НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАУКОВО-НАВЧАЛЬНИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

«До захисту допущено»
Завідувач кафедри
_____ Дмитро ЛАНДЕ
(підпис)
«___»_____2025 р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Системи, технології та математичні методи кібербезпеки»
спеціальності 125 «Кібербезпека»

на тему: «Модель дерева рішень для процедури кіберрозвідки »

Виконав (-ла): здобувач вищої освіти **IV** курсу, групи ФБ-13
(шифр групи)

Медвецький Давид Вадимович _____
(прізвище, ім'я, по батькові) (підпис)

Керівник Козленко Олег Віталійович , асистент _____
(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові) (підпис)

Рецензент Южакова Ганна Олексіївна, доцент кафедри ММЗІ _____

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище, ім'я, по батькові) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Здобувач вищої освіти _____
(підпис)

Київ – 2025 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

Рівень вищої освіти – перший (бакалаврський)
Спеціальність – 125 «Кібербезпека»
Освітньо-професійна програма «Системи, технології та математичні методи кібербезпеки»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Дмитро ЛАНДЕ
(підпис)
«___»_____2025 р.

ЗАВДАННЯ
на дипломну роботу здобувачу вищої освіти

Медвецького Давида Вадимовича
(прізвище, ім'я, по батькові)

1. Тема роботи: «Модель дерева рішень для процедури кіберрозвідки »

керівник роботи асистент кафедри інформаційної безпеки Козленко О.В..,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від « 26 » травня 2025 р. № 1761-С

2. Термін подання здобувачем вищої освіти роботи: 13 червня 2025 р.

3. Вихідні дані до роботи: літературні джерела, графові бази даних, мови запитів, формати зберігання даних.

4. Зміст роботи: надання теоретичних відомостей, огляд підходу до ідентифікації сутностей і зберігання даних, формування дерев, приклад застосування.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): презентація

6. Дата видачі завдання 23.02.2025

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів дипломної роботи	Примітка
1	Отримання завдання	23.02.2025	Виконано
2	Формулювання мети дослідження та постановка завдань	03.03.2025 – 08.03.2025	Виконано
3	Огляд та аналіз теоретичних джерел за темою роботи	10.03.2025 – 14.03.2025	Виконано
4	Аналіз предметних областей процедур розвідки	14.04.2025 – 18.04.2025	Виконано
5	Підбір інструментів для збереження даних і їх обробки.	21.04.2025 – 25.04.2025	Виконано
6	Збір даних.	28.04.2025 – 17.05.2025	Виконано
7	Ідентифікація сутностей і опис зв’язків між ними.	17.05.2025 – 19.05.2025	Виконано
8	Формування бази даних і візуалізація дерев.	20.05.2025 – 27.05.2025	Виконано
9	Опис отриманих результатів у дипломній роботі	28.05.2025 – 31.05.2025	Виконано
10	Підготовка презентації дипломної роботи	02.06.2025 – 11.06.2025	Виконано
11	Передзахист дипломної роботи	13.06.2025	Виконано
12	Фінальна підготовка до захисту дипломної роботи	14.06.2025 – 19.06.25	Виконано
13	Захист дипломної роботи	20.05.2025	

Здобувач вищої освіти _____ Давид МЕДВЕЦЬКИЙ
(підпис) **(Власне ім’я, ПРІЗВИЩЕ)**

Керівник роботи _____ Олег КОЗЛЕНКО
(підпис) **(Власне ім’я, ПРІЗВИЩЕ)**

РЕФЕРАТ

Обсяг роботи 64 сторінки, 25 ілюстрацій, 4 таблиці, 22 джерела літератури.

Об’єкт дослідження: процедура кіберрозвідки в контексті кібербезпеки.

Предмет дослідження: можливість застосування моделі дерева рішень для формалізації етапів кіберрозвідки з використанням вагових коефіцієнтів та графової бази даних

Мета дослідження: реалізувати дерева рішень для процесів кіберрозвідки з можливістю візуалізації, аналізу та оцінки доцільності певних дій.

Методи дослідження: аналіз літературних джерел, аналіз предметних областей методів розвідки, аналіз структур даних.

Отримані результати: побудовано три дерева рішень для методів кіберрозвідки.

Ключові слова: кіберрозвідка, візуалізація даних, дерево рішень, тип, метод, джерело, алгоритм

ABSTRACT

Volume of work 64 pages, 25 illustrations, 4 tables, 22 sources of literature.

Object of research: the procedure of cyber intelligence in the context of cybersecurity.

Subject of research: the possibility of applying a decision tree model to formalize the stages of cyber intelligence using weighted coefficients and a graph database.

Purpose of the study: to implement decision trees for cyber intelligence processes with support for visualization, analysis, and evaluation of the appropriateness of certain actions.

Research methods: analysis of literary sources, analysis of intelligence methods and domains, data structure analysis.

Obtained results: three decision trees were built for different cyber intelligence methods.

Keywords: cyber threat intelligence, data visualization, decision tree, type, method, source, algorithm

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

APT-група (APT group, Advanced Persistent Threat group) – організоване угруповання хакерів, які мають передові навички та ресурси, часто фінансуються державою чи іншими великими структурами.

CTI (Cyber Threat Intelligence) – процес збору, обробки та аналізу інформації як про існуючі, так і нові загрози, які можуть бути спрямовані на певний об’єкт безпеки.

IPS (Intrusion Prevention System) – система запобігання вторгнень.

IDS (Intrusion Detection System) – система виявлення вторгнень.

SIEM (Security Information and Event Management System) – система управління інформаційною безпекою.

TTP (Tactics, Techniques, Procedures) – опис поведінки акторів загроз.

RDF (Resource Description Framework) – стандарт для опису даних.

JSON (JavaScript Object Notation) – формат зберігання та обміну структурованими даними.

AQL (ArangoDB Query Language) – мова запитів, що використовується у ArangoDB.

ВСТУП

В загальному розумінні, кіберрозвідка (або кібершпіонаж) трактується як несанкціоноване отримання інформації з метою отримання особистої, економічної, воєнної або політичної переваги. Сам термін “кіберрозвідка” походить від англомовного cyber threat intelligence, який набрав популярності приблизно в 2000-их роках, коли ІБ-вендори почали публікувати свої звіти про деякі APT (Advanced Persistent Threat) групи. З плином часу, в умовах зростання кількості кібератак та їхньої складності, з’являється необхідність у розробці ефективних методів збору, аналізу та інтерпретації інформації і кіберрозвідка стає одним з інструментів не тільки нападу, а і захисту.

Актуальність роботи. В нашому сьогоденні актуальність дослідження зумовлена не тільки стрімким розвитком технологій і зростанням бажання ці технології захистити, а й вдосконаленням кіберзагроз, які стають дедалі складнішими і витонченішими. Це вимагає від держав, організацій та окремих осіб постійного вдосконалення засобів захисту інформації. В Україні ситуація з кібербезпекою також досить є актуальною. Згідно з даними Державної служби спеціального зв'язку та захисту інформації України [1], у 2024 році було зафіксовано 4315 кіберінцидентів, що на 69,8% більше порівняно з 2023 роком, коли було зареєстровано 2541 випадок.

Окрім кількісного зростання інцидентів, спостерігається збільшення кількості APT – угруповань. За даними MITRE кількість APT-груп зросла з 78 груп, задокументованих на момент квітня 2019 року [2], до 163 груп станом на 2024 рік [3].

Мета і завдання дослідження. Метою дослідження є побудова дерев рішень для проведення ефективної процедури кіберрозвідки з урахуванням різних її типів. Це дозволить структуровано підходити до збору і аналізу даних, а також визначати достатність інформації для завершення процедури. У ході проведення дослідження, виникають наступні завдання:

1. Визначити основні типи кіберрозвідки, проаналізувати мету і методи проведення кожного.
2. Дослідити можливість використання графових баз даних для збереження та обробки отриманих даних.
3. Визначити вагові коефіцієнти для вузлів та метрики оцінки ефективності процедури кіберрозвідки.
4. Відобразити дерева за допомогою відповідних інструментів.

