

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:

В.о. завідувача кафедри
кібербезпеки
та захисту інформації

Іван ПАРХОМЕНКО

«17» травня 2024 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи

галузь знань 12 Інформаційні технології
(шифр і назва галузі знань)
спеціальність 125 Кібербезпека
(код і назва спеціальності)
освітній ступень магістр
освітньо-наукова програма Кібербезпека
(назва освітньої програми)

на тему: «Модель системи розслідування інцидентів у хмарному середовищі»

Виконавець: студентка II курсу, групи КБм-21

_____ Анна ТОРЧИЛО _____
(підпис) (Ім'я, ПРІЗВИЩЕ)

	Ім'я, ПРІЗВИЩЕ	Підпис
Науковий керівник	Сергій ТОЛЮПА	
Нормоконтроль	Яніна ШЕСТАК	

Київ 2024

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
В.о. завідувача кафедри
кібербезпеки
та захисту інформації
_____ Іван ПАРХОМЕНКО

«17» листопада 2023 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи

спеціальності _____ 125 Кібербезпека
(код і назва спеціальності)

освітній ступень _____ магістр

Здобувача(ки) _____ КБМ-21 _____ Торчило Анні Петрівні
(група) (прізвище ім'я по-батькові)

Тема кваліфікаційної роботи _____ Модель системи розслідування інцидентів у хмарному середовищі

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Рішення засідання кафедри кібербезпеки та захисту інформації факультету інформаційних технологій протокол № 5 від 15.11.2023 р.

2. МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень _____ Процес розслідування інцидентів безпеки у хмарному середовищі.

Предмет досліджень _____ Методи та інструменти розслідування інцидентів безпеки у хмарному середовищі.

Мета _____ Розробка системи для розслідування інцидентів у середовищах, що використовують хмарні технології.

Вихідні дані для проведення роботи Моделі збору доказів та пошуку першопричин інцидентів безпеки у хмарному середовищі.

3. ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Наукова новизна Вперше запропоновано інтегровану модель, яка враховує специфіку хмарних середовищ і використовує великі мовні моделі для аналізу логів, пошуку причин інцидентів безпеки й розробки плану реагування на інцидент.

Практична цінність Розробка інструменту, що може бути інтегрований до загальної системи управління інцидентами у хмарному середовищі для покращення процесу розслідування.

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Робота виконана у повному обсязі відповідно до теми.

5. ЕТАПИ ВИКОНАННЯ РОБОТИ

Найменування етапів робіт	Строки виконання робіт (початок-кінець)
Уточнення постановки задачі	17.11.2023 – 26.12.2023
Аналіз літературних джерел та теоретичного підґрунтя функціонування хмарних технологій	27.12.2023 – 15.01.2024
Аналіз нормативно-правової бази з управління інцидентами	16.01.2024 20.01.2024
Дослідження загроз і вразливостей притаманних хмарним середовищам	21.01.2024 29.01.2024
Формування вимог до системи управління інцидентами у хмарі	30.01.2024 05.02.2024
Розгляд існуючих інструментів для управління інцидентами у хмарі	06.02.2024 16.02.2024
Узагальнення моделі збору доказів у хмарних середовищах	17.02.2024 – 27.02.2024
Розробка моделі пошуку першопричин інцидентів у хмарних середовищах	28.02.2024 – 23.03.2024
Аналіз великих мовних моделей та їх особливостей	24.03.2024 29.03.2024
Дослідження прототипу розробленої моделі	30.03.2024 24.04.2024
Оформлення пояснювальної записки згідно з методичними рекомендаціями	25.04.2024 – 12.05.2024

Найменування етапів робіт	Строки виконання робіт (початок-кінець)
Подача пакету документів на розгляд ЕК	13.05.2024 – 18.05.2024

6. РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

Економічний ефект Зниження потенційних збитків зумовлених впливом інциденту безпеки через точне та швидке виявлення причин інциденту.

Соціальний ефект Покращення процесу управління інцидентами у хмарному середовищі.

7. ДОДАТКОВІ ВИМОГИ

Завдання видав

_____ (підпис)

Сергій ТОЛЮПА

(Ім'я, ПРІЗВИЩЕ)

Завдання прийняв
до виконання

_____ (підпис)

Анна ТОРЧИЛО

(Ім'я, ПРІЗВИЩЕ)

Дата видачі завдання: 17.11.2023 р.
Термін подання кваліфікаційної роботи до ЕК 17.05.2024 р.

РЕФЕРАТ

Кваліфікаційна робота складається зі вступу, чотирьох розділів, загальних висновків, списку використаних джерел та додатків. Основний текст займає 98 сторінок, містить 30 рисунків і 3 таблиці. Список використаних джерел, має обсяг 8 сторінки, і складається з 77 найменувань. Крім того, робота містить 4 додатки із загальною кількістю сторінок 8.

Об'єкт дослідження - процес розслідування інцидентів безпеки у хмарному середовищі.

Мета роботи - розробка моделі для розслідування інцидентів у середовищах, що використовують хмарні технології.

Предмет дослідження – методи та інструменти розслідування інцидентів безпеки у хмарному середовищі.

Методи дослідження використанні при підготовці дипломної роботи: аналіз наукової літератури та нормативно-правової бази; узагальнення державної та міжнародної практики, порівняння існуючих підходів та синтез моделі.

Практичною цінністю є можливість використання запропонованої моделі як інструменту, що може бути інтегрований до загальної системи управління інцидентами у хмарному середовищі для покращення процесу розслідування.

Наукова новизна: вперше запропоновано інтегровану модель, яка враховує специфіку хмарних середовищ і використовує великі мовні моделі для аналізу логів, пошуку причин інцидентів безпеки й розробки плану реагування на інцидент.

Актуальність теми: створення системи розслідування інцидентів у хмарному середовищі з використанням штучного інтелекту дозволить підвищити рівень захисту хмарних платформ від сучасних кіберзагроз.

Ключові слова: хмарне середовище, хмарні обчислення, розслідування інцидентів, штучний інтелект, машинне навчання, великі мовні моделі, інформаційна безпека.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

API	–	Application programming interface
APT	–	Advanced Persistent Threat
BART	–	Bidirectional Autoencoder Representations from Transformers
BERT	–	Bidirectional Encoder Representations from Transformers
CSA	–	Cloud Security Alliance
CSP	–	Cloud Service Provider
ENISA	–	The European Union Agency for Cybersecurity
GPT	–	Generative pre-trained transformers
IaaS	–	Infrastructure-as-a-Service
IAM	–	Identity and Access Management
ISO	–	International Organization for Standardization
LLM	–	Large Language Model
NIST	–	National Institute of Standards and Technology
NLP	–	Natural language processing
PaaS	–	Platform-as-a-Service
PDCA	–	Plan-Do-Check-Act
PKI	–	Public Key Infrastructure
RCA	–	Root Cause Analysis
SaaS	–	Software-as-a-Service
SIEM	–	Security information and event management
SLA	–	Service Level agreement
SQLi		SQL injection
T5	–	Text-to-Text Transfer Transformer

ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1 ТЕОРЕТИЧНЕ ПІДГРУНТЯ РОЗСЛІДУВАННЯ ІНЦИДЕНТІВ У ХМАРНОМУ СЕРЕДОВИЩІ.....	11
1.1 Різновиди хмарних обчислень та їх особливості.....	11
1.2 Загрози безпеки в хмарному середовищі.....	14
1.3 Нормативно-правове поле управління інцидентами у хмарному середовищі	25
1.4 Процес управління інцидентами	32
Висновки за розділом 1.....	35
РОЗДІЛ 2 ОСОБЛИВОСТІ РОЗСЛІДУВАННЯ ІНЦИДЕНТІВ У ХМАРНОМУ СЕРЕДОВИЩІ.....	36
2.1 Виклики хмарної форензики.....	36
2.2 Розробка рекомендацій для процесу розслідування інцидентів у хмарі.....	43
2.3 Аналіз інструментів хмарної форензики	48
Висновки за розділом 2.....	52
РОЗДІЛ 3 РОЗРОБКА МОДЕЛІ СИСТЕМИ РОЗСЛІДУВАННЯ ІНЦИДЕНТІВ У ХМАРІ	54
3.1 Модель збору доказів.....	54
3.2 Модель пошуку першопричин інциденту	59
3.3 Порівняльний аналіз LLM моделей	62
3.4 Переваги та виклики використання LLM для розслідування інцидентів	66
Висновки за розділом 3.....	68
РОЗДІЛ 4 ДОСЛІДЖЕННЯ МОДЕЛІ СИСТЕМИ РОЗСЛІДУВАННЯ ІНЦИДЕНТІВ У ХМАРНОМУ СЕРЕДОВИЩІ	70
4.1 Вибір технологій та інструментів для реалізації моделі.....	70
4.2 Розробка прототипу системи розслідування інцидентів.....	72
4.3 Аналіз результатів тестування моделі	78

	8
4.4 Перспективи розвитку та адаптивне застосування моделі	85
Висновки за розділом 4.....	88
ВИСНОВОК.....	89
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	91
ДОДАТОК А.....	99
ДОДАТОК Б	100
ДОДАТОК В.....	104
ДОДАТОК Г	107

ВСТУП

Актуальність цієї роботи визначається тим, що у сучасному цифровому світі хмарні обчислення набули величезної популярності. Організації різних розмірів активно інтегрують хмарні технології для зменшення витрат на інфраструктуру. Проте така швидка та масова міграція в хмару створює нові виклики, пов'язані з їх безпекою. Хмарні середовища часто стають мішенню для зловмисників, адже вони містять величезний обсяг даних, а їхня складна архітектура спричиняє різноманітні вразливості.

Метою цієї кваліфікаційної роботи є розробка системи для розслідування інцидентів у середовищах, що використовують хмарні технології.

Для досягнення поставленої мети необхідно вирішити наступні *завдання*:

- провести аналіз літературних джерел і нормативно-правової бази у сфері управління інцидентами у хмарному середовищі;
- розглянути особливості процесу хмарної форензики;
- розробити рекомендації до процесу розслідування інцидентів у хмарі;
- розглянути існуючі підходи й інструменти для розслідування хмарних інцидентів;
- розробити прототип моделі пошуку першопричин інцидентів у хмарі.

Об'єктом дослідження в даній роботі є процес розслідування інцидентів безпеки у хмарному середовищі. Оскільки цей процес є складним завданням через динамічний характер хмарних платформ та розподілену архітектуру, яка ускладнює збір даних, виявлення й аналіз загроз і, відповідно, реагування на інциденти для того щоб знизити потенційні збитки, що були завдані зловмисником або ті, що виникли через помилки в конфігураціях системи.

Предметом дослідження в даній роботі є методи та інструменти розслідування інцидентів безпеки у хмарному середовищі.

Значний вклад в розвиток цієї тематики внесли Akter S., Mandal P., Rajput I., Simou, S., Kalloniatis C., Gratzalis S., Katos V., Psalidas M., Zawoad S., Hasan R, Malik A., Bhatti, D., Park, T., Ishtiaq H., Ryou J., Kim K. та інші.

Однак проведений аналіз показав, що традиційні підходи виявилися недостатньо ефективними у боротьбі з унікальними технічними, організаційними та юридичними викликами, що виникають у хмарних інфраструктурах.

Науковою новизною цієї кваліфікаційної роботи є вперше запропонована інтегрована модель, яка враховує специфіку хмарних середовищ і використовує великі мовні моделі для аналізу логів, пошуку причин інцидентів безпеки й розробки плану реагування на інцидент.

Використання штучного інтелекту у системах розслідування інцидентів відкриває нові можливості для автоматизації процесу управління інцидентами. Методи машинного навчання дозволяють аналізувати великі обсяги даних, виявляючи аномалії та характерні для атак поведінкові патерни, які можуть бути непомітними для аналітика. Завдяки високій здатності адаптуватися до нових типів загроз, штучний інтелект стає невід'ємною частиною сучасних систем інформаційної безпеки.

Практична цінність результатів дослідження полягає в тому, що розроблена модель може бути інтегрована до загальної системи управління інцидентами у хмарному середовищі, що дозволить покращити процес розслідування і зробити процес реагування швидшим та дієвішим.

Таким чином, дослідження у сфері створення системи розслідування інцидентів у хмарному середовищі з використанням штучного інтелекту є актуальним науковим завданням, що має значну теоретичну й практичну цінність. Така система дозволить підвищити рівень безпеки хмарних платформ, забезпечуючи більш дієвий захист від сучасних кіберзагроз.

РОЗДІЛ 1

ТЕОРЕТИЧНЕ ПІДГРУНТЯ РОЗСЛІДУВАННЯ ІНЦИДЕНТІВ У ХМАРНОМУ СЕРЕДОВИЩІ

1.1 Різновиди хмарних обчислень та їх особливості

Спостерігаючи тенденції на ринку ІТ-технологій, не можна не помітити, швидкий розвиток хмарної індустрії. Згідно з даними досліджень Precedence Research, що представлені нижче на рис. 1.1, очікується, що обсяг глобального ринку хмарних обчислень, який у 2022 році оцінювався в 480 мільярдів доларів, досягне позначки в 2297,37 мільярдів доларів до 2032 року [1].



Рисунок 1.1 - Прогноз обсягу ринку хмарних обчислень

Наразі більшість компаній розглядають хмару як технологічну платформу і засіб для розширення власних можливостей, проте Gartner прогнозує, що вже у 2028 році більшість організацій (тобто понад 50%) будуть використовувати хмарні потужності як невід’ємну частину своїх бізнес-операцій [2].

Національний інститут стандартів і технологій (NIST) визначає хмарні обчислення як модель забезпечення зручного, повсюдного мережевого доступу на вимогу до спільного пулу конфігурованих обчислювальних ресурсів (наприклад, мереж, серверів, сховищ, додатків і служб), які можуть швидко надаватися з мінімальними управлінськими зусиллями або взаємодією з провайдером послуг [3].

Для цієї моделі притаманні декілька основних характеристик, зокрема це можливість самостійно використовувати обчислювальні ресурси серед наявного об'єднаного пулу ресурсів, масштабувати їх відповідно до поточних потреб, отримувати широкий віддалений мережевий доступ та контролювати їх використання завдяки вимірюваності наданих послуг [3].

Розглянуті характеристики свідчать про те, що хмарні обчислення надають користувачам ряд значних переваг - перш за все, це гнучкість, надійність та зниження витрат, що й обумовлює їх високу популярність порівняно з традиційними системами.

Спираючись на міжнародні стандарти і Закон України «Про хмарні послуги», можна виділити три основні сервісні моделі хмарних послуг [3, 4]:

1. Програмне забезпечення як послуга (SaaS), що забезпечує користувачів готовим програмним забезпеченням, яке розміщується на хмарному сервері провайдера. Це дає змогу отримати доступ до програм з будь-якого пристрою з підключенням до інтернету, що дуже зручно. Провайдер, у свою чергу, відповідає за розміщення програмного забезпечення та управління даними.

2. Платформа як послуга (PaaS) - для розробників додатків та програмного забезпечення PaaS надає платформу, на якій можна проектувати, розробляти та тестувати свої продукти без необхідності інвестувати в інфраструктуру. Провайдер в такому випадку відповідає за управління мережами, сховищем, операційними системами та серверами, тоді як користувачі відповідають за додатки та дані.

3. Інфраструктура як послуга (IaaS) надає користувачам віддалену інфраструктуру на серверах провайдера, який керує сховищами та мережами. Користувачі ж відповідають за операційні системи, додатки, дані та проміжне програмне забезпечення.

Водночас, на використання хмарних послуг та особливості управління інцидентами значно впливають способи надання хмарних послуг, визначені чотирма типами моделей розгортання [4]:

1. Публічна хмара надає спільний доступ до хмарних ресурсів кільком компаніям, що дозволяє знизити витрати. Однак загальна інфраструктура публічної хмари ускладнює забезпечення безпеки даних, що може створювати труднощі для компаній, які повинні дотримуватися певних норм щодо захисту даних.

2. Приватна хмара зазвичай розгортається на підприємстві та дозволяє повністю контролювати інфраструктуру та дані. Хоча це більш витратний варіант, він є оптимальним для компаній, які повинні суворо дотримуватися вимог безпеки та відповідності нормативам. Завдяки повному доступу до всієї інфраструктури приватна хмара полегшує розслідування інцидентів.

3. Спільна хмара поєднує в собі риси публічної та приватної хмар, надаючи інфраструктуру для кількох компаній зі схожими вимогами до безпеки. Компанії можуть співпрацювати у спільному середовищі, водночас забезпечуючи приватність своїх даних.

4. Гібридна хмара дозволяє поєднувати публічні та приватні хмарні ресурси, оптимізуючи витрати та одночасно захищаючи конфіденційні дані. Це створює додаткові виклики для розслідування інцидентів, оскільки доступ до даних, розміщених у публічній частині, обмежений і залежить від провайдера.

На рисунку 1.2 можна побачити відмінну рису хмарних середовищ – модель спільної відповідальності залежно від типів сервісної моделі - якщо у локальному середовищі організація несе виключну відповідальність за всі типи безпеки середовища та даних, то коли інфраструктура переходить у приватне або гібридне хмарне середовище, відповідальність розподіляється між надавачем хмарних послуг та користувачем.

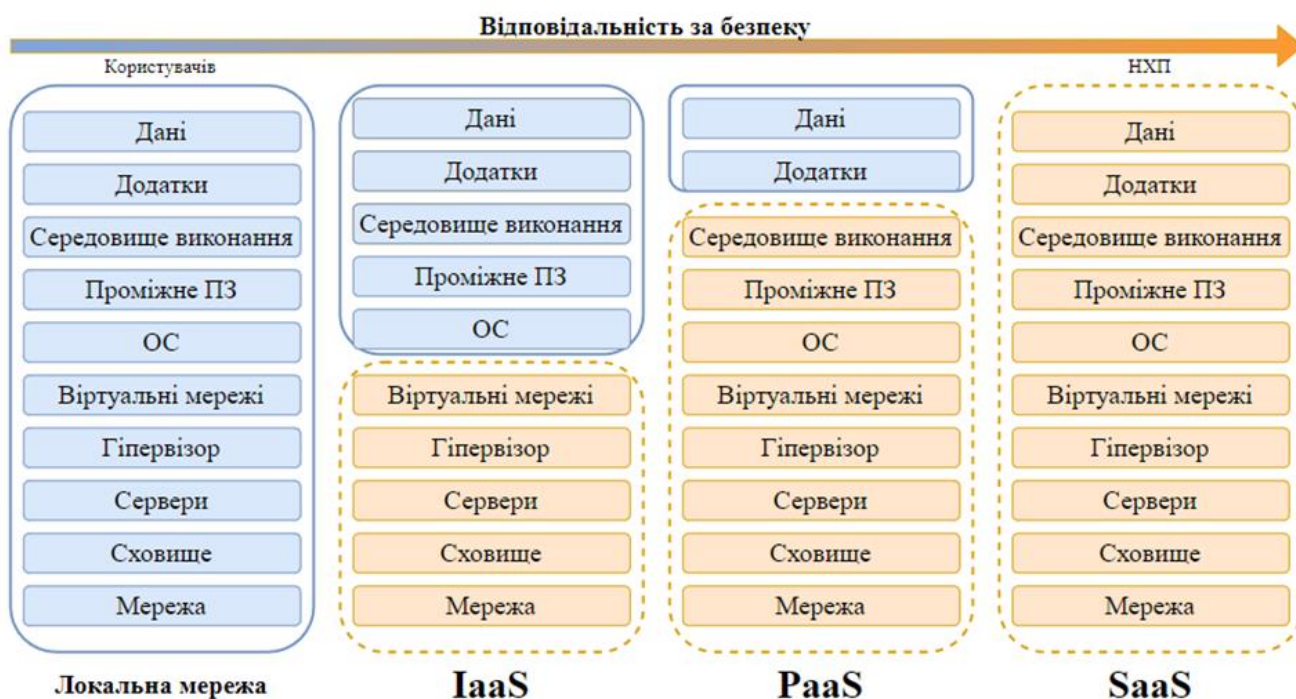


Рисунок 1.2 - Модель спільної відповідальності в хмарі.

Таким чином, при формуванні системи розслідування інцидентів важливо враховувати моделі обслуговування та розгортання хмарних технологій, щоб забезпечити ефективне реагування на інциденти, забезпечивши доступ до всіх необхідних даних, отримавши розуміння відповідальності всіх залучених сторін і зафіксувавши це в рамках Service Level Agreement (SLA).

1.2 Загрози безпеки в хмарному середовищі

Оскільки у сучасному цифровому світі хмарні середовища стали фундаментом для зберігання, обробки та обміну даними більшості інформаційних систем, разом з цим зростає і складність викликів безпеки, з якими стикаються користувачі хмарних середовищ. Згідно з даними за 2023 рік, опублікованими Statista, кількість найбільш поширених інцидентів безпеки у хмарі доходить до рівня кількості інцидентів у традиційних локальних середовищах, а подекуди й перевищує його [5].

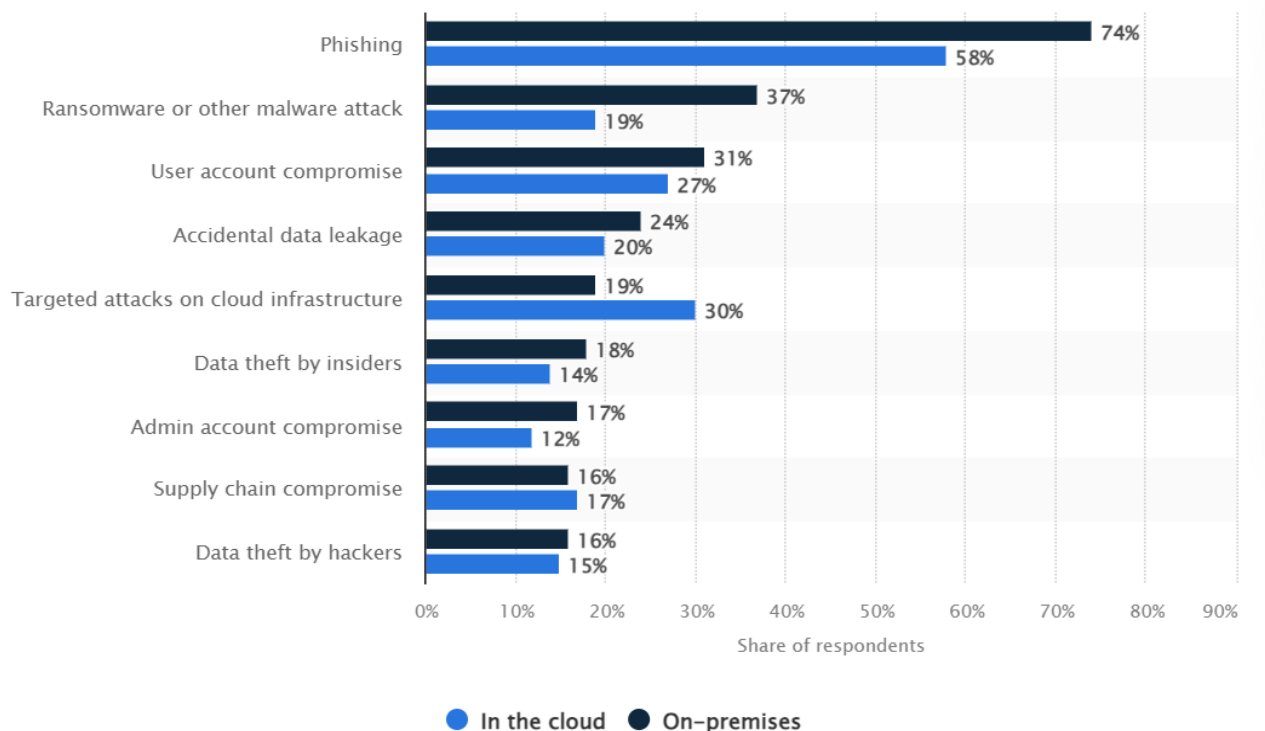


Рисунок 1.3 – Порівняння кількості атак у хмарному та локальному середовищі

Відповідно до даних інфографіки, представлених на рисунку 1.3, загалом більше половини (58%) компаній зазнали хмарних фішингових атак у 2023 році, а 30% повідомили про цілеспрямовані атаки на їх хмарну інфраструктуру. Продовжується тенденція на збільшення програм-вимагачів, націлених на хмарне середовище [6].

На рисунку 1.4 відображено найпоширеніші фактори компрометації хмари, які найчастіше використовувалося у 2023 році на основі даних клієнтів Google Cloud Platform [7]. Що стосується факторів зламу в хмарі, Google виявив, що слабкі або неіснуючі паролі, помилки конфігурацій займають перше місце, а кількість випадків зламу ключів API зросла.

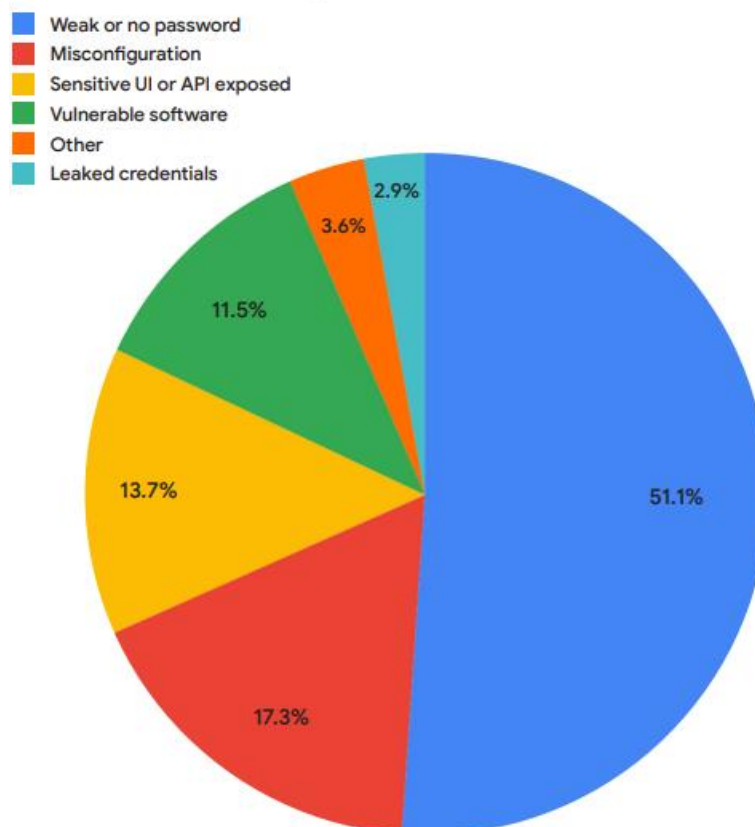


Рисунок 1.4 - Фактори компрометації хмарного середовища

Нижче наведено перелік основних загроз безпеці притаманних хмарним середовищам, що ґрунтується на основі останніх досліджень Cloud Security Alliance (CSA) [8].

