/is/content/microsoftcorp/microsoft/final/en-us/microsoft-brand/documents/Microsoft%20Digital%20Defense%20Report%202024%20%281%29.pdf

3. Cost of a Data Breach Report 2024 [Електронний ресурс]. – Режим доступу: https://www.ibm.com/reports/data-breach

4. ISO 27002-2022 [Електронний ресурс]. – Режим доступу: https://www.slideshare.net/ChristianAquino52/iso-270022022pdf

5. ISO 27002 essentials: a comprehensive overview [Електронний ресурс]. – Режим доступу: https://nordlayer.com/learn/iso/iso-27002/

6. The Ultimate Guide to ISO 27002 [Електронний ресурс]. – Режим доступу: https://www.isms.online/iso-27002/#:~:text=ISO%2FIEC%2027002%3A2022%20is,implementing%20an%20ISO%2027001%20ISMS

7. ISO/IEC27035-1 [Електронний ресурс]. – Режим доступу: https://www.amnafzar.net/files/1/ISO%2027000/ISO%20IEC%2027035-1-2016.pdf

8. ISO/IEC27035-2 [Електронний ресурс]. – Режим доступу: https://www.amnafzar.net/files/1/ISO%2027000/ISO%20IEC%2027035-2-2016.pdf

9. ISO/IEC27035-3 [Електронний ресурс]. – Режим доступу: https://cdn.standards.iteh.ai/samples/74033/21625771e477493a86b821ea41886c22/ISO-IEC-27035-3-2020.pdf

10. Security Incident Management according to ISO 27035 [Електронний ресурс]. – Режим доступу: https://www.linkedin.com/pulse/security-incident-management-according-iso-27035-dipen-das-/

11. ISO/IEC 27035-1:2023— Information Security Management [Електронний ресурс]. – Режим доступу: https://blog.ansi.org/iso-iec-27035-1-2023-information-security-management/

12. NIST.SP.800-61r2 [Електронний ресурс]. – Режим доступу: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf

13. What is threat intelligence? [Електронний ресурс]. – Режим доступу: https://www.ibm.com/think/topics/threat-intelligence#:~:text=Threat%20intelligence%E2%80%94also%20called%20cyberthreat,detecting%2C%20mitigating%20and%20preventing%20cyberattacks.

14. Cyber Threat Intelligence Explained [Електронний ресурс]. – Режим доступу: https://www.crowdstrike.com/en-us/cybersecurity-101/threat-intelligence/

15. Actionable threat intelligence at Google scale [Електронний ресурс]. – Режим доступу: https://cloud.google.com/security/products/threat-intelligence

16. Introduction to Cyber Threat Hunting [Електронний ресурс]. – Режим доступу: https://www.crowdstrike.com/en-us/cybersecurity-101/threat-intelligence/threat-hunting/

17. What is threat hunting? [Електронний ресурс]. – Режим доступу: https://www.ibm.com/think/topics/threat-hunting

18. Threat Hunting Definition [Електронний ресурс]. – Режим доступу: https://www.fortinet.com/resources/cyberglossary/threat-hunting

19. Threat Hunting vs. Threat Intelligence: Differences and Synergies [Електронний ресурс]. – Режим доступу: https://www.exabeam.com/explainers/information-security/threat-hunting-vs-threat-intelligence-differences-and-synergies/

20. What is cyber threat hunting? [Електронний ресурс]. – Режим доступу: https://www.microsoft.com/en-us/security/business/security-101/what-is-cyber-threat-hunting

21. Threat Modeling: Designing for Security [Електронний ресурс]. – Режим доступу:

https://public.magendanz.com/Temp/Threat%20Modeling%20-%20Shostack,%20Adam.pdf

22. Threat Modeling [Електронний ресурс]. – Режим доступу: https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling

23. Definition Of Threat Modeling [Електронний ресурс]. – Режим доступу: https://www.fortinet.com/resources/cyberglossary/threat-modeling#:~:text=Threat%20modeling%20involves%20identifying%20and,they%20may%20impact%20the%20network.