Об’єкт дослідження. Об’єктом дослідження є процеси кіберрозвідки, зокрема методи збору, аналізу та оцінки інформації, що використовується для виявлення кіберзагроз.

Предмет дослідження. Предметом дослідження є побудова дерева рішень, яке дозволяє оцінювати достатність зібраної інформації та визначати подальші кроки для її збору.

Методи дослідження. До методів проведення дослідження можемо віднести збір і аналіз інформації з відкритих джерел, огляд літератури на відповідну тему, підбір інструментів і розбір їх документації.

Апробація результатів роботи. Результати роботи були представлені на XXIII Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених «Теоретичні і прикладні проблеми фізики, математики та інформатики» (14 - 17.05.2025 р., м. Київ, Україна)

1 МЕТОДИ ТА МОДЕЛІ КІБЕРРОЗВІДКИ В КОНТЕКСТІ АНАЛІЗУ ЗАГРОЗ

1.1 Роль кіберрозвідки у сфері кібербезпеки

Кіберрозвідка (англ. Cyber Threat Intelligence, CTI) [4] – це процес збору, обробки та аналізу інформації як про існуючі, так і нові загрози, які можуть бути спрямовані на певний об’єкт безпеки. Вона є невід'ємною частиною кібербезпеки і дедалі частіше стає ефективним методом захисту, що спрямований на забезпечення організації або окремих користувачів актуальними даними про потенційні та реальні кібератаки. Такий підхід дозволяє ефективніше реагувати на загрози та запобігати їм. Говорячи про традиційні засоби кібербезпеки, такі як антивіруси, системи виявлення/запобігання вторгнень (Intrusion Detection/Prevention System, I(D/P)S), або системи захисту інформації і івент менеджменту (Security Information and Event Management Systems, SIEMs) ми розуміємо, що вони не покривають один важливий аспект – прогнозування. У цьому контексті кіберрозвідка виконує важливу роль, оскільки вона дозволяє не лише виявляти атаки, а й прогнозувати їх, аналізуючи поведінку загроз, тактики, техніки і процедури (Tactics, Techniques, Procedures, TTPs) зловмисників. Звідси випливають ключові особливості кіберрозвідки [5]:

- Проактивний підхід – перехід від реактивного до проактивного підходу боротьби з кіберзагрозами. Це означає що замість того, щоб чекати на кібератаку, її можливо запобігти і не дозволити зловмиснику завдати шкоди.
- Унікальність механізмів захисту – вивчаючи суб’єкти загроз, які спрямовані саме на вашу галузь чи регіон, можна створити більш індивідуальні і ефективні стратегії захисту.
- Підтримка процесу ухвалення рішень – кіберрозвідка надає аналітичні дані для керівників, допомагаючи їм ухвалювати стратегічні рішення щодо інвестицій у безпеку, мінімізації ризиків та підвищення операційної ефективності.

- Розуміння поведінки атакуючих – аналіз тактик, технік і процедур (TTPs) зловмисників (Threat Actors) допомагає краще передбачати їхні дії та розробляти стратегії захисту.

Оскільки кіберрозвідка не є одноразовою дією, а являє собою трудомісткий і тривалий процес, він має певні етапи, які можна уявити у вигляді життєвого циклу. Життєвий цикл кіберрозвідки спрямований на перетворення необроблених даних у корисну інформацію, яка допомагає командам кібербезпеки ухвалювати обґрунтовані рішення. Цей процес складається з кількох основних етапів, кожен з яких є взаємопов’язаним [6]:

- **Визначення методів і мети**
  На цьому етапі визначаються цілі та методологія проведення кіберрозвідки. Формулюються ключові питання, зокрема: які мотиви зловмисника та які заходи необхідно вжити для посилення захисту. Окреслюється поверхня атаки.
- **Збір даних**
  Отримання необхідних даних із різних джерел. Інформація збирається відповідно до попередньо визначених вимог.
- **Обробка даних**
  Приведення зібраної інформації до зручного для аналізу формату. Це може включати розшифрування файлів, переклад текстів іноземною мовою, структурування даних у таблиці тощо.
- **Аналіз**
  Дослідження та інтерпретація оброблених даних з метою отримання відповідей на запитання, поставлені на першому етапі. У результаті аналізу робляться конкретні висновки та розроблюються рекомендації щодо подальших дій.
- **Поширення результатів**
  Метою цього етапу є надання отриманої інформації у зручному для сприйняття форматі відповідно до цільової аудиторії.

- **Отримання зворотнього зв’язку**

  Є одним з методів оцінки ефективності розвідувальної операції. Це дозволяє коригувати стратегію, уточнювати пріоритети та покращувати формати подачі аналітичних матеріалів для подальшого використання.

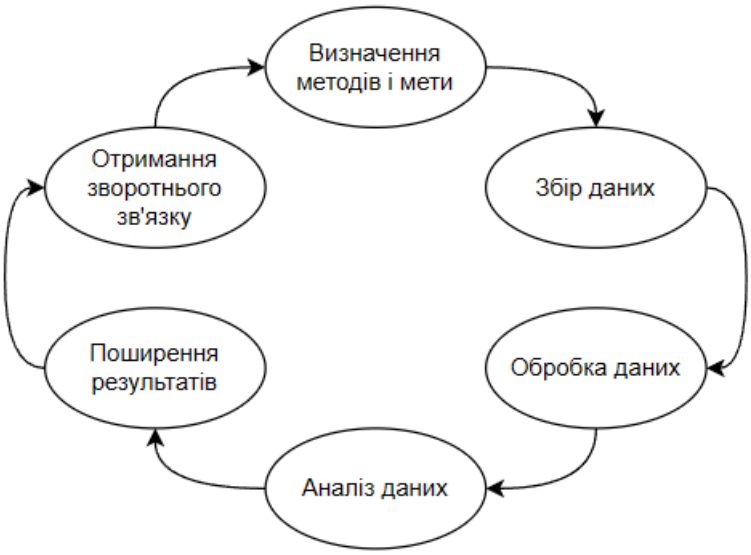

Рисунок 1.1 – Життєвий цикл процесу кіберрозвідки

Кожен етап, від початкового визначення цілей розвідки до збору, аналізу та поширення отриманих даних відіграє важливу роль у забезпеченні ефективного управління кіберзагрозами.

Важливо розуміти, що процес кіберрозвідки може виконувати не тільки захисну функцію, але й мати атакуючий вектор. У такому контексті йдеться про **кіберрекогностування** (англ. Cyber Reconnaissance) – початковий етап кібератаки, під час якого збирається якомога більше відомостей про об’єкт атаки, що дозволяє зловмисникам сформувати профіль своїх цілей, який вони згодом можуть використати. Сам термін “рекогностування” (від лат. Recognosco – оглядаю) згідно з джерелом [7] широко використовується у військовій справі і трактується як “візуальне вивчення супротивника і місцевості в районі майбутніх бойових дій особисто командиром і офіцерами штабів для отримання даних і прийняття рішення”. У кіберпросторі ж замість місцевості, “оглядають” доменні імена, IP-адреси, адреси електронної пошти, версії програмного забезпечення,

конфігурації апаратного забезпечення та навіть особисту інформацію, знайдену на просторах соціальних мереж. Виділяють два основних типи кіберрекогностування [8]:

- **Пасивний –** тип, за умов якого не відбувається прямої взаємодії з об’єктом атаки. Інформація збирається з відкритих джерел, тобто широко використовуються засоби і інструменти OSINT розвідки. Може доходити до того, що інформація дістається з неправильно утилізованих носіїв.
- **Активний –** тип, для якого характерна пряма взаємодія з системою чи мережею жертви. Одним із найпоширеніших засобів проведення цього типу є сканування портів. Відкриті порти слугують точками входу/виходу даних, а отже можуть допомогти зловмиснику знайти потенційні вразливості.

