МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ФАХОВИЙ БІЗНЕС-КОЛЕДЖ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

# КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему

« СИСТЕМА ВИЯВЛЕННЯ ТА РЕАГУВАННЯ НА ЗАГРОЗИ У ХМАРНИХ СЕРВІСАХ »

Виконав: студент групи 1КІ-23
спеціальності
123 «Комп'ютерна інженерія
Семизенко В.В.
Керівник роботи к.т.н., доцент
Захарова М.В.
Кількість балів:_____
Оцінка: ECTS_____

Черкаси, 2025

# АНОТАЦІЯ

У роботі досліджено підходи до виявлення та реагування на кіберзагрози у хмарних середовищах. Проведено аналіз сучасних інструментів безпеки, зокрема SIEM- і SOAR-систем, а також методів поведінкового аналізу та машинного навчання. Запропоновано архітектуру прототипу системи виявлення загроз з підтримкою інтеграції через API, візуального моніторингу та автоматизованого реагування. Реалізовано програмний модуль на мові Python з використанням моделі Isolation Forest для класифікації аномальної активності користувачів у логах. Проведено тестування на згенерованих даних, що засвідчило високу ефективність прототипу: виявлено понад 80% потенційних загроз при низькому рівні хибнопозитивних спрацювань. Результати роботи можуть бути застосовані для створення адаптивних систем кіберзахисту в хмарних платформах.

Ключові слова: хмарні сервіси, загрози інформаційної безпеки, вразливості, інцидент безпеки, несанкціонований доступ, DDoS-атаки, витік даних, штучний інтелект, машинне навчання.

## ABSTRACT

The qualification paper investigates approaches to threat detection and incident response in cloud environments. A comprehensive analysis of modern cybersecurity tools was conducted, including SIEM and SOAR systems, behavioral analysis, and machine learning techniques. A prototype architecture of a threat detection system was proposed, featuring API integration, a visual monitoring interface, and automated response capabilities. The system was implemented in Python using the Isolation Forest algorithm to identify anomalies in user log data. Testing on synthetic datasets demonstrated the system's high effectiveness, detecting over 80% of simulated threats with a low false positive rate. The developed solution can serve as a foundation for building adaptive cybersecurity systems in cloud-based infrastructures.



Keywords: cloud services, information security threats, vulnerabilities, security incident, unauthorized access, DDoS attacks, data leakage, artificial intelligence, machine learning.

# ЗМІСТ

# СПИСОК УМОВНИХ СКОРОЧЕНЬ

| | |
|---|---|
| **API** | Application Programming Interface – програмний інтерфейс прикладного рівня |
| **CSV** | Comma-Separated Values – значення, розділені комами |
| **FP** | False Positive – хибнопозитивне спрацювання |
| **GCP** | Google Cloud Platform – хмарна платформа Google |
| **HTTP** | HyperText Transfer Protocol – протокол передачі гіпертексту |
| **IP** | Internet Protocol – інтернет-протокол |
| **JSON** | JavaScript Object Notation – текстовий формат обміну даними |
| **ML** | Machine Learning – машинне навчання |
| **SIEM** | Security Information and Event Management – управління інформацією та подіями безпеки |
| **SMTP** | Simple Mail Transfer Protocol – простий протокол передачі пошти |
| **SOAR** | Security Orchestration, Automation and Response – оркестрація, автоматизація та реагування на інциденти безпеки |
| **TP** | True Positive – істинно позитивне спрацювання |
| **UI** | User Interface – інтерфейс користувача |
| **URL** | Uniform Resource Locator – уніфікований локатор ресурсу |
| **VPN** | Virtual Private Network – віртуальна приватна мережа |
| **XML** | eXtensible Markup Language – розширювана мова розмітки |

# ВСТУП

У сучасних умовах цифровізації хмарні сервіси стали критично важливою складовою інформаційної інфраструктури підприємств, державних установ і приватних користувачів. Їх використання забезпечує гнучкість, масштабованість та економічну ефективність, однак водночас створює нові вектори атак і підвищує ризики втрати або компрометації даних.

У зв'язку з цим зростає потреба в реалізації ефективних рішень з інформаційної безпеки, які забезпечують не лише контроль доступу, а й активне виявлення аномальної поведінки, автоматизоване реагування на інциденти та адаптацію до умов динамічного хмарного середовища.

**Актуальність теми** полягає у необхідності побудови систем, здатних забезпечити високий рівень кіберстійкості в умовах постійно змінюваних загроз. Традиційні підходи, засновані виключно на сигнатурному аналізі, не дають змоги ефективно протидіяти атакам нульового дня, інсайдерським загрозам та складним багатовекторним вторгненням. Тому особливої ваги набувають методи машинного навчання, поведінковий аналіз, SIEM- і SOAR-технології.

**Метою дослідження** є аналіз існуючих методів виявлення й реагування на загрози у хмарних сервісах, а також розробка прототипу системи моніторингу безпеки, що поєднує інтелектуальний аналіз даних і механізми автоматизованого реагування.

Для досягнення мети було визначено наступні **завдання**:

- провести огляд сучасних методів виявлення загроз і оцінити їх ефективність у хмарному середовищі;
- здійснити порівняльний аналіз інструментів автоматизованого реагування (SOAR, SIEM тощо);
- спроєктувати архітектуру системи, що підтримує інтеграцію з логами, API та сервісами хмари;
- реалізувати механізми збору, фільтрації та аналітики подій;
- протестувати прототип на згенерованих даних та оцінити точність виявлення аномалій.

**Об’єктом дослідження** є процеси виявлення, моніторингу та реагування на кіберзагрози в хмарних середовищах.

**Предметом дослідження** — методи та інструменти виявлення загроз, алгоритми поведінкової аналітики та компоненти автоматизованої реакції.

**Методи дослідження** включають: аналіз великих даних, методи машинного навчання (Isolation Forest, класифікатори), програмну реалізацію на Python, візуалізацію за допомогою Streamlit та тестування у моделюваному середовищі.

**Практичне значення** роботи полягає у створенні прототипу системи, яка демонструє ефективність поєднання сучасних аналітичних та автоматизованих засобів захисту, що можуть бути інтегровані у хмарну інфраструктуру та адаптовані під конкретні потреби організації.

**Апробація результатів бакалаврської роботи.** Матеріали до бакалаврської роботи апробувалися на XVII Студентській науково-практичній конференції «Тенденції розвитку ІТ-технологій в Україні» (Черкаси: ЧДБК, 2025).

**Публікації.**

1. Семизенко В.В. Аналіз вразливостей хмарних сервісів// XVII Студентська науково-практична конференція «Тенденції розвитку ІТ-технологій в Україні» 26-27 березня 2025 р. м.Черкаси. с. 41-45.

**Структура та обсяг роботи.** Дипломна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел (29 найменування) та додатків. Загальний обсяг роботи становить 60 сторінок основного тексту, 18 рисунка та 9 таблиць.

# РОЗДІЛ 1 АНАЛІЗ ЗАГРОЗ ТА ВРАЗЛИВОСТЕЙ ХМАРНИХ СЕРВІСІВ

З розвитком хмарних технологій питання кібербезпеки набуває все більшої актуальності. Хмарні сервіси, завдяки своїй масштабованості, гнучкості та економічній ефективності, широко використовуються у різних сферах діяльності. Проте їх популярність робить їх привабливою мішенню для кіберзлочинців, які прагнуть отримати несанкціонований доступ до конфіденційних даних, порушити роботу сервісів або використати хмарні ресурси у своїх цілях.

Однією з основних загроз для хмарних середовищ є несанкціонований доступ. Недостатній рівень аутентифікації та авторизації користувачів може спричинити витік або компрометацію даних. Також значну небезпеку становлять витоки інформації, оскільки хмарні платформи обробляють великі обсяги конфіденційної інформації, що робить їх привабливою мішенню для атак. Крім того, хмарні сервіси можуть зазнавати DDoS-атак, що спричиняє перебої у їх роботі. Ще одним ризиком є зловмисне використання ресурсів, коли кіберзлочинці застосовують хмарні обчислювальні потужності для проведення атак або незаконного майнінгу криптовалют [3]. Додатково, використання застарілого або ненадійного програмного забезпечення може призвести до експлуатації вразливостей системи (див. табл. 1.1).

Таблиця 1.1 - Класифікація загроз і вразливостей хмарних сервісів

| № | Категорія | Тип загрози / вразливості | Можливі наслідки | Методи протидії |
|---|---|---|---|---|
| **1** | **2** | **3** | **4** | **5** |
| 1 | Несанкціонований доступ | Недостатня аутентифікація | Компрометація облікових записів, витік даних | Використання MFA, сильні паролі, контроль доступу |
| 2 | Витік даних | Помилки в конфігураціях, зловмисні дії інсайдерів | Втрата конфіденційної інформації, репутаційні ризики | Шифрування даних, аудит безпеки |
| 3 | Атаки на сервіси | DDoS-атаки | Недоступність сервісів, фінансові збитки | Використання WAF, балансувальники навантаження |
| 4 | Експлуатація вразливостей | Вразливості API, застаріле ПЗ | Несанкціонований доступ, можливість атак | Регулярні оновлення, тестування на проникнення |

Продовження таблиці 1.1

| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| 5 | Зловмисне використання ресурсів | Використання ресурсів для майнінгу, ботнетів | Збільшення витрат, зниження продуктивності сервісів | Моніторинг активності, обмеження ресурсів |

Вразливості хмарних сервісів можуть бути зумовлені як технічними, так і організаційними чинниками. Недостатній контроль доступу та неналежна автентифікація користувачів створюють потенційні ризики для безпеки. Помилки в налаштуванні серверів та сховищ даних можуть зробити їх доступними для сторонніх осіб. Також слід враховувати інсайдерські загрози, коли співробітники організацій можуть ненавмисно або свідомо сприяти витоку чи компрометації даних. Окрему небезпеку становлять вразливості API, які можуть бути використані зловмисниками для атак на хмарну інфраструктуру [3].

Для ефективного захисту хмарних сервісів необхідно застосовувати різноманітні методи аналізу загроз. Моніторинг безпеки дозволяє виявляти аномальну активність, яка може свідчити про можливі атаки. Тестування на проникнення допомагає оцінити рівень захищеності системи, імітуючи реальні атаки на хмарну платформу. Аналіз журналів подій забезпечує збір та обробку даних про активність користувачів, що допомагає виявляти потенційні загрози. Окрім цього, використання систем штучного інтелекту дозволяє прогнозувати та запобігати атакам завдяки аналізу великих обсягів даних та виявленню закономірностей у діях зловмисників.

Аналіз загроз та вразливостей хмарних сервісів є ключовим етапом у забезпеченні їх безпеки. Виявлення потенційних ризиків дозволяє мінімізувати загрози та захистити критично важливі дані, забезпечуючи стабільну та безпечну роботу хмарної інфраструктури.

## 1.1 Особливості та архітектура хмарних сервісів

Хмарні сервіси є однією з ключових технологій сучасного інформаційного простору, що дозволяє користувачам отримувати доступ до обчислювальних ресурсів через Інтернет без необхідності їх локального розгортання та

адміністрування. Основними особливостями хмарних сервісів є масштабованість, гнучкість, доступність, економічна ефективність та централізоване управління ресурсами (див. рис. 1.1).

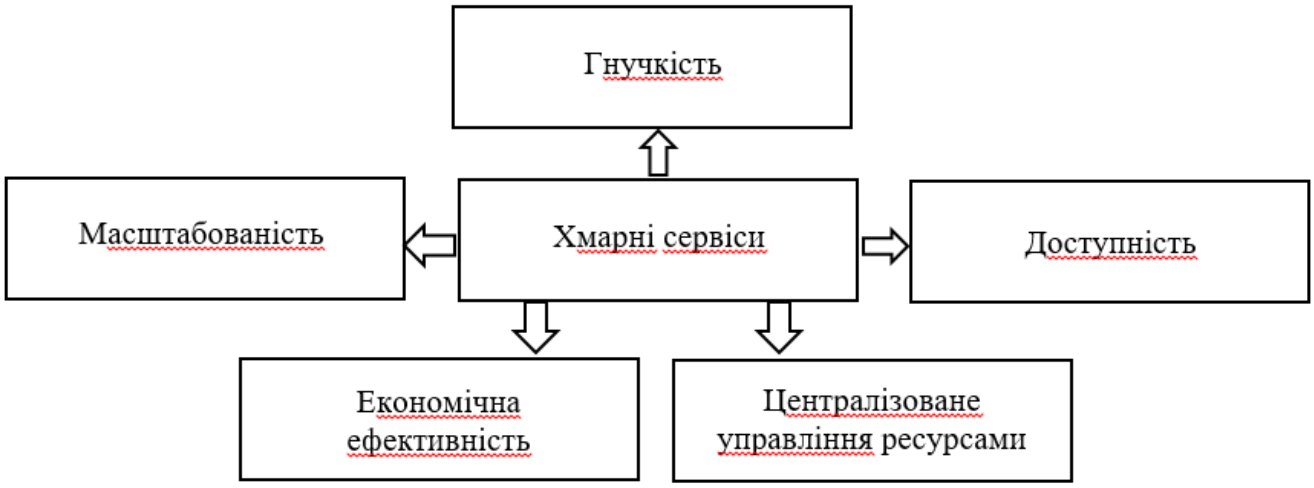


Рисунок 1.1 - Основні особливості хмарних сервісів

Хмарні сервіси можна класифікувати за моделями надання послуг і способами розгортання. Такий поділ дозволяє чітко зрозуміти особливості кожного типу сервісів, їх функціональні можливості та сфери застосування [4].

Перш за все, хмарні сервіси поділяються на кілька моделей залежно від рівня послуг, які вони надають.

*Модель IaaS (Infrastructure as a Service)* надає користувачам доступ до базових інфраструктурних ресурсів, таких як віртуальні машини, сховища даних і мережеві ресурси. Ця модель забезпечує гнучкість у налаштуванні обчислювальних потужностей, управлінні сховищами та конфігурації мережі. Вона ідеально підходить для хостингу веб-додатків, тестових середовищ або обробки великих обсягів даних.

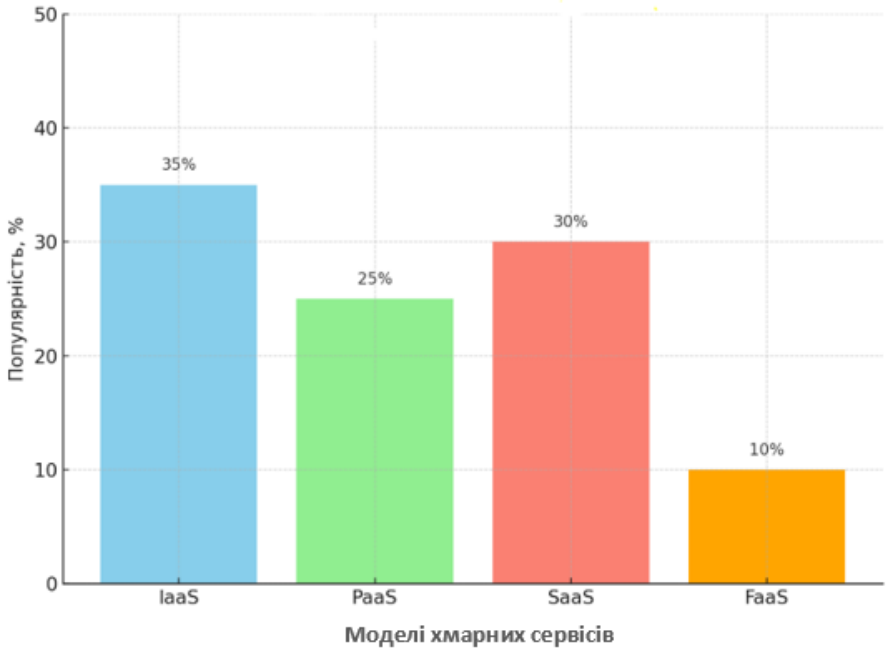


Рисунок 1.2 - Візуалізація класифікації ризиків інформаційної безпеки

*PaaS (Platform as a Service)* надає платформу для розробки, тестування та розгортання додатків. Користувачам не потрібно керувати інфраструктурою, що дозволяє зосередитися на створенні програмного забезпечення. Ця модель пропонує інструменти для інтеграції, підтримку кількох мов програмування та автоматичне масштабування. PaaS підходить для розробки веб- і мобільних додатків або побудови платформ для аналізу великих даних.