24. What Is Threat Modeling & What Are Its Advantages? [Електронний ресурс]. – Режим доступу: https://www.eccouncil.org/cybersecurity-exchange/incident-handling/what-is-threat-modeling-what-are-its-advantages-ecih-ec-council/

25. Threat Modeling [Електронний ресурс]. – Режим доступу: https://www.eccouncil.org/threat-modeling/

26. Practical Threat Modelling for SOCs [Електронний ресурс]. – Режим доступу: https://www.linkedin.com/pulse/practical-threat-modelling-socs-rob-van-os/

27. What is SOC automation? Optimize Your SOC Workflow [Електронний ресурс]. – Режим доступу: https://radiantsecurity.ai/learn/soc-automation/

28. How Machine Learning is Transforming Security Operations Centers (SOC) [Електронний ресурс]. – Режим доступу: https://industrywired.com/how-machine-learning-is-transforming-security-operations-centers-soc/

29. Role of Machine Learning in Modern SOC Operations [Електронний ресурс]. – Режим доступу: https://itbutler.sa/blog/role-of-machine-learning-in-modern-soc-operations-2/

30. Building a Machine Learning Model for the SOC, by the Input from the SOC, and Analyzing it for the SOC [Електронний ресурс]. – Режим доступу: https://doi.org/10.1109/VIZSEC.2018.8709231

31. SOAR vs. MDR vs. SOC: Choosing The Right Security Strategy [Електронний ресурс]. – Режим доступу: https://radiantsecurity.ai/learn/soar-vs-mdr-vs-soc/

32. What is SOAR? A Complete Guide to SOAR Platforms [Електронний ресурс]. – Режим доступу: https://swimlane.com/blog/what-is-soar/

33. What is UEBA (User and Entity Behavior Analytics)? [Електронний ресурс]. – Режим доступу: https://www.paloaltonetworks.com/cyberpedia/what-is-user-entity-behavior-analytics-ueba

34. SOC Metrics: Security Metrics & KPIs for Measuring SOC Success [Електронний ресурс]. – Режим доступу: https://www.splunk.com/en_us/blog/learn/security-operations-metrics.html

35. 6 Metrics & KPIs for measuring SOC success [Електронний ресурс]. – Режим доступу: https://www.digitalxraid.com/6-soc-metrics-kpis/

36. SOC Metrics: The Key Metrics & KPIs to Measure Your SOC Success [Електронний ресурс]. – Режим доступу: https://radiantsecurity.ai/learn/soc-metrics-and-kpis/

ДОДАТОК А

СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ РОБОТИ

Тези наукових доповідей

1. Апробація роботи: Костюченко В., Табаченко Д., Білоконь І. Зменшення втоми від тривог в SOC: Покращення реагування на інциденти за допомогою автоматизації та штучного інтелекту. VIII Міжнародна науково-практична конференція «Проблеми кібербезпеки інформаційно-комунікаційних систем» (PCSICS). 2025. С. 109-110

ДОДАТОК Б

Приклад форми для повідомлення про подію ІБ

Повідомлення про подію ІБ			
1. Дата події		3. Пов'язана подія та/або ідентифікаційний номер інцидента	
2. Номер події			
4. Дані про особу, яка повідомляє про подію			
4.1 Ім’я		4.2 Адреса	
4.3 Організація		4.4 Департамент	
4.5 Телефон		4.6 Електронна адреса	
5. Опис події інформаційної безпеки			
5.1 Опис події: Що сталося? Як сталося? Чому сталося? Які активи постраждали? Вплив на бізнес Які вразливості?			
6. Деталі про подію інформаційної безпеки			
6.1 Дата та час події			
6.2 Дата та час коли подія була виявлена			
6.3Дата та час повідомлення про подію			
6.4. Чи було реагування на цю подію	ТАК ☐	НІ ☐	
6.5. Якщо так, вкажіть, як довго тривала подія			