1.2 Класифікація та ключові аспекти кіберрозвідки

Кіберрозвідка, як сучасна дисципліна, має глибоке історичне коріння, яке уходить у традиційні методи збору та аналізу інформації, що використовуються століттями. Розуміння еволюції цих методів дозволить нам краще осмислити сучасні підходи до кіберзагроз та їх запобіганню. Історично склалося так, що розвідка грала ключову роль в забезпеченні безпеки держав і прийнятті стратегічних рішень. З переходом до цифрової епохи традиційні методи розвідки зазнали трансформації, адаптуючись до нових реалій. Поява Інтернету та глобальних комунікаційних мереж значно розширила можливості для збору інформації. Це призвело до виникнення кіберрозвідки як окремої дисципліни, яка має декілька типів і поєднує велику кількість методів її проведення з новими технологіями для протидії сучасним загрозам.

Як і для будь-якої задачі, важливим аспектом для проведення кіберрозвідки є розуміння мети, характеру вхідних даних і сфери застосування. За цими складовими і виокремлюють три основні типи кіберрозвідки [9]: тактичну,

оперативну та стратегічну, кожен з яких відрізняється за рівнем деталізації і функцією в загальній системі інформаційної безпеки.

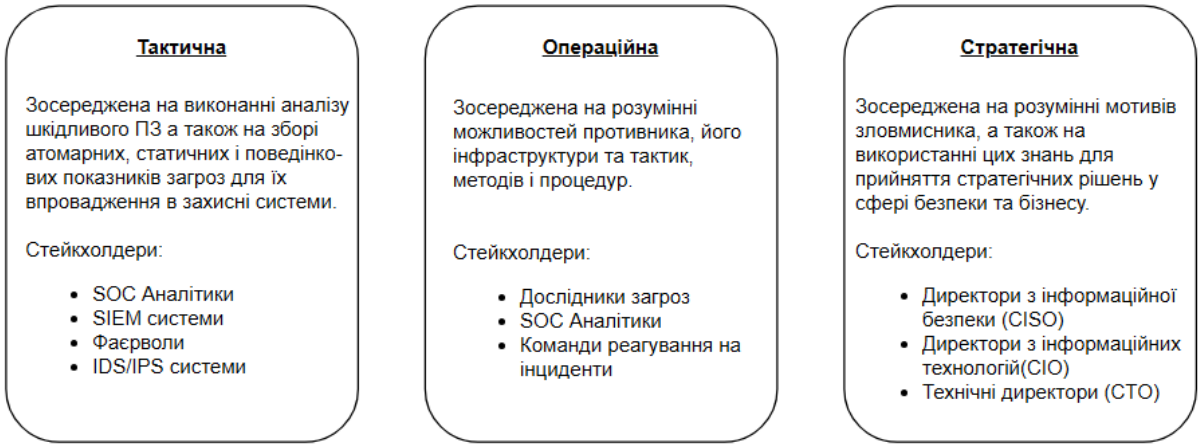

Рисунок 1.2 – Типи кіберрозвідки згідно CrowdStrike

Розберемо трохи детальніше кожен з типів [10]:

- **Тактична розвідка** – зосереджена саме на поточних загрозах. Основний фокус йде на забезпечення безпеки системи шляхом надання конкретних технічних даних про загрози. Полягає в аналізі індикаторів компрометації (Indicators of Compromise, IoC), таких як IP-адреси, хеші файлів, домени зловмисників, сигнатури шкідливого ПЗ тощо. Варто зазначити, що робота з індикаторами компрометації є як і перевагою, оскільки можна легко автоматизовувати процес їх збору, так і недоліком, через те, що вони швидко втрачають свою актуальність, через декілька днів і, навіть, годин.
- **Оперативна розвідка** – фокусується на дослідженні діяльності APT-груп для кращого розуміння зловмисника. Фокусування цього типу розвідки можна зрозуміти задавши наступні питання: “Хто?” – визначає атрибуцію, “Чому?” – мотивацію і “Як?” – окреслює TTPs зловмисників. Перевагою такого типу розвідки є тривалий термін актуальності зібраної інформації, у той час як недоліком є потреба у великій кількості, перш за все, людського ресурсу.

- **Стратегічна розвідка** – більш широкий погляд на процедуру розвідки. Оскільки зловмисники ніколи не діють в умовах “вакууму”, потрібно також брати до уваги геополітичні ризики, економічні фактори тощо. Основною метою цього типу розвідки є формування довгострокових стратегій захисту на рівні організацій, держав та міжнародних структур. Вона включає аналіз загальних тенденцій у сфері кіберзлочинності, прогнозування майбутніх загроз та оцінку їхнього потенційного впливу на критичну інфраструктуру, бізнес-сектор та національну безпеку. Перевагою стратегічної розвідки є можливість комплексної оцінки ситуації, але з цього випливає і недолік – у порівнянні з попередніми типами розвідки, потребує набагато більше ресурсів і часу.

Проведення кіберрозвідки неможливе без відповідних методів збору та аналізу інформації. Варто зауважити, що типи кіберрозвідки (тактична, оперативна, стратегічна), на відміну від методів, визначають рівень деталізації та цільове застосування отриманих даних, у той час як методи розвідки безпосередньо описують способи отримання цієї інформації. Методи розвідки, часто позначені як "INT" (від англ. Intelligence), поділяються залежно від джерел інформації. Оскільки нема стандартизованого списку, розглянемо лише декілька основних методів [11, 12]:

- **Human Intelligence (HUMINT)** – основний фокус йде на комунікацію людини і використання людського фактору. Це традиційна і найстаріша форма збору даних, що ґрунтується на прямому спілкуванні чи спостеріганні людської поведінки. Серед джерел інформації можуть бути інсайдери, агенти, співробітники оборонних структур, військовополонені тощо. Зібрана інформація може надати уявлення про думки, настрої або наміри окремих осіб або груп. В розвідувальних чи військових напрямах, цей метод використовується для збору інформації про іноземні уряди і політичні формування. Основними недоліками HUMINT є можливість спотворення інформації і маніпуляції зі сторони осіб, що її надають.

- **Imagery Intelligence (IMINT)** – метод збору інформації за допомогою фото, відео та інших видів зображень. Найчастіше дані отримуються із супутників, безпілотників та розвідувальних літаків. Основна цінність IMINT – надання візуальної інформації про об’єкти або території, що дозволяє виявляти і оцінювати інфраструктуру, переміщення військ і наслідки ударів, завдяки чому, метод знайшов широке застосування у військових операціях. Однак цей метод має і свої слабкі сторони: інтерпретація зображень потребує високої кваліфікації аналітиків, особливо в умовах сучасності, де технології дозволяють створювати фальшиві зображення, що ускладнює перевірку достовірності даних.
- **Signals Intelligence (SIGINT)** – метод зосереджений на перехопленні, дешифруванні та аналізі електронних сигналів і комунікаційних систем. Цей метод особливо важливий у сучасному світі, де електронні комунікації є основним способом обміну інформацією. Проте і SIGINT має недоліки: дуже великий обсяг перехоплених даних ускладнює їх аналіз. Також зростає різноманіття методів шифрування і технічного захисту, що обмежує доступ до цінної інформації.
- **Open-Source Intelligence (OSINT)** – назва говорить сама за себе – основою цього методу є загальнодоступні джерела інформації. Це можуть бути новини, соціальні мережі, державні звіти, форуми і багато тому подібного. Метод має широкий спектр застосування, оскільки відкриті джерела надають інформацію на будь-яку тематику, включно про політичні та економічні події, військовий потенціал і соціальні тенденції. Але треба бути уважним оскільки надійність відкритих джерел може бути сумнівною через суб'єктивність авторів і потенційні інформаційні маніпуляції.
- **Measurement and signature intelligence (MASINT) –** спеціалізована галузь розвідки, яка займається яка займається виявленням, відстеженням, ідентифікацією та описом унікальних характеристик (сигнатур) стаціонарних або динамічних об'єктів. MASINT має достатньо великий спектр напрямів діяльності:

- Electro-optical MASINT - аналіз електромагнітного випромінювання в оптичному та інфрачервоному діапазонах. Використовується для виявлення теплових слідів, лазерних сигналів, а також для розпізнавання об’єктів у темряві.
- Nuclear MASINT - виявлення та аналіз ядерного випромінювання. Дозволяє ідентифікувати ядерні вибухи, витоки радіоактивних матеріалів або діяльність, пов’язану з розробкою ядерної зброї.
- Radar MASINT - аналіз характеристик радіолокаційних сигналів, включаючи розсіювання, відбиття та поляризацію. Застосовується для виявлення об’єктів, які можуть бути непомітними для звичайних радарів, наприклад, стелс-літаків.
- Materials MASINT - аналіз хімічного складу матеріалів, викидів і залишків вибухових речовин. Дозволяє виявляти хімічні, біологічні та радіаційні загрози.
- Radiofrequency MASINT - дослідження електромагнітних сигналів, які випромінюють різні пристрої, наприклад, радари, системи зв’язку або електронні перешкоди. Використовується для виявлення незвичайної активності та локалізації джерел сигналів.

Описані вище типи та методи розвідки дозволяють класифікувати її за рівнем деталізації та джерелами отримання інформації. Однак для ефективного аналізу та реагування на кіберзагрози необхідно також розібратися з об’єктами розвідки.

При розслідувані, скоріш за все, першим постає питання мотиву і тут вони можуть бути найрізноманітніші, від звичайного дрібного шахрайства в інтернеті до спроб заволодіти військовими розробками іншої держави. Спираючись на мотиви і відбувається класифікація об’єктів розвідки. Згідно з джерелом [13] виділяють такі види суб’єктів загроз (Threat actors):

Таблиця 1.1 – Суб’єкти загрози і приклади їх мотивів

Threat Actor	Common Motivation Example
Держава (Nation State)	Збір розвідувальної інформації (наприклад, зловмисник, який намагається заволодіти держ таємницею).
Кіберзлочинець (Cybercriminal)	Фінансова вигода або прибуток (наприклад, група, що займається програмами-вимагачами, компрометує великий рітейлер, щоб вимагати викуп).
Хактивіст (Hacktivist)	Ідеологічні переконання (наприклад, хакери здійснюють атаку SQL-ін’єкції на вебсайт політичної партії).
Терористичні групи (Terrorist Groups)	Ідеологія/тероризм (наприклад, терористична група здійснює DDoS-атаку на урядовий вебресурс).
Любитель гострих відчуттів (Thrill Seeker)	Хизування або розвага (наприклад, “скрипт-кідді” змінює головну сторінку новинного вебсайту).
Інсайдерська загроза (Insider Threats)	Помста (наприклад, звільнений співробітник викрадає комерційні секрети компанії на USB-носії).

Проте, така діяльність рано чи пізно піддається ретельному аналізу. Фіксація цифрового сліду, що залишається після атак, дозволяє визначити інструменти, методи та мотиви зловмисників. Індикатори компрометації (IoC), техніки, тактики та процедури (TTP), а також характеристики шкідливого програмного забезпечення ретельно досліджуються аналітиками для встановлення зв’язків між окремими атаками та суб’єктами загроз.

Саме завдяки цьому і з’явились спеціалізовані бази даних і системи обміну інформацією про загрози. Вони містять узагальнені відомості, що, як мінімум, допомагає зрозуміти з ким або чим ми маємо справу і, як максимум, швидко реагувати на атаки та розробляти ефективні стратегії захисту. Яскравими прикладами таких ресурсів є:

- **MITRE ATT&CK** (Adversarial Tactics, Techniques, and Common Knowledge) – систематизована база знань, тактик і технік, які часто використовуються зловмисниками. Вона використовується як основа для розробки конкретних моделей та методологій протидії загрозам для компаній в різних галузях економіки, а також у спільноті розробників продуктів та послуг з кібербезпеки. Взагалі, Фреймворк ATT&CK розглядає дві основні концепції – тактики і техніки, тобто конкретну короткострокову мету атаки і засоби, які використовує суб’єкт загрози, відповідно. Він охоплює три різні сфери, звідси і три матриці: Enterprise Matrix описує тактики і техніки характерні для атак на корпоративні системи; Mobile Matrix - для атак на мобільні пристрої; ICS (Industrial Control Systems) Matrix – для атак на промислові системи.
- **AlienVault OTX** (Open Threat Exchange) – платформа для обміну інформацією про кіберзагрози, яка дозволяє запитувати, переглядати та фільтрувати дані, зібрані та форматовані AT&T AlienVault. Формат взаємодії з платформою полягає у підписці на “пульси” – набори даних, що пов’язані з певним шкідливим ПО або атаками на певні компанії. OTX поєднує соціальні внески з автоматизованими інструментами, які інтегруються з основними продуктами безпеки.
- **Shodan** – пошукова система, яка, використовуючи інструменти сканування, “опитує” значну частину адресного простору, тим самим цілеспрямовано індексує всі підключені до інтернету обчислювальні пристрої (веб-камери, маршрутизатори, сервери тощо) і дозволяє знаходити їх за допомогою різних пошукових запитів та фільтрів.

- **Common Vulnerabilities and Exposures** (CVE) – ще один продукт компанії MITRE, розроблений у 1999 році і досі підтримується, який являє собою стандартизований список відомих вразливостей у програмному забезпеченні та обладнанні.
- **VirusTotal** – зручний і відносно безкоштовний онлайн сервіс, для аналізу будь-яких проявів шкідливого ПЗ. Використовує понад 70 антивірусних двигунів для статичного і динамічного аналізу та має досить широкий інструментарій: від візуалізації інформації за допомогою графового представлення до розширеного пошуку у базі даних сервісу за допомогою складних запитів.

Також ці ресурси знайшли широке використання ще в одному з проактивних підходів виявлення загроз, про який треба згадати, а саме в **полюванні на загрози** (Threat Hunting).

Мабуть, найкращим визначенням, що описує threat hunting, буде – процес виявлення загроз, які, якимось чином, не були детектовані автоматизованими механізмами захисту. Згідно з джерелом [14], він складається з декількох кроків:

Рисунок 1.3 – Кроки процесу полювання на загрози

- Визначення цілі полювання (англ. Hunting Objectives and Purpose) – этап, на якому компанії співставляють план проведення полювання зі своїми бізнес-цілями. Формується деяка “гіпотеза”, яка буде направляти весь процес, а основним питанням буде: “Що саме ми шукаємо і для чого?”
- Визначення області полювання (англ. Hunting Scope Definition) – тут відбувається окреслення меж середовища, в якому шукають загрози. Це може бути як корпоративна мережа, так і деякі хости або навіть додатки.

- Ідентифікація джерел даних (англ. Hunting Data Selection) – важливий етап підбору джерел даних, оскільки від цього буде залежати інформованість всієї команди.
- Вибір технік (англ. Hunting Techniques Selection) – оскільки немає чіткої стандартизації полювання на загрози, тут все залежить від рівня спеціалістів і складності задачі. Дуже часто серед технік фігурують підходи зі статистики і аналітики даних.
- Репорт (англ. Report) – формування звітів з результатами проведення полювання, зі всіма аналізами і візуалізаціями. На цьому етапі визначається чи зібрана інформація є достатньою, чи спрацювали обрані техніки і чи досягнуті цілі полювання.