*SaaS (Software as a Service)* забезпечує доступ до готових програм через Інтернет. Користувачі можуть працювати з програмами без необхідності їх встановлення чи обслуговування. Основними перевагами цієї моделі є автоматичні оновлення програмного забезпечення та доступність з будь-якого пристрою. Вона використовується для електронної пошти, інструментів спільної роботи чи CRM-систем.

*FaaS (Function as a Service)* фокусується на виконанні окремих функцій без необхідності управління серверною інфраструктурою. Це ідеальне рішення для обробки даних у реальному часі, побудови serverless-додатків і автоматизації робочих процесів [4].

Порівняння моделей хмарних сервісів за ключовими характеристиками та аналіз популярності (див. табл. 1.2 та рис. 1.2).

Таблиця 1.2 - Порівняння моделей хмарних сервісів за ключовими характеристиками

| № | Модель | Опис | Функціональні можливості | Приклади застосування |
|---|---|---|---|---|
| 1 | **IaaS** | Інфраструктура як послуга | Масштабування ресурсів, налаштування мережі | Хостинг, тестові середовища, обробка даних |
| 2 | **PaaS** | Платформа для розробки додатків | Розробка, тестування, автоматичне масштабування | Веб- і мобільні додатки, аналіз великих даних |
| 3 | **SaaS** | Готове програмне забезпечення | Використання ПЗ на вимогу, автоматичні оновлення | Електронна пошта, CRM-системи, співпраця |
| 4 | **FaaS** | Функції як послуга | Виконання подієвих функцій, серверless-архітектура | Обробка даних у реальному часі, автоматизація |

Крім моделей надання послуг, хмарні сервіси розрізняються за типами розгортання.

Публічні хмари є загальнодоступними та обслуговуються сторонніми провайдерами. Вони пропонують масштабованість ресурсів і економічну вигоду завдяки спільному використанню інфраструктури. Такий тип часто використовується для резервного копіювання даних або розгортання веб-додатків.

Приватні хмари створюються для використання однією організацією. Вони забезпечують високий рівень контролю та безпеки, а також інтеграцію з внутрішніми системами компанії. Цей тип розгортання підходить для організацій, які працюють з критично важливими даними або мають жорсткі вимоги до безпеки.

Гібридні хмари поєднують у собі елементи публічних і приватних хмар. Це дозволяє організаціям гнучко розподіляти ресурси, збалансовуючи між доступністю і безпекою. Вони є оптимальними для компаній, що потребують одночасного використання приватних і загальнодоступних середовищ.

Мультихмари передбачають використання кількох хмарних провайдерів одночасно. Це дає змогу користувачам вибирати найкращі сервіси від різних постачальників, а також забезпечує підвищену стійкість до збоїв. Мультихмари широко використовуються глобальними корпораціями, які мають потребу в розподілі навантаження між різними регіонами.

Типологія хмарних сервісів дає змогу організаціям вибирати оптимальні рішення залежно від їхніх потреб. Різноманітність моделей і типів розгортання забезпечує гнучкість, функціональність і безпеку для користувачів у найрізноманітніших сферах діяльності [5].

## 1.2 Основні загрози інформаційної безпеки в хмарних середовищах

Сучасні хмарні технології значно розширюють можливості зберігання, обробки та управління даними, що робить їх незамінними для бізнесу, державних установ та окремих користувачів. Водночас, із зростанням популярності хмарних сервісів суттєво збільшується кількість атак, спрямованих на компрометацію даних та порушення безпеки. Основні загрози інформаційної безпеки в хмарних середовищах можна розділити на кілька категорій.

Однією з найбільш серйозних загроз є несанкціонований доступ до даних, що може виникати через відсутність надійної автентифікації або недостатній контроль доступу. Використання слабких паролів, витік облікових даних або недостатній рівень захисту може дозволити зловмисникам отримати контроль над конфіденційною інформацією [6].

Не менш критичною проблемою є витік даних, який може статися внаслідок атак на сервери хмарного провайдера або через помилки у налаштуваннях безпеки. Оскільки хмарні сервіси обробляють великі обсяги інформації, вони стають привабливою ціллю для кібератак, що спрямовані на отримання персональних, фінансових чи корпоративних даних.

DDoS-атаки (розподілені атаки на відмову в обслуговуванні) також залишаються серйозною загрозою для хмарних середовищ. Зловмисники генерують велику кількість запитів до серверів, перевантажуючи їх ресурси, що може спричинити збої у роботі критично важливих сервісів та значні фінансові втрати для компаній.

Ще однією загрозою є зловмисне використання обчислювальних ресурсів хмари. У деяких випадках хакери можуть використовувати сервери хмарних платформ для власних цілей, наприклад, для майнінгу криптовалют, запуску ботнет-мереж або проведення подальших атак на інші системи [6].

Слабким місцем можуть бути і вразливості API. Хмарні сервіси широко використовують програмні інтерфейси (API) для взаємодії між системами, однак недостатній рівень захисту цих інтерфейсів може стати точкою входу для кібератак. Наприклад, недостатньо перевірені запити до API можуть дозволити отримати несанкціонований доступ до критично важливої інформації.

Окрім зовнішніх атак, хмарні середовища можуть зазнавати впливу інсайдерських загроз. Працівники компаній або адміністратори хмарних сервісів можуть ненавмисно чи навмисно сприяти витоку даних, порушенню безпеки або саботажу систем [7].

Також значну небезпеку становлять фішингові атаки, під час яких зловмисники використовують методи соціальної інженерії, аби змусити користувачів передати свої облікові дані, відкриваючи тим самим доступ до хмарної інфраструктури. Графік, що показує частоту виникнення основних загроз у хмарних середовищах показано на (рис. 1.3). Він допомагає візуально оцінити, які загрози є найбільш поширеними та на що варто звернути особливу увагу при забезпеченні безпеки.

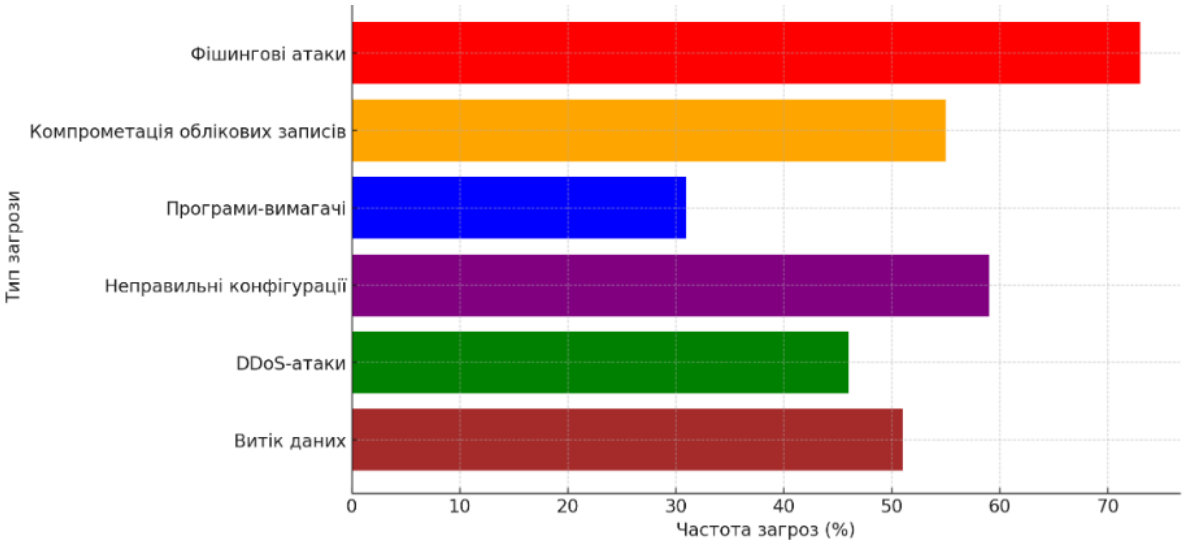


Рисунок 1.3 - Частота основних загроз інформаційної безпеки в хмарних середовищах

Пояснення до графіка:

- Фішингові атаки (73%): Найпоширеніша загроза, яка спрямована на отримання конфіденційних даних користувачів через обманні електронні листи або повідомлення.
- Компрометація облікових записів (55%): Включає несанкціонований доступ до хмарних ресурсів через викрадені або слабкі облікові дані.
- Програми-вимагачі (31%): Шкідливе програмне забезпечення, яке шифрує дані та вимагає викуп за їх розшифрування.
- Неправильні конфігурації хмарних сервісів (59%): Помилки в налаштуваннях, які можуть призвести до витоку даних або інших вразливостей.
- DDoS-атаки (46%): Атаки на відмову в обслуговуванні, які спрямовані на перевантаження хмарних сервісів, роблячи їх недоступними для користувачів.
- Витік даних (51%): Несанкціоноване розголошення або доступ до конфіденційної інформації, що зберігається в хмарі.

Важливим аспектом інформаційної безпеки є захист від шкідливого програмного забезпечення (Malware). Віруси, трояни та інші типи

зловмисного коду можуть проникати у хмарне середовище, що призводить до втрати або модифікації даних, а в деяких випадках – до повного виходу системи з ладу.

Неправильні конфігурації хмарних сервісів є ще однією поширеною загрозою. Ненавмисно відкриті для загального доступу бази даних, неправильно налаштовані права доступу або відсутність шифрування можуть зробити конфіденційну інформацію доступною для сторонніх осіб.

Крім технічних загроз, існують і юридичні та нормативні ризики. Зберігання конфіденційних даних у хмарі може не відповідати законодавчим вимогам певних країн або галузевим стандартам, що може призвести до штрафів або судових розглядів [1].

Для мінімізації ризиків та забезпечення надійного захисту хмарних середовищ необхідно впроваджувати ефективні методи безпеки.

Одним із ключових заходів є постійний моніторинг активності у хмарних системах. Використання засобів аналізу поведінки користувачів та автоматизованих алгоритмів для виявлення аномальних дій дозволяє вчасно реагувати на потенційні загрози.

Тестування на проникнення – це метод, що дозволяє оцінити рівень захищеності хмарної інфраструктури шляхом імітації реальних атак. Проведення регулярних перевірок дає змогу виявити вразливості до того, як ними скористаються зловмисники.

Шифрування даних є важливим механізмом захисту, оскільки воно забезпечує конфіденційність інформації навіть у разі її витоку. Як шифрування даних під час передавання, так і зберігання у зашифрованому вигляді мінімізує ризики несанкціонованого доступу [8].

Останнім часом все більше компаній впроваджують автоматизовані системи виявлення загроз, що базуються на алгоритмах штучного інтелекту та машинного навчання. Такі системи аналізують величезні обсяги даних, шукаючи закономірності, що можуть свідчити про потенційні атаки.

З метою захисту облікових записів користувачів важливо використовувати мультифакторну автентифікацію (MFA). Поєднання паролів із додатковими рівнями безпеки (наприклад, одноразові коди, біометрія) значно ускладнює компрометацію акаунтів.

Також важливим заходом є регулярне резервне копіювання даних, що дозволяє швидко відновити інформацію у разі її втрати через атаку або технічний збій [8].

Безпека хмарних сервісів – це багаторівневий процес, що включає як технічні, так і організаційні заходи захисту. Виявлення та усунення потенційних загроз, використання сучасних методів кібербезпеки та регулярний моніторинг дозволяють забезпечити стабільну та безпечну роботу хмарної інфраструктури. Тільки комплексний підхід до захисту даних дозволить компаніям та користувачам зберегти конфіденційність, цілісність та доступність інформації у хмарних середовищах.

## 1.3 Методи аналізу вразливостей та оцінки ризиків

Безпека хмарних сервісів є одним із найважливіших аспектів сучасних інформаційних технологій. Зростання популярності хмарних обчислень призводить до збільшення кількості атак та загроз, що можуть завдати шкоди компаніям та користувачам. Одним із ключових етапів забезпечення безпеки є аналіз вразливостей та оцінка ризиків. Ці процеси дозволяють виявляти слабкі місця у системах, оцінювати рівень потенційних загроз і впроваджувати ефективні заходи захисту.

У сучасних хмарних середовищах аналіз вразливостей та оцінка ризиків є ключовими етапами забезпечення безпеки. Це дозволяє своєчасно виявляти слабкі місця в системах та мінімізувати потенційні загрози [9].

Таблиця 1.3 - Методи аналізу вразливостей та оцінки ризиків

| № | Метод | Опис | Інструменти |
|---|---|---|---|
| 1 | Аналіз коду та архітектури | Перевірка вихідного коду, налаштувань та логіки роботи додатків на наявність вразливостей | SAST (Static Application Security Testing), DAST (Dynamic Application Security Testing) |
| 2 | Автоматизоване сканування вразливостей | Використання спеціальних програм для виявлення відомих вразливостей у хмарній інфраструктурі | Nessus, OpenVAS, Qualys, Acunetix |
| 3 | Тестування на проникнення | Імітація атак на хмарні сервіси для перевірки їхнього захисту | Metasploit, Burp Suite, Kali Linux |
| 4 | Аналіз журналів подій та моніторинг | Виявлення підозрілих дій та аномальної поведінки користувачів | SIEM-рішення (Splunk, IBM QRadar, ELK Stack), UEBA (User and Entity Behavior Analytics) |
| 5 | Аудит безпеки конфігурацій | Перевірка налаштувань безпеки хмарних сервісів на відповідність стандартам | ISO 27001, NIST, SOC 2 |

Основні методи аналізу вразливостей та оцінки ризиків у хмарних середовищах, їх особливості, принципи роботи та практичне застосування показано в таблицях (див табл. 1.3 та табл. 1.4).

Таблиця 1.4 - Методи оцінки ризиків

| № | Метод | Опис | Застосування |
|---|---|---|---|
| **1** | **2** | **3** | **4** |
| 1 | Кількісний та якісний аналіз ризиків | Оцінка ризиків у числових показниках або за рівнем загрози (низький, середній, високий) | Використовується для прийняття управлінських рішень у кібербезпеці |
| 2 | OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation) | Оцінка критичних активів та загроз для компанії | Стратегічне управління ризиками |
| 3 | NIST 800-30 | Структурований підхід до аналізу ризиків, що включає | Використовується у державних та комерційних хмарних системах |

Продовження таблиці 1.4

| 1 | 2 | 3 | 4 |
|---|---|---|---|
| | | виявлення загроз та оцінку їхнього впливу | |
| 4 | FAIR (Factor Analysis of Information Risk) | Кількісний підхід до оцінки кіберризиків, що дозволяє оцінювати потенційні фінансові втрати | Використовується для планування інвестицій у безпеку |
| 5 | SWOT-аналіз безпеки | Оцінка сильних та слабких сторін хмарного середовища, аналіз можливостей та загроз | Використовується для стратегічного планування безпеки |

Забезпечення безпеки хмарних сервісів вимагає комплексного підходу, який включає аналіз вразливостей та оцінку ризиків [2]. Використання сучасних методів дозволяє своєчасно виявляти загрози, мінімізувати потенційні ризики та забезпечувати стабільну роботу хмарних платформ. Інтеграція автоматизованих рішень із традиційними підходами до кібербезпеки сприяє підвищенню ефективності захисту даних та інфраструктури в умовах постійно зростаючих загроз.