ДОДАТОК В

Код файлу main.py

```
import time
import re
import sys
import os
import subprocess
from datetime import datetime

# Add 'modules' folder to Python's search path
current_dir = os.path.dirname(os.path.abspath(__file__))
modules_dir = os.path.join(current_dir, 'modules')
if modules_dir not in sys.path:
    sys.path.insert(0, modules_dir)

try:
    # Import necessary functions from modules
    from modules.port_scanning import process_log_line
    from modules import gemini_integration
    # Import ALL playbook functions
    from modules.scanning_response_playbook import block_ip, log_blocked_ip,
log_false_positive
except ImportError as e:
    print(f"Error importing modules: {e}")
    print("Ensure modules/__init__.py exists and modules folder is accessible.")
    sys.exit(1)
```

```
    # --- Settings ---
    LOG_FILE_PATH = "/var/log/kern.log"
    SCAN_THRESHOLD = 10          # Min number of connections
    MIN_DISTINCT_PORTS = 10      # Min number of unique ports
    SCAN_WINDOW_SECONDS = 600    # Time window
    ALERT_INTERVAL_SECONDS = 600  # Throttling interval

    DST_IP_PATTERN = re.compile(r'DST=([\d.]+)')

    # Helper functions
    def extract_dst_ip(log_lines):
        """Extracts the first found DST IP from the provided logs."""
        if not log_lines: return "Unknown"
        for line in log_lines:
            match = DST_IP_PATTERN.search(line)
            if match:
                return match.group(1)
        return "Unknown"

    def start_tail_process(log_file):
        """Starts the 'tail -f' process and returns it."""
        try:
            print(f"[{datetime.now()}] Starting 'tail -f -n 0 {log_file}'...")
            process = subprocess.Popen(
                ['tail', '-f', '-n', '0', log_file],
                stdout=subprocess.PIPE, stderr=subprocess.PIPE, bufsize=1,
universal_newlines=True
            )
```

```
            print(f"[{datetime.now()}] 'tail' process started successfully (PID:
{process.pid}).")
            return process
        except FileNotFoundError:
            print(f"[Error] 'tail' command not found.")
            sys.exit(1)
        except Exception as e:
            print(f"[Error] Failed to start 'tail' process: {e}")
            sys.exit(1)


    def main():
        """Main program function."""
        print("Starting Incident Response Automation (Real-time Mode)...")
        print(f"Monitoring log file: {LOG_FILE_PATH} using 'tail -f'")
        print(f"Detection Threshold: >={SCAN_THRESHOLD} connections AND
>={MIN_DISTINCT_PORTS} distinct ports within {SCAN_WINDOW_SECONDS}
seconds")
        print(f"Alert Interval per IP: {ALERT_INTERVAL_SECONDS} seconds")
        print("-" * 30)
        print("-" * 30)

        tail_process = start_tail_process(LOG_FILE_PATH)
        if not tail_process: sys.exit(1)

        # Main log processing loop
        try:
            print(); print(); print("#" * 60)
            print(f"[{datetime.now()}] Waiting for new log entries...")
            for line in iter(tail_process.stdout.readline, ''):
```

```
                if not line:
                    print(f"[{datetime.now()}] Tail process stream ended.")
                    break
                line = line.strip()
                if not line:
                    continue

                scan_details = process_log_line(
                    line,
                    threshold=SCAN_THRESHOLD,
                    window_seconds=SCAN_WINDOW_SECONDS,
                    alert_interval_seconds=ALERT_INTERVAL_SECONDS,
                    min_distinct_ports=MIN_DISTINCT_PORTS
                )

                # If potential scan detected
                if scan_details:
                    # --- TIMING START ---
                    t1_detect_trigger = time.perf_counter()
                    detection_time_log = datetime.now()
                    # --------------------

                    src_ip, logs = scan_details
                    print(f"\n[{detection_time_log}] Potential scan detected for {src_ip}.")

                    # Print logs before sending to Gemini
                    print(f"[{detection_time_log}] Preparing to send following {len(logs)}
logs to Gemini:")
                    for i, log_line in enumerate(logs): print(f"  Log [{i+1}/{len(logs)}]:
{log_line}")
```

```
                print("-" * 20)

                # Send logs to Gemini and get result
                print(f"[{detection_time_log}] Sending logs to Gemini...")
                t2_gemini_start = time.perf_counter()
                analysis_result = gemini_integration.analyze_logs_with_gemini(logs)
                t3_gemini_end = time.perf_counter()
                gemini_duration = t3_gemini_end - t2_gemini_start
                print(f"[{datetime.now()}] Received analysis from Gemini for {src_ip}.")
                print(f"    [TIMING] Gemini API call duration: {gemini_duration:.4f}
seconds")

                # Parse Gemini response
                # --- INITIALIZATION POINT ---
                verdict_line = ""
                explanation_lines = []
                explanation_start_index = -1
                is_attack = False
                is_not_attack = False
                # --------------------------

                lines = analysis_result.splitlines()
                for i, l in enumerate(lines):