Після того як ми розібрали загальні етапи процесу, пропоную ознайомитися з типами полювання на вразливості. Як вже згадувалося, процес threat hunting не є стандартизованим, а отже і чітко виокремити його типи доволі тяжко. Проте, команда IBM запропонувала деяке узагальнення, яке виділяє 3 основні типи [14]:

- **Структуроване полювання** (англ. Structured Hunting) – цей тип базується на формалізованих фреймворках, накшталт MITRE ATT&CK. Тобто особлива увага приділяється саме відомим індикаторам атак (IoA), або тактикам, технікам і процедурам.
- **Неструктурована полювання** (англ. Unstructured Hunting) – більш реактивний тип, аніж структурований, оскільки йде аналіз причини, що спричинила появу індикатору компометації (IoC).
- **Ситуаційне полювання** (англ. Situational or Entity-Driven Hunting) – тип, який залежить від внутрішньої оцінки ризиків. Основний фокус йде на критично важливі об’єкти, або об’єкти з високим ступенем ризику. Значною перевагю цього типу є можливість зосередити зусилля на найбільш важливих ділянках, що підвищує ефективність виявлення загроз.

1.3 Дерева рішень як інструмент

Оскільки людина, яка займається будь-яким типом кіберрозвідки, може не знати всієї сукупності чинників, виникають умови певної невизначеності. В основному, така ситуація невизначеності полягає у тому, що вибір конкретного плану дій і інструментів, може зумовити будь-який результат із певної множини варіантів, але ймовірності впливу випадкових факторів невідомі. Тому пропоную розглянути метод дерева рішень у цьому контексті. **Дерево рішень** [15] – це графічна модель, яка представляє собою деяку послідовність рішень і виступає у ролі засобу для розв'язання первісної проблеми. Інакше кажучи, це граф [16], вершини якого відповідають ключовим станам (листя або вузли), а дуги (гілки дерева) – різним подіям (рішенням, операціям), що можуть відбутися в ситуації, яка обумовлюється вершиною. Кожна дуга має деяку числову характеристику (вагу), яка визначає відбудеться перехід до одного вузла чи до іншого. Загальний вигляд дерева можна зобразити наступним чином:

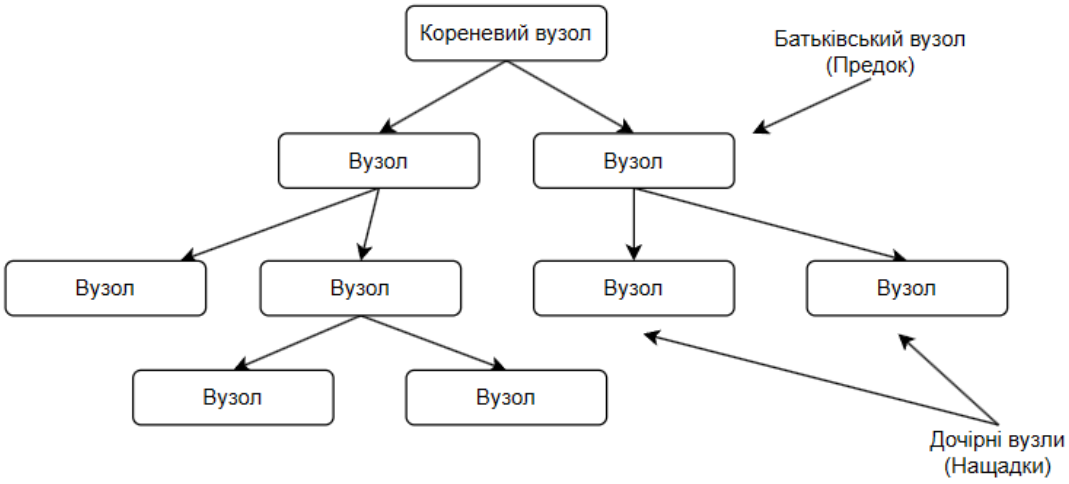

Рисунок 1.4 – Загальний приклад структури дерева рішень

Історія самої концепції дерева рішень бере початок у 50-их роках XIX століття у працях дослідників Ховленда (Hoveland) та Ханта (Hunt) [17]. У 1966 році, під їх авторством була написана "Experiments in Induction" в якій вони детально виклали суть дерев рішень, їх правила та ієрархію. З плином

часу, алгоритм вдосконавлювався і видозмінювався, так у 1979 році Джон Рос Куінлан (John Ross Quinlan) розробив алгоритм ID3 (Iterative Dichotomiser 3), який міг автоматично виокремлювати найважливіші ознаки з даних і використовувати їх для побудови оптимального дерева рішень. До речі, саме він послугував основою для алгоритму C4.5, де додались можливості роботи з числовими атрибутами, формування дерева на неповній навчальній виборці і відсікання гілок, що значно зменшує навантаження при машинному навчанні. Трохи згодом, у 1984 році, Лео Брейман (Leo Breiman) запропонував метод CART (Classification and Regression Trees), де вперше фокус йшов саме на побудову бінарного дерева рішень.

З розвитком методів побудови дерев рішень виникла потреба в їхньому строгому математичному обґрунтуванні. Оскільки сучасні алгоритми ухвалення рішень ґрунтуються на теоретичних поняттях, пропоную розглянути математичні концепції, які широко використовуються в різних підходах побудови дерев рішень [18]:

- **Ентропія** – показник невизначеності (хаосу) у певній вибірці. Використовується в деревах рішень при вирішенні задач класифікації і математично описується як:

$$H = -\sum_{i=1}^{n} p_i * log_2(p_i), \tag{1.1}$$

де n – число класів в початковій множині, p_i – ймовірність потрапляння випадкового об'єкта в клас i. Чим вища ентропія, тим більше різнорідна вибірка. При розбитті дерева алгоритм намагається зменшити ентропію.

- **Індекс Джині** – вимірює ймовірність неправильної класифікації об'єкта, якщо його клас обирати випадково. Також використовується при вирішенні задач класифікації і описується як:

$$G = 1 - \sum_{i=1}^{n} p^2\,, \tag{1.2}$$

де n – число класів в початковій множині, p_i – ймовірність потрапляння випадкового об'єкта в клас i.

Розглянемо ще один алгоритм, основою якого є концепція дерева рішень, а саме **алгоритм випадкового лісу (англ. Random Forest)** [19]. Випадковий ліс, вперше запропонований батьком алгоритму CART Лео Брейманом (Leo Breiman) за співпраці з Адель Катлер (Adele Cutler) у 2001 році, набув широкого використання у задачах класифікації і регресії, є одним з найпоширеніших алгоритмів машинного навчання.

Перед тим як переходити до суті алгоритму, потрібно згадати про наступні методи ансамблевого навчання – способу машинного навчання, де задіяні декілька моделей, за принципом того що вони краще працюють разом, аніж поодинці:

- **Беггінг** (англ. Bagging, скорочення від Bootstrap aggregating) – метод заснований на формуванні підвиборок даних, при чому вибрані елементи в підвиборках можуть повторюватися, і навчанні декількох моделей паралельно.
- **Бустинг** (англ. Boosting) – метод, ґрунтується на послідовному навчанні моделей з врахуванням помилок при навчанні попередніх.