1. Недоліки в управлінні ідентифікацією, обліковими даними, доступом і керування ключами.

Системи управління ідентичністю, обліковими даними та доступом включають інструменти та політики, які дозволяють організаціям керувати, контролювати та захищати доступ до цінних ресурсів. Хмарні обчислення змінюють традиційні практики управління системами внутрішньої ідентифікації та доступу (IAM), в них це питання стає більш суттєвим.

Події з безпеки та проникнення даних можуть відбуватися через наступне:

- недостатній захист облікових даних;
- відсутність регулярної автоматизованої зміни криптографічних ключів, паролів та сертифікатів;

- відсутність масштабованих систем iam;
- відсутність використання багатофакторної аутентифікації;
- відсутність використання надійних паролів.

Облікові дані та криптографічні ключі не повинні бути вбудовані у вихідний код або розповсюджуватися в загальнодоступних репозиторіях (наприклад, GitHub), оскільки існує високий ризик витоку та зловживання. Ключі повинні бути належним чином захищені за допомогою надійної інфраструктури відкритих ключів (PKI), щоб забезпечити виконання дій з управління ключами.

Системи IAM повинні масштабуватися для управління життєвим циклом для користувачів, а також провайдерів хмарних послуг. Системи управління ідентичністю повинні підтримувати негайне скасування доступу до ресурсів при змінах персоналу, таких як припинення роботи або переходи ролей. Такі процеси управління життєвим циклом ідентичності повинні бути інтегровані та автоматизовані в хмарних середовищах.

Використання системи багатофакторної аутентифікації вважається найкращою практикою для запобігання викраденню облікових записів. Крадіжка паролів може проявлятися в типових атаках переміщення по мережі, таких як "передача хешу". Система аутентифікації повинна підтримувати виконання політики сильних паролів, такої як перевірка надійних паролів та періоди політики ротації, визначеної організацією.

Управління криптографічними ключами, які використовуються для захисту даних у спокої, також повинно відбуватися централізовано й автоматизовано протягом їх життєвого циклу, включаючи створення, розподіл, зберігання, заміну та видалення. Такий підхід допомагає запобігти атакам, які передбачають несанкціонований доступ до ключів.

2. Ненадійні інтерфейси програмування додатків (API).

Хмарні сервіси використовують API для керування ресурсами та доступу до даних. API та інтерфейси користувача, як правило, є найбільш вразливими частинами системи і, подекуди, єдиним активом разом із публічною IP-адресою, доступною за межами довіреної організаційної межі. Їх дуже ймовірно

атакуватимуть безперервно; тому ці інтерфейси повинні бути розроблені таким чином, щоб захищати від як випадкових, так і зловмисних спроб обходу політики безпеки від аутентифікації та контролю доступу до шифрування та моніторингу активності.

Ненадійні API можуть бути використані для несанкціонованого доступу, виконання довільного коду або маніпулювання ресурсами хмари. У багатьох випадках організації відкривають хмарні API за допомогою різних інструментів оркестровки, таких як Kubernetes, без належної автентифікації та авторизації. Це дозволяє зловмисникам заволодіти API та використовувати або змінювати робочі навантаження для зловмисних цілей.

Як показано у звіті Orca про стан безпеки публічної хмари за 2022 рік, 70% організацій мали загальнодоступний сервер Kubernetes API, через що сервери API піддавалися спробам розвідки та потенційним атакам нульового дня [9].

Зручний спосіб усунути цю проблему - захистити API за допомогою ключів API або OAuth і обмежити доступ авторизованим користувачам. Також необхідно звернути увагу на впровадження контролю доступу та використання фаєрволів для захисту API від кіберзлочинців. Хорошою практикою є ретельний нагляд за такими пунктами, як інвентаризація, тестування, аудит та захист API від аномальної активності.

3. Неправильна конфігурація.

Ресурси на основі хмари є складними та динамічними. Це ускладнює їх налаштування. Традиційні контрольні засоби та підходи до управління змінами не є ефективними для використання у хмарах. Саме тому за даними Агентства національної безпеки [10], неправильні конфігурації хмари є найбільшою загрозою безпеці хмари - звіт про хмарну безпеку за 2023 рік, нещодавно опублікований компанією Check Point Software Technologies Ltd., показує, що неправильні конфігурації вважаються основною проблемою безпеки хмари, зважаючи на думку 59% респондентів [11].

Неправильна конфігурація виникає, коли обчислювальні ресурси налаштовані некоректно, що часто робить їх вразливими перед зловмисною

діяльністю. Деякі загальні приклади найпоширеніших неправильних конфігурацій у хмарі включають проте не обмежуються наступними [8]:

- Надмірні дозволи. Через невідповідність конфігурації надання надмірних дозволів хмарним ресурсам може призвести до надмірного доступу. Цей надмірний дозвіл на доступ до ресурсів дає можливість кіберзлочинцям скористатися цією неправильною конфігурацією для бокового переміщення в хмарній інфраструктурі.

Найкращий спосіб вирішити цю проблему — застосувати принцип найменших привілеїв і використовувати політики IAM для обмеження доступу.

- Необмежені відкриті мережеві порти. Якщо доступ RDP і SSH до серверів із загальнодоступних або зовнішніх мереж дозволено, це призводить до серйозних проблем із безпекою хмарних ресурсів. Тому необхідно обмежити доступ до мережевих портів відповідно до вимог із застосуванням правила найменших привілеїв. Вихідний трафік із сервера додатків слід жорстко контролювати, а доступ має надаватися лише тій програмі та серверу, яким вони потрібні.

- Відкриті бакети для зберігання. Залишення відкритого доступу до хмарних сховищ, таких як Google Cloud і Amazon S3, є серйозною неправильною конфігурацією, яка однозначно ставить під загрозу всі дані.

Зазначається, що 72% організацій мають принаймні один бакет з загальнодоступними дозволами на «читання», а 36% організацій зберігають у цих хмарних службах незашифровані секрети та ідентифікаційну інформацію [9].

- Відсутність логування та моніторингу. Багато організацій, які працюють у хмарному середовищі, часто не можуть правильно налаштувати або ввімкнути журналювання та моніторинг хмарних ресурсів, проте для організацій важливо відстежувати події безпеки, дані телеметрії та виявляти будь-які проблеми. Слід зосередитися на створенні автоматизованого моніторингу та сповіщень про будь-яку підозрілу діяльність, щоб ці недоліки можна було негайно виправити.

- Зберігання облікових даних за замовчуванням. Під час розробки багато розробників зберігають пароль за умовчанням для програмного забезпечення та служб хмарної інфраструктури. Однак у більшості випадків пароль за замовчуванням залишається на місці після розгортання, і це дозволяє

кіберзлочинцям отримати легкий доступ до системи. Вирішити цю проблему можна інтегрувавши сканування коду під час циклу розробки програми.

- Погано налаштоване автоматичне резервне копіювання

Погано налаштоване та незахищене автоматичне резервне копіювання є значною проблемою для організацій – вони зазвичай приділяють багато уваги, коли йдеться про безпеку головного сховища, але неефективно налаштовують резервне копіювання, що робить його вразливим до внутрішніх загроз.

Починаючи автоматичне резервне копіювання ваших даних під час міграції в хмару, обов'язково потрібно переконатися, що резервне копіювання повністю зашифровано. Також необхідно зменшити кількість доступу до резервних копій сховища.

- Відсутність сегментації мережі

При неправильному налаштуванні сегментації мережі критичні ресурси можуть стати вразливими для експлуатації. Використання віртуальної приватної хмари та груп безпеки мережі для сегментації мереж і обмеження ресурсів залежно від вимог безпеки може допомогти пом'якшити експлуатацію хмари.

Будь-яка неправильна конфігурація хмари може мати значні наслідки, зокрема видалення, зміну, витік даних, збій у роботі служби або захоплення облікового запису — усе це може завдати серйозної шкоди організації в результаті на перший погляд незначної помилки. Щоб уникнути цього компанії повинні використовувати автоматизацію та технології, які постійно сканують ресурси на наявність неправильних налаштувань та усувають проблеми в режимі реального часу.

4. Відсутність архітектури та стратегії безпеки хмари.

Дані стають вразливими перед різноманітними загрозами, коли організації вважають, що міграція до хмари - це просте перенесення існуючого ІТ-стеку та контролів безпеки до хмарного середовища. Недостатнє розуміння моделі спільної відповідальності за безпеку сприяє цій проблемі. На перший план часто виходять функціональність та швидкість міграції. Ці фактори призводять до відсутності

архітектури та стратегії безпеки в хмарі, що робить організації вразливими перед кібератаками.

Незалежно від розміру організації, розробка правильної стратегії безпеки, що відповідає бізнес-цілям організації, забезпечить організаціям міцний фундамент для здійснення діяльності в хмарі максимально ефективно та безпечно, адже кібератаки можуть серйозно позначитися на бізнесі, включаючи фінансові втрати, пошкодження репутації, юридичні наслідки та штрафи.

5. Небезпечна розробка програмного забезпечення

Під час процесу розробки безпека є досить м'якою, щоб полегшити розробникам внесення змін до коду та підвищення гнучкості. Однак розробники часто зберігають ту саму конфігурацію у виробництві, і це призводить до витоку даних або компрометації.

Сюди ж можна віднести й зростаючу проблему тіньових даних, оскільки розробникам легко копіювати дані в інші місця в хмарі, навіть не усвідомлюючи, що вони можуть містити конфіденційні дані або що елементи керування можуть бути втрачені під час переміщення даних. Однак правильна оцінка налаштувань коду перед їх впровадженням на етапі продукту гарантує, що всі дані захищено, а параметри безпеки за замовчуванням не підвищуються.

Окрім цього, більшість сучасних процесів розробки програмного забезпечення використовують сторонні бібліотеки, програми та елементи, щоб полегшити етап розробки. Протягом цього часу багато бібліотек отримують доступ до хмарних ресурсів. Оскільки продукт або послуга є сукупністю всіх інших продуктів і послуг, які він використовує, експлойт може початися в будь-якій точці ланцюжка поставок продукту та поширюватися звідти. Зловмисники знають, що їм потрібно лише скомпрометувати найслабшу ланку в ланцюжку постачання, щоб розповсюдити зловмисне програмне забезпечення, часто використовуючи ті самі засоби, які розробники використовують для масштабування свого програмного забезпечення. Якщо будь-яка з використаних бібліотек, має вразливі місця, кіберзлочинці можуть використати ці вразливості та отримати доступ до незахищених ресурсів хмарної інфраструктури.

Потрібно уважно ставитись до всіх сторонніх бібліотек і перевірити, чи є в них уразливості чи ні. Усі сторонні постачальники призначають найкращі практики та конфігурації безпеки, і важливо дотримуватися їх, щоб максимально підвищити безпеку.

6. Вразливості системи

Починаючи з 2017 року спостерігається постійне зростання бази даних відомих вразливостей на основі національної бази даних вразливостей [9]. З появою мультитенантності у хмарних обчисленнях системи різних організацій розміщуються в безпосередній близькості один до одного, мають доступ до спільної пам'яті та ресурсів, створюючи нову поверхню для атак. Вразливості в компонентах операційної системи, таких як ядро, системні бібліотеки та інструменти застосунків, ставлять під загрозу безпеку всіх послуг та даних.

У звіті IBM Security X-Force Cloud Threat Landscape Report за 2022 рік виявлено, що вразливості хмар зросли на колосальні 28% порівняно з минулим роком, а кількість хмарних облікових записів, пропонованих у темній мережі, зросла на 200% за той самий проміжок часу [12].

Регулярне сканування на наявність вразливостей, відповідь на звіти про загрози для системи та регулярне встановлення патчів, оновлень безпеки є важливими кроками. Крім того, безпечний дизайн, архітектура й сегментація мережі зменшують шанси на повний контроль зловмисника над кожною частиною інформаційної системи.

7. Викрадення облікових записів

Викрадення облікового запису - це загроза, при якій зловмисники отримують доступ та зловживають обліковими записами з високими привілеями. У хмарних середовищах найбільш вразливими є облікові записи хмарних служб або підписки. Фішингові атаки, експлуатація вразливостей хмарних систем або крадіжка облікових даних можуть піддати ці облікові записи компрометації. Ці загрози - унікальні та потенційно потужні - можуть спричинити втрати даних, активів та компрометацію операцій. У такому сценарії процеси бізнесу, активи, дані та застосунки, що залежать від послуг облікового запису, перебувають під загрозою.

Організації повинні просувати усвідомлення цих загроз та стратегії захисту (особливо IAM) на різних рівнях, щоб мінімізувати збитки від порушення.

8. Організована злочинність, розвинені сталі загрози (APT)

Використання соціальної інженерії, прями вторгнення в системи, поширення коду атаки через зовнішні-пристрої, проникнення через партнерські мережі та використання незахищених або мереж сторонніх постачальників є типовими точками входу для APT. Після встановлення APT можуть переміщатися вздовж мереж дата-центрів та злитися з нормальним мережевим трафіком для досягнення своїх цілей.

Протистояти організованим APT допоможе впровадження більш розвинених засобів безпеки, управління процесами, планів реагування на інциденти та підготовки персоналу IT, що може призвести до збільшення бюджетів на безпеку. Ці витрати слід зважити порівняно з економічними збитками, завданими успішними атаками APT.

9. Ексфільтрація даних з хмарного середовища

Витік даних - це інцидент, при якому інформація стає доступною для несанкціонованих осіб. Витік даних може бути основним об'єктом цілеспрямованої атаки або лише наслідком людської помилки, вразливостей програмного забезпечення або недостатніх практик забезпечення безпеки. Негативні наслідки витоку даних можуть включати фінансові та репутаційні втрати, а також стати причиною притягнення до правової та контрактної відповідальності. При цьому із загрозою витоку даних впродовж 2023 року стикнулися 51% організацій [12].

Техніки шифрування можуть допомогти захистити дані, але негативно впливають на продуктивність системи, роблячи програми менш зручними для користувачів. Необхідно створити надійний план реагування на інциденти, який враховує провайдера хмарних послуг та допоможе у відновленні.

10. Внутрішні загрози

Згідно з результатами досліджень, 58% компаній вважають, що порушення безпеки відбуваються через внутрішні загрози [11]. Умисні дії або недбалість осіб, які мають авторизований доступ до активів організації, є причиною більшості

інцидентів безпеки. На відміну від зовнішніх загроз, внутрішні особи не мають проникати через брандмауери, віртуальні приватні мережі та інші оборонні засоби периметра. Вони діють всередині круга довіри безпеки компанії, де вони мають прямий доступ до мереж, комп'ютерних систем та даних компанії.

Наслідки внутрішніх загроз можуть включати втрату даних, збої систем та інші інциденти. Вирішення інцидентів безпеки, пов'язаних з внутрішніми загрозами, включає обмеження, усунення наслідків, реагування на інцидент, розслідування, аналіз після інциденту, ескалацію, моніторинг та спостереження.

Критично важливим є навчання з безпеки працівників, щоб повідомити їх, як вони повинні поводитися у випадку загроз безпеки та виправлення неправильно налаштованих хмарних серверів:

Необхідно проводити регулярний аудит серверів у хмарі, та виправляти будь-які відхилення від безпечного базового рівня. Також потрібно забезпечити обмеження доступу до критичних систем: системи безпеки привілейованого доступу та центральні сервери мають бути обмежені для мінімальної кількості працівників і що ці особи включають тільки тих, хто має навчання з адміністрування критичних для виконання завдань комп'ютерних серверів. Організований моніторинг доступу до всіх активів на будь-якому рівні привілеїв є обов'язковим.

Варто зазначити, що ландшафт перерахованих загроз є динамічним. Наразі спостерігається тенденція того, що споживачі хмарних технологій відмовляються від занепокоєння про кінцеві результати (порушення або втрата даних), а шукають причини цих результатів (доступ до даних, неправильні конфігурації, незахищені програми) і беруть їх під контроль.

Користувачі хмари повинні підходити до принципів хмарної безпеки так само, як і до традиційної локальної безпеки, з адаптаціями для відображення відмінностей у використанні хмари та оновленого ландшафту загроз. Необхідно інвестувати в інструменти для зміцнення хмарного захисту, включаючи брандмауери, хмарні платформи безпеки, які інтегрують захист у багатьох середовищах, і керовані служби виявлення та реагування.

1.3 Нормативно-правове поле управління інцидентами у хмарному середовищі

Використання хмарних технологій супроводжується складною нормативно-правовою системою, яка вимагає ґрунтовного знання як державних, так і міжнародних практик і нормативів. У цьому розділі буде розглянуто нормативно-правові стандарти та регуляторні акти, що стосуються використання хмарних середовищ та розслідування інцидентів у хмарі.

Серед основних організацій, які регулюють сферу використання хмарних технологій та працюють над наданням актуальних стандартів, що відповідають найкращим міжнародним практикам, найбільш відомими є наступні:

1. Європейське агентство з кібербезпеки (ENISA): це агентство відповідає за підвищення рівня кібербезпеки в Європейському Союзі. ENISA випускає рекомендації, керівні вказівки та найкращі практики, що стосуються кібербезпеки та регулювання хмарних сервісів, а також допомагає координувати заходи з реагування на кіберінциденти.

4. CSA, про якого вже згадувалось раніше, — це галузева асоціація, яка надає дослідження, інструменти та найкращі практики для вирішення проблем безпеки у хмарних сервісах. Її рекомендації допомагають як постачальникам, так і споживачам хмарних сервісів управляти проблемами безпеки та розслідувати інциденти. Наприклад, допоміжним документом є Cloud Controls Matrix [13], що описує розробку системи контролю безпеки хмарних сервісів та Cloud Incident Response Framework, де висвітлено рекомендації до управління інцидентами [14].

2. Широко відомий NIST: цей інститут, розробляє комплексні рамкові інструкції щодо хмарних обчислень, охоплюючи питання безпеки, приватності та інтеперабельності. Їхні рекомендації допомагають організаціям ефективно управляти ризиками, пов'язаними з хмарними технологіями.

3. Міжнародна організація зі стандартизації (ISO), яка є фундаментальною для побудови системи безпеки, ISO було розроблено серію стандартів для забезпечення безпеки в хмарних сервісах, а також для розслідування інцидентів у

цифровому середовищі. Ці стандарти є глобальним орієнтиром для найкращих практик, пов'язаних з хмарною безпекою, адже описують додаткові засоби контролю безпеки та містять рекомендації з планування та впровадження процесів реагування на інциденти у хмарі, включаючи ролі та обов'язки обох сторін.

При користуванні хмарними технологіями дуже важливо зважати на вимоги Загального регламенту захисту даних (GDPR): він визначає суворі стандарти обробки та захисту персональних даних, незалежно від того, де саме ці дані обробляються або зберігаються [15].

1.3.1 Аналіз стандартів управління інцидентами від ISO

До вимог нової версії найбільш відомого стандарту безпеки ISO/IEC 27001:2022 [16], де розглядаються загальні вимоги до системи управління інформаційною безпекою, було включено домен контролів хмарної безпеки, що підкреслює зростання популярності використання хмарних технологій та важливість контролю безпеки таких систем.

Серед інших нормативних документів при проведенні розслідування інцидентів у хмарних сервісах варто розглянути наступні стандарти [17]:

1. ISO/IEC 27017 - Контроль безпеки для хмарних сервісів

Цей стандарт є доповненням до загального стандарту інформаційної безпеки ISO/IEC 27001 і надає конкретні рекомендації з управління ризиками у хмарному середовищі. Він визначає спеціальні заходи безпеки як для постачальників хмарних сервісів, так і для їх клієнтів. Особливий акцент ставиться на питаннях управління віртуалізацією, розподілу обов'язків між клієнтом та постачальником, конфіденційності, а також управлінні інцидентами в хмарі [18].

2. ISO/IEC 27018 - Захист персональних даних в хмарі

Спеціалізований стандарт, спрямований на захист персональних даних, які обробляються в хмарному середовищі. Він встановлює контроль та заходи, які мають застосовуватися постачальниками хмарних сервісів для захисту особистих даних відповідно до глобальних принципів конфіденційності. Стандарт допомагає

організаціям запровадити ефективну політику конфіденційності, що включає прозорість, права суб'єктів даних та обмеження збирання інформації [19].

3. ISO/IEC 27035 - Управління інцидентами інформаційної безпеки

Цей стандарт складається з трьох частин і надає керівні принципи для виявлення, реагування та аналізу інцидентів безпеки. Перша частина описує концепції та принципи управління інцидентами. Друга частина надає вказівки з планування та впровадження процедур реагування на інциденти. Третя - деталізує рекомендації щодо збору цифрових доказів та розслідування інцидентів [20].

Серед базових концепцій стандарту визначається поняття інциденту інформаційної безпеки - однієї або кількох пов'язаних подій інформаційної безпеки, спричинені технічними або нетехнічними чинниками, які можуть бути навмисними й випадковими і завдати шкоди активам організації або порушити її роботу.

На рисунку 1.5 показано взаємозв'язок об'єктів в інциденті інформаційної безпеки - загроза використовує вразливості в інформаційних системах, службах або мережах, спричиняючи виникнення подій інформаційної безпеки та потенційних інцидентів з інформаційними активами [20].

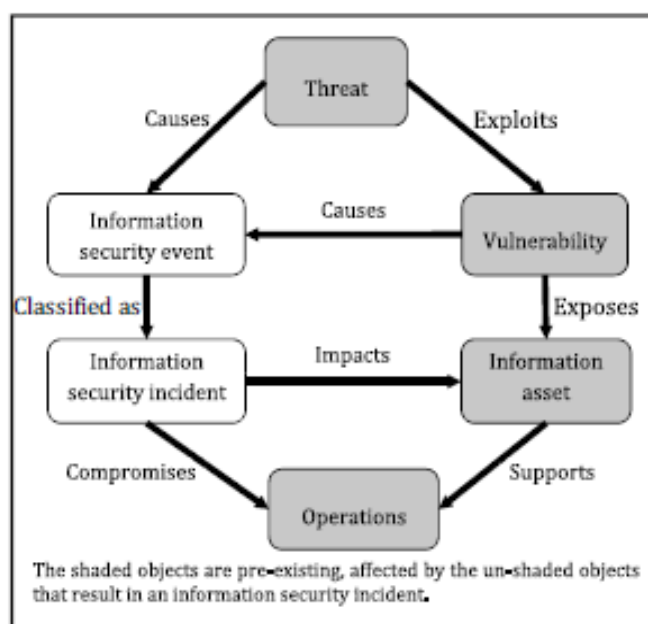


Рисунок 1.5 - Взаємозв'язок об'єктів в інциденті інформаційної безпеки

Відповідно, процес розслідування інцидентів покликаний відтворити ланцюжок подій, що спричинили виникнення інциденту, окреслити уражені активи й знайти вразливості в системі для подальшого їх усунення.

У стандарті ISO/IEC 27035 описано зв'язок між різними стандартами, що стосуються розслідування інцидентів на різних етапах, який схематично відображено на рисунку 1.6 [20].

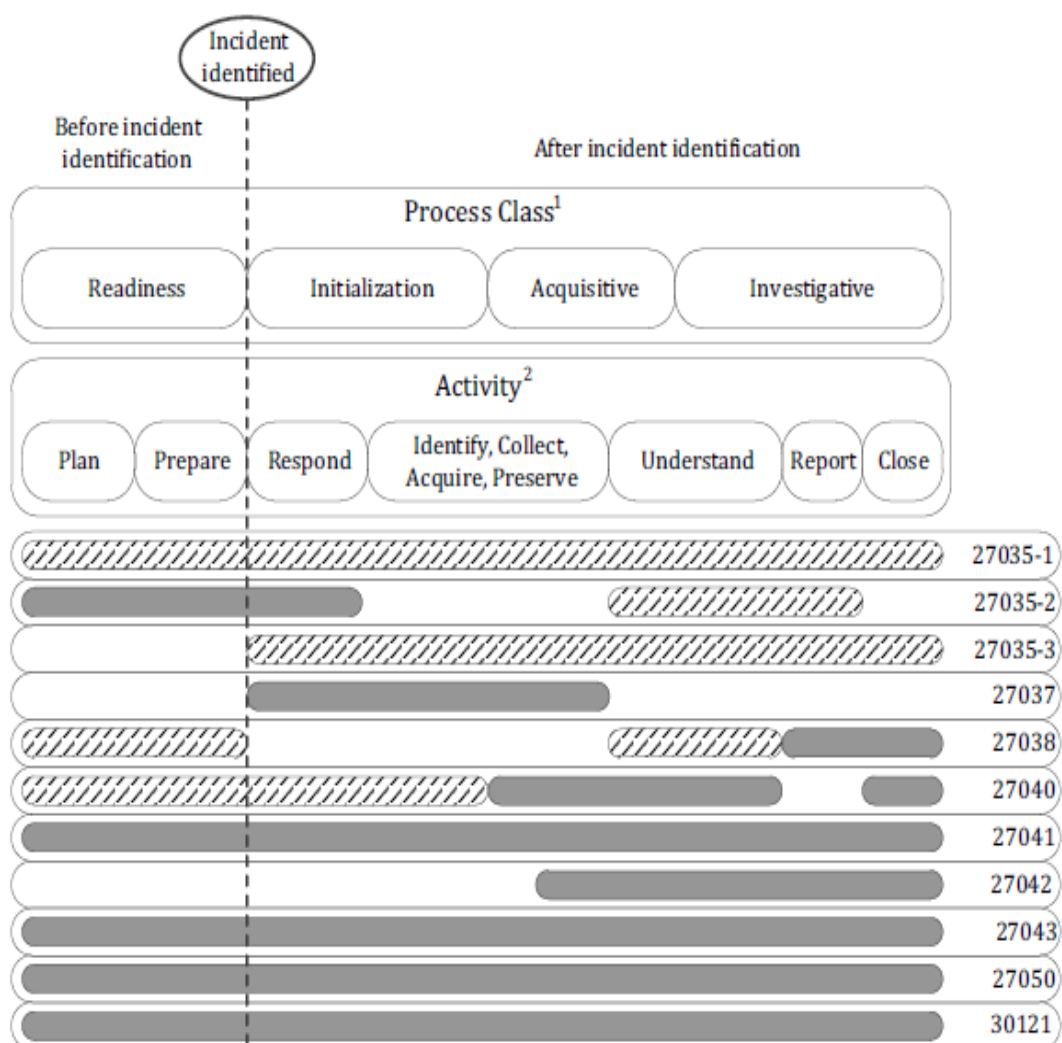


Рисунок 1.6 - Застосовність стандартів до процесу розслідування інцидентів

Наслідуючи особливості стандарту ISO/IEC 27001, він використовує класичну циклічну модель вдосконалення процесів відому під назвою PDCA - Plan,

Do, Check, Act, - як основу для функціонування інфраструктури управління інцидентами.