                    stripped_l = l.strip()
                    # Find verdict
                    if stripped_l.lower().startswith("verdict:"):
                        verdict_line = stripped_l
                        verdict_lower = verdict_line.lower()
                        if "attack" in verdict_lower and "not attack" not in verdict_lower:
                            is_attack = True
```

```
                elif "not attack" in verdict_lower:
                    is_not_attack = True
                elif stripped_l.lower().startswith("explanation:"):
                    explanation_start_index = i
                    break

            # Collect explanation lines
            if explanation_start_index != -1:
                explanation_lines = [line.strip() for line in
lines[explanation_start_index:]]

            # --- RESPONSE LOGIC ---S
            if is_attack:
                # Actions for confirmed attack
                target_ip = extract_dst_ip(logs)
                alert_title = f"🚨 ALERT: Potential Port Scan CONFIRMED by AI
({detection_time_log}) 🚨"
                # Print alert to console
                print("\n" + "="*40); print(alert_title);
                print(f"Source IP: {src_ip}");
                print(f"Target IP: {target_ip}");
                print(f"Trigger Logs Count: {len(logs)}");
                print("-" * 20);
                print("AI Analysis:"); print(verdict_line);
                [print(expl) for expl in explanation_lines];
                print("="*40 + "\n");
                # CALL BLOCKING PLAYBOOK
                print(f"[{datetime.now()}] Attack confirmed. Initiating response
playbook...")
                # --- TIMING: Block Action ---
```

```
                    block_successful = block_ip(src_ip)
                    if block_successful:
                        t4_block_end = time.perf_counter()
                        total_response_time = t4_block_end - t1_detect_trigger
                        print(f"    [TIMING] Total Response Time (Detection to Block): {total_response_time:.4f} seconds")
                    else:
                        print(f"    [TIMING] IP Blocking failed for {src_ip}. Cannot calculate full response time.")
                    # --------------------------
                    # Log to file
                    log_blocked_ip(alert_title, verdict_line, explanation_lines, src_ip, logs)

                elif is_not_attack:
                    # Actions for "Not Attack" verdict (False Positive)
                    target_ip = extract_dst_ip(logs)
                    print(f"\n[{datetime.now()}] AI classified event from {src_ip} as 'Not Attack'. Logging as False Positive.")
                    print("-" * 20); print("AI Analysis:"); print(analysis_result); print("-" * 20)
                    # Log False Positive to file
                    log_false_positive(detection_time_log, src_ip, target_ip, logs, analysis_result)

                elif verdict_line:
                    # Handle other verdicts (Error, Blocked etc.)
                    print(f"\n[{datetime.now()}] AI Analysis Result for {src_ip} (Verdict: Error/Blocked or other):")
                    print(analysis_result); print("-" * 20)
                else:
```

```
                    # If verdict could not be parsed at all
                     print(f"\n[{datetime.now()}] Could not parse AI verdict for {src_ip}.
Raw response:"); print(analysis_result); print("-" * 20)

                print(); print(); print("#" * 60)
                print(f"[{datetime.now()}] Waiting for new log entries...")

            # --- End of main for loop ---
            print(f"[{datetime.now()}] Main loop finished processing stdout.")
            stderr_output = tail_process.stderr.read()
            if stderr_output:
                print(f"[{datetime.now()}] 'tail' process error output:\n{stderr_output}")

        # --- Exception Handling ---
        except KeyboardInterrupt:
            print(f"\n[{datetime.now()}] Termination signal received (Ctrl+C). Shutting
down.")
        except Exception as e:
            print(f"\n[{datetime.now()}] An unexpected error occurred in the main loop:
{e}")
            import traceback
            traceback.print_exc()

        # Cleanup
        finally:
            if 'tail_process' in locals() and tail_process and tail_process.poll() is None:
                print(f"[{datetime.now()}] Terminating 'tail' process (PID:
{tail_process.pid})...")
                tail_process.terminate()
                try:
```

```
                tail_process.wait(timeout=5)
                print(f"[{datetime.now()}] 'tail' process terminated.")
            except subprocess.TimeoutExpired:
                print(f"[{datetime.now()}] 'tail' process did not terminate gracefully,
killing...")
                tail_process.kill()
                print(f"[{datetime.now()}] 'tail' process killed.")
        elif 'tail_process' in locals() and tail_process:
            print(f"[{datetime.now()}] 'tail' process already terminated (return code:
{tail_process.poll()}).")
        else:
            print(f"[{datetime.now()}] 'tail' process was not running or already
handled.")

        print(f"[{datetime.now()}] Script shutdown complete.")


if __name__ == "__main__":
    main()
```