# РОЗДІЛ 2 СИСТЕМИ ВИЯВЛЕННЯ ТА АНАЛІЗУ ЗАГРОЗ В ХМАРНИХ СЕРВІСАХ

У зв’язку з активним розвитком хмарних технологій та стрімким зростанням обсягів даних, що обробляються у віддалених дата-центрах, значно зростає і кількість кіберзагроз, які націлені на хмарну інфраструктуру. Традиційні підходи до захисту інформаційних систем не завжди є ефективними у динамічному, масштабованому та розподіленому середовищі, яким є хмара. Це створює нагальну потребу у впровадженні спеціалізованих систем виявлення та аналізу загроз.

Такі системи мають забезпечити постійний моніторинг, аналіз поведінки користувачів і мережевого трафіку, своєчасне виявлення шкідливої активності та реагування на інциденти безпеки. З огляду на специфіку хмарних сервісів, ці системи повинні бути високопродуктивними, масштабованими та здатними адаптуватися до швидкозмінних умов.

У межах цього розділу здійснюється комплексний аналіз підходів до побудови систем виявлення загроз у хмарних середовищах. Зокрема, розглядаються сигнатурний, поведінковий та гібридний методи, здійснюється оцінка їх ефективності, а також визначаються переваги та недоліки сучасних систем виявлення та запобігання вторгненням (IDS/IPS) у контексті хмарної інфраструктури. Особливу увагу приділено потенціалу використання методів штучного інтелекту та машинного навчання для підвищення ефективності процесів виявлення загроз та забезпечення інформаційної безпеки в хмарному середовищі.

## 2.1 Підходи до виявлення загроз

Сучасні системи виявлення загроз у хмарних середовищах базуються на різних концептуальних підходах до виявлення шкідливої активності. Основними серед них є сигнатурний, поведінковий та гібридний методи. Кожен із зазначених підходів має власні переваги та обмеження, що визначають їх ефективність у конкретних умовах експлуатації [10].

Сигнатурний метод (signature-based detection) полягає у порівнянні трафіку або поведінки користувача з попередньо визначеними зразками (сигнатурами) відомих загроз. Його головною перевагою є висока точність виявлення відомих атак при мінімальній кількості хибних спрацювань. Проте цей підхід є малоефективним щодо нових, раніше невідомих загроз, так званих «нульового дня» (zero-day threats), оскільки він не здатен виявляти невідомі шаблони.

Поведінковий метод (anomaly-based detection) ґрунтується на аналізі поточної активності користувачів та системи з метою виявлення відхилень від заздалегідь визначених норм. Він дозволяє виявляти нові загрози, що не мають відомих сигнатур. Такий підхід є особливо корисним у хмарному середовищі, де структури трафіку та дій можуть постійно змінюватися. Однак поведінкові системи часто генерують велику кількість хибнопозитивних спрацювань, що вимагає залучення додаткових ресурсів для обробки інцидентів.

Гібридний метод поєднує в собі переваги сигнатурного та поведінкового аналізу. Це дозволяє забезпечити більш високий рівень точності виявлення загроз і одночасно реагувати на як відомі, так і нові типи атак. Гібридні системи зазвичай використовуються у складних середовищах, де необхідна адаптивність та надійність в умовах змінної інфраструктури [11].

Вибір відповідного підходу залежить від специфіки хмарного середовища, рівня ризику, обсягів даних і доступних ресурсів. У багатьох випадках оптимальним рішенням є використання гібридного методу, який дозволяє досягти балансу між точністю, продуктивністю та ефективністю захисту. Порівняння методів виявлення загроз (див табл. 2.1).

Таблиця 2.1 – Порівняння методів виявлення загроз

| № | Метод | Переваги | Недоліки | Застосування |
|---|---|---|---|---|
| 1 | Сигнатурний | Висока точність для відомих атак | Неефективний проти нових або невідомих атак | Захист від поширених загроз |
| 2 | Поведінковий | Виявляє нові та атипові загрози | Часті хибні спрацювання, потреба в ресурсах | Моніторинг аномалій у поведінці |
| 3 | Гібридний | Поєднує точність сигнатурного та гнучкість поведінкового | Складність реалізації, потребує більше ресурсів | Комплексний захист хмарної інфраструктури |

Ефективне виявлення загроз у хмарних сервісах потребує використання комбінації різних методів. Сигнатурний підхід забезпечує високу точність при відомих атаках, поведінковий – дозволяє виявляти нові загрози, а гібридний метод забезпечує баланс між точністю і гнучкістю. Саме тому гібридні системи є найбільш доцільними для використання в умовах динамічного хмарного середовища.

## 2.2 Аналіз існуючих систем виявлення загроз у хмарному середовищі

У хмарному середовищі системи виявлення загроз, зокрема IDS (Intrusion Detection System) та IPS (Intrusion Prevention System), відіграють ключову роль у забезпеченні інформаційної безпеки. Вони дозволяють здійснювати моніторинг мережевого трафіку, виявляти підозрілу активність і запобігати вторгненням.

IDS – це система, яка здійснює пасивне спостереження за трафіком, виявляє аномалії або відомі загрози, але не здійснює активного блокування. Вона зазвичай використовується для інформування адміністраторів про потенційні інциденти безпеки. IPS, на відміну від IDS, має можливість автоматично блокувати або змінювати трафік у режимі реального часу для запобігання атаці [12].

У хмарному середовищі, на відміну від традиційної локальної інфраструктури, IDS/IPS системи повинні враховувати велику динамічність ресурсів, розподіленість даних, а також мультиорендну модель, яка передбачає одночасне використання ресурсів багатьма користувачами. Це ускладнює процес

виявлення загроз, оскільки типовий профіль поведінки системи може бути значно варіативним.

Серед популярних систем IDS/IPS, які можуть використовуватись у хмарному середовищі, варто відзначити такі:

- Snort – відкрите ПЗ, що підтримує сигнатурний аналіз. Використовується як IDS, так і IPS. Має широку спільноту та гнучкість у налаштуванні. Підтримується Cisco і є одним із найпоширеніших інструментів у світі.
- Suricata – високопродуктивна система з підтримкою багатопотокової обробки даних, можливістю протоколювання, роботи з TLS/SSL, HTTP/2 і JSON-форматами. Підтримує сигнатурний та поведінковий аналіз.
- OSSEC – хостова IDS (HIDS), що забезпечує аналіз журналів, перевірку цілісності файлів, виявлення rootkit, моніторинг реєстру та інтеграцію з SIEM-системами. Часто використовується для захисту окремих інстанцій в хмарному середовищі.
- Snort Cloud – хмарна реалізація Snort, оптимізована для роботи в масштабованому середовищі. Підтримує інтеграцію з хмарними сервісами через API та автоматичне оновлення сигнатур.
- AWS GuardDuty – власна служба Amazon, яка використовує машинне навчання, сигнатурний та поведінковий аналіз для виявлення загроз у середовищі AWS. Працює без встановлення агентів і автоматично інтегрується з іншими сервісами AWS.
- Microsoft Defender for Cloud – сервіс, який забезпечує захист ресурсів в Azure за допомогою комплексного моніторингу безпеки, рекомендацій, виявлення вразливостей та автоматизованого реагування [13].

Також варто відзначити системи типу Zeek (раніше Bro) – потужний інструмент мережевого моніторингу, що забезпечує глибокий аналіз трафіку, та Wazuh – розширену версію OSSEC з інтеграцією в Elastic Stack (див. табл. 2.2).

Таблиця 2.2 – Порівняння IDS/IPS систем

| № | Система | Тип | Підтримка хмари | Основні особливості |
|---|---|---|---|---|
| 1 | Snort | IDS/IPS | Так | Відкрите ПЗ, сигнатурний аналіз, підтримка Cisco |
| 2 | Suricata | IDS/IPS | Так | Багатопотоковість, TLS/SSL, сигнатурний + поведінковий |
| 3 | OSSEC | HIDS | Частково | Журнали, rootkit, інтеграція з SIEM |
| 4 | AWS GuardDuty | IDS | Так (AWS) | AI/ML, інтеграція з сервісами AWS |
| 5 | Microsoft Defender | IDS/IPS | Так (Azure) | Комплексний моніторинг і захист ресурсів |
| 6 | Zeek | IDS | Частково | Глибокий аналіз трафіку |
| 7 | Wazuh | HIDS | Частково | Розширення OSSEC, Elastic Stack |

Існуючі системи IDS/IPS адаптуються до специфіки хмарного середовища, забезпечуючи інтеграцію з платформами IaaS, PaaS та SaaS, підтримку API, масштабованість, автоматичне оновлення сигнатур, звітність та аналітику у реальному часі [14]. Особливу увагу приділяється можливості централізованого керування у багатокористувацьких середовищах та підтримці шифрованого трафіку. Класифікацію систем IDS/IPS у хмарних сервісах (див. рис. 2.1).

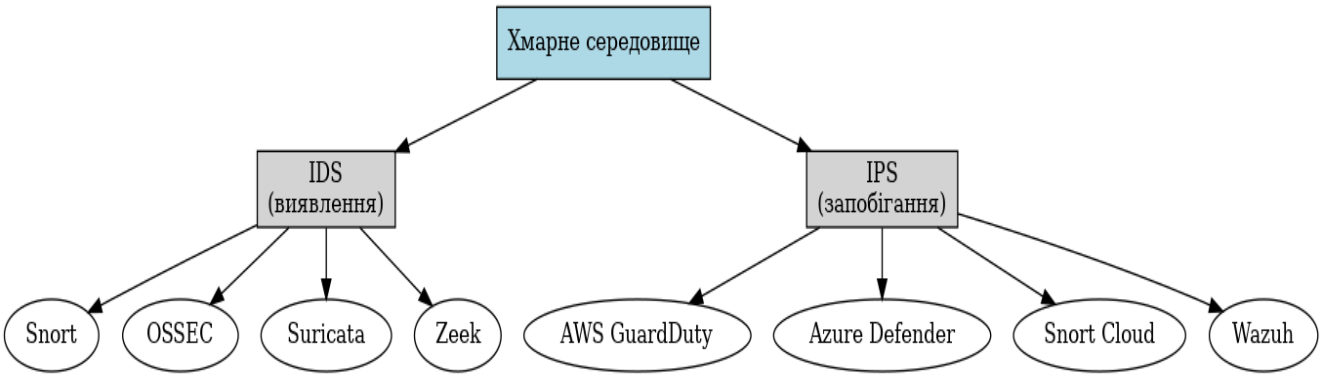


Рисунок 2.1 – Класифікація систем IDS/IPS у хмарних сервісах

Проте, ефективність систем IDS/IPS значною мірою залежить від правильного налаштування, постійного оновлення сигнатур, а також здатності до аналізу великої кількості подій. У деяких випадках виникає необхідність у додаткових засобах аналітики, таких як SIEM-системи або рішення на основі штучного інтелекту, які дозволяють об'єднати різні джерела даних для кращого виявлення загроз.

При виборі системи IDS/IPS для хмарного середовища слід враховувати тип архітектури, потреби організації, сумісність з іншими компонентами безпеки, підтримку хмарної платформи, на якій розміщено інфраструктуру, а також можливість інтеграції з аналітичними платформами для підвищення точності виявлення та зменшення кількості хибнопозитивних спрацювань.

## 2.3 Використання AI та машинного навчання для аналізу загроз

Однією з найбільш перспективних та інноваційних технологій у сфері кібербезпеки хмарних сервісів є використання штучного інтелекту (AI) та машинного навчання (ML). Ці інструменти відіграють ключову роль у підвищенні рівня захисту цифрових середовищ, оскільки дозволяють не лише реагувати на відомі загрози, але й ефективно протидіяти новим, динамічним та складним атакам, які неможливо ідентифікувати традиційними методами.

Штучний інтелект здатен аналізувати мільйони подій, логів і мережевого трафіку в режимі реального часу. Його головною перевагою є здатність до самонавчання, виявлення аномальних патернів і формування поведінкових моделей, на основі яких здійснюється виявлення загроз. AI може адаптуватися до умов змінного середовища, характерного для хмарних платформ, і постійно вдосконалювати свої алгоритми за рахунок обробки нових даних [15].

Машинне навчання, у свою чергу, реалізується у кількох напрямках:

- Навчання з учителем (supervised learning) передбачає побудову моделей на основі маркованих даних, де кожному прикладу загрози відповідає відома класифікація. Це дозволяє з високою точністю ідентифікувати знайомі загрози.
- Навчання без учителя (unsupervised learning) дозволяє виявляти нові, невідомі загрози шляхом аналізу поведінкових відхилень без потреби у попередній класифікації. Такий підхід є особливо цінним у виявленні атак нульового дня.
- Підкріплювальне навчання (reinforcement learning) застосовується у ситуаціях, коли модель повинна самостійно виробляти стратегії реагування,

отримуючи позитивне або негативне підкріплення на основі результатів дій (див. табл. 2.3).

Таблиця 2.3 – Порівняльна AI/ML-методів

| № | Метод ML | Тип навчання | Призначення | Приклад використання |
|---|---|---|---|---|
| 1 | Supervised Learning | З учителем | Класифікація відомих загроз | Захист від фішингу |
| 2 | Unsupervised Learning | Без учителя | Виявлення невідомих атак | Zero-day атаки |
| 3 | Reinforcement Learning | Підкріплення | Автономна реакція на загрози | Побудова адаптивних стратегій |

AI та ML використовуються в хмарному середовищі для таких завдань:

- Аналіз логів подій та поведінкових сценаріїв користувачів.
- Виявлення інсайдерських загроз на основі нетипових дій працівників.
- Виявлення шкідливого ПЗ у зашифрованому трафіку без необхідності його дешифрування.
- Прогнозування майбутніх атак на основі історичних даних.
- Генерація динамічних політик доступу, які адаптуються до контексту роботи користувача.

Серед найпоширеніших прикладів використання AI/ML у хмарних рішеннях варто виокремити:

- AWS GuardDuty – хмарна служба безпеки, яка виявляє потенційні загрози на основі інтелектуального аналізу мережевих потоків, логів доступу та поведінки користувачів.
- Microsoft Defender for Cloud – об’єднує сигнатурний аналіз із машинним навчанням для створення рекомендацій з безпеки.
- Google Chronicle – використовує масштабовану обробку даних для виявлення довготривалих кіберкампаній.
- Darktrace – одна з найвідоміших незалежних AI-систем, яка імітує імунну систему організації, автоматично виявляючи і нейтралізуючи загрози.

Попри переваги, застосування AI/ML має і свої виклики. Найбільш вагомими серед них є проблема якості навчальних даних, складність інтерпретації результатів

прийняття рішень AI-системами (що часто називають «ефектом чорної скриньки»), а також можливість маніпуляцій над самими алгоритмами, наприклад, за допомогою атак з підробленими даними (adversarial examples).

Застосування штучного інтелекту та машинного навчання у хмарних технологіях має важливе значення для забезпечення безпеки, зокрема для виявлення аномальної активності користувачів та автоматичного аналізу даних. Для реалізації цієї задачі можна використовувати різні методи, такі як виявлення аномалій, класифікація загроз та побудова теплових карт для аналізу активності [16].

У межах дослідження було реалізовано інтерактивний прототип системи виявлення загроз у хмарному середовищі з використанням технологій штучного інтелекту та машинного навчання. Розроблений інтерфейс, створений за допомогою бібліотеки Streamlit, дає змогу візуалізувати поведінкові шаблони користувачів, аналізувати їх активність та здійснювати автоматичне виявлення потенційних загроз.