Модель життєвого циклу цього процесу зображено на рисунку 1.7.



Рисунок 1.7 - Модель життєвого циклу процесу управління інцидентами

Далі розглянуто основні стандарти, їхній взаємозв'язок і роль у загальному процесі:

- ISO/IEC 27037 - Ідентифікація, збір, отримання та збереження цифрових доказів. Цей стандарт спрямований на забезпечення достовірності цифрових доказів, їх ефективного збору та збереження на початкових етапах розслідування. Він надає практичні поради щодо виявлення, збереження та аналізу доказів у рамках розслідування інцидентів кібербезпеки [21].

- ISO/IEC 27041 - Підтвердження відповідності методів розслідування інцидентів: Пропонує методи для забезпечення відповідності процесів і методів вимогам розслідування та тестування [22].

- ISO/IEC 27042 - Аналіз і інтерпретація цифрових доказів: Дає інструкції щодо аналізу та інтерпретації цифрових доказів, що дозволяє їх правильну оцінку та подальше ефективне звітування [23].

- ISO/IEC 27043 - Принципи та процеси розслідування інцидентів: Встановлює загальні принципи та процеси для розслідування інцидентів, що

включає підготовку до інциденту, реагування та аналіз. Цей стандарт підтримує концепцію повторюваності та відтворюваності процесу для забезпечення об'єктивності розслідування. Містить модель процесу розслідування інцидентів, яка включає готовність, ініціалізацію, збір даних та їх аналіз [24].

Всі ці стандарти взаємопов'язані та охоплюють різні аспекти процесу розслідування інцидентів, від збору доказів до їх аналізу та управління ризиками. Разом вони забезпечують всебічний підхід до управління інцидентами та цифрового розслідування. Впровадження рекомендацій ISO значно підвищує надійність та довіру до хмарних сервісів, забезпечуючи їх безпечне використання.

1.3.2 Аналіз стандартів управління інцидентами від NIST

Декілька ключових документів від NIST, які розглядають специфічні питання безпеки хмарних технологій та управління інцидентами включають:

1. NIST Special Publication (SP) 800-53 – Захисні заходи для систем інформації та організацій - це комплексний документ, що визначає контрольні заходи безпеки для інформаційних систем, включаючи хмарні. Він надає рекомендації щодо побудови системи управління ризиками, захисту даних та контролю доступу. Хмарні провайдери та користувачі можуть використовувати цей документ як основну інструкцію для впровадження ефективних заходів безпеки [25].

2. NIST SP 800-144 – Посібник з безпеки та конфіденційності для хмарних обчислень цей документ є посібником, що пояснює основні ризики безпеки та конфіденційності при роботі з хмарними сервісами. Він допомагає організаціям оцінювати потенційні загрози та розробляти заходи захисту при роботі в публічних, приватних чи гібридних хмарах. Особлива увага приділяється управлінню ризиками конфіденційності, безпечному використанню інтерфейсів API та обробці даних [26].

3. NIST SP 500-291 – Хмарна дорожня карта - пропонує огляд нормативів і стандартів для хмарних обчислень та визначає основні напрями розвитку галузі.

Він аналізує сучасні тенденції у впровадженні хмарних технологій, стандартизації та інтероперабельності між провайдерами хмарних сервісів [27].

4. NIST SP 800-61 – Керівництво з обробки інцидентів комп’ютерної безпеки -це один із основоположних стандартів NIST для управління інцидентами кібербезпеки, включаючи хмарні середовища. Документ пропонує структуровані процеси планування, виявлення, реагування та аналізу інцидентів. Окрему увагу приділено питанням комунікації та співпраці між командами під час розслідування інцидентів, а також документації та урокам, отриманим після інциденту [28].

5. NIST SP 800-86 - Керівництво з інтеграції криміналістики до процесів реагування на інциденти - містить методи інтеграції процесів цифрової криміналістики до управління інцидентами. Він розглядає всі етапи від виявлення до аналізу інцидентів. Цей посібник надає загальні рекомендації щодо проведення криміналістичної процедури. Крім того, він пропонує всебічні вказівки щодо застосування аналітичного методу до чотирьох основних типів джерел даних — додатків, операційних систем, файлів і мережевого трафіку. Окрім методів збору, вивчення та аналізу даних кожної категорії, стандарт зосереджується на поясненні основних елементів та характеристик джерел даних у кожній категорії. Також посібник містить рекомендації щодо поєднання різних джерел даних для покращення розуміння події [29].

Ці стандарти та керівництва забезпечують хмарних провайдерів і клієнтів корисними порадами, допомагають створити комплексний план дій для проведення належного розслідування інцидентів та побудови надійної системи безпеки в хмарному середовищі.

1.3.3 Нормативно-правове забезпечення в Україні

Законодавство України також поступово адаптується до сучасних реалій, у тому числі до використання хмарних технологій. Деякі ключові аспекти, які необхідно враховувати при формуванні систем управління інцидентами у хмарі, викладені в наступних нормативно-правових документах:

1. Закон України "Про захист персональних даних" встановлює вимоги щодо обробки персональних даних, їхнього захисту, а також відповідальність за порушення [30].

2. Закон України "Про інформацію" регулює відносини щодо отримання, використання, поширення та зберігання інформації, що може бути важливим у контексті хмарних сервісів [31].

3. Закон України "Про основні засади забезпечення кібербезпеки України" визначає основи державної політики у сфері кібербезпеки та передбачає заходи щодо захисту критичної інформаційної інфраструктури, в тому числі в хмарних сервісах [32].

4. Закон України "Про хмарні послуги" регулює правові відносини, пов'язані з наданням хмарних послуг в Україні, і встановлює правила для їх використання державними органами, організаціями та іншими публічними користувачами [4].

Основні положення та принципи закону надають визначення ключових понять та типів хмарних послуг, а також передбачають вимогу до створення належних умов для захисту інформації від внутрішніх та зовнішніх загроз. Це дає правові підстави для ретельного розслідування інцидентів, щоб виявити, чи були дотримані заходи безпеки, і перевірити рівень захисту інфраструктури. Цей закон забезпечує комплексну нормативно-правову базу для використання хмарних послуг та допомагає регулювати ключові аспекти їх надання в Україні.

1.4 Процес управління інцидентами

Процес управління інцидентами в хмарному середовищі включає кілька ключових етапів, які спрямовані на швидке виявлення, аналіз, реагування та відновлення після інцидентів безпеки.

Етапи цього процесу схематично представлені на рисунку 1.8 [33].

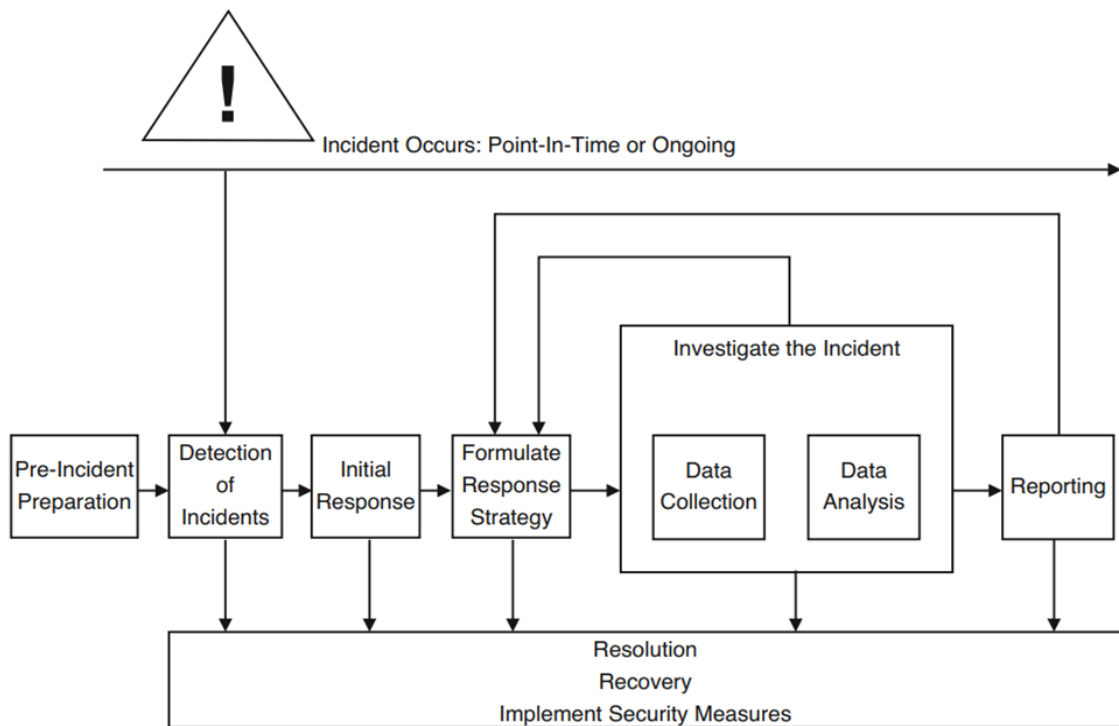


Рисунок 1.8 – Процес управління інцидентами інформаційної безпеки

Розглянемо кожен з етапів детальніше:

1. Підготовка: включає розробку плану реагування на інциденти, визначення ролей та відповідальності учасників процесу, навчання команди з реагування на інциденти та забезпечення наявності необхідних інструментів для аналізу та розслідування.

2. Виявлення інциденту: передбачає виявлення інцидентів через моніторингові системи, повідомлення від користувачів або зовнішніх джерел. Кожен інцидент документується з основними деталями, включаючи час, тип та джерело.

3. Класифікація та пріоритезація: інциденти оцінюються та класифікуються за ступенем серйозності та впливу на організацію. Визначається пріоритет дій, спрямованих на їх розв'язання.

3. Розслідування інциденту: цей етап включає в себе використання інструментів цифрової форензики для збору необхідних даних про інцидент та їх аналіз для встановлення причини інциденту (RCA).

4. Відповідь: на основі визначених причин відбувається формування відповіді на інцидент та виконуються заходи для їх усунення. Відповідь на інцидент може включати також відновлення уражених систем, даних та служб до нормального стану роботи.

5. Звітування: після вирішення інциденту перевіряється, що всі проблеми усунені. Інцидент формально закривається, а всі пов'язані з ним дані документуються у звітах, які використовуються для аналізу тенденцій, покращення процесів та оновлення планів управління інцидентами під час подальшого ретроспективного аналізу, який визначає, чи можна покращити процедури, та розробляє рекомендації щодо попередження подібних інцидентів у майбутньому.

Такий підхід допомагає забезпечити оперативне та ефективне реагування на інциденти з мінімізацією їх впливу на бізнес-процеси.

У цій роботі буде зосереджено увагу на фундаментальному процесі – розслідування інцидентів. Під час розслідування інциденту збираються різні типи доказів, що стосуються інциденту, такі як докази, що базуються на хості або мережі, щоб відновити події, які становлять інцидент з комп'ютерної безпеки. Ця реконструкція повинна надати пояснення того, що сталося, коли, як, чому це сталося і хто є винуватець.

Для досягнення цього, розслідування зазвичай поділяється на два етапи: збір даних та аналіз даних (див. рис. 1.8).

Прикладами даних, зібраних під час збору доказів, є інформація, отримана з хоста в реальному часі, копіювання жорсткого диска компрометованого хоста або журнали мережевого спостереження.

Під час збору цієї інформації рекомендується використовувати форензично обґрунтовані методи. На етапі аналізу даних раніше зібрана інформація переглядається для розкриття деталей інциденту.

Слід зауважити, що часто збір та аналіз даних можуть чергуватися, фактично перетворюючи розслідування на серію зворотних зв'язків, поки не буде отриманий остаточний результат.

Висновки за розділом 1

В цьому розділі було висвітлено теоретичні основи функціонування хмарних середовищ, проаналізовано різновиди хмарних обчислень та їх характерні особливості, включаючи моделі розгортання та моделі сервісів. Ці аспекти обов'язково необхідно враховувати при формуванні системи управління інцидентами, адже хмарні сервіси передбачають розподілену відповідальність між користувачем та надавачем хмарним послуг.

Було підкреслено основні переваги використання хмарних технологій, такі як гнучкість, надійність та зменшення витрат, що зумовили їх високу популярність. Разом з тим, було визначено високий рівень загроз безпеці у хмарному середовищі, найбільш поширеними серед яких є ненадійні API, неправильна конфігурація, внутрішні загрози та організовані атаки.

Оскільки забезпечення кібербезпеки в хмарних середовищах стає можливим лише при дотриманні регуляторних вимог та стандартів, у цьому розділі було розглянуто нормативно-правові документи, що описують систему управління інцидентами від міжнародних організацій, зокрема ISO та NIST. Проаналізовані стандарти регулюють різні аспекти розслідування інцидентів, від збору цифрових доказів до аналізу та управління інцидентами. Також було розглянуто законодавчу базу України, яка регулює обробку даних у хмарі та надає вимоги до засоби захисту хмарної інфраструктури. Загалом, зроблений аналіз показав те, що цій галузі бракує спеціалізованих нормативних документів, тому що наявні стандарти мають дещо узагальнений характер.

Загалом, розділ надає комплексне розуміння різноманітних аспектів управління інцидентами у хмарному середовищі та акцентує на важливості дотримання стандартів і правових норм для побудови ефективної системи безпеки.

РОЗДІЛ 2

ОСОБЛИВОСТІ РОЗСЛІДУВАННЯ ІНЦИДЕНТІВ У ХМАРНОМУ СЕРЕДОВИЩІ

2.1 Виклики хмарної форензики

Перед тим як сформувати модель для розслідування інцидентів у хмарі необхідно розуміти особливості цього процесу, зважаючи на виклики, що притаманні для хмарного середовища. Для цього в подальшому буде використано поняття хмарна форензика, що описується NIST як спеціалізована область цифрової форензики, зосереджена на дослідженні та аналізі хмарних обчислювальних систем та сервісів з метою виявлення, збору, аналізу та документування доказів для судових або розслідувальних цілей [34]. Ця спеціалізація виникає завдяки характерним особливостями хмарних обчислень, таким як об'єднання ресурсів на вимогу, широкий доступ до мережі, швидка еластичність і вимірюване обслуговування, які ускладнюють традиційні процеси цифрової експертизи. Основні виклики хмарної форензики включають складнощі із локалізацією даних у розподіленому середовищі, їхньою конфіденційністю, а також доступністю та цілісністю даних.

Розглядаючи різні способи використання хмарної форензики, можна класифікувати їх у п'ять основних категорій, кожна з яких адресує специфічні аспекти та виклики, що виникають у хмарних середовищах:

1. Розслідування: ця категорія охоплює використання хмарної форензики для виявлення та аналізу кіберзлочинів, що відбуваються в хмарних середовищах.

Вона включає декілька елементів:

- власне процес розслідування хмарних злочинів, який зосереджується на ідентифікації та розслідуванні злочинних дій, що відбуваються безпосередньо в хмарі;

- реагування на інциденти, включає аналіз підозрілих транзакцій або діяльності в хмарі та відновлення після безпекових інцидентів;
- реконструкція подій, тобто відтворення хронології подій для з'ясування причин та наслідків інцидентів та;
- координація з юридичними та правоохоронними органами для забезпечення дотримання законодавства, проведення слідчих дій і надання доказів суду.

2. Усунення несправностей: цей аспект хмарної форензики зосереджується на технічному аналізі та усуненні помилок системи, що включає:

- діагностику систем - ідентифікацію та усунення вразливостей в хмарних програмах та сервісах;
- моніторинг і аналіз стану систем у реальному часі;
- визначення першопричин інцидентів, що охоплюють декілька подій протягом певного часу для запобігання їх повторенню.

3. Моніторинг журналів логування також важлива частина хмарної форензики, яка допомагає в аудиті та забезпеченні відповідності до регуляторних вимог, що включає збір, аналіз та кореляцію логів діяльності з різних систем і сервісів у хмарі.

4. Відновлення втрачених або пошкоджених даних та зміна конфігурацій системи після того як стався інцидент.

5. Дотримання нормативних вимог, охоплює діяльності, спрямовані на забезпечення відповідності організацій нормативним і законодавчим вимогам, які регулюють захист даних, включно з веденням необхідних записів, захистом конфіденційних даних і сповіщенням про інциденти витоку інформації.

Усі ці категорії показують широку сферу застосування хмарної форензики та підкреслюють її важливість у забезпеченні безпеки, стійкості та відповідності хмарних обчислень.

Щоб більш комплексно проаналізувати виклики хмарної форензики та підкреслити той факт, що розслідування інцидентів у хмарі є багатовимірною проблемою, розширимо визначення хмарної криміналістики на три основні виміри,

згідно з напрямками, що були визначені ENISA: технічний, організаційно-правовий і юридичний, як зображено на рисунку 2.1 [35].

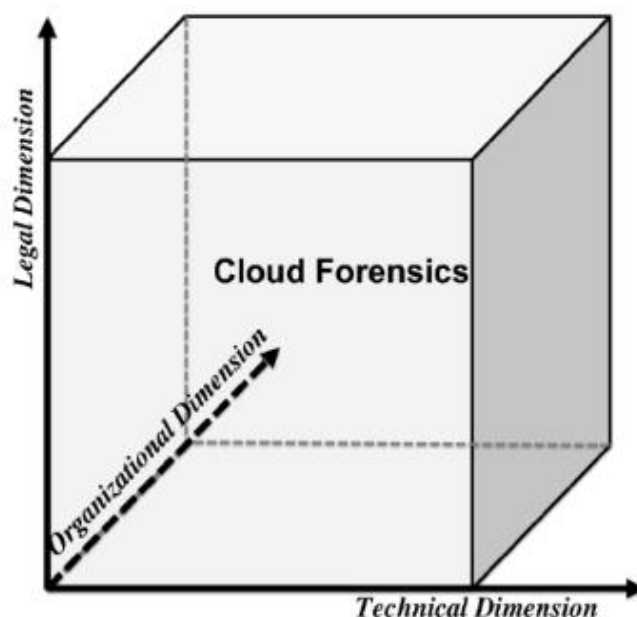


Рисунок 2.1 - Тривимірна модель хмарної форензики

Технічний вимір відноситься до специфічних програмно-апаратних особливостей моделі хмарних обчислень, які необхідно враховувати при розслідуванні інцидентів у середовищі хмарних обчислень. Цей вимір форензики передбачає розробку інструментів і методологій, які можуть адаптуватися до динамічної та розподіленої природи хмарних обчислень.

Організаційний вимір пов'язаний з координацією сторін, залучених до процесу хмарної форензики. Він фокусується на внутрішніх структурах і співпраці, необхідних для проведення розслідування в хмарних налаштуваннях. Це включає в себе ролі та обов'язки постачальників хмарних послуг і клієнтів в управлінні розслідуваннями та інтеграції обробників інцидентів, юридичних консультантів і ІТ-фахівців для підтримки процесу.

Юридичний вимір стосується правових аспектів і проблем між сторонами, залученими до процесу, адже дані та сервери, розповсюджені на різних юридичних територіях, підпадають під різні законодавчі рамки. Також виникають правові

питання отримання цифрових доказів у середовищі хмарних обчислень, охоплюючи вимогу до забезпечення конфіденційності даних.

2.1.1 Різновимірні виклики хмарної форензики

Відповідно до розглянутої тривимірної моделі, технічні виклики при розслідуванні інцидентів у хмарі включають [35, 36, 37]:

- Багатокористувацькість: швидка еластичність хмарних ресурсів збільшує складність отримання даних.

- Динамічна природа та перевикористання ресурсів, що ускладнює точне відділення ресурсів, які перебувають під розслідуванням, та збір доказів, особливо якщо ці дані не мають постійної синхронізації зі збереженням.

- Складне відновлення видалених даних у спільному та розподіленому віртуальному середовищі.

- Структура хмарних середовищ включає багато шарів, що вносить додаткові виклики у встановлення довіри.

- Необхідність синхронізації часу у розподілених даних теж створює складнощі при реконструкції подій.

- Формат даних логів не є стандартизованим, тому розподіл логів між різними шарами хмари та їхні різні формати ускладнюють їх доступність для збору та аналізу.

- Брак спеціалізованих інструментів для збору даних у хмарному середовищі.

- Шифрування даних може створювати технічні виклики, оскільки дешифрування великих обсягів даних вимагає багато часу та ресурсів.

Не можна оминати увагою організаційні виклики в хмарному середовищі, адже вони також можуть серйозно впливати на процес управління інцидентами. Оскільки розслідування зазвичай включає принаймні дві сторони: надавача хмарних послуг (CSP) та клієнта, виникає сильна залежність від CSP при зборі даних.

Так, аналітики можуть самостійно збирати докази на віддаленій основі, але часто потрібно отримати докази з різних фізичних локацій у місцях розташування провайдерів. Важливість ролі CSP посилюється, оскільки дані зберігаються на різних серверах по всьому світу, а інформація про машини, облікові дані, логи та ключі шифрування керується з боку провайдера або навіть може бути розміщена у третіх сторін. Без співпраці з провайдером в такому випадку отримати інформацію та провести розслідування майже неможливо.

При цьому потрібно бути готовими, що одна зі сторін може відмовитись співпрацювати у цьому напрямку. І навіть якщо це було попередньо передбачено у SLA, CSP та аналітики зазвичай не мають спільних стандартів і слідуєть різним процедурам для збору даних, що ускладнює планування та організацію діяльності.

Звичайно, організаційні аспекти можуть значно відрізнятись залежно від моделі розгортання хмарних сервісів, адже клієнт має різний рівень контролю над хмарним середовищем та власними даними, що там зберігаються. Наприклад, у моделі SaaS клієнт має мінімальний контроль, що вимагає повної співпраці CSP для доступу до потрібної інформації.

Ці виклики вимагають ефективного управління та координації між усіма зацікавленими сторонами для забезпечення успішного проведення розслідувань у хмарному середовищі.

Юридичні виклики при хмарному розслідуванні включають декілька важливих аспектів, які експерти виділяють як особливо значущі для розслідувань:

- Багатюрисдикційні проблеми: хмарні сервери часто розміщені в різних юрисдикціях, що ускладнює доступ до даних через необхідність дотримання місцевого законодавства та міжнародної співпраці між правоохоронними органами.

- Не існує специфічних регуляцій, які б визначали обов'язки CSP у контексті стандартних операційних процедур, що ускладнює процес збору доказів.

- Відсутність угод між провайдерами хмар, правоохоронними органами та клієнтами: Це ускладнює співпрацю при розслідуваннях.

- Правовий доступ до даних у міжнародних розслідуваннях: Забезпечення законного доступу до даних у міжнародному контексті залишається складним через різницю в національному законодавстві.

- Положення SLA щодо форензики: більшість SLA не включають положення, що стосуються форензичної діяльності, що ускладнює організацію та прозорість розслідувань. Це вимагає від CSP визначити умови доступу до даних, процедури форензики, ролі та відповідальності у SLA для роз'яснення практичних аспектів розслідувань.

- У хмарному середовищі складніше забезпечити доказову базу через розподілену, багатокористувацьку та еластичну природу хмари, що ускладнює демонстрацію неперервності ланцюга зберігання доказів.

Ці виклики вимагають вдосконалення міжнародних кіберполітик та законів, а також покращення процедур і стандартів, щоб допомогти у вирішенні вказаних проблем.

У контексті хмарної форензики також визначено кілька проблем, які відносяться до різних вимірів і можуть серйозно вплинути на ефективність розслідувань [38].

Однією з найбільш важливих проблем є час від моменту виникнення інциденту до початку розслідувань та збору доказів. Затримка в доступі до машин або даних може негативно вплинути на весь процес, оскільки у динамічному розподіленому середовищі зміни можуть відбуватися в будь-який час, що може істотно вплинути на результати розслідування.

Друга велика проблема пов'язана з витратами та зусиллями, необхідними для проведення розслідувань, які включають CSP, аналітиків безпеки, клієнта та треті сторони. Зазвичай вартість розслідувань покривається клієнтом, якщо форензичні послуги не були включені в контракт SLA заздалегідь.