ДОДАТОК Г

Код файлу gemini_integration.py

```
import os
import google.generativeai as genai
from google.generativeai import types
from google.generativeai.types import HarmCategory, HarmBlockThreshold
from dotenv import load_dotenv

load_dotenv()

GEMINI_API_KEY = os.environ.get("GEMINI_API_KEY")
IS_CONFIGURED = False

if not GEMINI_API_KEY:
    print("[Warning] GEMINI_API_KEY not found in environment variables.")
    print("          Gemini analysis will be skipped.")
else:
    try:
        genai.configure(api_key=GEMINI_API_KEY)
        IS_CONFIGURED = True
        print("[Info] Gemini API configured successfully.")
    except Exception as e:
        print(f"[Error] Failed to configure Gemini API: {e}")
        print("        Check your API key and environment setup.")
        print("        Gemini analysis will be skipped.")
        IS_CONFIGURED = False

MODEL_NAME = "gemini-1.5-flash"
```

```
def analyze_logs_with_gemini(log_lines):

    # Sends logs to Gemini for analysis.
    # Checking a checkbox instead of a non-existent function
    if not IS_CONFIGURED:
        return "Verdict: Not Checked (API Not Configured)\nExplanation: Gemini API key is missing or configuration failed during startup."

    if not log_lines:
        return "Verdict: Not Checked (No Logs)\nExplanation: No log lines were provided for analysis."

    logs_str = "\n".join(log_lines)

    # Prompt
    prompt = f"""You're a security analyst who got an alert about a potential host port scan.
    Here are the logs:
    --- LOGS START ---
    {logs_str}
    --- LOGS END ---

        Analyze the source IP, destination IP, session flags (e.g., SYN indicates connection attempts), number of distinct destination ports attempted (DPT value), the time frame over which the connections occurred (based on timestamps if available in logs, otherwise assume they happened close together), and any other relevant log parameters (like TCP flags, packet lengths).
```

```
You need to conclude whether this pattern strongly indicates a malicious port scan (e.g., many different ports targeted quickly from one source) or if it could potentially be a false alarm (e.g., normal application behavior, a single port connection repeated, misconfigured scanner). Answer ONLY in the following format:

Verdict: Attack / Not Attack

Explanation: [Your concise analysis and reasoning here. Mention key indicators like number of ports, SYN flags, timing if possible.]

"""

try:

model = genai.GenerativeModel(MODEL_NAME)

print(f"[Gemini Integration] Sending {len(log_lines)} log lines to model {MODEL_NAME}...")

# Making a request to Gemini
response = model.generate_content(
    prompt,
    generation_config=genai.types.GenerationConfig(
        temperature=0.5
    ),
)

print(f"[Gemini Integration] Received response from model.")

# Response handling
if response.parts:
    return response.text.strip()
```

```
            elif response.prompt_feedback and response.prompt_feedback.block_reason:
                block_reason_name = "Unknown"
                try:
                    block_reason_name = response.prompt_feedback.block_reason.name
                except AttributeError:
                    block_reason_name = str(response.prompt_feedback.block_reason)
                    print(f"[Warning] Gemini analysis blocked due to safety concerns: {block_reason_name}")
                return f"Verdict: Analysis Blocked\nExplanation: The analysis was blocked due to safety concerns: {block_reason_name}"