Таблиця 1.2 – Порівняння методів ансамблевого навчання

Беггінг	Бустинг
Декілька моделей навчаються паралельно, незалежно одна від одної	Моделі навчаються послідовно, кожна виправляє помилки попередньої
Для кожної моделі використовуються випадкові підвиборки даних	Концентрація на даних, які могли б спричинити помилки у навчанні
Зменшує дисперсію моделі, корисний при перенавчанні	Допомагає при донавчанні моделі, у випадку якщо вона занадто проста

Тепер більш детально розглянемо концепцію випадкового лісу. Його суть полягає використанні вихідних даних декількох дерев рішень, для вирішення певної проблеми або для отримання остаточного прогнозу. Інакше кажучи, це певне розширення беггінгу, але з використанням випадкового вибору ознак (feature randomness) – тобто, якщо в класичному варіанті, на кожному вузлі обирається найкраща ознака для розбиття даних, то в цьому – спочатку обираються рандомні ознаки і потім серед них обирається найкраща, що дозволяє досягти якнайменшої кореляції між деревами рішень.

При цьому, алгоритм має три змінні параметри:

- **Розмір вузла** (англ. node size) – визначає мінімальну кількість зразків у вузлі.
- **Кількість дерев** (англ. number of trees) – кількість дерев рішень, що буде використовуватися в ансамблі.
- **Кількість ознак для вибору** (англ. number of features sampled) – скільки ознак буде випадково обиратися для кожного розбиття.

Як і будь-який алгоритм, випадковий ліс має свої переваги і недоліки. Спираючись на джерело [19], можемо скласти наступну таблицю:

Таблиця 1.3 – Переваги і недоліки алгоритму Random forest

Переваги	Недоліки
Низька ймовірність перенавчання моделі через усереднення вихідних результатів.	Повільне навчання, через необхідність обчислень для кожного дерева в ансамблі в умовах великого об’єму даних.
Підходить як для задач класифікації, так і для задач регресії завдяки feature randomness.	Необхідність великої кількості ресурсів для зберігання і обробки великої кількості даних.
Дозволяє оцінювати внесок ознак в прогнози моделі.	Можлива складність інтерпретації через комплексність алгоритму.

Висновки до розділу 1

У цьому розділі було розглянуто базові поняття, пов’язані з кіберрозвідкою, її особливості і роль у сфері кібербезпеки, а також основні етапи життєвого циклу цього процесу.

Проаналізовано класифікацію кіберрозвідки залежно від глибини аналізу та цілей використання. Розглянуто основні методи розвідки. Також досліджено ключові підходи до класифікації загроз і суб’єктів атак, що дозволяє краще розуміти мотиви зловмисників.

Надано теоретичні відомості про концепцію дерев рішень. Проаналізовано переваги дерев рішень як графічної моделі, що дозволяє ієрархічно структурувати можливі сценарії дій, призначати вагові коефіцієнти та приймати рішення на основі наявних даних. Через умови деякої невизначеності при прийнятті рішень, виникла потреба в інструменті, який дозволяє формалізувати та структурувати цей процес кіберрозвідки – таким інструментом і стало дерево рішень.

2 ФОРМАЛІЗАЦІЯ І ОПИС МЕТОДОЛОГІЙ

2.1 Вибір середовища зберігання та обробки моделі

Оскільки перед нами стоїть задача побудови дерева рішень, що має структуру графа, потрібно обрати відповідне середовище зберігання та обробки даних, яке, крім цього, дасть змогу ще й візуалізувати ці дані. У такому випадку, вибір падає на графові бази даних.

Графова база даних [20] – це тип нереляційних баз даних, який використовує графову структуру для семантичних запитів задля представлення даних. Через те, що вони базуються на теорії графів, такі БД мають наступні компоненти:

- **Вузли** (англ. Nodes) – сутності в даних
- **Ребра** (англ. Edges) – зв’язки між сутностями
- **Властивості** (англ. Properties) – додаткова інформація чи атрибути, для того, щоб додати більше контексту. Можуть надаватися як вузлам, так і ребрам.

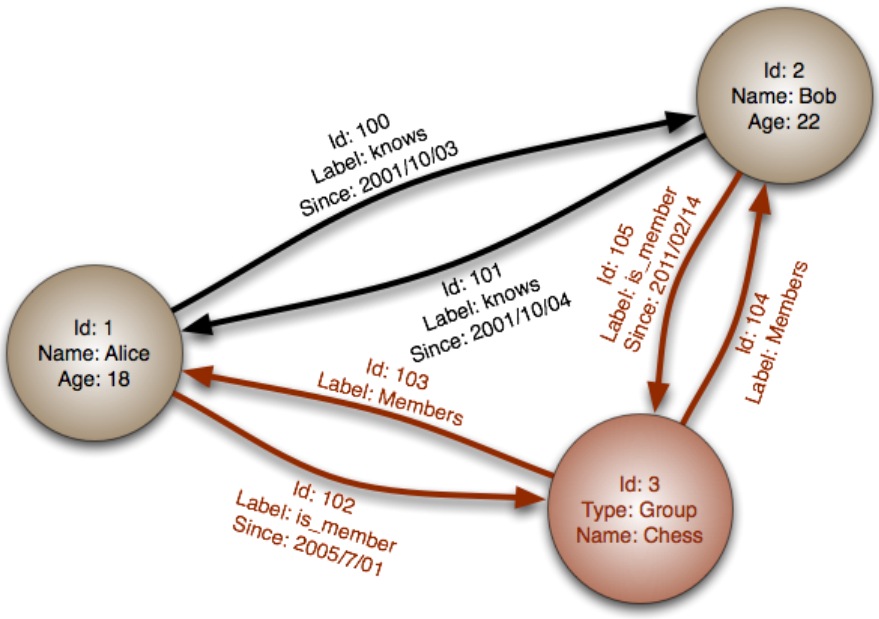

Рисунок 2.1 – Приклад компонент графової бази даних

Графові бази даних широко використовуються для роботи з даними, де важливий контекст взаємозв'язків, наприклад: соціальні мережі, рекомендації товарів, системи виявлення шахрайства та інші сфери, де зв'язки між об'єктами мають ключове значення. В свою чергу, ці зв’язки можна зручно описувати у вигляді триплетів, що відсилає нас до абстрактної моделі представлення даних RDF (Resource Description Framework) [21]. Сам триплет являє собою просту стуктуру даних “суб’єкт – предикат – об’єкт” (subject–predicate–object).

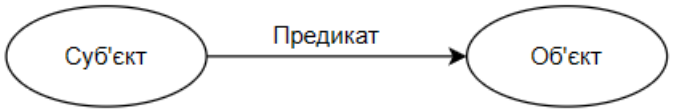

Рисунок 2.2 – Структура триплета

Кожен триплет описує окреме твердження про об’єкт:

- Суб’єкт - це ресурс, про який йде мова
- Предикат - властивість або відношення
- Об’єкт - значення властивості або інший ресурс

Ця модель зручна для представлення пов’язаних даних, оскільки дозволяє будувати орієнтований граф, де вузли - це суб’єкти та об’єкти, а ребра - це предикати.

Враховуючи актуальність використання графових баз даних для вирішення поставленої задачі, було використано ArangoDB. **ArangoDB** [22] – мультимодельна база даних з відкритим кодом, що підтримує три моделі даних:

- **Графова модель**
- **Документна модель** – дані зберігаються у вигляді JSON-об'єктів, які називаються документами.
- **Модель формату “Ключ - Значення” –** дані зберігаються як сукупність пар “ключ-значення”, причому як ключі так і значення можуть являти собою прості або складені об’єкти.

Окрім цього ArangoDB має ще низку особливостей. Базу даних можна розгорнути будь-де, в залежності від потреб і можливостей. Якщо ресурси дозволяють, то можна використовувати on-premise рішення, в іншому ж випадку можна скористатися послугами хмарних провайдерів і використовувати ArangoGraph Insights Platform, що працює в дата-центрах провайдера, таких як Google Cloud Platform (GCP), Amazon Web Services (AWS) чи Microsoft Azure. Платформа забезпечує постійну доступність бази даних шляхом цілодобового моніторингу, встановлення нових версій без переривання сервісу та забезпечує безпеку даних через шифрування та аудит, а також регулярне створення резервних копій.