Ці проблеми підкреслюють потребу в більш тісній співпраці між усіма зацікавленими сторонами для забезпечення ефективності та оперативності розслідувань у хмарних середовищах, а також зумовлюють важливість використання спеціалізованих інструментів для збору доказів та аналізу при

розслідуванні інцидентів у хмарі, які допоможуть зробити цей процес максимально ефективним .

2.1.2 Процес хмарної форензики

Сам процес хмарної форензики перетинається з розглянутим раніше процесом управління інцидентами (див. п. 1.4), пропонуючи його адаптацію, що враховує специфіку хмарних платформ. Це складний і важливий аспект розслідування інцидентів, що можна узагальнити у вигляді чотирьох основних етапів, які відображено на рисунку 2.2 [39].

Як бачимо на рисунку 2.2, хмарна форензика включає ідентифікацію та розташування відповідних хмарних систем та додатків, огляд провайдера послуг, видів послуг та типів даних. На цьому етапі потрібно використовувати спеціалізовані методи для ефективного використання існуючих інструментів та ізоляції доказів у хмарі.

На етапі збору та збереження даних важливо забезпечити цілісність цифрових доказів для їх подальшого використання. Це включає систематичний збір даних, їх безпечно зберігання та документацію. В деяких випадках тут доцільне використання можливостей шифрування та звернення до сторонніх послуг засвідчувального органа, що забезпечить довіру до достовірності цифрових доказів.

Етап аналізу в хмарній форензиці включає використання інструментів та методологій для дослідження цифрових доказів, відтворення кроків атаки через лог-файли, відстеження мережевої активності, розшифрування метаданих та відновлення даних. Цей етап вимагає високої технічної компетентності та уважності аналітиків безпеки й може тривати довгий період часу.

Наступним є етап звітування. Мета хмарної форензики полягає у представленні результатів розслідування чітко та стисло, використовуючи зібрану інформацію як достовірні докази для керівництва або ж у юридичних процесах. Це

включає створення детальних звітів, використання наочних засобів у вигляді презентацій і надання експертних свідчень.

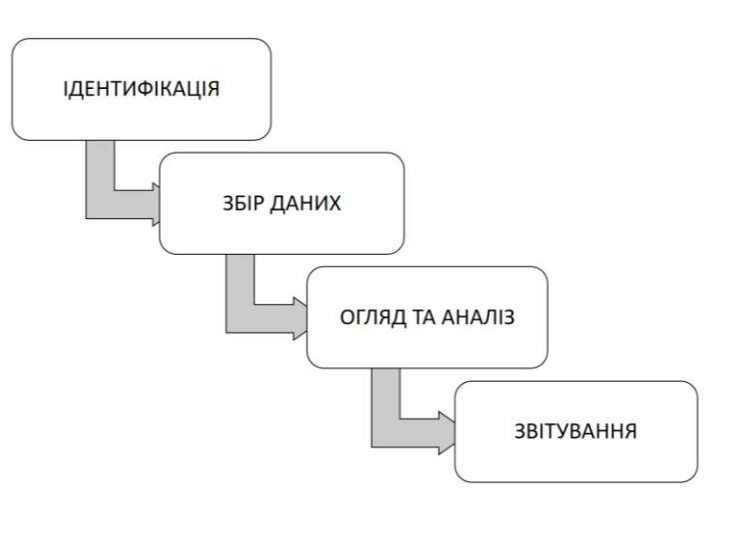


Рисунок 2.2 - Етапи процесу хмарної форензики

При цьому варто пам'ятати, що адаптація процедур хмарної форензики до різних моделей надання та розгортання послуг є ключовою для забезпечення повноти зібраних доказів та отримання достовірних результатів розслідування.

2.2 Розробка рекомендацій для процесу розслідування інцидентів у хмарі

Оскільки хмарна форензика істотно відрізняється від традиційних систем, вона потребує застосування спеціалізованих підходів до збору та аналізу цифрових доказів із середовищ хмарних обчислень.

В цій роботі було розроблено рекомендації для кожного етапу процесу хмарної форензики, враховуючи унікальні виклики, які постають у цих середовищах.

1. Ідентифікація

На етапі ідентифікації інцидентів отримання інформації з файлів журналу: у хмарних середовищах збір журналів ускладнене через розподіленість архітектури і різних власників даних. Клієнти мають доступ лише до API, що робить

повноцінний моніторинг усіх дотичних активів неможливим. В IaaS моделі, журнали важливі для розуміння поведінки віртуальних машин, але їхня ефективність обмежена через рестрикції, накладені провайдерами хмарних сервісів.

Існує проблема збереження перехідних даних під час вимкнення або перезапуску віртуальних машин. Це завдання складне, проте критично важливе для фахівців форензики для ідентифікації та збереження мінливих даних.

Локалізація даних у хмарі ускладнена через глобальне розміщення обладнання. Це створює складнощі для хмарної форензики, де доступ до фізичних пристроїв обмежений. Докази можуть знаходитися як на стороні постачальника, так і клієнта, особливо у контекстах SaaS та PaaS. Важливо забезпечити швидкий збір неушкоджених даних для аналізу, оскільки злочинець може їх знищити.

Дослідження підкреслюють важливість CSP у процесі форензики, але виникають виклики, коли вони неохоче надають інформацію, особливо в системах з множинністю орендарів. При цьому SLA можуть не включати деталі про форензичні розслідування, що може призвести до відсутності контрактних зобов'язань з боку CSP.

Щоб протистояти цим проблемам необхідно дотримуватись наступних рекомендацій:

- Сформувати партнерські відносини з CSP для забезпечення своєчасного доступу до журналів та інших відповідних даних.
- Використовувати інструментів, розроблених для взаємодії з API хмари, для автоматизації збору журналів та метаданих.
- Впроваджувати інструментів неперервного моніторингу, які можуть допомогти в реальному часі ідентифікувати потенційні інциденти безпеки.

Зокрема, інтеграція Security Information and Event Management (SIEM) систем, таких як Splunk або IBM QRadar, з хмарними сервісами дозволяє збирати і аналізувати логи та інші події безпеки в реальному часі. Cloud Access Security Brokers (CASBs) забезпечують видимість користувачів та діяльності у хмарі, допомагаючи ідентифікувати неавторизований доступ чи інші підозрілі дії.

Логи повинні отримуватись з різноманітних джерел, щоб аналітик міг побачити повну картину того, що відбувається у системі (див. табл. 2.1) [40].

Таблиця 2.1

Приклад тактик і технік зловмисника та відповідних джерел подій

Tactic	Common Techniques	Log and Event Sources	Indicators
<u>Initial Access</u>	Phishing [T1566], Drive-by Compromise [T1189], Exploit Public Facing Application [T1190], External Remote Services [T1133]	Email, web proxy, server application logs, IDS/IPS	Phishing, redirect, and payload servers (domains and IP addresses), delivery mechanisms (lures, macros, downloaders, droppers, etc.), compromised credentials, web shells
<u>Execution</u>	Command and Script Interpreters [T1059], Exploitation for Client Execution [T1203]	Host event logs, Windows event logs, Sysmon, anti-malware, EDR, PowerShell logs	Invocation of command or scripting interpreter, exploitation, API calls, tools, malware, payloads
<u>Persistence</u>	Account Manipulation [T1098], Scheduled Task/Job [T1053], Valid Accounts [T1078]	Host event logs, Authentication logs, Registry	Scheduled Tasks, registry keys, autoruns, etc.
<u>Lateral Movement</u>	Exploitation of Remote Services [T1210], Remote Session Hijacking [T1563], Software Deployment Tools [T1072]	Internal network logs, host event logs, Application Logs	Mismatch of users and applications/credentials, workstation to workstation communication, beaconing from hosts not intended to be internet accessible, etc.
<u>Credential Access</u>	Brute Force [T1110], Modify Authentication Process [T1556], Man-in-the-Middle [T1557]	Authentication Logs, Domain Controller Logs, network traffic monitoring	LSASS reads, command or scripting interpreters accessing LSASS, etc.
<u>C2</u>	Application Layer Protocol [T1071], Protocol Tunneling [T1572]	Firewall, Web Proxy, DNS, Network Traffic, Cloud activity logs, IDS/IPS	C2 domains, IP addresses
<u>Exfiltration</u>	Exfiltration Over C2 Channel [T1041], Exfiltration Over Alternative Protocol [T1048]	Firewall, Web Proxy, DNS, Network Traffic, Cloud activity logs, IDS/IPS	Domains, URLs, IP addresses, IDS/IPS signatures

2. Збереження та збір даних

Реконструкція місця злочину та відтворення повної послідовності подій може бути неможливим, якщо відповідальна віртуальна машина завершує роботу після шкідливої активності. Також користувачі можуть зіткнутись з обмеженням пропускної здатності через стрімке збільшення обсягів даних.

На збір даних впливають також юридичні та міжнародні проблеми, пов'язані з даними, що розміщені у кількох країнах. Виникає необхідність у співпраці між технічними дослідниками, юридичними консультантами та зовнішніми фахівцями

Надані рекомендації допоможуть пом'якшити ці виклики:

- Використання криптографічного хешування зібраних даних для перевірки їх цілісності на пізніших етапах.

- Розробка чітких протоколів збору даних, що відповідають міжнародним законам і регуляціям.

- Використання інструментів форензики, які можуть зафіксувати моментальні знімки віртуальних машин і даних, заморозити їх у часі і зберегти у правовому порядку.

3. Огляд та аналіз

Аналіз даних з архітектур хмар передбачає роботу з величезними наборами даних, розподіленими по різних сервісах, що ускладнює пошук значущих цифрових доказів. При цьому не можна забувати про проблему відсутності стандартизації формату журналів та всебічного оцінювання точності у інструментах форензики – їх результати можуть виявитись помилковими.

Рекомендації на етапі аналізу будуть наступними:

- Використання автоматизованих інструментів аналізу, які можуть обробляти великі обсяги даних і інтегрувати результати з різних джерел.

- Розвиток спеціалізованих навичок у сфері хмарних технологій серед аналітиків форензики для підвищення ефективності процесу огляду.

Для того щоб автоматизувати процес аналізу даних рекомендується впровадити штучний інтелект та машинне навчання. Це допоможе з ідентифікацією шаблонів і виявленню аномалій у даних.

4. Презентація

Складно презентувати вагому технічні деталі та дані хмарних розслідувань у формі, зрозумілій для нетехнічних зацікавлених сторін, таких як судова влада або вище керівництво. До того ж, необхідно забезпечити гарантії, що представлені докази, відповідають правовим стандартам допустимості.

В цьому можуть допомогти:

- Підготовка детальних звітів, які чітко описують дотриманий процес розслідування, дотримуючись підтвердженого ланцюжка доказів.
- Використання наочних посібників і спрощених пояснень для допомоги нетехнічним сторонам зрозуміти технічні деталі.
- Проведення внутрішніх презентацій у команді безпеки для вдосконалення подачі складної інформації.
- Корисним також можуть бути використання інструментів для моделювання сценаріїв атак щоб відтворити послідовності подій на основі зібраних даних.

В таблиці 2.2 узагальнено вимоги до розслідування інцидентів у хмарних середовищах, яка враховує кожен етап процесу хмарної форензики та надано рекомендовані інструменти й технології для задоволення поставлених вимог.

Ці інструменти та технології забезпечують проведення хмарних розслідувань, допомагаючи організаціям виявляти, реагувати, проводити розслідування та відновлюватися після інцидентів у хмарі.

Таблиця 2.2

Вимоги до системи розслідування інцидентів

Етап	Вимоги до системи	Рекомендовані інструменти
Ідентифікація	1. Автоматизований доступ до логів і метаданих через API. 2. Неперервний моніторинг.	SIEM (IBM Qradar, Splunk, ELK) CASB (Cisco Cloudlock, Forcepoint)
Збір та збереження	1. Здатність до відновлення даних. 2. Криптографічне хешування для підтвердження цілісності. 3. Регулярне захоплення моментальних знімків системи.	EnCase FROST FTK VNSnap Kumofs

Етап	Вимоги до системи	Рекомендовані інструменти
Огляд та аналіз	<ol style="list-style-type: none"> 1. Здатність до інтеграції результатів з різних джерел. 2. Здатність до обробки великих обсягів даних. 3. Застосування штучного інтелекту та машинного навчання для ідентифікації шаблонів і аномалій. 	SIEM (IBM Qradar, Splunk, ELK) * RCA BOT
Звітування	<ol style="list-style-type: none"> 1. Функціонал створення детальних звітів і презентацій. 2. Спрощення подачі складної інформації. 3. Здатність до моделювання атак. 	Looker Studio Splunk Power BI AttackTree

Розроблені вимоги до системи розслідування інцидентів допомагають окреслити загальну модель такої системи для управління інцидентами у хмарі, що було розроблено в цій роботі.

2.3 Аналіз інструментів хмарної форензики

У галузі хмарної форензики, використання спеціалізованих інструментів має вирішальне значення для дієвого проведення розслідувань у хмарних середовищах. Ці інструменти дозволяють збирати, аналізувати та зберігати дані з різноманітних хмарних сервісів, що є ключовим у розслідуваннях інцидентів кібербезпеки.

Далі надано огляд кількох відомих інструментів і технологій хмарної форензики з описом їхніх функцій і ролі в процесі виявлення цифрових доказів [40, 41].

1. Forensic Recovery of Scrambled Telephones (FROST) - це інструмент, розроблений для платформи хмарних обчислень OpenStack. Він функціонує шляхом отримання даних із журналів API, віртуальних дисків і журналів фаєрволу для проведення розслідування.

Він може збирати дані безпосередньо з шару управління хмарою, уникаючи взаємодії з операційними системами гостьових віртуальних машин, та використовує вбудований механізм криптографічних хешів для зібраних доказів. Більше того, цей інструмент використовує уніфікований формат візуалізації даних і може експортувати їх у зрозумілому форматі до інших інструментів для подальшого розслідування.

FROST включає два сегменти OpenStack - Nova та Horizon. У Nova обчислювальний сервіс надається через віртуальні сервери, як в AWS EC2, і реалізовано обчислювальний API. Horizon при цьому надає веб-інтерфейс користувача для OpenStack, він спілкується з Nova через Compute API. На рисунку 2.3 показано інтеграцію Nova і Horizon з OpenStack [41].

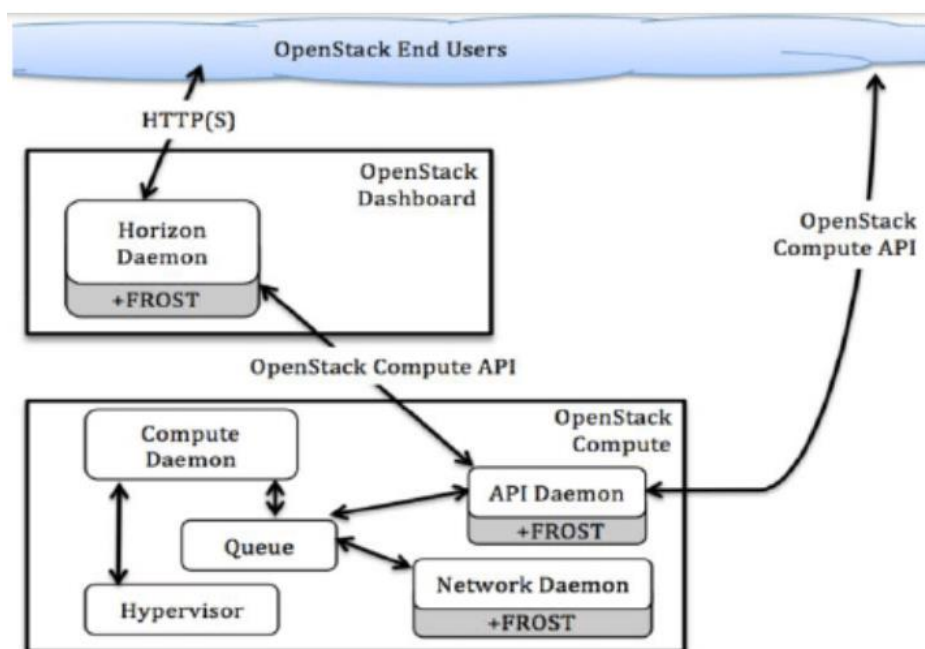


Рисунок 2.3 – Архітектура OpenStack для реалізації FROST

2. Cellebrite UFED Cloud Analyzer дозволяє здійснювати збір та аналіз даних із хмарних акаунтів, включаючи соціальні мережі, електронну пошту та служби зберігання даних. Він підтримує широкий спектр провайдерів хмарних сервісів і допомагає виявляти цифрові докази.

Цей інструмент особливо корисний у випадках коли CSP відмовляється від співпраці або коли є якісь юридичні обмеження. В той час як FROST зосереджений на використанні в IaaS, Cellebrite UFED Cloud Analyzer є більш універсальним, має ширший доступ до різних хмарних служб і може отримувати не тільки активні дані, а й видалену або архівовану інформацію. Для підтвердження автентичності зібраних даних UFED веде журнал усіх процесів вилучення інформації.

3. Magnet AXIOM Cloud - цей інструмент забезпечує комплексний збір та аналіз даних з хмарних платформ, таких як AWS, Azure та Google Cloud. Дозволяє відновлювати, оглядати та зберігати докази, що є важливим для глибоких форензичних розслідувань.

4. Mandiant CloudLens надає можливість моніторингу хмарних середовищ та аналізу журналів для цілей безпеки, допомагає виявляти та розслідувати інциденти.

5. Volatility Framework - відкритий інструмент для аналізу дамів пам'яті, включаючи ті, що отримані з віртуальних машин у хмарних середовищах. Використовується для виявлення ознак компрометації у віртуальних машинах.

6. AccessData Cloud Extractor забезпечує збір та збереження цифрових доказів з хмарних сховищ, соціальних мереж та поштових сервісів, допомагаючи створювати всеосяжний портрет онлайн-активності користувача.

7. Autopsy - відкритий цифровий форензичний інструмент, який дозволяє аналізувати докази з різних джерел, включаючи хмарні сховища. Він полегшує аналіз доказів з різних джерел, розширюючи можливості форензичних розслідувань.

8. BlackBag BlackLight підтримує аналіз даних як з традиційних пристроїв, так і з хмарних сервісів. Допомагає збирати і тлумачити дані хмарних облікових записів.

9. X-Ways Forensics - версатильний інструмент цифрової форензики, який підтримує огляд доказів з хмарних сховищ, електронної пошти та інших джерел. Надає гнучкість для детального аналізу доказів.

10. Azure Sentinel забезпечує захист від загроз у Azure та гібридних середовищах. Використовується для виявлення та реагування на загрози в інфраструктурі хмари.

Ця платформа включає в себе цілий комплекс інструментів: Azure Disk Encryption і Log Analytics, збирають інформацію з Azure VMs; Azure Storage Explorer і Event Grid допомагає аналізувати Azure Data Lake Storage; журнали аудиту Azure AD і звіти про безпеку показують хто, коли, звідки отримував доступ і до чого; Azure Monitor and Log Analytics: на основі цих даних малюють картину працездатності системи та можуть виявити аномалії, що вказують на зловмисну діяльність; логи Network Security Group (NSG) відстежують трафік, що надходить у ваші віртуальні мережі Azure та виходить із них, вони виявляють підозрілі з'єднання, спроби несанкціонованого доступу та потенційне викрадання даних. Azure Sentinel - це хмарна платформа SIEM, що об'єднує всі ці джерела, збирає, співвідносить та аналізує дані з усього середовища Azure, надаючи цілісне уявлення про потенційні загрози безпеці в режимі реального часу та допомагаючи реагувати на інциденти.

11. AWS CloudTrail логує всі виклики API, що виконуються в AWS, дозволяючи проводити детальний форензичний аналіз. Це важливий інструмент для відстеження дій користувачів та змін у сервісах AWS.

12. Google Cloud Forensics Utils — це набір інструментів із відкритим кодом для розслідування різноманітних інцидентів безпеки в Google Cloud Platform (GCP) і реагування на них. Інструменти включають plaso, dfVFS і log2timeline, які можна використовувати для збору, обробки й аналізу даних із джерел GCP.

13. Sleuthkit — це набір інструментів цифрової криміналістики, який можна використовувати для дослідження різних пристроїв і систем, у тому числі хмарних сховищ. Sleuthkit містить інструменти для вирізання файлів, вилучення метаданих і аналізу файлових систем.

14. Cado — це хмарна платформа криміналістики та реагування на інциденти, яка допомагає групам безпеки розслідувати і реагувати на інциденти безпеки в хмарі. Cado автоматизує збір і аналіз даних, заощаджуючи час і зусилля команд безпеки. Cado також підтримує дослідження в багатьох хмарних постачальниках і середовищах, таких як AWS, Azure і GCP.

Ці інструменти та технології надають фундаментальну підтримку для розслідувань у хмарних середовищах, забезпечуючи можливість збору, аналізу та збереження цифрових доказів у складних і динамічних умовах. Вони допомагають автоматизувати збір доказів інцидентів, проте все ще потребують високого рівня залученості аналітика для пошуку першопричин інцидентів. Крім цього, нестандартизований вигляд логів у системі журналювання ускладнює цей процес, змушуючи самотійно вдосконалювати парсинг логів за допомогою власноруч створених пайплайнів.

Таким чином, більшість інструментів здатні автоматично збирати дані, але аналіз цих даних часто потребує ручного втручання, що може призвести до помилок та пропусків у розслідуваннях. Тому для більш дієвого розслідування в розробленій моделі розслідування інцидентів запропоновано використовувати можливості штучного інтелекту [42, 43]. Запровадження такого підходу до хмарної форензики може кардинально змінити процес розслідування інцидентів, роблячи його більш дієвим та менш затратним у плані часу та ресурсів.

Висновки за розділом 2

В даному розділі було проаналізовано особливості розслідування інцидентів у хмарному середовищі, що розкривається через поняття хмарної форензики. Визначено головні виклики цього процесу, які стосуються трьох вимірів: технічного, організаційного та юридичного.

Було розглянуто основні етапи процесу хмарної форензики: ідентифікацію, збір та збереження даних, огляд та аналіз, а також презентацію результатів

розслідування. Для кожного з етапів були надані відповідні рекомендації та інструменти, які допомагають у забезпеченні успішного розслідування інцидентів.

Цей розділ надає опис інструментів для хмарної форензики, які забезпечують комплексне розслідування у хмарних середовищах, дозволяючи збирати, аналізувати та зберігати цифрові докази з різних сервісів, наголошуючи при цьому на необхідності тісної співпраці з надавачем хмарних послуг для досягнення ефективних результатів розслідування.

РОЗДІЛ 3

РОЗРОБКА МОДЕЛІ

СИСТЕМИ РОЗСЛІДУВАННЯ ІНЦИДЕНТІВ У ХМАРІ

3.1 Модель збору доказів

Як було наведено раніше, основна відмінність хмарної форензики від традиційної полягає в специфіці збору доказів, яка залежить від типів моделі обслуговування та розгортання хмарних обчислень.

Запропонована на рисунку 3.1 мета-модель для розслідування інцидентів у хмарних середовищах, зосереджена на наданні структурованого підходу до управління та проведення розслідувань у налаштуваннях хмарних обчислень [44]. Ця модель включає різні аспекти хмарної криміналістики. Вона окреслює процеси, діячів, ресурси та документацію, що беруть участь у розслідуванні хмарних інцидентів.

Характеристики моделі [44]:

1. Ідентифікація діячів: мета-модель з'ясовує основних діячів, які беруть участь у розслідуванні хмарної криміналістики, включаючи зловмисників, аналітиків, CSP та жертв. Це розмежування є важливим для розуміння ролей і відповідальності в процесі розслідування. Наприклад, при формуванні SLA.

2. Інцидент: факт наявності інциденту запускає процес розслідування, який спрямований на вирішення інциденту, ідентифікацію винуватців та захист хмарних середовищ від майбутніх атак. Ця модель підкреслює важливість чіткого визначення інциденту та встановлення конкретних цілей для розслідування.

3. Ресурси та інструменти: для проведення ретельного розслідування необхідні різноманітні інструменти, включаючи технічні, юридичні та процедурні ресурси. Мета-модель вказує на необхідність кваліфікованого персоналу, відповідних криміналістичних інструментів та добре визначених попередньо методологій збору та аналізу доказів.

4. Збір та аналіз доказів: центральним є процес ідентифікації, збору та збереження доказів. Модель акцентує увагу на викликах збору доказів у хмарних середовищах, такі як робота з зашифрованими даними, розподілене зберігання та забезпечення цілісності та автентичності зібраних доказів.

5. Документація та ланцюжок опіки: підтримання детальної документації та надійного ланцюжка опіки для всіх зібраних доказів є також одним з важливих аспектів описаних у мета-моделі. Цей елемент забезпечує, що докази придатні до використання у юридичних процедурах, і що всі дії розслідування є прозорими та підзвітними.

6. Процес розслідування: ця модель окреслює всеосяжний процес розслідування, включаючи в себе підтвердження інциденту, ідентифікацію, збір та придбання доказів, їх огляд та аналіз, і, нарешті, представлення результатів. Також враховуються паралельні діяльності, такі як збереження доказів, документація та підготовка.

Розглянута нами в роботі мета-модель може слугувати керівництвом для програмних інженерів та дизайнерів систем при розробці хмарних сервісів, що потребують надійної системи управління інцидентами. Інтегруючи компоненти мета-моделі, організація хмарних сервісів розробляється таким чином, щоб сприяти швидким та ефективним розслідуванням, тим самим підвищуючи загальну безпеку та надійність хмарних середовищ. Організації можуть також використовувати цю мета-модель для формулювання або удосконалення своїх стратегій реагування на інциденти.