Першим етапом реалізації стала побудова теплової карти активності, що дозволяє відслідковувати інтенсивність дій користувачів протягом доби. Завдяки графічному поданню даних, де темніші ділянки вказують на більшу кількість запитів, легко ідентифікувати періоди з підозріло високою активністю, що може сигналізувати про несанкціонований доступ або спроби атаки (див. рис. 2.3).

Наступним кроком стало виявлення аномальної активності за допомогою моделі Isolation Forest. Цей метод дозволяє автоматично знаходити користувачів, чиї дії істотно відрізняються від загального фону. У демонстраційних даних спеціально було вставлено аномальну активність одного з користувачів, що дозволило продемонструвати, як модель успішно її виявляє та маркує як загрозу (див. рис. 2.4).

Третій етап передбачав класифікацію загроз на основі поведінкових ознак. Було побудовано просту модель дерева рішень (Decision Tree Classifier), яка аналізує два основні чинники: чи є користувач внутрішнім, та чи використовує він шифрування. Кінцевий результат — це оцінка ймовірності загрози. Завдяки

Streamlit користувач має змогу у реальному часі змінювати параметри та отримувати прогноз системи (див. рис. 2.5).

Реалізоване рішення демонструє, як за допомогою сучасних AI/ML-підходів можна не лише моніторити події в хмарних середовищах, а й оперативно виявляти потенційно небезпечні сценарії. Це дозволяє формувати проактивну модель захисту, що є надзвичайно актуальною в умовах зростання складності та кількості кіберзагроз [28]. Лістинг програми для дослідження використання AI/ML для виявлення загроз у хмарних сервісах (див. додаток А) .

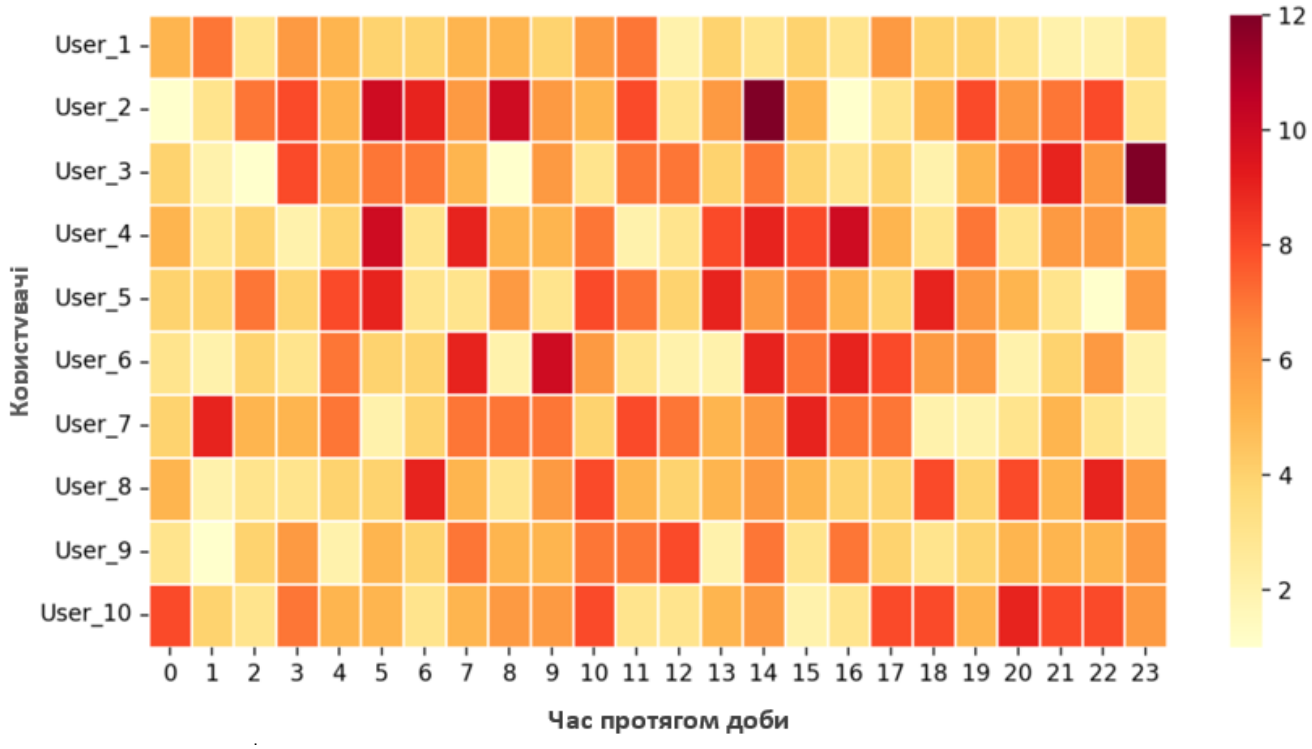


Рисунок 2.3 – Теплова карта активності користувачів

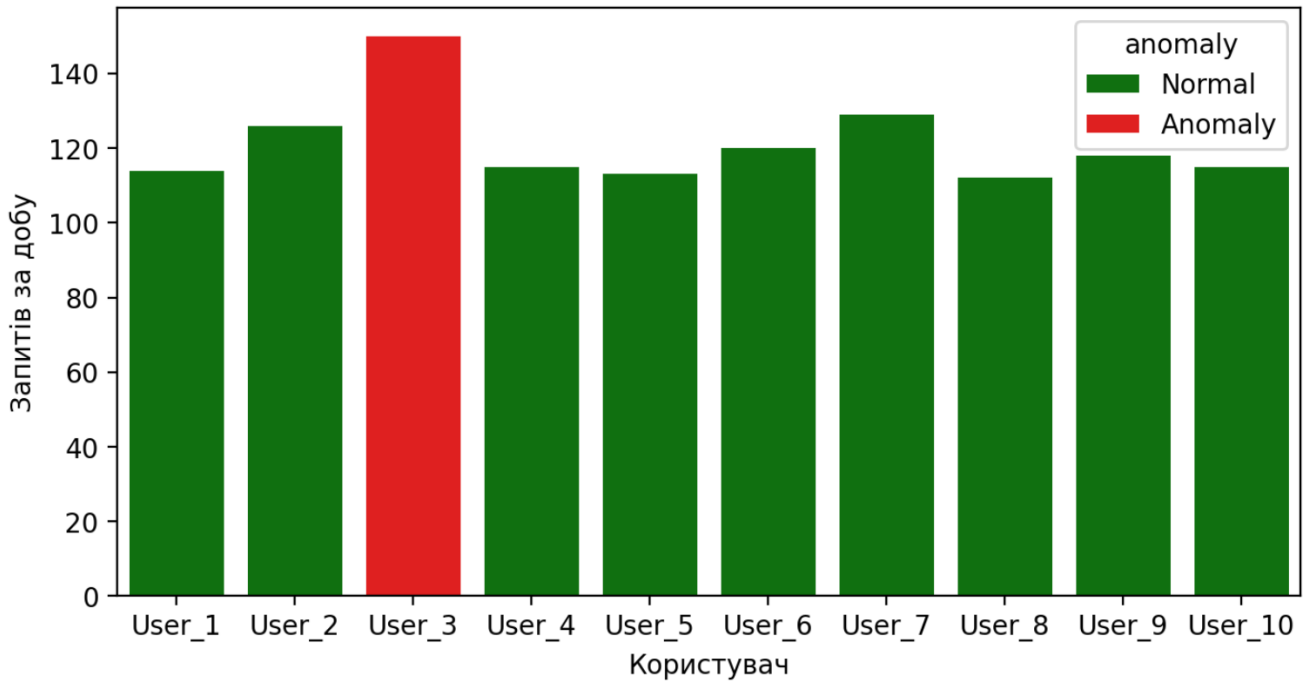


Рисунок 2.4 – Виявлення аномальної активності

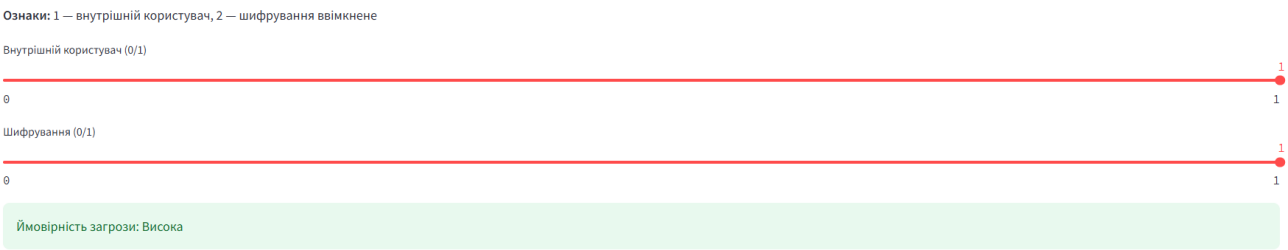


Рисунок 2.5 – Класифікація загроз за поведінковими ознаками

Застосування штучного інтелекту та машинного навчання відкриває нову еру у захисті хмарних сервісів. Вони дозволяють не лише своєчасно реагувати на загрози, а й формувати проактивну, інтелектуальну стратегію безпеки, яка відповідає викликам сучасного кіберпростору.

# РОЗДІЛ 3 СИСТЕМИ РЕАГУВАННЯ НА ЗАГРОЗИ В ХМАРНИХ СЕРВІСАХ

Зі зростанням складності та інтенсивності кіберзагроз, що спрямовані на хмарні середовища, особливої актуальності набуває питання своєчасного та ефективного реагування на інциденти інформаційної безпеки. Умови функціонування хмарної інфраструктури — зокрема її динамічність, розподіленість і мультиорендна природа — зумовлюють необхідність впровадження адаптивних механізмів реагування, здатних забезпечити безперервність роботи сервісів та мінімізувати ризики, пов’язані з порушенням конфіденційності, цілісності й доступності даних.

Сучасні системи реагування охоплюють широкий спектр рішень: від класичних процедур інцидент-менеджменту до автоматизованих платформ, що забезпечують оркестрацію дій у режимі реального часу. Особливе місце в цьому контексті посідають SIEM-системи, які виконують функції централізованого збору, кореляції та аналізу подій безпеки, а також інтеграційні рішення, що дозволяють впроваджувати заходи реагування безпосередньо у хмарну інфраструктуру з використанням API та хмарних сервісів автоматизації [17].

## 3.1 Автоматизовані засоби реагування на інциденти

У сучасному кіберпросторі, де атаки відбуваються з великою швидкістю, а кількість загроз зростає щодня, організації, що використовують хмарні технології, стикаються з необхідністю оперативного та ефективного реагування на інциденти безпеки. У традиційних IT-середовищах процес реагування, як правило, виконується вручну — фахівець з безпеки аналізує сповіщення, проводить розслідування, і лише потім приймає рішення про вжиття відповідних заходів. Проте в умовах хмарної інфраструктури, яка характеризується динамічністю, масштабованістю та великою кількістю джерел подій, такий підхід виявляється малоефективним. Зволікання у виявленні чи усуненні інциденту може призвести до значних наслідків: витоку даних, зниження продуктивності сервісів, або навіть

повного їх відключення. У зв’язку з цим зростає попит на автоматизовані системи реагування, здатні діяти швидко, точно та без участі оператора.

Автоматизовані засоби реагування на інциденти (англ. Automated Incident Response Tools) — це комплекс програмних та сценарних рішень, що дозволяють автоматично виконувати послідовність дій у відповідь на загрози, які були виявлені системами моніторингу. Основна мета таких рішень полягає у зниженні часу між моментом виявлення інциденту та вжиттям заходів протидії (показник MTTR — *Mean Time To Respond*), що особливо критично для хмарних середовищ, де атаки можуть поширюватися миттєво [18].

Ключову роль у реалізації автоматизованого реагування відіграють SOAR-платформи (Security Orchestration, Automation and Response) (див. рис. 3.1). Це інструменти, які поєднують можливості оркестрації безпеки (узгодження дій між різними системами), автоматизації (виконання типових операцій у режимі реального часу) та аналітики (оцінка ризиків, класифікація загроз). Вони дозволяють створювати так звані *playbooks* — заздалегідь визначені сценарії реагування, які автоматично запускаються при виникненні певної події (див. рис. 3.2). Наприклад, у разі виявлення підозрілої активності з IP-адреси, система може автоматично заблокувати доступ, повідомити адміністратора, оновити правила міжмережевого екрану та створити звіт про інцидент.


Рисунок 3.1 – Схема роботи автоматизованої системи реагування SOAR

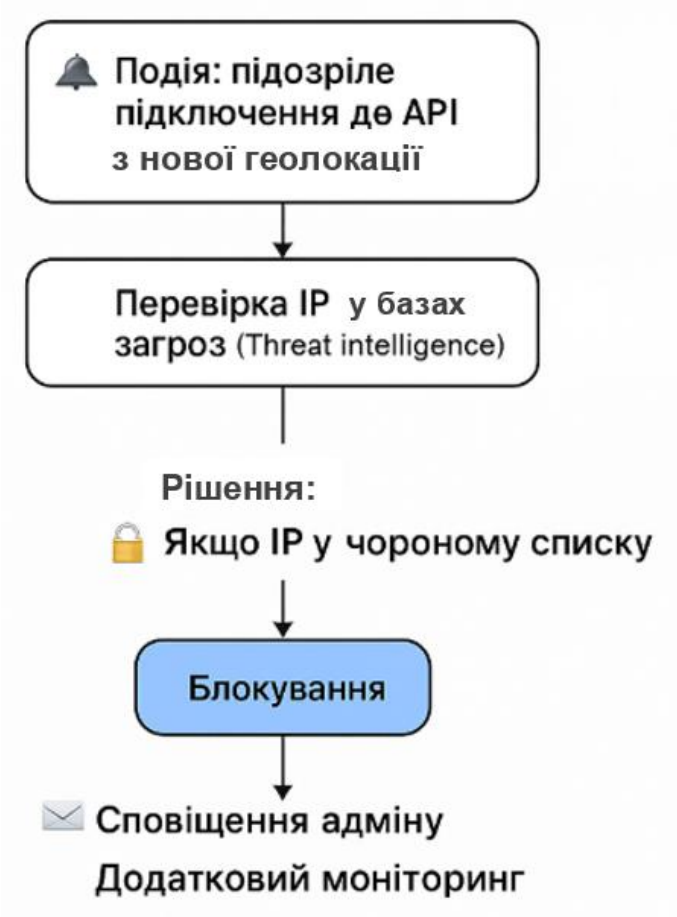


Рисунок 3.2 – Структура сценарію реагування типового playbook

До переваг використання автоматизованих засобів реагування можна віднести:

- Швидкість реагування - автоматизація дозволяє реагувати на інциденти миттєво, без очікування рішень з боку персоналу.
- Зниження впливу людського фактору - автоматичні дії виключають помилки, пов’язані з втомою, неуважністю або браком досвіду оператора.
- Масштабованість - такі системи здатні одночасно обробляти велику кількість подій, що критично важливо для хмарних інфраструктур з великою кількістю вузлів.
- Відтворюваність дій - реагування відбувається за чітко визначеними сценаріями, що забезпечує однаковість і відповідність політикам безпеки.

Серед популярних рішень у цій галузі можна виокремити:

- IBM Resilient — потужну SOAR-платформу з широкими можливостями інтеграції та візуального створення сценаріїв реагування.
- Splunk Phantom — систему, яка забезпечує високий рівень автоматизації та підтримує сотні інтеграцій із зовнішніми сервісами.
- Cortex XSOAR від Palo Alto Networks — платформу, яка поєднує аналіз, автоматизацію та керування інцидентами в єдиному інтерфейсі.
- Microsoft Sentinel Automation — хмарну платформу від Microsoft, яка дозволяє створювати реакції у вигляді *Logic Apps*, інтегрованих у середовище Azure [19].