При локальному розгортанні, ArangoDB пропонує достатню кількість інструментів для взаємодії з базою даних. Яскравими прикладами можуть слугувати web-інтерфейс і командна оболонка arangosh, яка підтримує функції імпорту/експорту даних, створення резервних копій, відновлення даних тощо. Запити до бази даних відбуваються за допомогою SQL подібної мови запитів **AQL** (ArangoDB Query Language), яка дозволяє комбінувати підтримувані моделі даних в одному запиті.

В моєму випадку, через відносно невелику кількість даних, ArangoDB була розгорнута локально і взаємодія відбувалася через web-інтерфейс. Основною перевагою використання web-інтерфейсу, є саме можливість візуалізація даних, це дає змогу легко орієнтуватися у зв’язках між даними і розуміти їх структуру. Також, варто згадати про налаштування відображення графа: колористика, відображення деяких атрибутів, кастомні найменування тощо. Це дає змогу покращити рівень сприйняття і регулювати степінь деталізації відображення даних.

2.2 Опис даних

Перейдемо до визначення сутностей і їх зв’язків. Сутності – це абстракції, яку ми хочемо змоделювати й зберегти в базі даних. В контексті графової моделі даних сутності відображаються як вузли. Кожен вузол може мати свій тип і

атрибути, які додають контексту і можуть описувати його характеристики. Зв’язки ж, у свою чергу - це направлене або ненаправлене відношення між двома сутностями. В графовій моделі зв’язки відображаються як ребра, і вони теж можуть мати тип і власні атрибути.

Проаналізувавши предметні області методів кіберрозвідки, можна запропонувати відносно універсальну модель ідентифікації сутностей:

- Кореневий вузол - визначає основний напрямок збору інформації;
- Техніка - узагальнені методи розвідки в межах напрямку;
- Сабтехніка – конкретизація вузлів технік;
- Інструмент - технічний засіб;
- Процедура – певний порядок дій.

Така модель дозволяє прослідкувати деяку ієрархію, або ж рівневість при побудові графа. З типів вузлів випливають наступні зв'язки:

- “hasTechnique” – метод кіберрозвідки має наступні техніки. зв’язок кореневого вузла з техніками;
- “hasSubTechnique” – зв’язок технік з сабтехніками;
- “hasTool” – зв’язок сабтехнік з інструментами;
- “hasProcedure” – зв’язок сабтехнік і процедур.

Щоб відобразити важливість і практичну цінність кожного елемента, було введено для кожного вузла атрибут ваги - числове значення від 1 до 5. Цю вагу розраховано шляхом експертних оцінок, щоб врахувати доречність і ефективність процедури або інструменту в контексті повної сабтехніки.

В якості наочного прикладу запропонованої моделі, можемо розглянути техніку активного сканування в межах процедури кіберрекогностування. У цьому прикладі чітко простежується ієрархічна структура: від загальної техніки - до її конкретизацій, інструментів та процедур.

<Active_Scanning> <type> "RecTechnique" .

<Active_Scanning> <weight> "4.5" .

<Active_Scanning> <hasSubTechnique> <Scanning_IP_Blocks> .

<Active_Scanning> <hasSubTechnique> <Vulnerability_Scanning> .

<Active_Scanning> <hasSubTechnique> <Wordlist_Scanning> .

<Scanning_IP_Blocks> <type> "RecSubTechnique" .

<Scanning_IP_Blocks> <hasTool> <Nmap> .

<Scanning_IP_Blocks> <hasTool> <Routersploit> .

<Scanning_IP_Blocks> <weight> "4.2" .

<Nmap> <type> "tool" .

<Nmap> <weight> "4.7" .

<Routersploit> <type> "tool" .

<Routersploit> <weight> "3.8" .

<Vulnerability_Scanning> <type> "RecSubTechnique" .

<Vulnerability_Scanning> <hasTool> <Nessus> .

<Vulnerability_Scanning> <hasTool> <OpenVAS> .

<Vulnerability_Scanning> <weight> "4.5" .

<Nessus> <type> "tool" .

<Nessus> <weight> "4.8" .

<OpenVAS> <type> "tool" .

<OpenVAS> <weight> "4.3" .

<Wordlist_Scanning> <type> "RecSubTechnique" .

<Wordlist_Scanning> <hasProcedure> <SubdomainEnumeration> .

<Wordlist_Scanning> <hasTool> <GoBuster> .

<Wordlist_Scanning> <weight> "3.5" .

<SubdomainEnumeration> <type> "procedure" .

<SubdomainEnumeration> <weight> "3.7" .

<GoBuster> <type> "tool" .

<GoBuster> <weight> "3.6" .

Техніка <Active_Scanning> позначає загальний метод збору інформації шляхом активної взаємодії з цілю. Цей вузол має тип "RecTechnique" і вагу 4.5, що відображає його важливість і доцільність у межах розвідки. До неї прив’язані три сабтехніки:

- <Scanning_IP_Blocks>
- <Vulnerability_Scanning>
- <Wordlist_Scanning>

Кожна з сабтехнік має тип "RecSubTechnique" та власну вагу - наприклад, сканування IP-діапазонів (Scanning_IP_Blocks) має вагу 4.2, що трохи нижче за основну техніку, оскільки є конкретизацією, яка застосовується залежно від контексту. Далі сабтехніки пов’язані з конкретними інструментами за допомогою зв’язку hasTool. Наприклад сабтехніка <Scanning_IP_Blocks> використовує інструменти <Nmap> і <Routersploit>. Під час <Vulnerability_Scanning> можуть застосовуватися інструменти <Nessus> та <OpenVAS>.

2.3 Агрегація даних

Для створення графа необхідно було не тільки зібрати інформацію і структуризувати її, а й сформувати датасет, з урахуванням можливостей і нюансів роботи з графовою базою даних ArangoDB.

У процесі переносу даних у базу, формувалися колекції, а саме колекції зв’язків (у форматі Edge) і колекції документів (у форматі Document). Якщо

проводити аналогії з реляційними базами даних, то колекції можна порівняти з таблицями, але, на відміну від таблиць, де всі записи мають певну структуру і визначений тип даних, у колекціях можуть зберігатися записи довільної структури. В моєму випадку, вони зберігаються в JSON файлах.

Розглянемо деякий приклад для ночності. Якщо ми хочемо сформувати таблицю під назвою “Техніки” в реляційній базі даних, вона буде мати вигляд таблиці 2.1. При цьому, тип даних у комірках буде заздалегідь визначений, наприклад для назви вузла – char, а для ваги – float.

Таблиця 2.1 – Приклад таблиці технік

ID	Вузол	Тип	Вага
1	Google_Dorking	Technique	4.6
2	Geolocation_Tracking	Technique	4.7

У випадку з колекціями документів, формат зберігання буде наступний:

[

{

"_key": "Google_Dorking",

"_id": "technique/Google_Dorking",

"_rev": "_jtG0C1O---",

"type": "Technique",

"weight": 4.6

},

{

"_key": "Geolocation_Tracking",

"_id": "technique/Geolocation_Tracking",

"_rev": "_jtxS2G----",

"type": "Technique",

"weight": 4.7

}

]

Колекції зв’язків мають схожий формат. Зв’язки будуються за допомогою полів _from (джерело) і _to (ціль). В них зберігаються значення полів “_id” з документів. Так ми розуміємо, між якими об’єктами з якої колекції є зв’язок. Наприклад:

[

{

"_key": "18308",

"_id": "start-tech/18308",

"_from": "start/OSINT",

"_to": "technique/Google_Dorking",

"_rev": "_jtHHQjO---",

"relation": "hasTechnique"

},

{

"_key": "36725",

"_id": "start-tech/36725",

"_from": "start/OSINT",

"_to": "technique/Geolocation_Tracking",

"_rev": "_jtd0qmK---",

"relation": "hasTechnique"

}

]

Бачимо, зв'язки кореневого вузла з техніками Geolocation_Tracking і Google_Dorking. Тип цих зв'язків, відповідно, буде “Має техніку”, про що свідчить атрибут "hasTechnique".