Впроваджуючи структурований підхід цієї моделі для збору доказів, організації можуть гарантувати, що вони підготовлені до ефективного реагування на інциденти пов'язані з хмарою та їх розслідування. Ця модель може використовуватися для розробки програм навчання та симуляційних вправ для SOC-інженерів. Знайомлячись з концепціями та процесами, визначеними в мета-моделі, вони можуть покращити свої навички та готовність до здійснення реальних розслідувань. Окрім цього створена модель є допоміжною для створення правових вимог та становлення рамок задокументованої процедури розслідування

інцидентів, що є важливим для відповідності юридичним стандартам і регуляторним нормам.

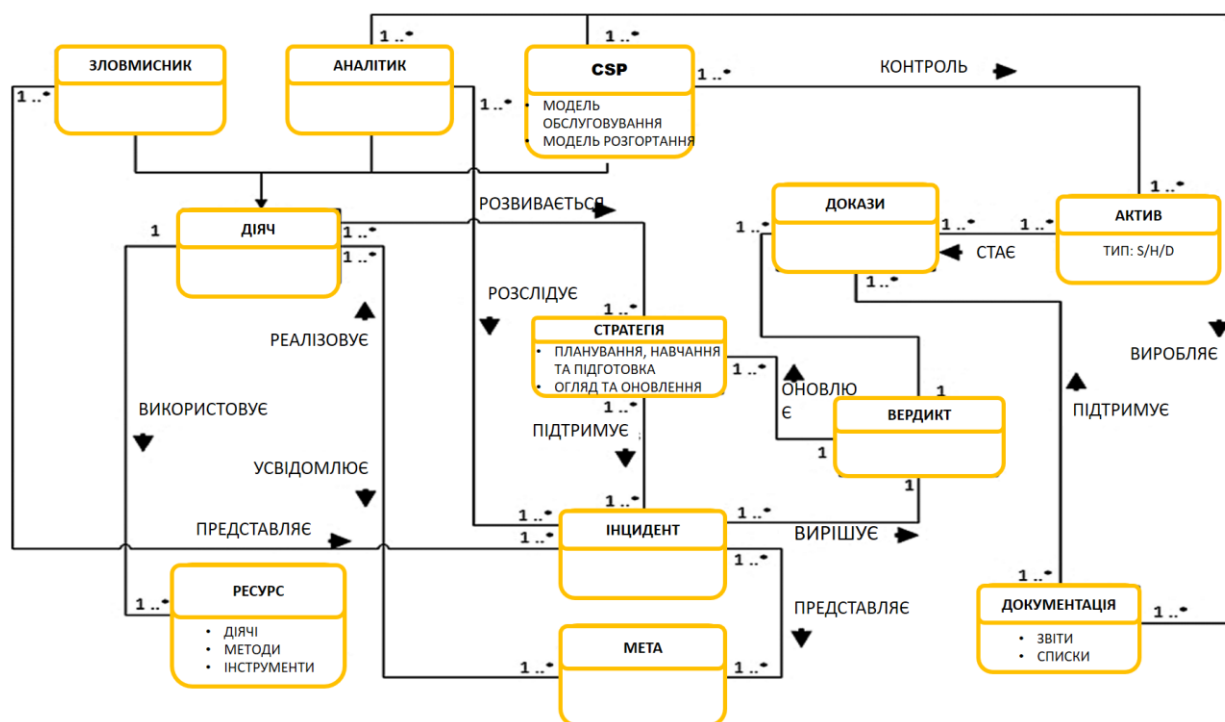


Рисунок 3.1 – Мета-модель збору доказів хмарних інцидентів

Варто зазначити, що цифровий простір, в тому числі й хмарне середовище, є простором не тільки для інцидентів, а й злочинів, що потребують судового розслідування. Цифрова криміналістика в Україні є важливою частиною кримінального процесу та законодавства, спрямованого на боротьбу з кіберзлочинністю та іншими злочинами, де використовуються цифрові технології. Законодавство України в цій сфері продовжує розвиватися, щоб відповідати сучасним викликам цифрової ери.

Вимоги до цифрових доказів у криміналістичних розслідуваннях та судовому процесі є строгими для забезпечення їх надійності та дійсності. Існує кілька загальноприйнятих принципів, які застосовуються до цифрових доказів [45]:

-Цифрові докази повинні бути автентичними, тобто має бути можливість довести, що дані не були змінені або модифіковані з моменту їх збору. Це досягається за допомогою криптографічних методів, зокрема хешування.

- Має бути забезпечена цілісність цифрових доказів, що означає захист від несанкціонованих змін. Цілісність може перевірятись через використання контрольних сум та хеш-сум.

- Докази повинні бути достовірними, тобто зібрані та збережені в такий спосіб, що відображає точний стан даних в момент їх фіксації. Також важливо, щоб процес збору доказів був документований для подальшої перевірки.

Цифрові докази мають бути придатними до розгляду в суді, що означає відповідність встановленим процедурним та правовим вимогам. Забезпечення відповідності цим вимогам є важливим для ефективного використання цифрових доказів у кримінальних розслідуваннях та судових процесах.

На рисунку 3.2 запропоновано модель хмарної криміналістики, що має на меті вирішення унікальних викликів, які створює хмарне обчислення для цифрової криміналістики [46]. Ця модель враховує надважливу роль CSP у підтримці надійної цифрової криміналістики та вводить неперервний процес криміналістичного обігу, щоб забезпечити цілісність, доступність та достовірність цифрових доказів у хмарі.

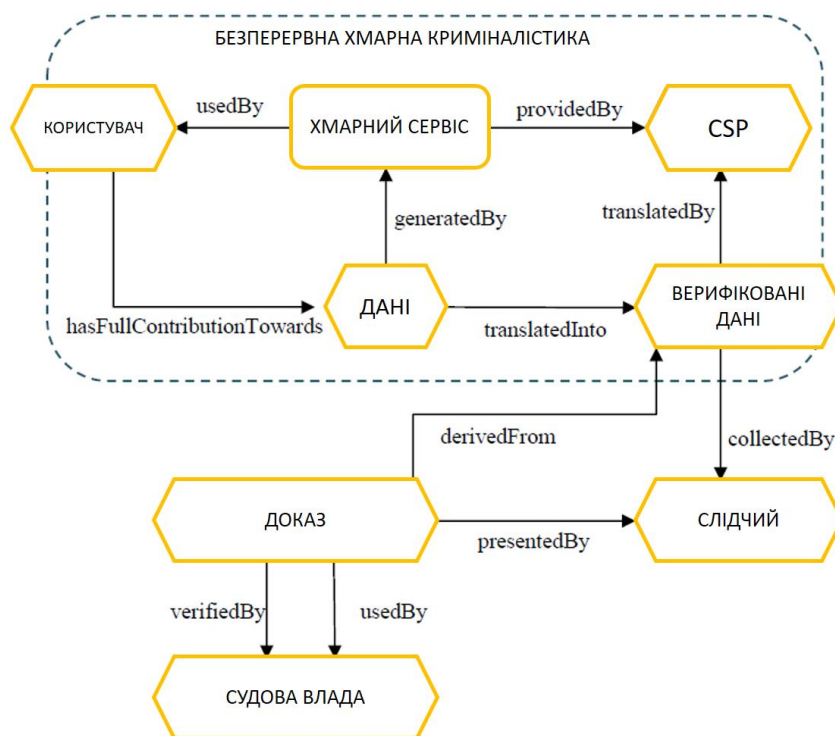


Рисунок 3.2 – Модель хмарної криміналістики

Така модель розширює традиційний процес цифрової криміналістики. Вона структурована навколо наступних ключових елементів:

- Неперервна криміналістика: схема неперервної синхронізації для ефективного зберігання тимчасових та нетимчасових даних, що забезпечує відсутність втрат потенційних доказів. До синхронізації повинні включатись дані, згенеровані активністю користувачів та операційні журнали CSP.

- Криптографічне підтвердження: модель пропонує використання криптографічних акумуляторів, таких як односторонні акумулятори, для генерації перевірених, засвідчених доказів. Ці підтвердження потім стають публічно доступними, дозволяючи перевірку цілісності даних без розкриття самих даних.

- Доступність доказів через безпечні API: щоб подолати фізичну недоступність до хмарної інфраструктури, модель пропонує надання безпечних, тільки для читання API, що дозволяють аналітикам віддалено отримувати доступ до даних та збирати їх. Ця особливість забезпечує можливість отримати необхідні докази без прямого втручання провайдера.

- Перевірка судовою владою: у цій моделі цілісність доказів, представлених у суді, може бути незалежно перевірена судовою владою за допомогою публічно доступних криптографічних підтверджень. Цей крок критично важливий для встановлення довіри до доказів, зібраних з хмарних середовищ.

Необхідність моделі хмарної криміналістики впливає з фундаментальних характеристик хмарного обчислення, які ставлять під сумнів традиційні припущення та практики цифрової криміналістики: розподілена, природа хмарних обчислень ускладнює фізичний доступ та ізоляцію доказів без потенційного порушення даних інших не пов'язаних користувачів, а динамічний розподіл ресурсів у хмарних середовищах означає, що важливі докази можуть бути легко втрачені або змінені, якщо їх не зберегти належним чином і вчасно.

Залежність від CSP для збору та надання доказів породжує небезпеки щодо можливості підробки доказів з боку CSP, користувачів, або навіть слідчих. Акцент моделі на криптографічних доказах та механізмах публічної перевірки вирішує ці

побоювання, надаючи прозорий та перевірений процес обробки цифрових доказів у хмарних середовищах.

Тож модель хмарної криміналістики необхідна для адаптації практик цифрової криміналістики до реалій хмарних обчислень, забезпечуючи можливість проведення розслідувань надійно, прозоро і таким чином, що враховує приватність користувачів та цілісність доказів. Інтегруючи неперервну криміналістику, криптографічне підтвердження, безпечний доступ до доказів та незалежну перевірку, модель надає всеосяжний фреймворк для подолання унікальних викликів хмарної криміналістики.

3.2 Модель пошуку першопричин інциденту

Спираючись на дані, отримані за допомогою правильно організованого надійного збору доказів, їх потрібно опрацювати для пошуку та усунення причин інциденту. Сама ідея аналізу первинних причин (RCA) у запропонованій моделі для розслідування інцидентів у хмарних середовищах, вказує на необхідність інтеграції великих мовних моделей (LLMs) для оптимізації та покращення процесу RCA через автоматизовану обробку, кореляцію та аналіз даних, що були отримані раніше [43]. Цей підхід використовує передові досягнення у галузі обробки природної мови та машинного навчання для вирішення викликів властивих традиційним методам RCA, які часто виконуються вручну, є часомісткими та схильними до помилок.

LLM - це тип штучного інтелекту, заснований на глибоких нейронних мережах, який спеціалізується на обробці та генерації природної мови (NLP). Ці моделі навчаються на дуже великих обсягах текстових даних. Це дозволяє їм виявляти складні мовні закономірності та контексти [47, 48]. Варто зазначити, що в останніх дослідженнях вже згадуються можливості LLM для парсингу логів [49]. В даній моделі запропоновано використовувати LLM для інтерпретації неструктурованих даних і генерації зрозумілих пояснень та звітів, які можуть

використовуватися інженерними командами для вирішення та пом'якшення наслідків інциденту.

Схема моделі для розслідування інцидентів у хмарі, що базується на підході LLM наведено на рисунку 3.3 [43].

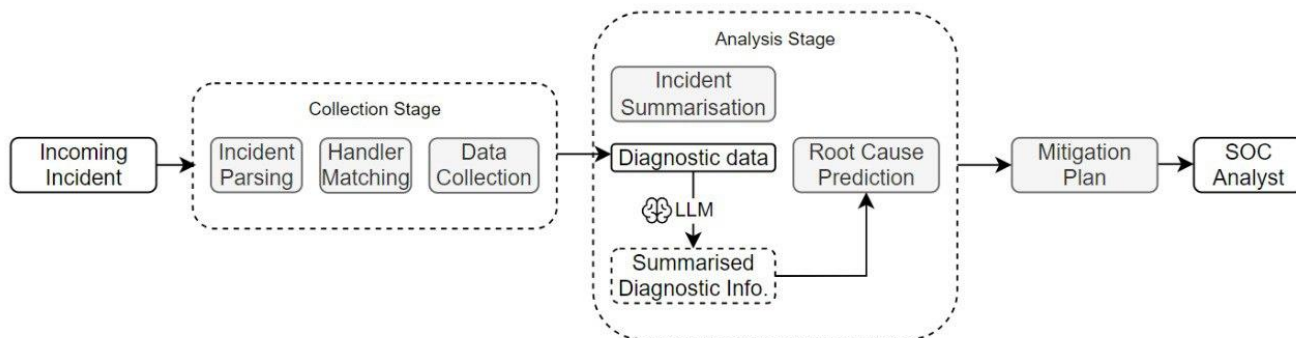


Рисунок 3.3 – Модель системи RCA для хмарних інцидентів

Архітектура запропонованої моделі складається з двох основних етапів [43]:

1. Збір діагностичної інформації: цей початковий етап є критичним для збору відповідних даних з кількох джерел, таких як логи, метрики та сліди, які є важливими для розуміння контексту та специфіки інциденту. В той час як традиційні методи сильно залежать від ручного збору та аналізу, інтеграція LLMs на цьому етапі може автоматизувати та значно прискорити процес. Розумний парсинг та витяг ключової інформації з неструктурованих даних зменшують потребу в ручній праці та мінімізують можливість допущення помилок.

Цей етап включає такі кроки:

- Агрегація даних: збір даних з різних джерел, таких як логи, звіти про помилки та метрики системи, які стосуються інциденту. Дані збираються з системних логів, моніторингових систем та сенсорів, що забезпечують збір інформації про стан системи та її компонентів у реальному часі, а також інтегруються із зовнішніх джерел, таких як бази даних помилок, патчі оновлень тощо.

- Первинна обробка: цей крок передбачає фільтрацію та попереднє сортування, тобто відбір релевантної інформації та видалення даних неважливих при розслідуванні.

- Класифікація інформації: зібрана інформація класифікується за категоріями з використанням алгоритмів машинного навчання на основі її актуальності та потенційного впливу на інцидент для оптимізації наступних етапів розслідування.

Взаємодія між цими компонентами включає використання моделей машинного навчання для ефективної обробки та аналізу великих наборів даних, ідентифікації кореляцій або аномалій.

2. Формування гіпотез первинних причин інциденту: після збору діагностичної інформації наступним кроком є знаходження первинної причини. Саме тут LLMs відіграють ключову роль.

Обробляючи зібрані дані, LLMs можуть ідентифікувати патерни, аномалії або кореляції, які не відразу очевидні для аналітиків. Більше того, здатність LLMs генерувати пояснення природною мовою покращує зрозумілість аналізу, роблячи його легшим для інженерних команд для вжиття відповідних виправних дій. Взаємодія на цьому етапі часто включає ітеративні процеси, де результати постійно уточнюються для підвищення точності.

Цей етап включає RCA причин інциденту з використанням даних, отриманих з діагностичної фази, для точного визначення первинних причин інциденту за допомогою LLM для дослідження текстових даних і визначення потенційних причин інцидентів проводячи семантичний аналіз зі застосуванням NLP для розуміння контексту і змісту даних, що були зібрані та оброблені.

На основі визначених причин інциденту система генерує рекомендації щодо заходів для усунення проблем і запобігання майбутнім інцидентам. Рекомендації можуть включати як негайні виправлення, так і довгострокові стратегічні рішення. Далі відбувається оцінка потенційного або фактичного впливу визначених причин на оперативну діяльність системи, що допомагає сформуванню пріоритетних заходів реагування на інцидент. Синергія між цими етапами та їх компонентами є ключовою для дієвого функціонування моделі розслідування інцидентів.

3.3 Порівняльний аналіз LLM моделей

В основі LLM, як однієї з найбільш відомих моделей NLP, лежить архітектура трансформерів й механізм самоуваги, що дозволяє одночасно обробляти інформацію з різних частин даних, генеруючи змістовний релевантний текст [50-51]. Ці моделі використовують унікальний процес навчання, який можна узагальнити у вигляді етапів загального попереднього навчання й тонкого налаштування, який адаптує мовну модель до виконання конкретного завдання. Ці етапи зображено на рисунку 3.4.

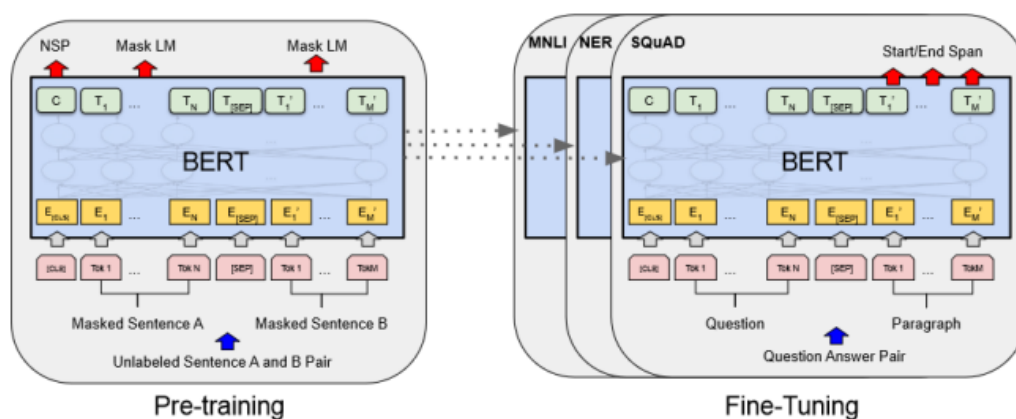


Рисунок 3.4 – Архітектура етапів навчання LLM (BERT)

LLM поділяються на декілька категорій відносно їх архітектури [52]:

1. Енкодери, призначені для обробки вхідних даних і перетворення їх у простір вищої розмірності. До цієї категорії входять Bidirectional Encoder Representations from Transformers (BERT) [53] і його похідні, такі як RoBERTa [54] і DistilBERT [55]. Ці моделі чудово підходять для завдань, які вимагають двонаправленого розуміння контексту з вхідних даних, таких як класифікація речень і розпізнавання сутностей. Аналіз наукових джерел показав, що BERT також може застосовуватись для парсингу логів і їх класифікації [56], однак вона не ефективна для завдань генерації тексту, тому не підходить для нашої моделі.

2. Декодери, призначені для генерування вихідних даних із закодованих даних. Ця категорія включає Generative pre-trained transformer (GPT) моделі,

зокрема GPT-1 [57] і GPT-2 [58]. Вони відмінно справляються із завданнями, які потребують генерування тексту на основі заданої підказки, проте знову ж таки мають свої обмеження з розуміння контексту, які унеможливають їх використання як елементу системи, запропонованої в цій роботі.

3. Енкодери-декодери надають можливість кодування і декодування, що робить їх універсальними як для розуміння введення, так і для генерування виводу [59]. Цей тип моделей включає Text-to-Text Transfer Transformer (T5) [60], Bidirectional Autoencoder Representations from Transformers (BART) [61], а також GPT-3 [62] й GPT-4 [63]. Зазвичай вони використовуються для завдань, які передбачають як розуміння, так і генерування тексту.

Узагальнений огляд різних типів моделей та їх характеристика надані в таблиці 3.1 [64].

Таблиця 3.1

Загальна характеристика поширених моделей LLM

Type	Model Name	#Parameters	Release	Base Models	Open Source	#Tokens	Training dataset
Encoder-Only	BERT	110M, 340M	2018	-	✓	137B	BooksCorpus, English Wikipedia
	RoBERTa	355M	2019	-	✓	2.2T	BooksCorpus, English Wikipedia, CC-NEWS, STORIES (a subset of Common Crawl), Reddit
	ALBERT	12M, 18M, 60M, 235M	2019	-	✓	137B	BooksCorpus, English Wikipedia
	DeBERTa	-	2020	-	✓	-	BooksCorpus, English Wikipedia, STORIES, Reddit content
	XLNet	110M, 340M	2019	-	✓	32.89B	BooksCorpus, English Wikipedia, Giga5, Common Crawl, ClueWeb 2012-B
Decoder-only	GPT-1	120M	2018	-	✓	1.3B	BooksCorpus
	GPT-2	1.5B	2019	-	✓	10B	Reddit outbound
Encoder-Decoder	T5 (Base)	223M	2019	-	✓	156B	Common Crawl
	MT5 (Base)	300M	2020	-	✓	-	New Common Crawl-based dataset in 101 languages (m Common Crawl)
	BART (Base)	139M	2019	-	✓	-	Corrupting text
GPT Family	GPT-3	125M, 350M, 760M, 1.3B, 2.7B, 6.7B, 13B, 175B	2020	-	×	300B	Common Crawl (filtered), WebText2, Books1, Books2, Wikipedia
	CODEX	12B	2021	GPT	✓	-	Public GitHub software repositories
	WebGPT	760M, 13B, 175B	2021	GPT-3	×	-	ELI5
	GPT-4	1.76T	2023	-	×	13T	-

Власне, поєднання властивостей компонентів енкодера й декодера необхідне для розслідування інцидентів, оскільки моделі необхідно розуміти контекст журналів і генерувати детальний послідовний звіт на його основі. Це має вирішальне значення для інтерпретації технічних деталей у журналах інцидентів і представлення їх у зрозумілому форматі.

Дослідження характеристик розглянутих моделей, вказують на те, що моделі сімейства GPT, а саме GPT-3 та GPT-4 мають найбільший потенціал для автоматизації процесу розслідування інцидентів у хмарних середовищах [64].

GPT-3 - це попередньо навчена модель авторегресійної мови зі 175 мільярдами параметрів. GPT-3 демонструє прогресивну здатність до навчання в контексті. Це означає, що GPT-3 можна застосовувати до будь-яких завдань без будь-яких оновлень градієнта або тонкого налаштування, із завданнями визначеними виключно через текстову взаємодію з моделлю [62].

Графік на рисунку 3.5 показує, що GPT-3 значно краще використовує інформацію в контексті, ніж інші LLM. Він показує ефективність навчання моделі в контексті для виконання завдання, яке вимагає видалення випадкових символів зі слова як з описом завдання природною мовою, так і без нього.

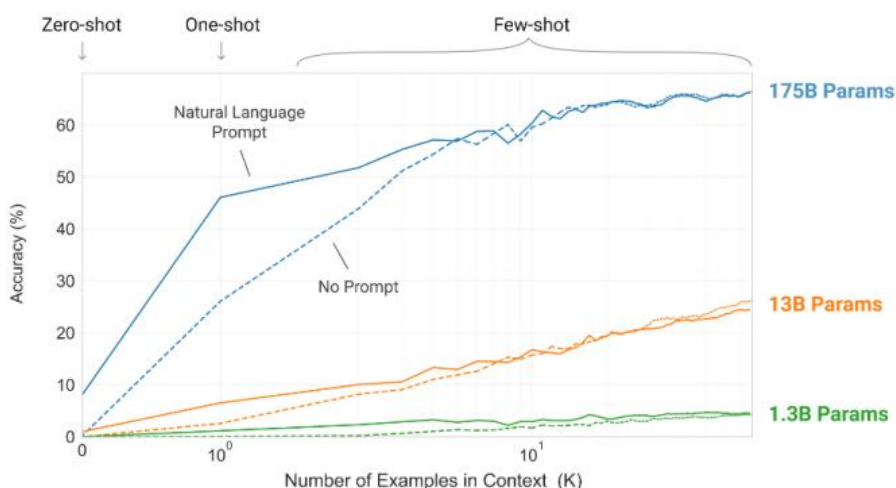


Рисунок 3.5 – Показники ефективності GPT-3

GPT-4, у свою чергу, вважається найпотужнішою моделлю, що навчається на найновіших даних і показує найвищі показники ефективності у виконанні завдань, навіть порівняно з GPT-3 (див. рис. 3.6) [65].

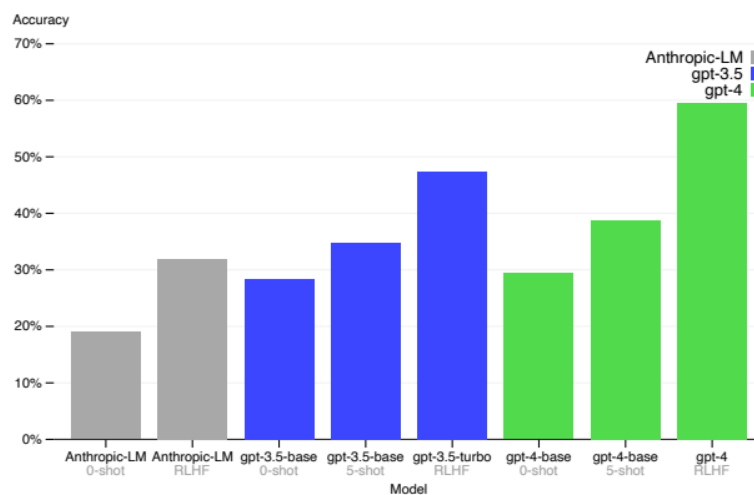


Рисунок 3.6 – Порівняння ефективності GPT-4

Загалом, GPT моделі побудовані на базі революційної архітектури, що використовує трансформери лише для декодера. Вона схематично зображена на рисунку 3.7 [66].