            elif response.candidates:
                finish_reason = response.candidates[0].finish_reason
                finish_reason_name = "Unknown"
                try:
                    finish_reason_name = finish_reason.name
                except AttributeError:
                    finish_reason_name = str(finish_reason)

                if finish_reason_name != 'STOP':
                    print(f"[Warning] Gemini response generation finished unexpectedly. Reason: {finish_reason_name}")
                    return f"Verdict: Error\nExplanation: Gemini response generation finished unexpectedly. Reason: {finish_reason_name}"
                else:
                    print("[Warning] Gemini returned an empty response despite normal completion (STOP).")
                    return "Verdict: Error\nExplanation: Gemini returned an empty response despite normal completion."
            else:
```

```
                print("[Error] Gemini response structure is unexpected (no parts, no candidates).")
                return "Verdict: Error\nExplanation: Gemini returned an unexpected response structure."

        except Exception as e:
            error_message = f"Gemini API call failed: {type(e).__name__}: {e}"
            print(f"[Error] {error_message}")
            # Check for 404 errors
            if isinstance(e, Exception) and "404" in str(e) and "is not found" in str(e):
                error_message += f"\n(Model '{MODEL_NAME}' might not be available in your region or for your API key. Check Google AI documentation for available models.)"

            return f"Verdict: Error\nExplanation: {error_message}"
```

ДОДАТОК Д

Код файлу port_scanning.py

```
# modules/port_scanning.py
import re
import time
from collections import defaultdict, deque
from datetime import datetime, timedelta
import os

# --- Module State ---
ip_connections = defaultdict(lambda: deque())
alert_timestamps = {}
potential_attack_logs = defaultdict(lambda: deque())
ip_ports = defaultdict(set)

# Regular Expressions for SRC and DST IPs
LOG_PATTERN = re.compile(r'SRC=([\d.]+)')
DPT_PATTERN = re.compile(r'DPT=(\d+)')
# Filter
REQUIRED_STRING = "All connections:"

def parse_log_line(line):
    """Extracts the source IP address from the log line."""
    match = LOG_PATTERN.search(line)
    if match:
        return match.group(1)
    return None
```

```
def parse_destination_port(line):
    """Extracts the destination port (DPT) from the log line."""
    match = DPT_PATTERN.search(line)
    if match:
        try:
            return int(match.group(1))
        except ValueError:
            return None
    return None

def clean_old_connections(ip, window_seconds):
    """Removes old timestamps and corresponding logs for the specified IP."""
    now = datetime.now()
    cleaned_count = 0
                while ip_connections[ip] and ip_connections[ip][0] < now -
timedelta(seconds=window_seconds):
        ip_connections[ip].popleft()
        cleaned_count += 1
    for _ in range(cleaned_count):
        if potential_attack_logs[ip]:
            potential_attack_logs[ip].popleft()

def process_log_line(line, threshold=10, window_seconds=600,
alert_interval_seconds=600, min_distinct_ports=10):
    """
     Processes ONE log line, updates state, and checks for port scanning from ANY source
IP.
    Filters by: presence of 'All connections:'.
```