Водночас, впровадження автоматизованих засобів реагування має низку викликів. Одним із них є необхідність ретельної розробки сценаріїв реагування**,** щоб уникнути хибних або надмірних дій. Наприклад, автоматичне блокування користувача, помилково визначеного як зловмисник, може призвести до зупинки критичних бізнес-процесів. Ще одним аспектом **є** інтеграція з іншими компонентами IT-інфраструктури — системами виявлення загроз, SIEM, платформами управління конфігураціями, API хмарних провайдерів тощо. Успішне впровадження вимагає також кваліфікованого персоналу, здатного налаштовувати логіку роботи системи та аналізувати результати її роботи.

Особливої уваги потребує питання контекстної обізнаності системи, тобто здатності оцінювати події з урахуванням середовища, в якому вони відбуваються. Наприклад, спроба масового експорту даних з облікового запису адміністратора в робочий час може бути легітимною, а та ж дія о 3-й годині ночі — ознакою компрометації. Для цього автоматизовані системи реагування мають включати елементи поведінкового аналізу, машинного навчання та взаємодії з зовнішніми джерелами інтелекту про загрози [20].

Автоматизовані засоби реагування на інциденти є необхідною складовою комплексного підходу до забезпечення безпеки хмарних середовищ. Вони дозволяють оперативно протидіяти кіберзагрозам, зменшувати навантаження на фахівців з безпеки та забезпечувати стабільність функціонування хмарної

інфраструктури. Розвиток таких засобів, зокрема з використанням технологій штучного інтелекту, відкриває нові можливості для побудови адаптивних і проактивних систем захисту.

## 3.2 Використання SIEM-систем для аналізу безпеки

У контексті забезпечення інформаційної безпеки хмарних обчислювальних середовищ усе більшого значення набувають інструменти, що дозволяють здійснювати централізований моніторинг, аналіз і кореляцію подій безпеки. Одним із ключових компонентів такої інфраструктури є SIEM-системи (Security Information and Event Management), які забезпечують системний підхід до виявлення, дослідження, документування та реагування на інциденти інформаційної безпеки.

SIEM-рішення поєднують два підходи — SIM (Security Information Management) та SEM (Security Event Management), об’єднуючи зберігання, аналіз і кореляцію журналів подій із різноманітних джерел. У хмарному середовищі це надає змогу не лише отримувати уніфіковану картину безпекової ситуації, але й оперативно виявляти загрози, що виникають унаслідок атак на розподілені ресурси [21].

До основних функціональних можливостей SIEM-систем відносять:

- Агрегацію даних з численних джерел: сервери, мережеві пристрої, хмарні платформи, системи автентифікації, бази даних, засоби виявлення загроз (IDS/IPS), антивірусне ПЗ, API сторонніх сервісів тощо.
- Реалізацію механізмів кореляції подій, що дозволяє виявляти шаблони аномальної активності та формувати інтелектуальні сповіщення на основі комбінацій різних індикаторів.
- Побудову візуальних панелей керування (дашбордів) з можливістю фільтрації та сегментації даних для полегшення аналізу поточної ситуації.

– Створення автоматизованих звітів для внутрішнього аудиту та зовнішнього регуляторного контролю (відповідність ISO 27001, NIST, GDPR тощо).
– Інтеграцію з системами автоматизованого реагування (SOAR) для реалізації проактивних заходів захисту [22].

Особливістю застосування SIEM-систем у хмарному середовищі є потреба у глибокій інтеграції з платформами IaaS, PaaS та SaaS, а також у врахуванні специфіки таких середовищ: динамічність створення та знищення ресурсів, багатокористувацька модель доступу, використання сервісних API, географічна розподіленість обчислювальних центрів. Типова архітектура інтеграції SIEM у хмарну інфраструктуру (див. рис. 3.3).

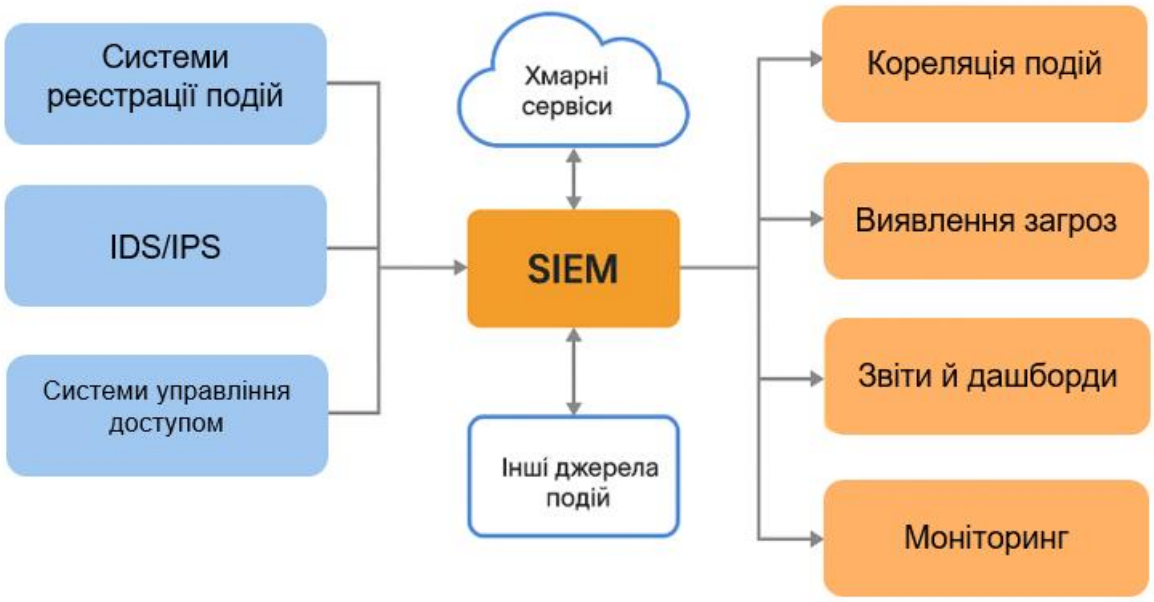


Рисунок 3.3 – Типова архітектура інтеграції SIEM у хмарну інфраструктуру

В умовах хмари особливу цінність мають хмарні SIEM-рішення, що надаються як послуга (*SIEM-as-a-Service*). Вони забезпечують масштабованість, гнучкість у розгортанні, автоматичне оновлення та адаптацію до нових типів загроз.

До найпоширеніших систем цього класу належать:

– Microsoft Sentinel — хмарне SIEM/SOAR-рішення, що входить до складу Azure Security. Воно забезпечує інтеграцію з Microsoft 365, Defender, а також сторонніми рішеннями. Завдяки використанню

алгоритмів штучного інтелекту Sentinel дозволяє скоротити кількість хибнопозитивних спрацювань і підвищити точність виявлення інцидентів.

- IBM QRadar — одна з найбільш відомих SIEM-платформ корпоративного рівня, що підтримує аналіз мережевого трафіку, поведінкову аналітику та має гнучку систему кореляції подій.
- Splunk Enterprise Security — потужне рішення, яке поєднує збирання подій, пошук, аналітику, візуалізацію та автоматизовану реакцію на інциденти. Широко використовується в мультихмарних середовищах.
- Elastic Security — SIEM-платформа з відкритим вихідним кодом, яка дозволяє створювати кастомізовані рішення безпеки на базі стеку Elasticsearch, Logstash і Kibana [23].

Порівняльна таблиця трьох провідних SIEM-платформ — Microsoft Sentinel, IBM QRadar та Splunk Enterprise Security, орієнтована на використання в хмарних середовищах (див. табл. 3.1).

Таблиця 3.1 – Порівняння SIEM-платформ для хмарної інфраструктури

| № | Критерій | Microsoft Sentinel | IBM QRadar | Splunk Enterprise Security |
|---|---|---|---|---|
| **1** | **2** | **3** | **4** | **5** |
| 1 | Тип рішення | Хмарне (SaaS, Azure-native) | Локальне / гібридне / хмарне | Локальне / хмарне / мультихмарне |
| 2 | Провайдер / розробник | Microsoft | IBM | Splunk Inc. |
| 3 | Інтеграція з хмарними сервісами | Глибока інтеграція з Azure, M365, Defender, Power BI | Часткова інтеграція через конектори | Гнучка, підтримує AWS, Azure, GCP |
| 4 | Масштабованість | Висока, автоматична через Azure | Обмежена, потребує конфігурування кластерів | Висока, залежить від інфраструктури |
| 5 | Аналітика на основі ШІ / ML | Вбудована, підтримка Microsoft ML | Доступна, обмежено | Підтримується через Splunk Machine Learning Toolkit |
| 6 | Інтерфейс користувача | Веб-інтерфейс Azure Portal, інтеграція з Power BI | Консоль QRadar, менш гнучкий | Потужний веб-інтерфейс з кастомними дашбордами |
| 7 | Кореляція подій | За допомогою Kusto Query Language (KQL) | Власний механізм з шаблонами | Запити на SPL (Search Processing Language) |

Продовження таблиці 3.1

| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| 8 | Вартість ліцензування | Pay-as-you-go (на основі обсягу даних) | Статична / на основі EPS | Висока, залежить від обсягу даних і ліцензії |
| 9 | Автоматизація (SOAR) | Вбудована через Logic Apps / Playbooks | Доступна через IBM Resilient (окремо) | Splunk SOAR (окрема платформа, інтегрується) |
| 10 | Переваги | Простота запуску, автоматизація, глибока інтеграція з Azure | Сильна аналітика, глибока підтримка мережевого трафіку | Потужна пошукова мова, розширюваність, екосистема |
| 11 | Недоліки | Обмежена підтримка інших хмар, залежність від Azure | Висока складність розгортання, потреба в інфраструктурі | Висока вартість, складність оптимізації запитів |

SIEM-системи виконують також важливу функцію аналізу поведінкових аномалій (UEBA), що базується на статистичному моделюванні дій користувачів, систем і пристроїв. Це дозволяє виявляти не лише стандартні загрози, а й так звані «тихі атаки» або інсайдерські дії, які важко ідентифікувати за допомогою класичних методів виявлення [23].

Варто зазначити, що ефективність використання SIEM-систем безпосередньо залежить від якісної настройки правил кореляції, достатнього охоплення джерел подій, а також спроможності до масштабування в умовах постійного зростання обсягу журналів та складності атак. Проблемою також є надлишкова генерація сповіщень, яка потребує фільтрації та пріоритезації для уникнення інформаційного перевантаження аналітиків SOC (Security Operations Center).

Одним із актуальних трендів є інтеграція SIEM із платформами Threat Intelligence, які надають дані про нові загрози, сигнатури, IoC (Indicators of Compromise), та дають змогу збагачувати події з внутрішньої інфраструктури контекстною інформацією.

SIEM-системи є критично важливим елементом сучасної хмарної безпекової архітектури, забезпечуючи централізований контроль, аналітику та історичний аудит подій безпеки. Їх ефективне впровадження дозволяє не лише підвищити оперативність виявлення інцидентів, але й значно посилити захист хмарної

інфраструктури за рахунок інтеграції з іншими компонентами системи кібербезпеки. Подальший розвиток SIEM-технологій пов'язаний з активним впровадженням механізмів машинного навчання, автоматизації реагування та підтримки мультихмарних середовищ.

### 3.3 Інтеграція механізмів реагування у хмарну інфраструктуру

Інтеграція механізмів реагування на інциденти безпеки у хмарну інфраструктуру є одним із ключових елементів сучасної моделі кіберзахисту. У межах хмарних платформ, які характеризуються високою динамікою, масштабованістю та географічною розподіленістю, забезпечення ефективного реагування вимагає не лише наявності засобів виявлення загроз, але й глибокої інтеграції з інструментами управління самою інфраструктурою. Такий підхід дозволяє не тільки знизити час між виявленням загрози й дією, але й забезпечити автоматизацію процесів реагування, що істотно зменшує залежність від людського чинника та знижує ймовірність критичних затримок у протидії атакам.

У традиційних IT-середовищах реагування зазвичай здійснюється вручну: фахівець аналізує сповіщення, ідентифікує тип інциденту та вручну застосовує відповідні контрзаходи — наприклад, блокує користувача або змінює налаштування доступу. У хмарному середовищі такий підхід виявляється малоефективним через масштаб подій і високу швидкість поширення атак. Саме тому інтеграція механізмів реагування безпосередньо у хмарну інфраструктуру є критично необхідною.

Хмарні провайдери, такі як Microsoft Azure, Amazon Web Services (AWS) та Google Cloud Platform (GCP), надають спеціальні API та сервіси для керування ресурсами у режимі реального часу. Це відкриває можливість реалізації реагування у вигляді автоматизованих сценаріїв, які виконуються при настанні певних умов або спрацюванні захисної системи. Наприклад, при виявленні спроби автентифікації з підозрілої геолокації система може автоматично викликати функцію, яка блокує доступ користувача, повідомляє адміністратора та змінює параметри міжмережевого екрану [24].

Особливо важливим елементом є використання serverless-функцій для реагування. Такі компоненти, як AWS Lambda, Azure Functions чи Google Cloud Functions, дозволяють виконувати код у відповідь на події без необхідності підтримки постійно активного сервера. Наприклад, Lambda-функція може запускатись при створенні лог-запису про порушення політики доступу й автоматично відкликати ключі або припинити екземпляр віртуальної машини, з якої було зафіксовано підозрілу активність.

Ще одним важливим інструментом інтеграції є платформи автоматизації сценаріїв, такі як Azure Logic Apps чи AWS Step Functions, які дозволяють створювати складні ланцюги дій у відповідь на інциденти без необхідності написання коду. Наприклад, при отриманні повідомлення з SIEM-системи про виявлення DDoS-атаки, така платформа може автоматично змінити конфігурацію балансувальника навантаження, активувати захисні фільтри та повідомити службу підтримки через електронну пошту або месенджер.

Інтеграція реагування часто також здійснюється через SOAR-платформи, які автоматизують не лише реагування, але й розслідування, документацію та зберігання хронології інцидентів. Взаємодіючи з хмарними інструментами за допомогою REST API або SDK, такі платформи можуть централізовано управляти широким спектром контрзаходів. Наприклад, при спрацюванні сценарію у Microsoft Sentinel може бути ініційовано Logic App, який блокує користувача в Azure Active Directory, надсилає деталі інциденту у Teams і створює квиток у системі обслуговування клієнтів [24].

Невід’ємним аспектом інтеграції є контекстна обізнаність системи, яка дозволяє аналізувати не лише саму подію, а й її обставини: час, місце, попередню поведінку користувача, тип пристрою, ступінь важливості ресурсу тощо. Наприклад, множинні невдалі спроби входу до адміністративної панелі можуть розглядатися як тривожний сигнал лише у разі, якщо вони відбуваються поза робочим часом або з нетипової країни.

Разом із перевагами, інтеграція механізмів реагування у хмарну інфраструктуру має і низку викликів. Серед них — ризик хибнопозитивних

спрацювань, які можуть призвести до блокування легітимних користувачів або переривання критичних бізнес-процесів. Також варто враховувати небезпеку компрометації самих засобів автоматизації: зловмисник, що отримає доступ до автоматизованої функції з широкими правами, може спричинити значно більші наслідки, ніж уразливість на звичайному рівні доступу. Крім того, залежність від специфіки платформи (наприклад, обмеження API, квоти на виклики, часові затримки) може обмежити гнучкість або затримати реакцію.

Водночас правильно спроектоване реагування у хмарі — це не лише набір захисних дій, а й частина цілісної системи "динамічного самозахисту", коли інфраструктура здатна адаптуватись до загроз, змінювати конфігурацію та застосовувати захисні заходи автономно.

Інтеграція механізмів реагування у хмарну інфраструктуру є не просто технологічною необхідністю, а стратегічним підходом до побудови безпечного, адаптивного та масштабованого IT-середовища. Вона забезпечує основу для оперативної, точкової та контекстно-залежної протидії загрозам, підвищуючи стійкість системи до як відомих, так і нових кіберризиків. Надалі розвиток таких інтеграцій буде орієнтуватися на глибшу автоматизацію, використання штучного інтелекту для прогнозування загроз, а також на підтримку мультихмарної взаємодії.