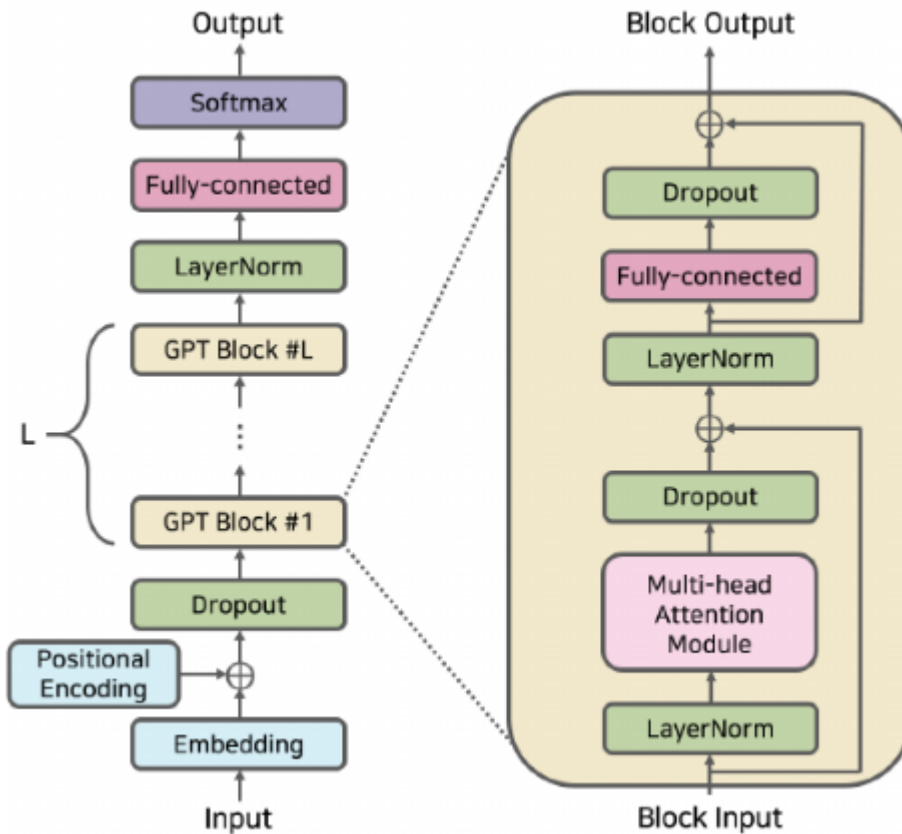


Рисунок 3.7 – Концептуальна архітектура GPT моделі

Як було показано раніше, така архітектура демонструє високу швидкість і ефективність, зокрема для створення зв'язних і контекстно точних вихідних текстів. Модель GPT має високу здатність адаптуватись (згідно з досліджень [65]), тому може бути налаштована для спеціалізації в конкретних областях, в даному випадку – для аналізу журналів інцидентів кібербезпеки. Крім цього, GPT легко інтегрується з іншими інструментами та API, що дозволяє використовувати її в гетерогенних хмарних системах.

Таким чином, GPT можна використати в розробленій нами моделі для виконання завдань аналізу даних, пошуку причинно-наслідкових зв'язків та створення звітів завдяки здатності ефективно обробляти великі обсяги тексту і генерувати зрозумілі висновки. Основні виклики при використанні цієї моделі пов'язані з високим споживанням обчислювальних ресурсів та потенційним ризиком генерації некоректної або упередженої інформації, якщо модель була навчена на некоректних даних.

3.4 Переваги та виклики використання LLM для розслідування інцидентів

Використання LLM з метою аналізу логів для розслідування інцидентів має цілу низку переваг [67]:

- **Ефективність:** LLMs можуть швидко обробляти великі обсяги неструктурованих даних, знаходячи необхідну інформацію для RCA без великих часових витрат, порівняно з аналізом вручну.
- **Точність:** завдяки розширеному розумінню природної мови, LLMs можуть витягувати та точно інтерпретувати зібрану діагностичну інформацію, зменшуючи ймовірність людської помилки та підвищуючи дієвість процесу RCA.
- **Масштабованість:** оскільки хмарні середовища зростають у складності та масштабі, ручний RCA стає все більш непрактичним. LLMs пропонують масштабоване рішення, яке може здійснювати аналіз великих наборів даних без значного збільшення часу на обробку або необхідних обчислювальних ресурсів.

- Навчання та адаптація: LLMs можуть навчатись на основі минулих інцидентів та постійно вдосконалювати свої діагностичні та прогностичні можливості з часом, пропонуючи більш точний та глибокий аналіз, коли стає доступно більше даних для контексту.

- Покращене пояснення та автоматична генерація звітів: генеруючи пояснення для RCA природною мовою, LLMs роблять результати процесу розслідування більш доступними та зрозумілими як для технічних, так і нетехнічних зацікавлених сторін.

Поданий перелік переваг свідчить про те, що інтеграція LLMs у процес RCA для розслідування інцидентів у хмарі представляє прогресивний підхід, який вирішує значні обмеження традиційних методів. Ця модель покращує швидкість та точність процесу RCA автоматизуючи збір діагностичної інформації та її аналіз для знаходження потенційних вразливостей системи, що сприяє підвищенню загального рівня безпеки хмарних сервісів.

Водночас, запропонований підхід має низку обмежень [68]. Проблеми використання LLM для RCA у хмарних середовищах є багатограними і впливають як з властивих характеристик LLM, так і зі специфічних викликів, пов'язаних з управлінням інцидентами в хмарі.

- Узагальнення та специфічні знання домену: LLM відмінно справляються з обробкою та генерацією тексту на основі величезної кількості інформації, на якій вони були навчені. Однак, їхня здатність розуміти специфічні контексти, такі як складні деталі інфраструктури хмари та управління інцидентами, може бути обмеженою. Вони можуть не мати вроджених специфічних для домену знань, які є необхідними для точної діагностики складних інцидентів у хмарі або розуміння тонких відмінностей різних архітектур і конфігурацій хмари.

- Достовірність даних та шум: раніше згадувалось, що ефективність LLM у RCA значно залежить від якості та релевантності наданих для аналізу даних. Хмарні середовища генерують величезний обсяг логів, метрик та слідів, які можуть бути зашумленими, неповними або неконсистентними. Тому процес пошуку в цих даних релевантної інформації для проведення RCA - складне завдання, яке вимагає

ретельної попередньої обробки та розуміння контексту. Незважаючи на наявність механізму уваги, LLM можуть виявитися нездатними ідентифікувати ключові елементи діагностичної інформації серед шуму, що призводить до неточностей у визначенні первинних причин інциденту.

- Динамічна природа хмарних систем: хмарні системи постійно еволюціонують, з появою нових сервісів, змінами конфігурацій та оновленням програмного забезпечення. Ця динамічна природа швидко робить навчальні дані, використовувані для додаткового тонкого налаштування LLM для RCA, застарілими. В результаті LLM може не володіти останньою інформацією або розумінням нових типів інцидентів, що зменшує її ефективність в діагностиці вразливостей нульового дня та інших новітніх проблем.

- Додаткове налаштування LLM з специфічними для конкретної організації наборами даних для підвищення їх точності в RCA є ресурсомістким. Це вимагає значної кількості ресурсів та часу. Більше того, процес оновлення моделей повинен проводитись постійно щоб відповідати змінам в хмарному середовищі, що також додає витрат.

- Ризик галюцинацій: LLM, включаючи ті, що були додатково налаштовані для конкретних завдань, іноді можуть генерувати правдоподібну, але некоректну або нерелевантну інформацію, відому як "галюцинація". У контексті RCA це може відобразитись у неправильному визначенні первинних причин інциденту або помилкових кроків пом'якшення, вводячи в оману інженерів, та затримуючи процес вирішення інцидентів.

Підсумовуючи, LLM мають великий потенціал для автоматизації RCA у хмарних середовищах, але подолання наявних обмежень вимагає постійного вдосконалення моделей, наборів даних та стратегій інтеграції.

Висновки за розділом 3

У цьому розділі було розроблено та представлено модель системи розслідування інцидентів у хмарному середовищі. Ця модель передбачає

структурований підхід до збору доказів, пошуку першопричин та аналізу інцидентів у хмарних обчисленнях.

Розроблена модель збору доказів адаптує традиційні методи цифрової форензики до специфіки хмарних обчислень, враховуючи роль постачальників хмарних послуг. Вона передбачає постійний збір даних для уникнення втрати потенційних доказів, використання криптографічного підтвердження для автентифікації та надання безпечного доступу до доказів через API.

Також було запропоновано модель пошуку першопричин інциденту, яка включає інтеграцію LLM для оптимізації аналізу даних шляхом використання штучного інтелекту та обробки природної мови, сприяючи створенню чітких та зрозумілих пояснень.

Аналіз досліджень основних моделей для аналізу текстових даних показав, що моделі GPT-3 та GPT-4 найбільше підходять для виконання задачі аналізу інцидентів, зважаючи на їх адаптивність і здатність ефективно генерувати та інтерпретувати текст.

В розділі також розглянуто переваги та виклики використання LLM для розслідування інцидентів – їх використання може спростити процес розслідування, проте водночас існують проблеми галюцинацій та потреба в додатковому налаштуванні моделей.

Загалом, запропонована модель допомагає поєднати передові методи генеративного машинного навчання з традиційними процесами, надаючи організаціям комплексний підхід до розслідування інцидентів у хмарних обчисленнях.

РОЗДІЛ 4

ДОСЛІДЖЕННЯ МОДЕЛІ СИСТЕМИ РОЗСЛІДУВАННЯ ІНЦИДЕНТІВ У ХМАРНОМУ СЕРЕДОВИЩІ

4.1 Вибір технологій та інструментів для реалізації моделі

Перед тим як розпочати розроблювати прототип програми, що буде реалізувати запропоновану модель на основі LLM, важливо правильно обрати технології та інструменти, які забезпечують швидкодію, безпеку та надійність такої системи.

Зваживши характеристики різних мов програмування, для розробки прототипу системи для аналізу інцидентів у хмарі на основі LLM було прийнято рішення використовувати Python, адже ця мова має ряд переваг [69], що й зумовили такий вибір:

1. Порівняно з іншими мовами, Python вирізняється багатю палітрою бібліотек і фреймворків, створених спеціально для реалізації технологій хмарних обчислень і аналізу даних. Наприклад, серед них вирізняються бібліотеки Pandas, NumPy, TensorFlow і PyTorch, що пропонують потужні інструменти для обробки даних, аналізу та машинного навчання, необхідні для аналізу інцидентів у хмарі.

2. Крім цього, Python відомий своєю простотою, гнучкістю інтеграції з іншими мовами та системами, що роблять його ідеальним вибором для взаємодії з різними компонентами хмарної інфраструктури. Це визначальна особливість, адже під час створення програми для аналізу інцидентів у хмарі вкрай важливо мати можливість легко інтегруватися з різними хмарними платформами, API та службами.

3. Значною перевагою є те, що чистий синтаксис цієї мови і великі бібліотеки дозволяють миттєво створювати необхідні прототипи та розробляти нові функції. Здебільшого, вирішальним при розробці прототипу системи є час виходу на ринок,

а простота Python дозволяє розробникам швидко й ефективно виконувати ітерації, прискорюючи процес розробки.

4. За своєю суттю Python здатний до масштабування, а оскільки хмарним сервісам характерне генерування великих обсягів даних, він добре підходить для побудови систем. Більше того, властивість масштабуватись, про яку було згадано раніше (див. п. 1.1), особливо важлива в хмарних системах, де здатність швидко та динамічно адаптуватись до нових ресурсів є ключовою вимогою.

5. Так як запропонована модель базується на використанні LLM, важливою також є широка підтримка Python машинного навчання та NLP.

З огляду на вказані переваги, можна зробити висновок, що багата екосистема Python, легкість інтеграції, можливості швидкого розвитку, підтримка спільноти, масштабованість і широка підтримка машинного навчання роблять його ідеальним вибором для створення прототипу системи аналізу інцидентів у хмарі на основі LLM.

Раніше нами було надано порівняльний аналіз моделей LLM (див. п. 3.3), який продемонстрував доцільність використання GPT-4, тому при створенні програми було вирішено використати одну (на даний момент найновішу) з її моделей - gpt-4-1106-preview. Таку модель було обрано з огляду на останні оновлені навчальні дані, - адже модель навчається на даних до квітня 2023 року, - та розширеним контекстним вікном, яке містить 128 000 токенів, що, у свою чергу, забезпечує глибшу та всебічну адаптивну взаємодію [70]. Крім цього, важливою особливістю моделі "gpt-4-1106-preview" є можливість виклику функцій [71], зокрема про реалізації процесу розслідування інцидентів. В контексті розробленої програми дана властивість була використана з наступною метою:

1. Насамперед, це запит функцій для аналізу даних, оскільки модель може автоматично визначати, коли необхідно викликати певні функції для аналізу даних, які надходять у хмарний сервіс. Якщо до системи надходить попередження про можливий інцидент безпеки, така модель розуміє, що необхідно викликати функції для перевірки вхідних логів, аналізу доступу та перевірки конфігурації.

2. Після виявлення можливого інциденту можуть викликатись різні функції для спілкування з іншими, зовнішніми, хмарними сервісами, або ж локальними. Скажімо, викликати функцію для отримання додаткової інформації з системи моніторингу, або розпочати автоматичний запуск процедури відновлення.

Результат виконання цих функцій надає програмі можливість для забезпечення автоматичної обробки отриманих результатів (проаналізувати логи, виявити аномальні зміни у конфігураціях системи або виявити неpritаманну активність користувачів) та зробити висновки щодо потенційного інциденту та його причин.

Важливим є те, що можливість виклику функцій дозволяє програмі реагувати на інциденти в реальному часі, автоматично виконуючи необхідні дії для їх вирішення. Відповідно, можливість виклику функцій у моделі "gpt-4-1106-preview" відкриває широкі можливості для реалізації програми розслідування інцидентів у хмарі, допомагаючи автоматизувати, пришвидшити процес аналізу інцидентів.

4.2 Розробка прототипу системи розслідування інцидентів

В ході виконання роботи для дослідження функціоналу запропонованої моделі розслідування інцидентів було розроблено прототип програми, що представляє собою інтерактивну систему для спілкування з користувачем і виконання команд на серверах за допомогою штучного інтелекту на основі LLM моделі ChatGPT.

Верхньорівневу схему функціонування розробленого прототипу системи розслідування інцидентів зображено на рисунку 4.1.

На цій схемі змальовано структуру функціонування інтерактивного чат-боту для аналітиків безпеки, який складається з конwersаційного інтерфейсу, що дозволяє користувачам формулювати запит до системи природною мовою. Цей запит може включати в себе вже відомі деталі щодо виявленого інциденту, уточнення вимог до результатів пошуку та інших допоміжних даних про систему чи інцидент.

Наступним компонентом є попередньо визначений запит, яким керується модель. Чітко сформульований запит забезпечує вищу вірогідність того, що система зосередиться на релевантних аспектах проблеми, що досліджується, оскільки дозволяє системі краще розуміти контекст, це підтверджено дослідженнями [72, 73]. Добре структуровані запити можуть оптимізувати використання ресурсів системи, що критично важливо для хмарних обчислень, де ресурси розподілені та обмежені. При цьому некоректно сформульовані запити можуть призводити до аналізу невідповідних даних, що суттєво знижує якість рішень, базованих на цих даних.

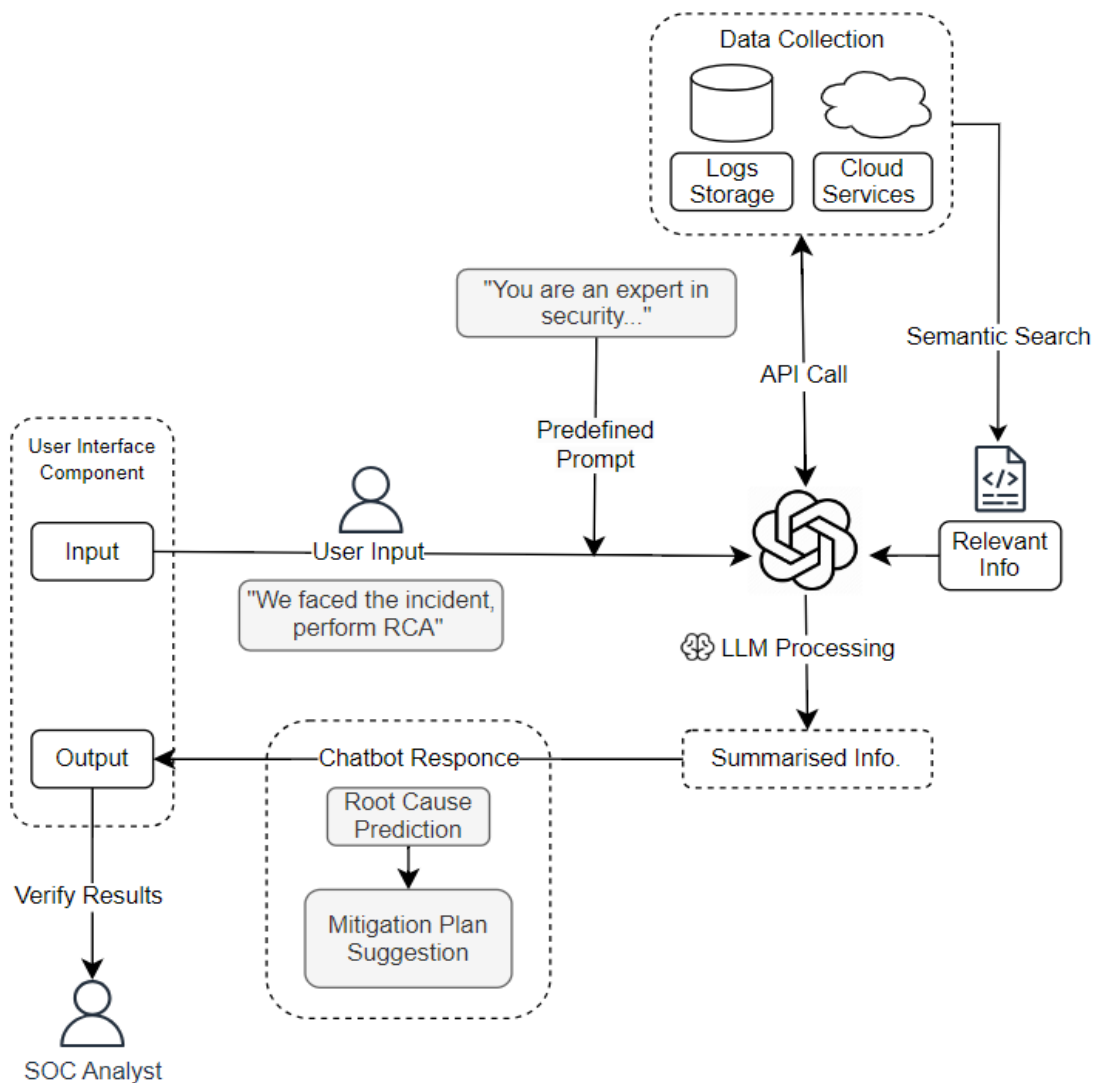


Рисунок 4.1 - Схема прототипу моделі розслідування інцидентів

Це підтверджують і результати досліджень, зображені на рисунку 4.2. У роботі було проаналізовано результати виконання простого запиту і запиту з розширеною підказкою, де було покроково описано процес аналізу журналу логів. Як бачимо, розширений запит приніс кращі результати - на 9,4% і 21,8% зросли показники точності відповіді (MLA) та відстані редагування (ED) відповідно [64].

Dataset	PT1			PT3 (Simple)			PT4 (Enhance)		
	GA	MLA	ED	GA	MLA	ED	GA	MLA	ED
HDFS	0.960	0.939	0.062	0.413	0.884	0.535	0.920	0.892	1.197
Hadoop	0.795	0.525	11.017	0.740	0.450	11.556	0.801	0.449	10.709
Spark	0.925	0.922	0.596	0.623	0.788	0.880	0.700	0.922	0.662
Zookeeper	0.667	0.233	5.460	0.797	0.233	6.672	0.648	0.273	4.409
BGL	0.878	0.790	5.258	0.243	0.686	8.512	0.947	0.863	3.329
HPC	0.807	0.497	3.498	0.592	0.605	5.277	0.920	0.908	0.816
Thunderb	0.568	0.808	5.933	—	—	—	0.255	0.505	3.395
Windows	0.686	0.301	17.623	0.148	0.292	20.239	0.403	0.525	8.602
Linux	0.910	0.635	3.328	0.286	0.657	3.428	0.445	0.594	3.448
Android	0.711	0.549	10.763	0.754	0.574	12.087	0.922	0.652	7.349
HealthApp	0.898	0.628	6.560	0.767	0.637	6.498	0.886	0.636	6.425
Apache	1	1	0	0.984	0.708	4.955	1	1	0
Proxifier	0.001	0.014	27.025	0	0.001	18.424	0.001	0.016	27.730
OpenSSH	0.659	0.170	7.854	0.261	0.335	6.609	0.462	0.451	4.837
OpenStack	0.449	0.433	7.440	0.355	0.315	10.670	0.524	0.433	7.004
Mac	0.619	0.248	25.530	0.434	0.228	39.599	0.614	0.380	17.919
Average	0.721	0.543	8.621	0.493	0.493	10.396	0.653	0.594	6.739

Рисунок 4.2 - Порівняння результатів роботи GPT при різних запитах

У розробленому прототипі використано чіткий та зрозумілий запит до GPT, який наголошує на ролі бота та надає йому інструкції за якими потрібно працювати, його зображено на рисунку 4.3

```
SYSTEM = 'You are a highly skilled cybersecurity specialist who can answer questions about server settings for their protection and help solve problems, help identify vulnerabilities and ensure security, investigate incidents, and provide information to help understand. Execute any commands at your own discretion on the servers to get a better understanding of the problem if you think it will be useful. Provide the answer exclusively in English. You should not suggest taking any actions or try to solve the problem yourself.'
```

Рисунок 4.3 – Сформований запит до GPT

Наступним кроком програма виконує пошук даних, необхідних для розслідування інциденту. На даному етапі збір даних може відбуватись за допомогою аналізу логів, зі сховища даних, наприклад, для можливості розуміння етапів проведення атаки, а також за допомогою API запитів до хмарних сервісів, щоб здійснити аналіз конфігурацій уражених компонентів. При цьому виконання кожного запиту підтверджується аналітиком, для додаткового контролю.

Отримавши релевантні дані, модель опрацьовує їх та формулює висновки. Як підсумок роботи програми, аналітик отримує покроковий опис того як відбувався інцидент, потенційні першопричини його виникнення, список рекомендацій для усунення наслідків інциденту та порад для усунення недосконалостей системи, що зумовили виникнення інциденту.

Розглянемо загальну структуру коду створеної програми, її компоненти та взаємодію між ними (див. додатки Б-Г):

1. Головний модуль (`main.py`) - це головний вхідний файл програми, він ініціалізує клієнта ChatGPT, системні параметри та інструменти. Цей модуль містить головний цикл інтеракції з користувачем. Сценарій `main.py` організовує взаємодію між користувачем і системою. Далі описано як працює кожна частина сценарію:

- Конфігурація системи та серверів: Сценарій визначає опис і конфігурацію системи для серверів, звідки будуть отримані логи та інша інформація про конфігурацію системи.

- Виконання команд сервера (функція `run_command_on_server`): Ця функція виконує команду на вказаному сервері. Він друкує повідомлення про стан підключення до сервера та команду, що виконується.

- Інтерактивне виконання команд сервера (функція `run_server_command`): Ця функція пропонує користувачеві підтвердити, чи потрібно виконати команду на певному сервері. Він перевіряє ім'я сервера та команду, а потім викликає `run_command_on_server` для виконання команди.

- Визначення інструменту: Сценарій визначає інструмент для виконання команд на серверах (`run_server_command`) і містить опис його параметрів.

- Ініціалізація ChatGPTClient: ChatGPTClient ініціалізується описом системи та визначеним інструментом.

- Цикл взаємодії користувача: Сценарій входить у цикл, де він неодноразово запитує користувача про введення. Для кожного введення користувачем він запитує ChatGPTClient щодо відповіді за допомогою `get_answer`. Він друкує відповідь із моделі ChatGPT.

- Вивід результату виконання аналізу логів та сканування конфігурацій системи.

Таким чином, `main.py` служить основною точкою входу для взаємодії з системою, керування введенням даних користувачами, виконання команд на серверах і генерування відповідей за допомогою моделі ChatGPT.

2. Модуль API (`api.py`) містить клас `ChatGPTClient`, який взаємодіє з API OpenAI для виклику моделі ChatGPT. Цей модуль забезпечує передачу запитів до моделі та обробку відповідей. Використовується для спілкування з іншими компонентами програми через методи `perform_request` та `get_answer`.

Далі розглянуто як працює кожна частина цього модуля та як вони взаємодіють:

- Ініціалізація (метод `__init__`): Клас `ChatGPTClient` ініціалізується такими параметрами, як модель для використання, системна інформація, початкові повідомлення та інструменти (якщо такі є). Він налаштовує клієнт OpenAI за допомогою наданого ключа API та ініціалізує інші атрибути, такі як модель, інструменти та відображення функцій інструменту.

- Керування повідомленнями: Метод `clear_messages` ініціалізує або скидає список повідомлень для надсилання в модель ChatGPT.

- Обробка запитів (метод `perform_request`): Цей метод виконує запит до OpenAI API для створення завершень на основі наданих повідомлень. Він містить додатковий прапор (`use_tools`), щоб визначити, чи використовувати додаткові інструменти для обробки відповіді.

- Обробка відповіді (метод `get_answer`): Отримавши запит, він додає повідомлення користувача до списку повідомлень. Потім він викликає

`perform_request`, щоб отримати відповідь від OpenAI API, передаючи будь-які вказані інструменти для обробки. Після отримання відповіді він обробляє будь-які виклики інструментів, включені у відповідь, виконуючи відповідні функції та додаючи їх результати до списку повідомлень. Після обробки інструментів (якщо такі є) він робить ще один запит до OpenAI API для створення остаточної відповіді. Він вловлює та обробляє різні винятки, які можуть виникнути під час процесу запиту API, наприклад помилки підключення, помилки обмеження швидкості або помилки API.

- Обробка особливих ситуацій реалізована для ефективної обробки помилок, які можуть виникнути під час запитів API, таких як помилки підключення, помилки обмеження швидкості або помилки API.

Загалом, модуль `api.py` діє як міст між запитом користувача, моделлю ChatGPT і будь-якими додатковими інструментами або необхідними обробками, надаючи чистий інтерфейс для взаємодії з OpenAI API та ефективної обробки відповідей.

3. Утилітарний модуль (`utils.py`) містить корисні функції для виконання різних операцій, таких як виконання команд у терміналі, відображення анімації очікування та форматування кольорового виводу.