Condition: requires >= threshold connections AND >= min_distinct_ports unique destination ports.

Args:

line (str): The log line to process.

threshold (int): Connection count threshold.

window_seconds (int): Time window in seconds.

alert_interval_seconds (int): Alert throttling interval.

min_distinct_ports (int): Min number of unique destination ports for an alert.

Returns:

tuple or None: (ip, logs_to_send) if detected, otherwise None.

"""

```
# Filter 1: Required string
if REQUIRED_STRING not in line:
    return None

# Parse source IP
ip = parse_log_line(line)
if not ip:
    return None # Could not find source IP
# Parse destination port
destination_port = parse_destination_port(line)
if destination_port is None:
    return None # Ignore lines without DPT for port scan detection



now = datetime.now()
# Clean old records for this specific IP
clean_old_connections(ip, window_seconds)
# Add current data for this specific IP
```

```
    ip_connections[ip].append(now)
    potential_attack_logs[ip].append(line.strip())
    ip_ports[ip].add(destination_port)

    # Check conditions for alert for this specific IP
    current_connection_count = len(ip_connections[ip])
    current_distinct_ports = len(ip_ports[ip])

            if current_connection_count >= threshold and current_distinct_ports >=
min_distinct_ports:
        last_alert_time = alert_timestamps.get(ip)

        # Check throttling for this specific IP
                                if not last_alert_time or now - last_alert_time >
timedelta(seconds=alert_interval_seconds):
            # Alert triggered for IP
            print(f"[Port Scanner] Potential scan detected from Source IP {ip}.")
                    print(f"   Conditions met: Connections={current_connection_count} (>=
{threshold}), Distinct Ports={current_distinct_ports} (>= {min_distinct_ports})")

            logs_to_send = list(potential_attack_logs[ip]) # Get logs to send
            alert_timestamps[ip] = now # Update last alert time for this IP

            # Clear state for this specific IP after alert
            ip_connections[ip].clear()
            potential_attack_logs[ip].clear()
            ip_ports[ip].clear()

            return (ip, logs_to_send) # Return detected IP and logs
        else:
```

```
            # Throttled
            return None
    return None


# Deprecated function check_port_scan
def check_port_scan(log_file_path, threshold=10, window_seconds=600, alert_interval_seconds=600):
     print("[Warning] Using check_port_scan is inefficient and does not check distinct port count.")
    return {}
```