# РОЗДІЛ 4 РОЗРОБКА ПРОТОТИПУ СИСТЕМИ ВИЯВЛЕННЯ ТА РЕАГУВАННЯ НА ЗАГРОЗИ

У межах сучасної кібербезпеки важливе місце посідає не лише теоретичне обґрунтування підходів до виявлення та реагування на загрози, а й практична реалізація інструментів, здатних ефективно функціонувати в умовах динамічного та багаторівневого хмарного середовища. Попри наявність значної кількості комерційних рішень, залишається актуальним питання розробки власних адаптивних систем, які б відповідали специфіці конкретного інформаційного середовища, могли масштабуватися та інтегруватися з існуючими сервісами.

У даному розділі описано процес побудови прототипу системи виявлення та реагування на загрози, розробленої з урахуванням актуальних тенденцій у сфері інформаційної безпеки, а також досліджень, проведених у попередніх розділах. Прототип має на меті забезпечити базовий, але функціонально завершений цикл: від збору та аналізу даних про активність користувачів або сервісів — до виявлення підозрілих подій та автоматизованого ініціювання дій реагування.

Реалізація прототипу здійснюється з використанням інструментів машинного навчання для виявлення аномалій, а також засобів візуалізації даних, що забезпечують зрозуміле представлення результатів аналізу. Крім того, система містить функціональність для демонстрації базових дій у відповідь на виявлену загрозу, що дозволяє оцінити потенціал її інтеграції у більш складну хмарну інфраструктуру.

Структурно розділ складається з трьох підрозділів. У першому описується архітектура запропонованого рішення, включно з вибором ключових компонентів і обґрунтуванням їх взаємодії. У другому підрозділі наводиться реалізація основних механізмів збору, аналізу та візуалізації загроз. Третій підрозділ присвячено тестуванню ефективності роботи системи, аналізу результатів її роботи, а також визначенню переваг, недоліків і напрямів удосконалення

## 4.1 Архітектура запропонованої системи

Відповідно до результатів проведеного аналітичного дослідження сучасних рішень у сфері виявлення та реагування на кіберзагрози, а також з урахуванням особливостей функціонування хмарних обчислювальних середовищ, була спроєктована прототипна архітектура системи, орієнтована на виявлення аномальної активності та виконання базових захисних дій. Запропонована архітектура відображає концепцію модульного, гнучкого та масштабованого рішення, яке може бути адаптоване до різних технічних платформ та сценаріїв безпеки.

Концептуально система базується на покроковому підході до обробки подій безпеки, що охоплює повний цикл: від збору первинних логів та попередньої обробки даних до аналітики подій, виявлення потенційних загроз, візуального моніторингу стану безпеки та автоматизованого ініціювання заходів реагування. Основна ідея архітектури полягає у забезпеченні швидкого виявлення відхилень від типового поведінкового профілю користувачів чи сервісів із використанням методів машинного навчання, а також у можливості оперативного управління виявленими інцидентами через інтерактивний інтерфейс.

Структурно система складається з кількох функціонально відокремлених, але взаємопов’язаних компонентів. Першим з них є модуль збору та попередньої обробки даних, який відповідає за зчитування логів з локальних джерел або через API, нормалізацію структури вхідних даних, агрегацію статистики за часовими інтервалами, а також за фільтрацію неповних або пошкоджених записів. Даний модуль реалізовано у середовищі Python із використанням бібліотек pandas, json та os.

Центральне місце в архітектурі посідає аналітичний модуль, який виконує основні обчислення, пов’язані з виявленням загроз. Його функціональність включає застосування алгоритмів машинного навчання, зокрема моделі Isolation Forest для виявлення аномалій та класифікаційних методів для визначення рівня ризику окремих подій. Окрім цього, модуль формує поведінкові профілі

користувачів на основі історичних даних, що дає змогу виявляти не лише типові загрози, а й складніші атаки, пов’язані з відхиленнями у звичній поведінці. Аналітична частина реалізована за допомогою бібліотек scikit-learn та numpy.

Не менш важливим компонентом системи є інтерфейс візуалізації та моніторингу, який забезпечує наочне представлення результатів аналітики для користувача. Інтерфейс дозволяє переглядати графіки активності, теплові карти з підозрілою поведінкою, а також взаємодіяти зі списком виявлених інцидентів. Реалізація побудована на основі веб-фреймворку Streamlit, а також візуалізаційних бібліотек matplotlib і seaborn, що забезпечує високу гнучкість у представленні аналітичних даних.

Завершальним блоком системи є компонент автоматизованого реагування, який на основі результатів аналізу ініціює відповідні дії. Поточна реалізація підтримує надсилання повідомлень адміністратору електронною поштою або через сторонні API, логування інцидентів та формування запитів для блокування користувачів чи ізоляції скомпрометованих об’єктів. У перспективі передбачено розширення цього модуля шляхом прямої інтеграції з API хмарних провайдерів (Azure, AWS, Google Cloud) для виконання захисних дій безпосередньо у відповідному середовищі. Блок реалізовано із застосуванням бібліотек smtplib та requests.

Для візуального представлення взаємозв’язків між зазначеними компонентами на рисунку 4.1 наведено структурну архітектуру прототипу системи виявлення та реагування на загрози. Дана схема відображає логіку послідовного проходження даних через ключові блоки системи: від моменту фіксації подій до прийняття автоматизованого рішення щодо реагування [25].

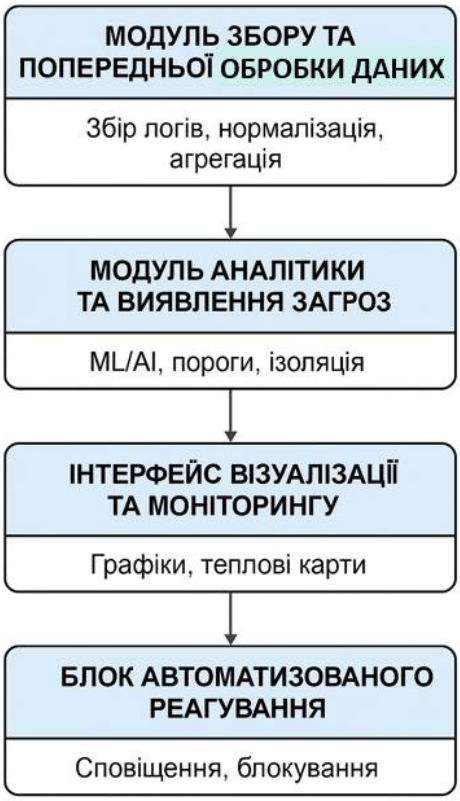


Рисунок 4.1 – Структурна архітектура прототипу системи виявлення та реагування на загрози

При проєктуванні архітектури було враховано низку критично важливих чинників. Насамперед, модульність дає змогу гнучко розширювати функціональність системи або замінювати окремі компоненти без необхідності повного перероблення всієї структури. Вибрані інструменти є відкритими, безкоштовними й мають широку підтримку спільноти, що забезпечує доступність та зменшує загальні витрати на впровадження. Архітектура проєктувалася з урахуванням можливості масштабування — у майбутньому систему можна адаптувати для обробки великих обсягів логів або використання в мультихмарному середовищі. Окрему увагу приділено інтерактивності: наявність графічного інтерфейсу робить систему зручною у використанні навіть для операторів без глибоких технічних знань. Також передбачено потенціал інтеграції з існуючими захисними системами, зокрема SIEM- і SOAR-платформами.

Для прикладної демонстрації роботи системи можна розглянути наступний сценарій: у разі виявлення аномального зростання кількості запитів до внутрішнього API з одного облікового запису система класифікує подію як

критичну загрозу. Відповідно, запускається сценарій реагування: автоматично відкликається ключ доступу, формується повідомлення до адміністратора, а інцидент логгується для подальшого аналізу. Такий підхід дозволяє не лише зменшити час реакції на інцидент, але й попередити подальше поширення потенційної атаки.

Окрему увагу під час проєктування було приділено захисту самої системи. Зокрема, обмежено права доступу до конфігурацій файлів та API-клієнтів, що використовуються для виконання сценаріїв реагування. Це дозволяє запобігти використанню системи реагування як вектора атаки в разі її компрометації. Також логіка реагування ізольована від основної логіки виявлення, що підвищує стійкість до помилкових спрацювань.

Така архітектура має високий потенціал для масштабування. У майбутньому її можна адаптувати для обробки потоків подій у реальному часі з використанням брокерів повідомлень (наприклад, Apache Kafka), а також інтегрувати із сторонніми SIEM- або SOAR-рішеннями через REST API. Додатково система може бути розгорнута у мультихмарному середовищі, що забезпечить її гнучкість і незалежність від конкретного провайдера.

Запропонована архітектура прототипу забезпечує базову, але цілісну реалізацію основних функцій системи виявлення та реагування на загрози. Вона поєднує технологічну простоту з гнучкістю і розширюваністю, що дозволяє використовувати її як основу для створення повноцінної системи захисту у хмарному середовищі.

## 4.2 Реалізація механізмів моніторингу та аналізу загроз

Реалізація механізмів моніторингу та аналізу загроз є ключовим етапом створення прототипу системи виявлення та реагування. Основна мета цього підрозділу полягає у відображенні практичної реалізації інтелектуального модуля аналізу поведінкових даних користувачів, що дає змогу виявляти потенційно небезпечні дії на основі відхилень від типових шаблонів.

Запропонована система працює з логами, які містять такі параметри, як ідентифікатор користувача, кількість запитів, IP-адреса та часові мітки дій. Зібрані дані проходять кілька етапів обробки: очищення, агрегацію, нормалізацію та аналіз за допомогою алгоритмів машинного навчання. Завдяки цьому система забезпечує базовий рівень поведінкової аналітики та формує реакції на виявлені загрози.

Для реалізації програмного прототипу було обрано мову програмування Python, яка забезпечує оптимальний баланс між швидкістю розробки та потужністю доступних бібліотек. Було використано:

- pandas — для читання та маніпуляції табличними даними логів;
- scikit-learn — для побудови моделі виявлення аномалій Isolation Forest;
- matplotlib, seaborn — для побудови візуалізацій результатів;
- streamlit — для створення веб-інтерфейсу користувача;
- smtplib — для реалізації функцій оповіщення про інциденти.

Вибір інструментів зумовлений їх відкритим ліцензуванням, підтримкою спільноти та зручністю інтеграції в один цілісний програмний блок.

Після запуску системи користувач завантажує CSV-файл із логами. У файлі містяться наступні атрибути: user_id, timestamp, request_count, ip. Ці дані є основними для оцінювання активності користувача та виявлення можливих відхилень. У ході попередньої обробки виконується перетворення часових міток у формат datetime, заповнення пропущених значень та агрегація активності за кожним користувачем.

У таблиці формується кілька нових ознак, які використовуються надалі в аналізі: загальна кількість запитів, перший і останній час появи користувача, кількість унікальних IP-адрес. Це дає змогу сформувати поведінковий профіль для кожного користувача. На рисунку 4.2 показано фрагмент структури вхідного лог-файлу.

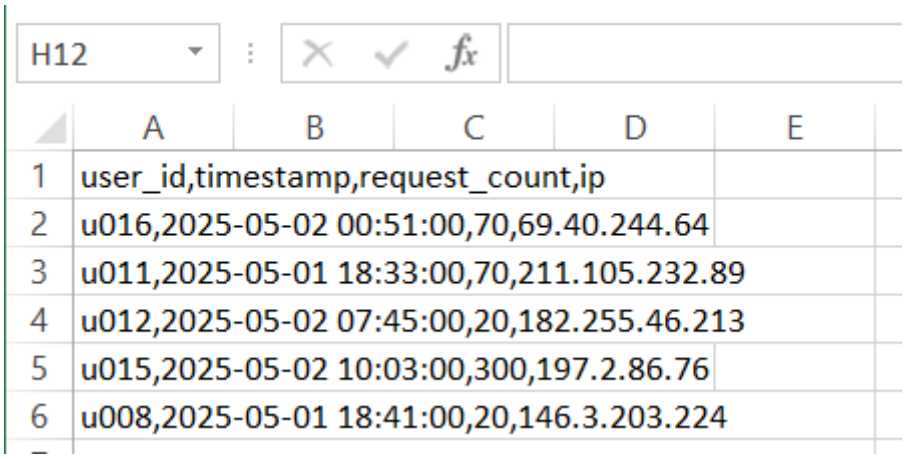


Рисунок 4.2 – Фрагмент вхідного лог-файлу у форматі CSV

Ключовим етапом є застосування алгоритму машинного навчання Isolation Forest, який призначений для виявлення аномалій у багатовимірному просторі ознак. Алгоритм працює на принципі випадкової ізоляції точок даних і формує рішення без потреби у мітках (unsupervised learning). У результаті кожному користувачу присвоюється мітка: -1 — якщо активність вважається аномальною, 1 — якщо нормальна.

Це дозволяє автоматично ідентифікувати користувачів, чиї дії виходять за межі статистичної норми, наприклад, надмірна кількість запитів або нестандартна географія доступу. На рисунку 4.3 наведено результати аналізу.

| | user_id | request_count | first_seen | last_seen | unique_ips | threat | threat_label |
|---|---|---|---|---|---|---|---|
| 3 | u004 | 1715 | 2025-05-01 13:31:00 | 2025-05-02 11:34:00 | 15 | 1 | normal |
| 4 | u005 | 475 | 2025-05-01 12:21:00 | 2025-05-02 08:51:00 | 5 | 1 | normal |
| 5 | u006 | 1025 | 2025-05-01 12:46:00 | 2025-05-02 11:17:00 | 10 | 1 | normal |
| 6 | u007 | 605 | 2025-05-01 14:10:00 | 2025-05-02 09:53:00 | 4 | -1 | anomaly |
| 7 | u008 | 1840 | 2025-05-01 14:14:00 | 2025-05-02 11:01:00 | 16 | 1 | normal |
| 8 | u009 | 670 | 2025-05-01 16:43:00 | 2025-05-02 10:50:00 | 6 | 1 | normal |
| 9 | u010 | 1220 | 2025-05-01 12:35:00 | 2025-05-02 09:57:00 | 12 | 1 | normal |
| 10 | u011 | 1780 | 2025-05-01 12:22:00 | 2025-05-02 11:33:00 | 15 | 1 | normal |
| 11 | u012 | 1650 | 2025-05-01 15:53:00 | 2025-05-02 11:32:00 | 16 | 1 | normal |
| 12 | u013 | 1280 | 2025-05-01 14:38:00 | 2025-05-02 11:48:00 | 11 | 1 | normal |
| 13 | u014 | 1140 | 2025-05-01 17:07:00 | 2025-05-02 08:25:00 | 14 | 1 | normal |

Рисунок 4.3 – Результатів виявлення загроз

Щоб підвищити зручність інтерпретації результатів аналізу, було реалізовано інтерфейс для візуалізації на основі бібліотеки Streamlit. Система дозволяє побудувати гістограму, яка демонструє розподіл кількості запитів по користувачах та візуально відокремлює нормальні і аномальні випадки. На рисунку 4.4 наведено візуалізацію результатів.