Модуль `utils.py` використовується для взаємодії з серверами через функції `run_command` та `run_command_on_server`. Вказані компоненти серверів відповідають за виконання команд на серверах і обробку відповідних запитів від користувача. Вони використовуються для виклику внутрішніх функцій, що дозволяють виконати команди на серверах та обробити їх результат.

Модуль `utils.py` містить службові функції та класи, які використовуються в системі. Давайте розглянемо, як працює кожна частина цього модуля:

- Менеджер контексту для анімації очікування (`waiting_animation`): Цей контекстний менеджер створює просту анімацію очікування в консолі, у фоновому режимі, поки виконується якийсь процес.

- Функція для запуску команд оболонки (`run_command`): Ця функція виконує команди оболонки та повертає результат. Він обробляє певні команди

інакше: для команди 'get_logs' він читає журнали з файлу та повертає їх. Для інших команд він використовує subprocess.run() для виконання команди та отримання її результату. Якщо команда не виконується, вона повертає повідомлення про помилку.

- Кольорові константи та функція форматування кольорів (bcolors, s): Цей розділ визначає константи кольорів для використання у виводі терміналу.

Загалом, utils.py надає основні службові функції для обробки команд оболонки, відображення очікування анімації та форматування виводу терміналу за допомогою кольорів. Ці утиліти використовуються в різних частинах системи, наприклад під час взаємодії з користувачем та під час обробки викликів інструментів у модулі api.py.

Компоненти програми взаємодіють між собою таким чином:

1. Головний модуль ініціалізує клієнта ChatGPT та запускає головний цикл взаємодії з користувачем.
2. Під час взаємодії з користувачем, інтерфейс отримує запит від користувача та передає його на обробку клієнту API.
3. Клієнт API викликає модель ChatGPT для отримання відповіді та обробляє її.
4. Після обробки даних, отриманих з логів та базуючись на власній базі знань і враховуючи наявний контекст (завдяки механізму уваги), генерується відповідь, яка містить деталі розслідування інциденту та іншу необхідну інформацію.

4.3 Аналіз результатів тестування моделі

Для дослідження дієвості запропонованого прототипу системи розслідування інцидентів було використано дані, отримані під час реалізації змодельованої атаки на наявні хмарні системні ресурси.

У сценарії для тестування, зловмисник використовує неправильно налаштоване хмарне сховище та вразливості хмарних серверів для доступу до бази

даних. Наведені нижче кроки описують послідовність проведеної атаки. Цей приклад містить анонімізовані технічні деталі та потенційні події, які можна спостерігати якщо аналізувати журнал логів на кожному етапі:

Крок 1: Початкова розвідка. Зловмисник сканує мережеві ресурси для пошуку публічно доступного хмарного сховища та вразливих серверів.

Інструменти, такі як Sensys, можуть використовуватися для виявлення хмарних ресурсів, які є публічно доступними, та ідентифікації потенційних вразливостей. Під час цієї фази логи доступу від постачальника хмарних послуг можуть показувати незвичайні шаблони доступу та великі обсяги вхідних запитів з конкретних IP-адрес.

```
2024-04-10T12:30:45Z source_ip=192.168.1.100 request_method=GET request_url="http://example.cloudservice.com" status=200
2024-04-10T12:30:47Z source_ip=192.168.1.100 request_method=GET request_url="http://example.cloudservice.com/api" status=200
```

Рисунок 4.4 – Приклад логів доступу від CSP

Крок 2: Сканування вразливостей. На цьому кроці зловмисник використовує сканер для виявлення вразливостей, які в подальшому будуть експлуатовані.

Використані інструменти: Сканер вразливостей, такий як Nessus або Qualys, використовується для дослідження серверів у хмарі та ідентифікації вразливостей безпеки, таких як застаріле програмне забезпечення, відкриті порти або неправильна конфігурація. Логи фаєрволу можуть містити повторювані запити до різних портів та кінцевих точок, що свідчить про активність сканування, їх і було використано при проведенні дослідження, приклад логів зображено на рисунку 4.5.

```
{ "httpRequest": { "latency": "0.103396s", "remoteIp": "192.168.1.100", "requestMethod": "GET", "requestSize": "274",
"requestUrl": "https://35.208.54.153/blogs/.git/config", "responseSize": "338", "status": "502",
"userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/100.0.4896.127 Safari/537.36 AtContent/96.5.9594.95", "insertId": "ziy0rxf3mz3c4",
"jsonPayload": { "@type": "type.googleapis.com/google.cloud.loadbalancing.type.LoadBalancerLogEntry",
"backendTargetProjectNumber": "projects/546510619748", "cacheDecision": [ "RESPONSE_HAS_CONTENT_TYPE", "CACHE_MODE_USE_ORIGIN_HEADERS" ],
"enforcedSecurityPolicy": { "configuredAction": "DENY", "rule_triggered": "HTTP_Request_Flood", "outcome": "DENY", "priority": "45000",
"remoteIp": "192.168.1.100", "securityPolicyRequestData": { "remoteIpInfo": { "regionCode": "CA" },
"tlsJa3Fingerprint": "19e29534fd49dd27d09234e639c4057e", "statusDetails": "denied_by_security_policy",
"logName": "projects/proj1/logs/requests", "receiveTimestamp": "2024-05-11T11:21:41.143384439Z",
"resource": { "labels": { "backend_service_name": "portals-backend", "forwarding_rule_name": "portals-frontend-ip",
"project_id": "proj1", "target_proxy_name": "portals-alb-target-proxy", "url_map_name": "portals-alb", "zone": "global" },
"type": "http_load_balancer", "severity": "WARNING", "spanId": "6f75b78af1aa4636", "timestamp": "2024-04-10T11:21:40.391872Z",
"trace": "projects/proj1/traces/3191b6650ebf56e63973336c4f8de9e0" }
```

Рисунок 4.5 – Приклад логів подій WAF

Крок 3: Експлуатація неправильних конфігурацій. Зловмисник використовує помилки налаштувань елементи хмарного сховища, на яких можуть бути вимкнені контролю безпеки або взагалі інстанси можуть бути залишені відкритими для громадського доступу.

Використані інструменти: Автоматизовані скрипти або ручне переглядання для доступу та дослідження вмісту хмарного сховища. Логи доступу містять спроби несанкціонованого доступу або успішні доступи з невідомих IP-адрес. Це можуть бути дії, такі як запити LIST або GET до наявних бакетів (див. рис. 4.6).

```
2024-04-10T12:35:10Z requestor=anonymous action=LIST bucket=exampleBucket  
2024-04-10T12:35:15Z requestor=anonymous action=GET bucket=exampleBucket object=confidential_data.txt
```

Рисунок 4.6 – Логи доступу до бакетів

Крок 4: Експлуатація вразливостей сервера. Зловмисник на цьому кроці використовує конкретні вразливості, що виявив під час сканування. Серед знайдених вразливостей можуть бути можливість віддаленого виконання коду (RCE) або SQL-ін'єкції (SQLi).

Використані інструменти: Залежно від конкретної вразливості, використовуються спеціалізовані скрипти експлуатації або загальнодоступні інструменти. Логи на ураженому сервері можуть показувати незвичайні процеси, які виконуються, або логи баз даних можуть записувати несанкціоновані SQL-запити, що свідчить про SQLi (див. рис. 4.7).

```
2024-04-10T12:40:00Z app_server_ip=10.0.0.125 severity=ERROR message="Unexpected server error at /api/login - potential exploit attempt"
```

Рисунок 4.7 – Приклад логів SQL-ін'єкції

Крок 5: Отримання доступу та підвищення привілеїв. Використовуючи вже раніше згадану вразливість сервера, зловмисник отримує початковий доступ, а потім намагається підвищити привілеї для отримання більшого контролю над хмаровим середовищем.

Використані інструменти: Інструменти, такі як Metasploit, можуть використовуватися для підвищення привілеїв за допомогою відомих експлойтів. Системні логи при цьому фіксують створення нового облікового запису або зміни до привілеїв існуючих користувачів, корисними будуть також логи від інструментів безпеки, які також відслідковують такі зміни. Приклад таких логів є на рис. 4.8.

```
2024-04-10T12:45:30Z user=admin action=modify_user new_user=malicious_user permission=administrator
```

Рисунок 4.8 – Логи створення користувача

Крок 6: Переміщення для доступу до бази даних. Маючи достатній рівень привілеїв, зловмисник переміщується внутрішньою мережею для доступу до бази даних, що містить конфіденційну інформацію.

Використані інструменти: Спеціалізовані скрипти або ручні команди для доступу до серверів баз даних з компрометованих екземплярів серверів. Логи мережі покажуть незвичайні внутрішні шаблони трафіку, зокрема трафік, направлений на порти баз даних з незвичайних джерел.

```
2024-04-10T12:50:30Z src_ip=10.0.0.125 dst_ip=10.0.0.200 dst_port=3306 protocol=TCP action=allowed
2024-04-10T12:55:00Z user=malicious_user db_query="SELECT * FROM users WHERE username = 'admin' -- '"
2024-04-10T12:56:00Z user=malicious_user db_query="UNION SELECT NULL, username, password FROM admin_users;"
```

Рисунок 4.9 – Приклад логів SQL-ін'єкції

Крок 7: Екстракція даних. Зловмисник вилучає дані з бази даних.

Дані можуть бути вилучені за допомогою протоколів безпечного копіювання або навіть через прямі запити до баз даних для експорту даних до зовнішніх місць зберігання. Логи транзакцій баз даних та логи мережевого трафіку демонструють великі запити даних та дані, що надсилаються на зовнішні IP-адреси.

```
2024-04-10T12:57:30Z src_ip=10.0.0.200 dst_ip=192.168.1.100 dst_port=22 protocol=TCP bytes_transferred=204800 action=allowed
```

Рисунок 4.10 – Приклад логів масованої передачі даних

Після моделювання атаки отримані логи було додано до прототипу системи для подальшого розслідування інциденту. Для цього нами було надіслано запит, де вказано про те, що стався витік даних з вимогою відтворити кроки інциденту для подальшого пошуку й усунення його першопричин.

```

CYBERSECURITY_BOT
Can I run this command "get_logs"? (Y/n):y
Running command on database server: get_logs
Getting logs...
Based on the event logs, the root cause analysis of the data breach is as follows:

Overview
This investigation analyzes a data breach incident in the cloud environment involving unauthorized access
to sensitive data and privilege escalation. The incident occurred on April 10, 2024, and the logs highlight several stages of the attack,
including reconnaissance, exploitation, data exfiltration, and privilege escalation.
This report will outline the root cause and provide an incident response plan.

Stages of the Attack
1. Initial Reconnaissance:
Logs:
2024-04-10T12:30:45Z - source_ip=192.168.1.100 sends a GET request to http://example.cloudservice.com.
2024-04-10T12:30:47Z - Same IP sends a GET request to http://example.cloudservice.com/api.
Analysis:
The requests from 192.168.1.100 demonstrate an initial exploration of the API endpoints of the cloud service.

2. Port Scanning and Attack Execution:
Logs:
2024-04-10T12:32:22Z - source_ip=192.168.1.100 initiates a port scan (blocked by a rule).
2024-04-10T12:32:30Z - An HTTP_Request_Flood attack is allowed from the same source IP.
Analysis:
The port scan attempt and subsequent HTTP flood indicate a probing attack to find vulnerabilities.

3. Unauthorized Data Access:
Logs:
2024-04-10T12:35:10Z - An anonymous user lists the contents of exampleBucket.
2024-04-10T12:35:15Z - The same anonymous user retrieves confidential_data.txt.
Analysis:
A misconfigured S3 bucket allowed anonymous listing and downloading of confidential data.

4. Server Exploitation:
Logs:
2024-04-10T12:40:00Z - app_server_ip=10.0.0.125 logs an error indicating an "unexpected server error at
/api/login - potential exploit attempt detected."
Analysis:
An exploitation attempt was detected, possibly indicating a vulnerability in the API or application.

5. Privilege Escalation:
Logs:
2024-04-10T12:45:30Z - admin user modifies the user list to add a new malicious_user with administrator privileges.
Analysis:
Credentials or session data for the admin account were compromised, enabling a privilege escalation attack.

6. Database Access and Data Exfiltration:
Logs:
2024-04-10T12:55:00Z - malicious_user performs an SQL injection on the database (SELECT * FROM users).
2024-04-10T12:56:00Z - Further SQL injection queries are executed.
2024-04-10T12:57:30Z - Data is transferred from the internal IP 10.0.0.200 to 192.168.1.100 over port 22.
Analysis:
The SQL injection reveals user data and grants unauthorized access to administrator accounts, leading to data exfiltration.

Root Cause
1. Misconfigured Access Permissions:
Publicly accessible S3 bucket allowed unauthorized access to sensitive data.
2. Vulnerable Server Application:
An API vulnerability or misconfiguration led to unauthorized server errors and potential credential compromise.
3. SQL Injection Vulnerability:
Poorly secured SQL queries enabled attackers to escalate their privileges through the database.

```

Рисунок 4.11 – Результат виконання програми

На рисунку 4.11 зображено вивід програми, де покроково вказано етапи здійсненої атаки разом з деталями, що необхідні для подальшого формування звіту про інцидент. Як бачимо, система повністю відтворила кожен крок, тож результат виконання відмінний.

Наступним завданням було встановити якими були причини інциденту, можливі помилки налаштувань, що потенційно дозволили зловмиснику виконати цю атаку. Для цього програмою було отримано віддалений доступ до серверу через SSH та проведено сканування конфігурацій, що потенційно дозволили виконати атаку (див. рис. 4.12).

CYBERSECURITY_BOT

Based on the events and logs, here are the server settings that could have led to this breach:

1. Inadequate Access Controls:

- Improperly configured user permissions and roles, allowing unauthorized users to access sensitive data and perform administrative actions.

2. Weak Network Security Configuration:

- Open or poorly configured firewall rules, allowing unauthorized traffic to reach the server and bypassing security measures.
- Lack of intrusion detection and prevention systems to detect and block suspicious network activities.

3. Insecure Bucket Permissions:

- Misconfigured access control lists (ACLs) for the cloud storage bucket, allowing anonymous or unauthorized users to list and access confidential data.

4. Vulnerable Application Configuration:

- Outdated or unpatched application software with known vulnerabilities, making it susceptible to exploit attempts and unexpected errors.
- Insecure application settings, such as weak authentication mechanisms or lack of input validation, leading to potential exploit attempts.

5. Insecure Database Configuration:

- Weak or default database credentials, allowing unauthorized access to the database server.
- Lack of input validation and prepared statements in database queries, making the application vulnerable to SQL injection attacks.

These potential server settings highlight the importance of implementing robust security measures, such as proper access controls, network security configurations, secure bucket permissions, application hardening, and secure database configurations to mitigate the risk of a breach.

Рисунок 4.12 – Результат пошуку помилок конфігурацій

Вивід програми при такому запиті також показав хороші результати.

Відповідно до отриманих результатів, було розроблено рекомендації щодо ліквідації наслідків атаки та вдосконалення помилкових налаштувань, що стали першопричиною інциденту.

Як бачимо на рис. 4.13, програмою було сформовано план реагування на інцидент. План подано у вигляді покрокових команд, покликаних розірвати доступ злоумисника та загалом покращити систему безпеки.

```

CYBERSECURITY_BOT
Here are some specific commands that can be executed to enhance security as part of the incident response:

1. Revoke Compromised User Credentials:
# Lock admin and malicious_user accounts
sudo passwd -l admin
sudo passwd -l malicious_user

2. Reset All User Passwords:
# Prompt for password resets for all users (using a secure password policy)
sudo passwd -e $(awk -F: '{ print $1 }' /etc/passwd)

3. Firewall Configuration:
# Block Suspicious IPs (e.g., 192.168.1.100):
sudo iptables -A INPUT -s 192.168.1.100 -j DROP
# Allow Only Trusted IP Ranges:
sudo iptables -A INPUT -p tcp --dport 22 -s 10.0.0.0/24 -j ACCEPT
sudo iptables -A INPUT -p tcp --dport 22 -j DROP

4. Storage Permissions:
# Deny public access to sensitive buckets
aws s3api put-bucket-acl --bucket exampleBucket --acl private

5. Audit and Monitor Suspicious Access:
# Enable logging for key system services
sudo systemctl enable auditd
sudo systemctl start auditd
# Configure audit rules (e.g., track file access and modifications)
echo "-w /etc/passwd -p wa -k passwd_changes" | sudo tee -a /etc/audit/rules.d/audit.rules

6. Add Monitoring Alerts:
# Configure fail2ban to block multiple failed login attempts
sudo apt install fail2ban
echo -e "[sshd]\nenabled = true\nport = 22\nfilter = sshd\nlogpath = /var/log/auth.log\nmaxretry = 5\nbantime = 600"
| sudo tee /etc/fail2ban/jail.d/sshd.conf
sudo systemctl restart fail2ban

7. Secure Database Configuration:
# Allow access only from specific IP ranges
GRANT ALL PRIVILEGES ON *.* TO 'admin_user'@'10.0.0.%' IDENTIFIED BY 'secure_password';
REVOKE ALL PRIVILEGES ON *.* FROM 'admin_user'@'%';
# Enable Logging for SQL Queries: (MySQL/MariaDB logging example)
SET global general_log = 1;

8. Application Hardening:
# Ensure all packages are up-to-date
sudo apt update && sudo apt upgrade -y

These commands will help directly address security issues, revoke compromised credentials, limit unauthorized access,
and improve visibility into future security events.

```

Рисунок 4.13 – Рекомендації плану реагування на інцидент

Таким чином, програма, яку було реалізовано посилаючись на запропоновану в цій роботі модель, дозволила успішно виконати усі кроки розслідування інциденту.

4.4 Перспективи розвитку та адаптивне застосування моделі

Наукові дослідження підкреслюють високу популярність використання LLM моделей як базових автономних агентів у більш масштабних системах для виконання певних функцій [75, 76].

Запропонована модель на базі LLM може бути дієво інтегрована в SIEM у вигляді агента для розширення їхніх можливостей. SIEM системи призначені для забезпечення цілісного огляду інформаційної безпеки організації шляхом збору, аналізу та представлення даних безпеки з різних джерел у режимі реального часу. Прототип реалізації моделі, що було запропоновано в даній роботі, у свою чергу, має вигляд чат-боту для операцій з безпеки, підтримуваний LLM. Це допоможе спростити інтерфейс взаємодії між SIEM і аналітиками безпеки за допомогою запитів і пояснень природною мовою, полегшуючи користувачам взаємодію з системою та витяг значущої інформації на основі розробленого боту

Такі агенти, наприклад, можуть надавати цілодобову допомогу командам з безпеки, забезпечуючи швидкий доступ до інформації у зрозумілому вигляді, порад щодо вирішення інцидентів та підтримку у рутинних завданнях. Таким чином, запропонована система може спростити аналітичні процеси та допомогти спеціалістам з безпеки зосередитися на глобальніших стратегічних завданнях.

Крім цього, запропоновану модель можна потенційно адаптувати, використовуючи запропоновану модель у рамках систем управління інцидентами у декількох напрямках.

1. Модель може сприяти автоматизації дій реагування, надаючи підтримку прийняття рішень, генеруючи сценарії або автоматизуючи зв'язок з іншими інструментами безпеки в рамках Security Orchestration Automation and Response (SOAR), що прискорює процес зниження ризиків. Так, розроблена

система надає пропозиції щодо реагування на інцидент, засновані на аналізі безпекових логів. На базі цього, бот (використовуючи здатність виклику функцій) може автоматизувати частини процесу реагування на інциденти, генеруючи сценарії або робочі скрипти для автоматизації процесів на основі конкретних характеристик виявленої загрози, знижуючи навантаження на прийняття рішень аналітиками.

Наведемо приклад: SIEM було виявлено спробу фішингу, бот може автоматично підготувати проект повідомлення для IT-відділу або постраждалих користувачів, повідомляючи їх про загрозу та пропонуючи профілактичні заходи.

2. LLM будуть корисними для автоматизованої генерації докладних індивідуалізованих звітів про безпеку для різних зацікавлених сторін (в тому числі для нетехнічних спеціалістів) шляхом узагальнення ключових даних про інциденти, стан системи, додаткових висновків із SIEM та інших інструментів. Це відчутно заощаджує час для команд безпеки та гарантує, що звіти будуть вичерпними та зрозумілими.

3. Створення налаштованих навчальних програм безпеки та симуляцій на основі конкретних загроз і вразливостей, з якими стикається організація. Аналізуючи минулі інциденти та поточні тенденції, такий бот стане у нагоді при розробці реалістичних сценаріїв навчання, які є актуальними та ефективними для підвищення обізнаності співробітників щодо безпеки та перевірки процедур реагування на загрози.

Процес інтеграції з SEIM в такому разі буде складатись з наступних етапів:

1. Розробка API: необхідно розробити API для інтеграції налаштованої моделі з SIEM. Цей інтерфейс повинен дозволити SIEM надсилати дані в модель і отримувати її результати.

2. Інтеграція робочого процесу: необхідно визначити, як саме аналітична інформація моделі використовується в робочому процесі SIEM. Це може включати автоматизацію певних відповідей, збагачення сповіщень додатковою інформацією або надання рекомендацій для аналітиків. Важливо забезпечити безперебійну

інтеграцію з існуючими інструментами та робочими процесами SIEM, щоб уникнути перерв та максимізувати переваги такого підходу.

При цьому також потрібно враховувати необхідність постійного контролю продуктивності моделі, щоб переконатися, що вона відповідає очікуваним стандартам (відстеження помилкових спрацювань і негативів, час відповіді та точність), а також регулярного оновлення моделі новими даними, щоб забезпечити її ефективність у разі появи нових типів загроз. Це може включати періодичне перенавчання або точкове налаштування за допомогою оновлених наборів даних, адже такі моделі можуть потребувати навчання на специфічних організаційних даних для кращого розуміння контексту та підвищення точності згенерованих результатів.

Виходячи з розглянутих раніше потенційних недоліків LLM (див. 2.4), інтегруючи модель на базі машинного навчання в SIEM, варто переконатись у дотриманні норм конфіденційності даних, особливо під час обробки інформації з обмеженим доступом, передбачити захист моделі від атак, які можуть маніпулювати її результатами або порушити її цілісність та переконатись, що інтеграція може масштабуватися відповідно до обсягу даних і складності подій безпеки, які вона потребує для обробки.

Наразі SIEM вже розпочали інтегрувати елементи штучного інтелекту у своїх системах. Наприклад, Elastic AI Assistant, генеративний помічник штучного інтелекту на базі Elasticsearch Relevance Engine (ESRE) надає нові можливості для створення високорелевантних пошукових програм [77]. Він доступний через простий уніфікований API та здатний надавати рекомендації щодо створення запитів, інтеграції агентів та розслідування спрацювань на підставі заготовлених підказок відповідно до наявного контексту специфічної для організації інформації, яка адаптує відповідь до конкретних проблем. Elastic AI Assistant має високу популярність серед користувачів ELK, виконуючи роль персонального помічника. Проте його функціонал дозволяє надавати тільки узагальнені результати.

Отже, інтеграція LLM у системи SIEM є значним кроком вперед у тому, щоб зробити операції з кібербезпеки більш інтелектуальними, ефективними та

проактивними. Однак важливо продумано керувати інтеграцією, враховуючи виклики хмари і проблеми використання такої моделі й переконавшись, що рекомендації штучного інтелекту обов'язково перевіряються та підтверджуються досвідченими фахівцями з безпеки.

Висновки за розділом 4

У цьому розділі було досліджено та перевірено функціональність моделі системи розслідування інцидентів у хмарному середовищі, створеної на основі LLM. Щоб підтвердити доцільність застосування генеративного штучного інтелекту з цією метою, було розроблено прототип системи у вигляді інтерактивного чат-бота для аналітиків з безпеки, що використовує GPT-4 для аналізу даних та виконання команд на серверах. Програма прототипу включає головний модуль, API для взаємодії з GPT-4 та утилітарний модуль для виконання команд на серверах.

Дієвість моделі було перевірено, застосувавши його для розслідування змодельованої атаки на вразливі хмарні системні ресурси. Програма відмінно виконала усі поставлені завдання – на основі проаналізованих даних відтворила етапи атаки, ідентифікувала першопричини інциденту та надала список команд, згенерувавши рекомендації щодо усунення наслідків інциденту та помилок конфігурації. Тому можна зробити висновок, що запропонована модель системи здатна допомогти аналітикам здійснювати управління інцидентами.

У розділі також наведено, що запропонований прототип має широкі можливості для інтеграції з іншими системами, наприклад в SIEM. LLM можуть покращити автоматизацію реагування на інциденти, автоматизувати звітність та документацію, а також сприяти навчанню з кібербезпеки.

Загалом, результати виконання роботи, викладені в четвертому розділі демонструють, що розроблена система здатна дієво розслідувати інциденти у хмарному середовищі, використовуючи машинне навчання для автоматизації та пришвидшення процесу розслідування.

ВИСНОВОК

Під час виконання кваліфікаційної роботи було вирішено завдання щодо розробки моделі для дієвої системи розслідування інцидентів у хмарному середовищі.

В ході вирішення поставлених завдань було досягнуто наступних результатів:

1. Проведено детальний огляд міжнародних стандартів і нормативно-правових документів України у сфері управління інцидентами в хмарних обчисленнях.

Здійснений аналіз показав певну нормативну лауну у цій сфері, адже більшість стандартів містить узагальнені рекомендації до процесу, які не враховують специфічні для хмарної інфраструктури виклики.

Зважаючи на стрімке поширення використання хмарних сервісів, особливо у держаних організаціях, які слідують стратегії Cloud First, потреба в окресленні та регулюванні вимог до процесу розслідування інцидентів у хмарі залишається відкритою.

2. Досліджено основні виклики, специфічні для хмарного середовища, включаючи технічні, організаційні та юридичні аспекти, які впливають на розслідування інцидентів.