ДОДАТОК Е

Код файлу scanning_response_playbook.py

```
# modules/scanning_response_playbook.py
import subprocess
import os
from datetime import datetime

# --- Settings ---
BLOCK_LOG_FILE = "scans-ip-block.txt"
FALSE_POSITIVE_LOG_FILE = "port_scanning_falsepositives.txt"
IPTABLES_COMMAND = ['sudo', 'iptables', '-I', 'INPUT', '1', '-s', '{ip}', '-j', 'DROP']



def block_ip(ip_address):
    """Executes the iptables command to block the specified IP address."""
     if not ip_address: print("[Playbook Error] No IP address provided for blocking."); return False
    command = [part.replace('{ip}', ip_address) for part in IPTABLES_COMMAND]
    try:
            print(f"[Playbook Action] Attempting to block IP {ip_address} using command: {' '.join(command)}")
             result = subprocess.run(command, check=True, capture_output=True, text=True)
        print(f"[Playbook Success] Successfully blocked IP: {ip_address}")
        if result.stdout: print(f"[Playbook Info] iptables stdout: {result.stdout.strip()}")
```

```
            return True
        except FileNotFoundError: print(f"[Playbook Error] 'sudo' or 'iptables' command not found."); return False
            except subprocess.CalledProcessError as e: print(f"[Playbook Error] Failed to block IP {ip_address}.\n  "
                                                              f"Command: {' '.join(e.cmd)}\n  Return Code: {e.returncode}\n  "
                                                                  f"Stderr: {e.stderr.strip()}\n  (Ensure sudo privileges)"); return False
            except Exception as e: print(f"[Playbook Error] An unexpected error occurred during IP blocking: {e}"); return False

    def log_blocked_ip(alert_title, ai_verdict, ai_explanation, ip_address, logs):
        """
        Writes blocking information, including trigger logs, to the scans-ip-block.txt log file.

        Args:
            alert_title (str): Alert title (with timestamp).
            ai_verdict (str): Verdict string from AI ("Verdict: ...").
            ai_explanation (list): List of strings with AI explanation.
            ip_address (str): The blocked IP address.
            logs (list): The list of log lines that triggered the analysis. # New argument
        """
        log_entry = f"{alert_title}\n"
        log_entry += "AI Analysis:\n"
        log_entry += f"{ai_verdict}\n"
        log_entry += "\n".join(ai_explanation) + "\n" # Using list of strings

        # --- Add Trigger Logs ---
```

```
        log_entry += "Trigger Logs:\n"
        if logs: # Check if logs list is not empty
            for i, log_line in enumerate(logs):
                log_entry += f"  [{i+1}] {log_line}\n"
        else:
            log_entry += "  (No trigger logs provided)\n"
        log_entry += ("-" * 20) + "\n"
        # ------------------------

        log_entry += f"Action: {ip_address} Blocked\n"
        log_entry += ("-" * 40) + "\n\n"

        try:
            with open(BLOCK_LOG_FILE, 'a') as f:
                f.write(log_entry)
                 print(f"[Playbook Log] Block action and logs for {ip_address} logged to {BLOCK_LOG_FILE}")
        except IOError as e:
                        print(f"[Playbook Error] Failed to write to block log file {BLOCK_LOG_FILE}: {e}")
        except Exception as e:
                print(f"[Playbook Error] An unexpected error occurred during block logging: {e}")

    def log_false_positive(detection_time, src_ip, target_ip, logs, analysis_result):
            """Writes false positive information to the port_scanning_falsepositives.txt file."""
            log_entry = ("=" * 40) + "\n"; log_entry += f"False Positive detected by AI ({detection_time})\n"; log_entry += f"Source IP: {src_ip}\n"; log_entry += f"Target IP: {target_ip}\n"; log_entry += "Logs:\n"; [log_entry := log_entry + f"  [{i+1}] {log_line}\n"
```

```
for i, log_line in enumerate(logs)]; log_entry += ("-"*20) + "\n"; log_entry += "AI Analysis:\n"; log_entry += f"{analysis_result}\n"; log_entry += ("=" * 40) + "\n\n"
        try:
            with open(FALSE_POSITIVE_LOG_FILE, 'a') as f: f.write(log_entry)
                print(f"[Playbook Log] False positive event for {src_ip} logged to {FALSE_POSITIVE_LOG_FILE}")
        except IOError as e: print(f"[Playbook Error] Failed to write to false positive log file {FALSE_POSITIVE_LOG_FILE}: {e}")
        except Exception as e: print(f"[Playbook Error] An unexpected error occurred during false positive logging: {e}")
```