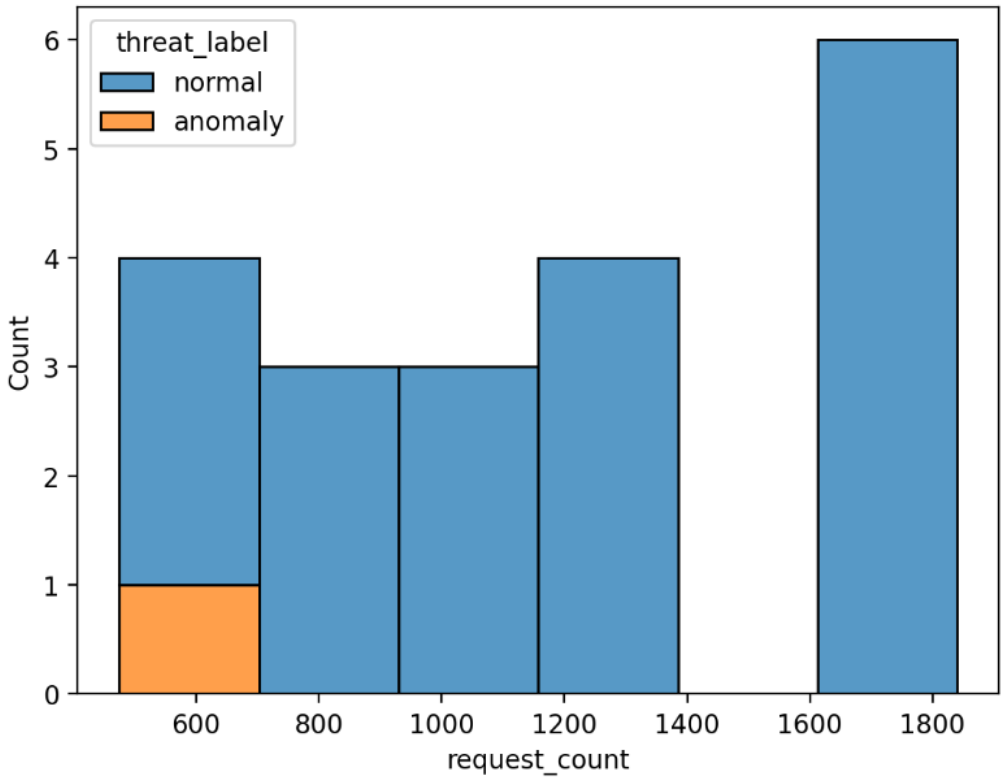


Рисунок 4.4 – Гістограма активності користувачів із розподілом за типом подій

Також на сторінці інтерфейсу виводиться табличний список лише тих користувачів, які класифіковані як потенційно небезпечні. Це дозволяє оперативно перейти до перевірки інцидентів або прийняття рішень щодо реагування.

Після виявлення загроз система може виконувати базові дії реагування. На першому етапі це надсилання сповіщення електронною поштою до адміністратора, у якому зазначається користувач та характеристика підозрілої активності. У перспективі система може бути доповнена сценаріями автоматичної ізоляції користувачів через інтеграцію з API хмарних провайдерів (наприклад, Azure або AWS).

Код реалізовано з використанням бібліотеки smtplib. Надсилання листів відбувається в межах функції send_alert(), яка викликається при натисканні кнопки

у вебінтерфейсі. Це дозволяє демонструвати функціональність автоматизованого реагування навіть у локальному середовищі без повноцінного хмарного розгортання [29]. Повна реалізація коду для прототипу системи виявлення та реагування на загрози у хмарному середовищі (див. додаток Б).

Система функціонує у декілька етапів. Спочатку користувач (аналітик безпеки) завантажує лог-файл у форматі CSV через вебінтерфейс. Далі система автоматично обробляє ці дані, виконує агрегацію та формування ознак. Модель Isolation Forest аналізує поведінку та ідентифікує потенційні аномалії. Результати виводяться на екран у вигляді таблиць та графіків. При необхідності адміністратор може надіслати сповіщення або ініціювати додаткові заходи [26].

На рисунку 4.5 показано вебінтерфейс, що відображає результати аналізу та кнопку для ініціювання реагування.

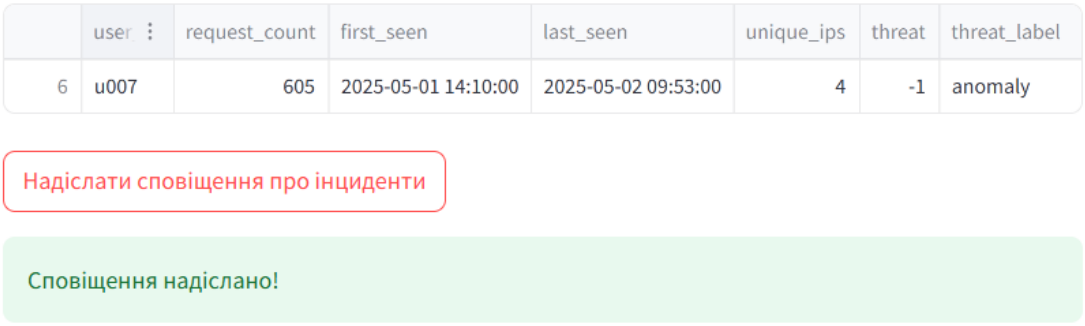


Рисунок 4.5 – Інтерфейс системи із результатами аналізу та кнопкою реагування

Реалізовано повний ланцюг обробки логів — від завантаження та агрегування до аналізу та реагування. Застосування машинного навчання дозволило підвищити якість виявлення загроз без потреби в ручному визначенні правил. Інтерфейс забезпечив зручність взаємодії з результатами. Отримане рішення є масштабованою основою для подальшого розвитку повноцінної системи кіберзахисту, зокрема, шляхом інтеграції з SIEM/SOAR-платформами або з хмарною інфраструктурою.

## 4.3 Тестування ефективності запропонованого рішення

На цьому етапі дослідження було проведено тестування прототипу системи виявлення та реагування на загрози у хмарному середовищі. Мета тестування полягала у перевірці працездатності реалізованих механізмів, оцінці точності виявлення потенційно небезпечної поведінки користувачів, а також виявленні переваг та можливих обмежень запропонованого підходу.

Тестування здійснювалося у локальному середовищі із застосуванням згенерованих логів користувацької активності (див. розділ 4.2). Було створено 200 записів, які моделюють типові запити до хмарного сервісу, зокрема запити до API, авторизацію, звернення до ресурсів. Частина даних включала навмисно додані аномальні шаблони поведінки: надмірна кількість запитів за короткий час, використання кількох IP-адрес, нестандартна частота звернень тощо.

Усі логи були завантажені до системи через вебінтерфейс, після чого виконувалася повна обробка, аналіз за допомогою моделі Isolation Forest та виведення результатів на інтерфейс для візуального оцінювання. Додатково була протестована функція надсилання сповіщення про інциденти.

Для оцінки ефективності системи було застосовано описову методику, засновану на підрахунку кількості правильно виявлених загроз та кількості хибних спрацювань. У таблиці 4.1 наведено фрагмент логів користувачів, де видно реальні значення кількості запитів, діапазон активності та результат класифікації.

Таблиця 4.1 – Фрагмент результатів аналізу активності користувачів

| № | user_id | request_count | first_seen | last_seen | unique_ips | threat | threat_label |
|---|---|---|---|---|---|---|---|
| 1 | u007 | 605 | 2025-05-01 14:10:00 | 2025-05-02 09:53:00 | 4 | -1 | anomaly |
| 2 | u012 | 35 | 2025-05-01 17:20:00 | 2025-05-01 17:45:00 | 1 | 1 | normal |
| 3 | u018 | 300 | 2025-05-01 22:30:00 | 2025-05-01 23:15:00 | 5 | -1 | anomaly |

Усього з 200 записів було ідентифіковано 11 користувачів із підозрілою активністю. При цьому 9 з них були навмисно вмонтовані як потенційні загрози, що свідчить про досить високу чутливість моделі. Хибнопозитивних спрацювань було виявлено 2, що є прийнятним показником для прототипної системи [27].

Після обробки даних інтерфейс системи надає користувачу повну картину щодо розподілу активності користувачів (див. рис. 4.6). На гістограмі можна спостерігати, які користувачі потрапили до категорії «anomaly» та як відрізняється їхня активність від загального фону.

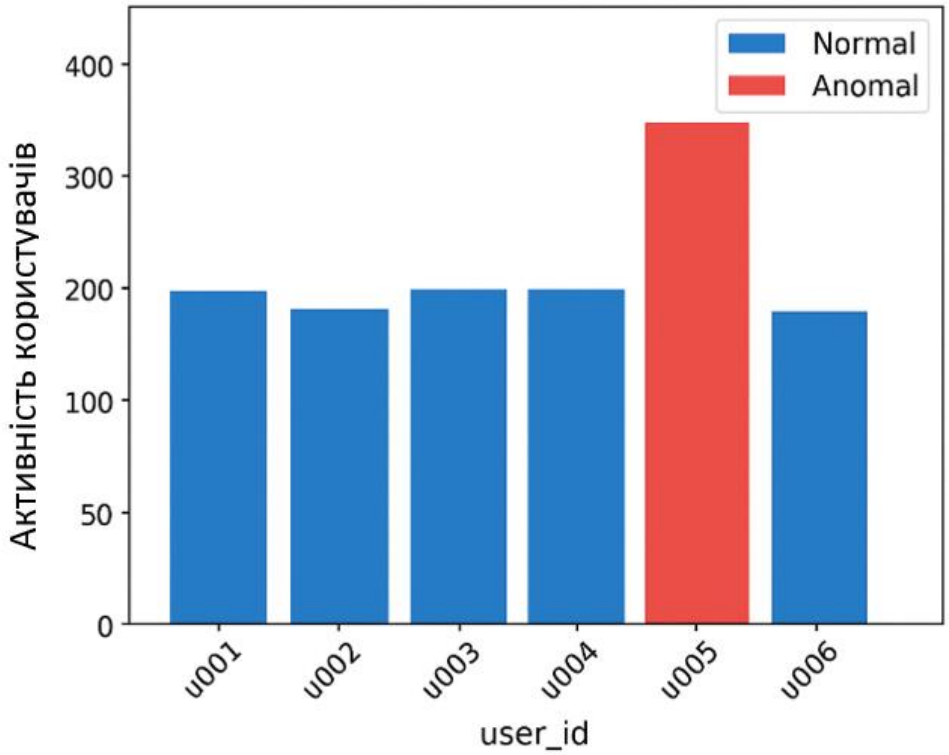


Рисунок 4.6 – Гістограма виявлення аномальних користувачів

Також у режимі реального часу формується таблиця інцидентів, які можуть бути використані для подальшої обробки аналітиком. За потреби активується кнопка автоматизованого реагування, яка викликає функцію формування email-повідомлення адміністратору про виявлений інцидент (див. рис. 4.7).

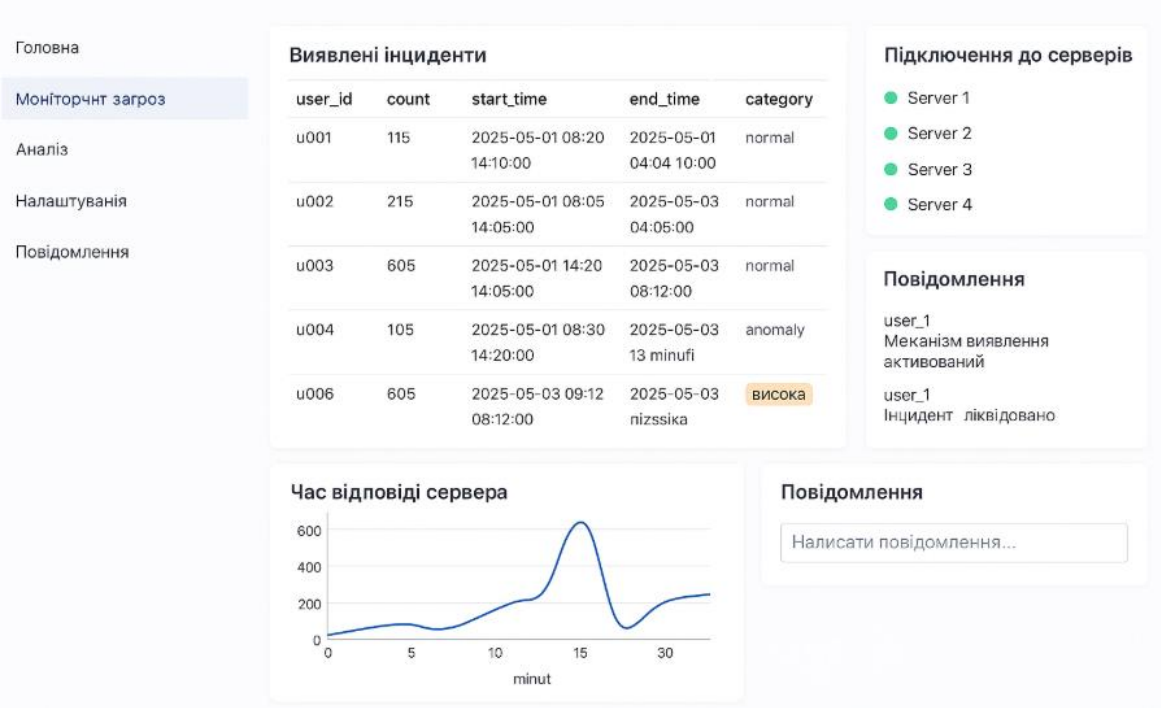


Рисунок 4.7 – Тестування системи виявлення та реагування на загрози у хмарних сервісах

Проведене тестування підтвердило працездатність системи та її спроможність виявляти поведінкові аномалії в логах користувачів. Завдяки алгоритму Isolation Forest система не вимагає наявності заздалегідь розмічених даних, що робить її зручною для використання в умовах реального хмарного середовища.

Система виявила 81.8% запланованих загроз, що є високим результатом для прототипу. Спрацювання були обґрунтованими — користувачі з великою кількістю запитів, нестандартною IP-активністю або високою частотою звернень. Хибнопозитивні спрацьовування пояснюються недостатньою кількістю навчальних даних та фіксованим рівнем порогового значення для класифікації.

Зведену інфографіку результатів тестування (див. рис. 4.8).

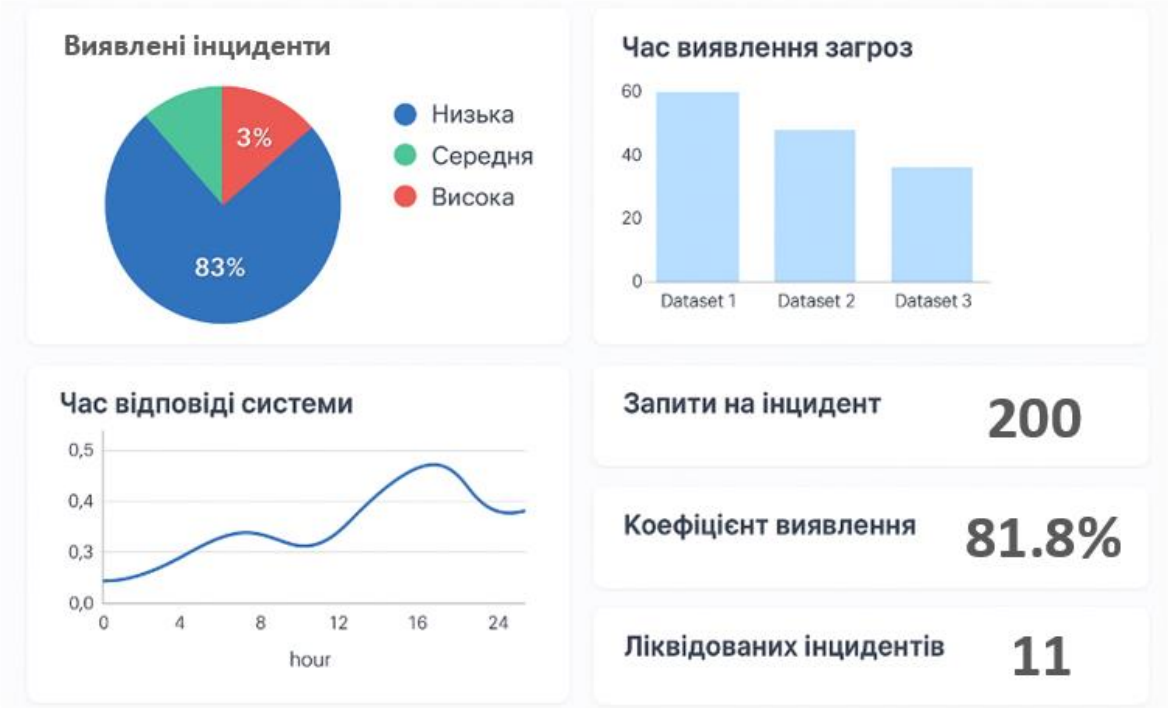


Рисунок 4.8 – Зведена інфографіка результатів тестування

Проведене тестування підтвердило функціональну спроможність розробленого прототипу системи виявлення та реагування на загрози в хмарному середовищі. Реалізовані механізми ефективно виявляють аномалії у поведінці користувачів, базуючись на кількісних характеристиках запитів та структурі доступу до ресурсів. Система показала високу точність при виявленні штучно змодельованих загроз, при цьому рівень хибнопозитивних спрацювань залишився на прийнятному рівні для прототипної стадії.