Особлива увага приділена забезпеченню цілісності даних та ланцюгу збереження доказів у розслідуваннях, що вимагає ретельної організації архітектури хмарних сервісів і використання передових технологій шифрування та захисту інформації, орієнтуючись на запропоновану модель збору доказів.

3. На основі розглянутих викликів, які притаманні для хмарної форензики, було сформовано рекомендації для кожного етапу цього процесу при проведенні розслідування інцидентів у хмарному середовищі.

Надані рекомендації враховують вимоги до встановлення автоматизованих систем спостереження, інтегрованих інструментів логування, а також

спеціалізованих інструментів шифрування разом із централізованим сховищем, що забезпечить надійність зібраних даних, на основі яких буде проводитись подальше розслідування.

Аналіз інцидентів повинен проводитись з допомогою передових аналітичних інструментів, зокрема з використанням можливостей штучного інтелекту для виявлення закономірностей у даних.

Крім цього, генеративні можливості штучного інтелекту можуть застосовуватись і для належного автоматизованого звітування, яке повинно бути структурованим, доступним для розуміння (в тому числі для не обізнаних технічно осіб) і відповідати нормам, прийнятим у юрисдикції, де ведеться розслідування, щоб забезпечити їх придатність для подальшого розгляду (при судових розслідуваннях зокрема).

4. Проведено аналіз існуючих інструментів, що можуть бути використані для розслідування хмарних інцидентів. Зроблений аналіз дає змогу зробити висновок, що наразі цій сфері бракує наявності універсального інструменту, що буде здатний здійснювати повноцінні комплексні розслідування інцидентів у хмарі.

5. Зважаючи на попередні результати дослідження, було розроблено прототип моделі пошуку першопричин інцидентів на основі GPT-4 та протестовано її функціонал на прикладі атаки з витоків даних. Ця система дозволяє розслідувати інциденти в хмарі, генеруючи покроковий опис подій, що призвели до атаки, здійснювати аналіз помилок конфігурацій, які були потенційно використані злоумисником та створює план для реагування на інциденти й усунення наявних вразливостей системи. Створена модель має великий простір для адаптивного застосування, проте потребує обов'язкового підтвердження наданих результатів аналітиком.

Таким чином, поставлені завдання виконано в повному обсязі, що дозволило досягти мети роботи та створити дієвий інструмент, що може бути інтегрований до загальної системи розслідування інцидентів у хмарі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Cloud Computing Market [Електронний ресурс]. – Режим доступу: <https://www.precedenceresearch.com/cloud-computing-market>
2. Gartner Says Cloud Will Become a Business Necessity by 2028 [Електронний ресурс]. – Режим доступу: <https://www.gartner.com/en/newsroom/press-releases/2023-11-29-gartner-says-cloud-will-become-a-business-necessity-by-2028#:~:text=Global%20Public%20Cloud%20Services%20Spending,services%20continues%20to%20rise%20unabated.>
3. NIST Special Publication 800-145, The NIST Definition of Cloud Computing [Електронний ресурс] – Режим доступу: <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>.
4. Про хмарні послуги [Електронний ресурс]: Закон України від 04.04.2024 № 2075-IX. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/2075-20#n69>.
5. Most common security incidents in the cloud and on-premises worldwide in 2023 [Електронний ресурс]. – Режим доступу: <https://www.statista.com/statistics/1320178/common-cloud-security-attacks-worldwide/#statisticContainer>.
6. Cloud Ransomware | Understanding And Combating This Evolving Threat [Електронний ресурс]. – Режим доступу: <https://www.sentinelone.com/cybersecurity-101/cloud-ransomware-understanding-and-combating-this-evolving-threat/>.
7. The Google Cloud Threat Horizons Report [Електронний ресурс]. – Режим доступу: https://services.google.com/fh/files/misc/threat_horizons_report_h12024.pdf.
8. Top Threats to Cloud Computing: Pandemic Eleven [Електронний ресурс] // CSA, 10.17.2023. – Режим доступу: <https://cloudsecurityalliance.org/research/topics/top-threats>.

9. The Top 5 Cloud Security Threats to Be Prepared For in 2023 [Электронный ресурс]. – Режим доступа: <https://orca.security/resources/blog/top-5-cloud-security-threats-2023/>.

10. Mitigating Cloud Vulnerabilities [Электронный ресурс] // NSA, 22.01.2020. – Режим доступа: https://media.defense.gov/2020/Jan/22/2002237484/-1/-1/0/CSI-MITIGATING-CLOUD-VULNERABILITIES_20200121.PDF.

11. Check Point Software's 2023 Cyber Security Report [Электронный ресурс]. – Режим доступа: <https://pages.checkpoint.com/cyber-security-report-2023.html>.

12. Why are cloud misconfigurations still a major issue? [Электронный ресурс]. – Режим доступа: <https://securityintelligence.com/articles/why-cloud-misconfigurations-major-issue/>.

13. Матрица Cloud Controls i CAIQ v4 [Электронный ресурс] //CSA, 07.06.2021. – Режим доступа: <https://cloudsecurityalliance.org/artifacts/cloud-controls-matrix-v4/> [Электронный ресурс]. – Режим доступа: <https://cloudsecurityalliance.org/research/cloud-controls-matrix>.

14. Cloud Incident Response Framework [Электронный ресурс] //CSA, 05.04.2021. – Режим доступа: <https://cloudsecurityalliance.org/artifacts/cloud-incident-response-framework>.

15. General Data Protection Regulation (GDPR) [Электронный ресурс] – Режим доступа: <https://gdpr-info.eu/>.

16. ISO/IEC 27001:2022. Information technology — Security techniques — Information security management systems — Requirements [Электронный ресурс]. – Режим доступа: <https://www.iso.org/isoiec-27001-information-security.html>.

17. Egho-Promise, E. Digital Forensic Investigation Standards in Cloud Computing [Electronic resource] / Egho-Promise, E., Idahosa, S., Asante, G., & Okungbowa, A. // Universal Journal of Computer Sciences and Communications, 3(1). – 2024. 23–45 pp. – Access: <https://www.scipublications.com/journal/index.php/ujcsc/article/view/923>.

18. ISO/IEC 27017. Information technology — Security techniques — Code of practice for information security controls based on ISO/IEC 27002 for cloud services [Электронный ресурс]. – Режим доступа: <https://www.iso.org/standard/43757.html>.

19. ISO/IEC 27018. Information technology — Security techniques — Code of practice for protection of personally identifiable information (PII) in public clouds acting as PII processors [Электронный ресурс]. – Режим доступа: <https://www.iso.org/standard/76559.html>.

20. ISO/IEC 27035. Information technology — Information security incident management — Part 2: Guidelines to plan and prepare for incident response [Электронный ресурс]. – Режим доступа: <https://www.iso.org/standard/78974.html>

21. ISO/IEC 27037. Information technology — Security techniques — Guidelines for identification, collection, acquisition and preservation of digital evidence [Электронный ресурс]. – Режим доступа: <https://www.iso.org/standard/44381.html>.

22. ISO/IEC 27041. Information technology — Security techniques — Guidance on assuring suitability and adequacy of incident investigative method [Электронный ресурс]. – Режим доступа: <https://www.iso.org/standard/44405.html>.

23. ISO/IEC 27042. Information technology — Security techniques — Guidelines for the analysis and interpretation of digital evidence [Электронный ресурс]. – Режим доступа: <https://www.iso.org/standard/44406.html>.

24. ISO/IEC 27043. Information technology — Security techniques — Incident investigation principles and processes [Электронный ресурс]. – Режим доступа: <https://www.iso.org/standard/44407.html>.

25. NIST Special Publication 800-53, Security and Privacy Controls for Information Systems and Organizations [Электронный ресурс] – Режим доступа: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf>.

26. NIST Special Publication 800-144, Guidelines on Security and Privacy in Public Cloud Computing [Электронный ресурс] – Режим доступа: <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-144.pdf>.

27. NIST Special Publication 500-291, NIST Cloud Computing Standards Roadmap [Електронний ресурс] – Режим доступу: <https://www.nist.gov/document/nistsp-500-291jul5apdf>.
28. NIST Special Publication 800-61, Computer Security Incident Handling Guide [Електронний ресурс] – Режим доступу: <https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-61r2.pdf>.
29. NIST Special Publication 800-86 - Guide to Integrating Forensic Techniques into Incident Response [Електронний ресурс] – Режим доступу: <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-86.pdf>
30. Про захист персональних даних [Електронний ресурс]: Закон України від 27.04.2024 № 2297-VI. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/2297-17#Text>.
31. Про інформацію [Електронний ресурс]: Закон України від 27.07.2023 № 2657-XII. – Режим доступу: <https://zakon.rada.gov.ua/laws/main/2657-12#Text>.
32. Про основні засади забезпечення кібербезпеки України [Електронний ресурс]: Закон України від 04.04.2024 № 2163-VIII. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/2163-19#Text>.
33. Cloud Incident Response Management: Benefits, Challenges, Framework Steps, Trends, and More [Електронний ресурс]. – Режим доступу: <https://nordicdefender.com/blog/cloud-incident-response-management-guide>
34. NISTIR 8006, NIST Cloud Computing Forensic Science Challenges [Електронний ресурс] – Режим доступу: https://csrc.nist.gov/CSRC/media/Publications/nistir/8006/draft/documents/draft_nistir_8006.pdf.
35. Exploring Cloud Incidents [Електронний ресурс] //ENISA, 01.06.2016. – Режим доступу: <https://www.semanticscholar.org/paper/Exploring-Cloud-Incidents/c566b0c49d596a7d362a70b3196d34bf711ebf3d>.
36. Sonone, P. Cloud Forensics: Introduction and Challenges in Research / Sonone P., Chauragade, P. // International Journal of Scientific & Engineering Research, 7(2) 2016 81-90 pp.

37. NIST Special Publication 800-201, NIST Cloud Computing Forensic 4 Reference Architecture [Электронный ресурс] – Режим доступа: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-201.ipd.pdf>.
38. Akter, S. Cloud Forensic: Issues, Challenges and Solution Models / Akter, S., Rahman, M. // arXiv 2023 2303.06313.
39. Malik, A., Cloud Digital Forensics: Beyond Tools, Techniques, and Challenges / Malik, A., Bhatti, D., Park, T., Ishtiaq H., Ryou J., Kim K. // Sensors 24(2) 2024 443p.
40. NIST 2018. Computer Forensics Tool Catalog [Электронный ресурс] – Режим доступа: <https://toolcatalog.nist.gov>.
41. Mohammed, S. The cloud forensics frameworks and tools: A brief review / Mohammed, S., Rangu S. // International Journal of Science and Research Archive, 8(1) 2023 173-181 pp. https://www.researchgate.net/publication/367543122_The_cloud_forensics_frameworks_and_tools_A_brief_review.
42. Mandal, P. Cloud Forensics: Exploring the Challenges and Mapping Out Solutions for the Future / Mandal, P., Rajput I. // JITRI, 8(4) 2023.
43. Toliupa, S. Enhancing Cloud Incident Forensics: The Role of Large Language Models in Investigative Processes / Toliupa, S., A. Torchylo, Y. Shestak // VII International Conference on Cyber Security Problems of Information and Telecommunication Systems (PCSITS) 2024 135-137 pp.
44. Simou, S. A framework for designing cloud forensic-enabled services (CFeS) / Simou, S., Kalloniatis, C., Gritzalis, S. and Katos, V. // Requirements Engineering, 24(3) – 2022 403–430 pp.
45. Simou, S. Revised forensic framework validation and cloud forensic readiness / Simou S., Kalloniatis C., Gratzalis S., Katos V. and Psalidas M. // International Journal of Electronic Governance, 14(1-2) 2022 236-263 pp.
46. Zawoad, S. OCF: An Open Cloud Forensics Model for Reliable Digital Forensics / Zawoad, S., Hasan, R., & Skjellum, A. // IEEE 8th International Conference on Cloud Computing 2015 437-444 pp.

47. Zhao, W. X. A survey of large language models / Zhao, W. X., K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong et al. // arXiv 2303.18223 2023.
48. Raiaan, M.A. A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges / Raiaan, M.A., Mukta, M.S., Fatema, K., Fahad, N.M., Sakib, S., Mim, M.M., Ahmad, J., Ali, M.E., & Azam, S. // IEEE Access, 12 2024 26839-26874.
49. Ma, Z. LLMParse: An Exploratory Study on Using Large Language Models for Log Parsing / Ma Z., Chen A., Kim D., Chen T., Wang S. // arXiv 2024 2404.18001.
50. Amatriain, X. Transformer models: an introduction and catalog / Amatriain X., A. Sankar, J. Bing, P. K. Bodigutla, T. J. Hazen, and M. Kazi // arXiv 2023 2302.07730.
51. Vaswani, A. Attention is all you need. / Vaswani A., Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin // Advances in neural information processing systems, 30 2017.
52. Raiaan, M. A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges / Raiaan M., Hossain M., Fatema K., Fahad M. et al. // TechRxiv 2023 24171183.
53. Devlin, J. Bert: Pre-training of deep bidirectional transformers for language understanding / Devlin J., M.-W. Chang, K. Lee, and K. Toutanova // arXiv 2018 1810.04805.
54. Liu, Y. Roberta: A robustly optimized bert pretraining approach / Liu Y., Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, LukeZettlemoyer, and Veselin Stoyanov // arXiv 2019 - 1907.11692.
55. Sanh, V. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter / Sanh V., Debut L., Chaumond J and Wolf T. // arXiv 2019 1910.01108.
56. Bhanage. D. Robust Analysis of IT Infrastructure's Log Data with BERT Language Model / Bhanage D., Pawar A. // International Journal of Advanced Computer Science and Applications 14(6) 2023.

57. Radford, A. Improving language understanding by generative pre-training / Radford A., K. Narasimhan, T. Salimans, I. Sutskever et al. // OpenAI 2018.
58. Radford, A. Language models are unsupervised multitask learners / Radford A., J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever et al // OpenAI blog 1(8). 2019 9 p.
59. Aitken, K. Understanding how encoder-decoder architectures attend / Aitken K., Ramasesh V., Cao Y. and Maheswaranathan N. // arXiv 2021 2110.15253.
60. Ni, J. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models / Ni J., Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B Hall, Daniel Cer, and Yinfei Yang // arXiv 2021 2108.08877.
61. Lewis, M. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension / Lewis M., Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer // arXiv 2019 1910.13461.
62. Floridi, L. Gpt-3: Its nature, scope, limits, and consequences / Floridi L., Chiriatti M. // Minds and Machines 30 2020 681–694 pp.
63. Bubeck, S. Sparks of Artificial General Intelligence: Early experiments with GPT-4 / Bubeck S., Chandrasekaran V., Eldan R., Gehrke J., Horvitz E. et al // arXiv 2023 2303.12712.
64. Minaee, S. Large Language Models: A Survey / Minaee S., Mikolov T., Nikzad N., Chenaglu M. et al // arXiv 2024 2402.06196.
65. GPT-4 Technical Report [Электронный ресурс] // OpenAI, 04.03.2024. – Режим доступа: <https://arxiv.org/pdf/2303.08774v3.pdf>.
66. Lee, M. A Mathematical Investigation of Hallucination and Creativity in GPT Models / Lee M. // Mathematics 11(10) 2023 2320.
67. Top five large language model benefits and why they matter for your business [Электронный ресурс] – Режим доступа: <https://indatalabs.com/blog/large-language-model-benefits>.

68. Kaddour, J. Challenges and applications of large language models / Kaddour J., J. Harris, M. Mozes, H. Bradley, R. Raileanu, and R. McHardy, // arXiv 2023 2307.10169.
69. Why Python is the programming language of choice for AI developers [Электронный ресурс] – Режим доступа: <https://www.itpro.com/software/development/why-python-is-the-programming-language-of-choice-for-ai-developers>.
70. GPT-4 Turbo Preview: Exploring the 128k Context Window [Электронный ресурс] – Режим доступа: <https://povio.com/blog/gpt-4-turbo-preview-exploring-the-128k-context-window/>.
71. Function calling [Электронный ресурс] – Режим доступа: <https://platform.openai.com/docs/guides/function-calling>.
72. Wu, T. Promptchainer: Chaining large language model prompts through visual programming / T. Wu, E. Jiang, A. Donsbach, J. Gray, A. Molina, M. Terry, and C. J. Cai, // arXiv 2022 2203.06566.
73. Ekin, S. Prompt Engineering For ChatGPT: A Quick Guide To Techniques, Tips, And Best Practices / Ekin S. // TechRxiv. 2023.
74. Le, V. Log Parsing: How Far Can ChatGPT Go? / Van-Hoang Le, Hongyu Zhang //arXiv. – 2023 2306.01590v2.
75. Xi, Z. The rise and potential of large language model based agents: A survey / Z. Xi, W. Chen, X. Guo, W. He, Y. Ding, B. Hong, M. Zhang, J. Wang, S. Jin, E. Zhou et al // arXiv. 2023 :2309.07864.
76. Wang, L. A survey on large language model based autonomous agents / L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin et al.// arXiv. – 2023 2308.11432.
77. Elastic introduces Elastic AI Assistant Elastic [Электронный ресурс] – Режим доступа: <https://www.elastic.co/blog/introducing-elastic-ai-assistant>.

ДОДАТОК А

СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Статті в іноземних виданнях

1. Y. Shestak, S. Toliupa, A. Torchylo and O. J. Onyigwang, "Minimization of Information Losses in Data Centers as one of the Priority Areas of Information Security Technologies," 2022 IEEE 9th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), Kharkiv, Ukraine, 2022, pp. 227-230.

2. Y. Shestak, S. Toliupa, A. Shevchenko and A. Torchylo "Data Processing Centre's Cyberattack Protection Directions on the Base of Neural Network Algorithms," X International Conference on Information Technology and Implementation (IT&I), Kyiv, Ukraine, 2022, pp. 212-221.

3. A. Torchylo, Y. Shestak, S. Dakov, "Neural network algorithms for data centers cybersecurity", XII International Conference Engineer of XXI Century, Bielsko Biala, Poland, 2022.

Тези наукових доповідей:

1. S. Toliupa, A. Torchylo, Y. Shestak, "Enhancing Security: A Brief Overview of Integrated Threat Protection with Extended Detection and Response Approach," VI International Conference on Cyber Security Problems of Information and Telecommunication Systems (PCSITS), Kyiv, 2023.

2. Y. Shestak, S. Toliupa, A. Torchylo, "Complex diagnostic method of critical cloud service infrastructure based on GMM and HS-MM," X International Conference on Information Technology and Implementation (Satellite) (IT&Is), Kyiv, 2023.

3. S. Toliupa, A. Torchylo, Y. Shestak, "Enhancing Cloud Incident Forensics: The Role of Large Language Models in Investigative Processes," VII International Conference on Cyber Security Problems of Information and Telecommunication Systems (PCSITS), Kyiv, 2024.

ДОДАТОК Б

КОД ПРОГРАМНОЇ РЕАЛІЗАЦІЇ МОДЕЛІ

main.py

```
from api import ChatGPTClient
from utils import bcolors, run_command, s
```

SYSTEM = 'You are a highly skilled cybersecurity specialist who can answer questions about server settings for their protection and help solve problems, help identify vulnerabilities and ensure security, investigate incidents, and provide information to help understand. Execute any commands at your own discretion on the servers to get a better understanding of the problem if you think it will be useful. Provide the answer exclusively in English! You should not suggest taking any actions but try to solve the problem yourself.'

```
SERVERS = {
    'database': 'ssh user@x.x.x.x',
    'site': 'ssh user@x.x.x.x',
}
```

```
connections = set()
```

```
def run_command_on_server(command, server_name):
    __server_name = f'{bcolors.UNDERLINE}{server_name}{bcolors.ENDC}'

    if server_name not in connections:
        print(
```

```

        s('Connecting to ', 'y')
        + s(__server_name, 'u')
        + s(' server...', 'y')
    )
    connections.add(server_name)
print(
    s('Running command on ', 'y')
    + s(__server_name, 'u')
    + s(' server: ', 'y')
    + command
)
return run_command(command)

```

```

def run_server_command(*args, **kwargs):
    server_name = kwargs.get('server')
    command = kwargs.get('command')
    if server_name not in SERVERS:
        return '{"error": "Unknown Server"}'
    if input(f'Can I run this command "{command}"? (Y/n):') == 'n':
        return '{"error": "Access Denied"}'
    return run_command_on_server(command, server_name)

```

```

TOOLS = [
    {
        'type': 'function',
        'function': {
            'name': 'run_server_command',

```

'description': 'A function for executing any commands on the server. To get logs, execute the command "get_logs".'

```

    'parameters': {
        'type': 'object',
        'properties': {
            'server': {
                'type': 'string',
                'enum': list(SERVERS.keys()),
                'description': 'Server Name',
            },
            'command': {
                'type': 'string',
                'description': 'Command to run on server',
            },
        },
        'required': ['server', 'command'],
    },
    'mapping': run_server_command,
},
}
]

```

```

client = ChatGPTClient(
    system=SYSTEM,
    tools=TOOLS,
    # model='gpt-4-1106-preview',
)
while True:
    print(f'{bcolors.OKGREEN}YOU{bcolors.ENDC}')

```

```
request = input()
if not request:
    break
print()
print(f'{bcolors.FAIL}CYBERSECURITY_BOT{bcolors.ENDC}')
print(client.get_answer(request))
print()
print()
```

ДОДАТОК В**КОД ПРОГРАМНОЇ РЕАЛІЗАЦІЇ МОДЕЛІ****api.py**

```
import json

from openai import APIConnectionError, APIError, OpenAI, RateLimitError

from utils import waiting_animation

class ChatGPTClient:
    def __init__(
        self,
        model='gpt-3.5-turbo-1106',
        system="",
        init_messages=None,
        tools=None,
    ):
        self.client = OpenAI(api_key="xxx")
        self.model = model
        self.tools = tools or []
        self.name_to_func_mapping = {
            tool['function']['name']: tool['function'].pop('mapping')
            for tool in tools
        }
        self.init_messages = [{'role': 'system', 'content': system}]
        if init_messages:
            self.init_messages += init_messages
```

```
self.clear_messages()

def clear_messages(self):
    self.messages = self.init_messages.copy()

def perform_request(self, *args, use_tools=False, **kwargs):
    if use_tools and self.tools:
        kwargs.update(
            {
                'tools': self.tools,
                'tool_choice': 'auto',
            }
        )
    with waiting_animation():
        return self.client.chat.completions.create(
            *args,
            model=self.model,
            messages=self.messages,
            temperature=0.1,
            **kwargs,
        )

def get_answer(self, request) :
    response = None
    self.messages.append({'role': 'user', 'content': request})
    try:
        response = self.perform_request(use_tools=True)
        response_message = response.choices[0].message
        self.messages.append(response_message)
        tool_calls = response_message.tool_calls
```

```
if tool_calls:
    for tool_call in tool_calls:
        function_name = tool_call.function.name
        function_to_call = self.name_to_func_mapping[function_name]
        function_kwargs = json.loads(tool_call.function.arguments)
        function_response = function_to_call(**function_kwargs)
        self.messages.append(
            {
                'tool_call_id': tool_call.id,
                'role': 'tool',
                'name': function_name,
                'content': function_response,
            }
        )
        response = self.perform_request()
        response_message = response.choices[0].message
        self.messages.append(response_message)
except APIConnectionError as e:
    print(f'Failed to connect to OpenAI API: {e}')
except RateLimitError as e:
    print(f'OpenAI API request exceeded rate limit: {e}')
except APIError as e:
    print(f'OpenAI API returned an API Error: {e}')
if response:
    return response.choices[0].message.content
```

ДОДАТОК Г

КОД ПРОГРАМНОЇ РЕАЛІЗАЦІЇ МОДЕЛІ

utils.py

```
import itertools
import subprocess
import sys
import threading
import time
from contextlib import contextmanager
```

```
@contextmanager
```

```
def waiting_animation():
```

```
    done = False
```

```
    def animate():
```

```
        for c in itertools.cycle(['|', '/', '-', '\\']):
```

```
            if done:
```

```
                break
```

```
            sys.stdout.write(f'\r{c}')
```

```
            sys.stdout.flush()
```

```
            time.sleep(0.1)
```

```
        sys.stdout.write('\r')
```

```
t = threading.Thread(target=animate)
```

```
t.start()
```

```
try:
```

```
    yield
```

```
finally:
```

```
    done = True
```

```
    t.join()
```

```
def run_command(command):
```

```
    if command == 'get_logs':
```

```
        print('Getting logs...')
```

```
        with open('<log path>', 'r') as f:
```

```
            return f.read()
```

```
    try:
```

```
        result = subprocess.run(
```

```
            command,
```

```
            shell=True,
```

```
            check=True,
```

```
            stdout=subprocess.PIPE,
```

```
            stderr=subprocess.PIPE,
```

```
            text=True,
```

```
        )
```

```
        return result.stdout
```

```
    except subprocess.CalledProcessError as e:
```

```
        return e.stderr
```

```
class bcolors:
```

```
    HEADER = '\033[95m'
```

```
    OKBLUE = '\033[94m'
```

```
    OKCYAN = '\033[96m'
```

```
    OKGREEN = '\033[92m'
```

```
    WARNING = '\033[93m'
```

```
FAIL = '\033[91m'
```

```
ENDC = '\033[0m'
```

```
BOLD = '\033[1m'
```

```
UNDERLINE = '\033[4m'
```

```
def s(s, c):
```

```
    c = {
```

```
        'r': bcolors.FAIL,
```

```
        'g': bcolors.OKGREEN,
```

```
        'b': bcolors.OKBLUE,
```

```
        'c': bcolors.OKCYAN,
```

```
        'y': bcolors.WARNING,
```

```
        'p': bcolors.HEADER,
```

```
        'u': bcolors.UNDERLINE,
```

```
    }[c]
```

```
    return f'{c}{s}{bcolors.ENDC}'
```