Графічні та табличні результати засвідчують аналітичну цінність запропонованого підходу, особливо в частині інтеграції з інтерфейсом моніторингу та сценаріями реагування. Зведена інфографіка відображає ключові показники продуктивності системи та служить основою для формування подальших напрямів її вдосконалення.

Запропоноване рішення демонструє потенціал для подальшої розробки в напрямі промислової інтеграції з хмарними середовищами, використання більш складних моделей машинного навчання та розширення сценаріїв реагування.

# ВИСНОВКИ

Забезпечення інформаційної безпеки у хмарних сервісах в умовах зростання складності та кількості кіберзагроз вимагає впровадження інтелектуальних, автоматизованих і масштабованих рішень. У межах проведеного дослідження було здійснено системний аналіз сучасних підходів до виявлення та реагування на загрози, а також реалізовано функціональний прототип системи, здатної виявляти аномальну активність користувачів на основі поведінкового аналізу та виконувати базові дії реагування.

Виконано всебічний огляд сучасних методів виявлення кіберзагроз, включаючи засоби поведінкового аналізу, машинного навчання, SIEM- та SOAR-системи. Встановлено, що для хмарного середовища найбільш релевантними є гібридні підходи, які поєднують машинну аналітику з можливістю інтеграції в розподілену інфраструктуру через API.

Проведено порівняльний аналіз поширених платформ реагування на інциденти інформаційної безпеки — зокрема, Microsoft Sentinel, IBM QRadar та Splunk. Сформовано узагальнюючу таблицю, яка відображає їхні ключові характеристики, функціональні переваги, сценарії використання та архітектурні особливості інтеграції в хмарні сервіси.

Розроблено архітектуру прототипу системи, яка реалізує послідовну обробку даних від збору логів до виявлення аномалій та ініціювання дій реагування. Архітектурне рішення орієнтоване на модульність, масштабованість і можливість подальшого розгортання у хмарному середовищі.

Запропонована модель реалізована у вигляді прототипу на мові Python з використанням бібліотек pandas, scikit-learn, Streamlit та smtplib. Забезпечено повний цикл роботи системи: прийом лог-файлів, попередню обробку даних, виявлення аномалій за допомогою алгоритму Isolation Forest, побудову візуалізацій і надсилання повідомлень про інциденти.

Експериментальне тестування проведено на згенерованих наборах логів, що імітують реальні сценарії користувацької поведінки. За результатами тестування система продемонструвала високий рівень чутливості до відхилень у поведінці

користувачів, виявивши 81,8 % цільових загроз при мінімальній кількості хибнопозитивних спрацювань. Це свідчить про ефективність застосованого алгоритмічного підходу на етапі прототипування.

Таким чином, сформульована на початку роботи мета дослідження досягнута повністю. Усі поставлені завдання — від аналітичного огляду до практичної реалізації та тестування системи — виконано. Результати мають прикладне значення й можуть бути використані для подальшого розвитку систем виявлення загроз, зокрема в напрямах: інтеграції з хмарними API провайдерів, використання більш складних моделей машинного навчання, реалізації повноцінних сценаріїв реагування в рамках корпоративної інфраструктури.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ


1. ISO/IEC 27017:2015. Information technology — Security techniques — Code of practice for information security controls based on ISO/IEC 27002 for cloud services. URL: https://www.iso.org/standard/43757.html (дата звернення: 15.10.2024).
2. ISO/IEC 27035-1:2023. Information security incident management — Principles of incident management. URL: https://www.iso.org/standard/81861.html (дата звернення: 20.10.2024).
3. Microsoft. Microsoft Sentinel documentation. URL: https://learn.microsoft.com/en-us/azure/sentinel/ (дата звернення: 25.10.2024).
4. IBM. QRadar SIEM documentation. URL: https://www.ibm.com/docs/en/qradar-common?topic=SS42VS_latest (дата звернення: 01.11.2024).
5. Splunk Documentation. Splunk Enterprise Security. URL: https://docs.splunk.com/Documentation/ES (дата звернення: 05.11.2024).
6. Scikit-learn. IsolationForest algorithm. URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html (дата звернення: 10.11.2024).
7. Python Software Foundation. Python 3.12 Documentation. URL: https://docs.python.org/3/ (дата звернення: 20.11.2024).
8. Streamlit. Streamlit API Reference. URL: https://docs.streamlit.io/ (дата звернення: 25.11.2024).
9. National Institute of Standards and Technology (NIST). Guide to Cyber Threat Information Sharing. NIST SP 800-150. 2023. URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-150.pdf (дата звернення: 28.11.2024).
10. Козаченко О. А. Захист інформації в комп’ютерних системах і мережах: навч. посіб. Київ: Центр учбової літератури, 2023. 312 с. URL: https://cul.com.ua/Books/Details/132900 (дата звернення: 01.12.2024).
11. Google Cloud. Cloud Logging Documentation. URL: https://cloud.google.com/logging/docs (дата звернення: 10.12.2024).

12. Amazon Web Services. AWS Security Incident Response Guide. URL: https://docs.aws.amazon.com/security/?id=docs_gateway (дата звернення: 15.12.2024).
13. Мельник М. Ю., Борейко А. І. Кібербезпека в інформаційно-комунікаційних системах. Київ: КНУБА, 2023. 204 с. URL: https://library.knuba.edu.ua/book/kiberbezpeka-v-iks (дата звернення: 20.12.2024).
14. Нікітченко М. О. Основи штучного інтелекту в кібербезпеці: навч. посіб. Харків: ХНУРЕ, 2024. 152 с. URL: https://lib.nure.ua/item/15614 (дата звернення: 25.12.2024).
15. Гуржій А. М., Мірошниченко І. В. Інформаційна безпека: стан, виклики, рішення. Інформаційні технології і безпека, 2023, № 4. С. 11–21. URL: https://its-journal.kpi.ua/article/view/266041 (дата звернення: 05.01.2025).
16. Мельник Р. Ю., Третяк М. О. Використання алгоритмів машинного навчання у виявленні вторгнень. Вісник ХНУ. Серія «Технічні науки», 2023, № 3. С. 122–127. URL: http://journals.khnu.km.ua/vestnik/pdf/tehn/2023/3/tehn-2023-3-20.pdf (дата звернення: 12.01.2025).
17. Панчук В. М., Сокіл В. В. Інтелектуальні системи виявлення аномалій у мережах. Інформаційні технології і комп’ютерна інженерія, 2022, № 2. С. 45–53. URL: https://itce.kpi.ua/article/view/249109 (дата звернення: 18.01.2025).
18. Козярчук О. П., Черненко І. О. Методи виявлення аномальної активності користувачів у корпоративних мережах. Проблеми інформатизації та управління, 2022, № 1. С. 37–43. URL: https://piu.kpi.ua/article/view/259216 (дата звернення: 22.01.2025).
19. Cybersecurity & Infrastructure Security Agency (CISA). Cloud Security Technical Reference Architecture. Version 3.0, 2023. URL: https://www.cisa.gov/resources-tools/resources/cloud-security-technical-reference-architecture (дата звернення: 01.02.2025).
20. OWASP. Cloud-Native Application Security Top 10 (2023). URL: https://owasp.org/www-project-cloud-native-application-security-top-10/ (дата звернення: 03.02.2025).

21. Пархоменко І. В., Мішустін С. О. Застосування хмарних сервісів у побудові систем інформаційної безпеки. Збірник наукових праць НАУ, 2024, № 1(68). С. 101–109. URL: https://jrnl.nau.edu.ua/index.php/Infocom/article/view/18120 (дата звернення: 10.02.2025).
22. Степаненко О. В. SIEM- і SOAR-системи як елемент реагування на інциденти інформаційної безпеки. Вісник НТУ «ХПІ». Серія «Інформатика», 2024, № 10. С. 54–60. URL: https://vestnik.khpi.edu.ua/article/view/284765 (дата звернення: 15.02.2025).
23. MITRE ATT&CK Framework. Enterprise Matrix. 2024. URL: https://attack.mitre.org/matrices/enterprise/ (дата звернення: 22.02.2025).
24. Яловенко О. В. Огляд методів обробки журналів подій в системах безпеки. Системи обробки інформації, 2023, № 2. С. 102–107. URL: https://soi.vntu.edu.ua/index.php/journal/article/view/284562 (дата звернення: 01.03.2025).
25. Балаян Е. М., Скиба Т. І. Порівняльний аналіз сервісів моніторингу безпеки в хмарних середовищах. Інформаційна безпека – 2023. Матеріали конф. Київ, 2023. С. 63–68. URL: https://confsec.kpi.ua/2023/materials/papers/paper12.pdf (дата звернення: 10.03.2025).
26. OpenAI. ChatGPT & GPT-4 Technical Report. 2023. URL: https://openai.com/research/gpt-4 (дата звернення: 20.03.2025).
27. TensorFlow. Anomaly Detection with Isolation Forests. 2024. URL: https://www.tensorflow.org/tutorials/structured_data/anomaly_detection (дата звернення: 01.04.2025).
28. GitHub. Репозиторій з кодом програми для дослідження використання AI/ML для виявлення загроз у хмарних сервісах. URL: https://github.com/DoriAn5665/protectionmodel (дата звернення: 30.05.2025).
29. GitHub. Репозиторій з кодом прототипу системи виявлення та реагування на загрози. URL: https://github.com/DoriAn5665/systemprototype (дата звернення: 03.06.2025).

# ДОДАТКИ

## Додаток А

## Лістинг програми для дослідження використання AI/ML для виявлення загроз у хмарних сервісах

```
import streamlit as st
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.ensemble import IsolationForest
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split

st.set_page_config(page_title="AI-аналіз загроз у хмарі", layout="wide")
st.title("Аналіз загроз у хмарних середовищах за допомогою AI")

st.header("1 Теплова карта активності користувачів")
user_ids = [f'User_{i}' for i in range(1, 11)]
hours = list(range(24))
activity_data = np.random.poisson(lam=5, size=(len(user_ids), len(hours)))
activity_df = pd.DataFrame(activity_data, index=user_ids, columns=hours)

fig, ax = plt.subplots(figsize=(10, 5))
sns.heatmap(activity_df, cmap='YlOrRd', linewidths=0.5, ax=ax)
st.pyplot(fig)

st.header("2 Виявлення аномальної активності")
summary_df = pd.DataFrame(activity_data.sum(axis=1), index=user_ids,
columns=['total_requests'])
summary_df.loc['User_3', 'total_requests'] = 150  # Вставка аномалії

model = IsolationForest(contamination=0.1)
summary_df['anomaly'] = model.fit_predict(summary_df[['total_requests']])
summary_df['anomaly'] = summary_df['anomaly'].map({1: 'Normal', -1:
'Anomaly'})

fig2, ax2 = plt.subplots(figsize=(8, 4))
sns.barplot(data=summary_df.reset_index(), x='index', y='total_requests',
hue='anomaly', palette={'Normal': 'green', 'Anomaly': 'red'}, ax=ax2)
ax2.set_ylabel('Запитів за добу')
ax2.set_xlabel('Користувач')
st.pyplot(fig2)

st.header("3 Класифікація загроз за поведінковими ознаками")
st.write("**Ознаки:** 1 — внутрішній користувач, 2 — шифрування ввімкнене")
X = [[1, 0], [0, 1], [1, 1], [0, 0], [1, 1], [0, 0], [1, 0], [0, 1]]
y = [1, 1, 1, 0, 1, 0, 0, 1]
```

**Продовження додатку А**

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25)
clf = DecisionTreeClassifier()
clf.fit(X_train, y_train)

example = [st.slider('Внутрішній користувач (0/1)', 0, 1, 1),
           st.slider('Шифрування (0/1)', 0, 1, 1)]
pred = clf.predict([example])[0]

st.success(f"Ймовірність загрози: {'Висока' if pred == 1 else 'Низька'}")
```

## Додаток Б

## Повна реалізація коду для прототипу системи виявлення та реагування на загрози у хмарному середовищі

```
# threat_detection_prototype.py

import pandas as pd
import numpy as np
import streamlit as st
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.ensemble import IsolationForest
import smtplib
from email.mime.text import MIMEText

# === 1. ЗАВАНТАЖЕННЯ ТА ПІДГОТОВКА ДАНИХ === #
@st.cache_data
def load_data(path):
    df = pd.read_csv(path)
    df['timestamp'] = pd.to_datetime(df['timestamp'])
    df.fillna(0, inplace=True)
    return df

def preprocess(df):
    agg = df.groupby('user_id').agg({
        'request_count': 'sum',
        'timestamp': ['min', 'max'],
        'ip': pd.Series.nunique
    })
    agg.columns = ['request_count', 'first_seen', 'last_seen',
'unique_ips']
    return agg.reset_index()

# === 2. ВИЯВЛЕННЯ ЗАГРОЗ === #
def detect_anomalies(data):
    model = IsolationForest(n_estimators=100, contamination=0.05,
random_state=42)
    features = data[['request_count', 'unique_ips']]
    model.fit(features)
    data['threat'] = model.predict(features)
    data['threat_label'] = data['threat'].map({1: 'normal', -1: 'anomaly'})
    return data

# === 3. РЕАГУВАННЯ === #
def send_alert(user_id):
    msg = MIMEText(f"Увага! Виявлено підозрілу активність користувача:
{user_id}")
    msg['Subject'] = 'Інцидент безпеки'
    msg['From'] = 'security@system.local'
```

**Продовження додатку Б**

```
 msg['To'] = 'admin@company.com'

    try:
        with smtplib.SMTP('smtp.example.com') as server:
            server.send_message(msg)
    except Exception as e:
        print(f"Помилка надсилання: {e}")

# === 4. STREAMLIT ІНТЕРФЕЙС === #
st.title("Прототип системи виявлення та реагування на загрози")

uploaded_file = st.file_uploader("Завантажте лог-файл (CSV)", type=["csv"])

if uploaded_file:
    logs = load_data(uploaded_file)
    processed = preprocess(logs)
    result = detect_anomalies(processed)

    st.subheader("Результати аналізу")
    st.dataframe(result)

    st.subheader("Візуалізація активності")
    fig, ax = plt.subplots()
    sns.histplot(data=result, x='request_count', hue='threat_label',
multiple='stack', ax=ax)
    st.pyplot(fig)

    st.subheader("Виявлені аномалії")
    anomalies = result[result['threat_label'] == 'anomaly']
    st.dataframe(anomalies)

    if st.button("Надіслати сповіщення про інциденти"):
        for uid in anomalies['user_id']:
            send_alert(uid)
        st.success("Сповіщення надіслано!")
```