Висновки до розділу 2

У розділі 2 було проведено ознайомлення з поняттям графічних баз даних, зокрема з структурою RDF триплетів. Обґрунтовано вибір графової бази даних ArangoDB в якості основного інструменту для реалізації моделі через її мультимодельний підхід, можливості зручної взаємодії з базою даних а також можливість візуалізації даних через web-інтерфейс.

Запропоновано універсальну модель ідентифікації сутностей, яка включає кореневий вузол (метод кіберрозвідки), техніки, сабтехніки, інструменти та процедури. Визначено основні типи зв’язків (hasTechnique, hasSubTechnique, hasTool, hasProcedure), які забезпечують чітке розуміння залежностей між сутностями. Розглянуто відповідний приклад для процедури кіберрекогностування.

Описано процес агрегації даних у порівнянні з реляційними базами даних на конкретних прикладах.

3 РЕАЛІЗАЦІЯ МОДЕЛІ

Проаналізувавши предметні області кожного методу розвідки, було вирішено, у рамках дипломної роботи, будувати дерева для трьох методів в контексті кіберрозвідки, а саме OSINT (Open-Source Intelligence), HUMINT (Human Intelligence) та SIGINT (Signal Intelligence). Такий вибір зумовлений цифровою трансформацією суспільства, де технології набули широкого використання і вже є неодмінною частиною людства. Навіть HUMINT, який, здавалося б, полягає у прямій комунікації з людьми, активно використовується в умовах кіберпростору. Яскравим і життєвим прикладом є дистанційне навчання, в умовах якого, за допомогою месенджерів і освітніх платформ, була налагоджена повноцінна комунікація між студентами та викладачами.

3.1 Загальний опис дерева

Дерево має кореневий вузол, який і визначає метод кіберрозвідки. Від нього розходяться гілки, до технік, які притаманні цьому методу. Так само будуються зв’язки з сабтехніками, процедурами і інструментами. Тобто відображується певна ієрархія, і конкретизація. Кожен тип вузла позначено кольором для зручності сприйняття:

- Жовтий колір – техніки. Основні підходи до збору та аналізу даних у рамках методу розвідки.
- Червоний колір – сабтехніки. Більш деталізовані етапи, які конкретизують техніки.
- Фіолетовий колір – інструменти. Програмне забезпечення або технології, що використовуються для реалізації сабтехнік.
- Зелений колір – процедури. Конкретні дії, які виконуються в межах сабтехнік.

3.2 OSINT метод

Зі стрімким ростом об’єму відкритої інформації і появі великого різноманіття її джерел, збільшуються і можливості і актуальність OSINT методу розвідки. Він, як вже згадувалося, полягає у зборі інформації з загальнодоступних джерел інформації, ними можуть бути вебсайти, форуми, соціальні мережі, реєстри, архіви, звіти компаній тощо. Отже, використавши запропоновану модель ідентифікації сутностей і визначивши зв’язки між ними можна побудувати граф (Рисунок 3.1):

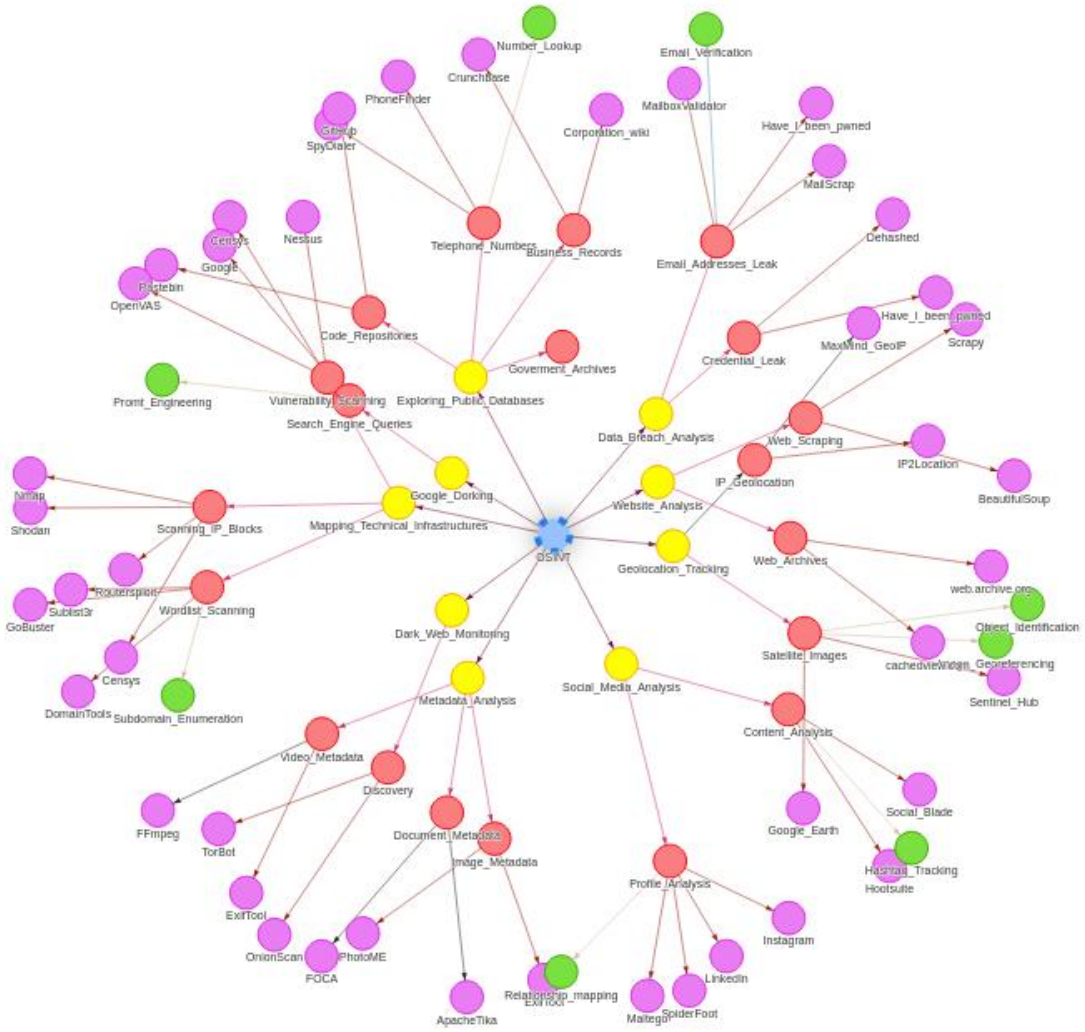

Рисунок 3.1 – Дерево рішень для OSINT методу

Перейдемо до наповнення дерева. Було виокремлено і візуалізовано, притаманні для OSINT методу, наступні техніки, сабтехніки, інструменти і процедури:

Техніка **Google_Dorking** полягає у використанні розширених пошукових запитів у пошукових системах для знаходження специфічної інформації. Техніці притаманна сабтехніка Search_Engine_Queries, яка має процедуру Promt_Engineering і браузер Google у якості інструменту.

Техніка **Mapping_Technical_Infrastructures** спрямована на ідентифікацію та аналіз технічної інфраструктури цілі. До неї належать сабтехніки Scanning_IP_Blocks – сканування IP діапазонів (з інструментами nmap, routersploit, censys, shodan), Vulnerability_Scanning – сканування вразливостей мережевих ресурсів і пристроїв (openvas, nessus, censys) та Wordlist_Scanning – сканування за допомогою словників, що включає процедуру SubdomainEnumeration і інструменти gobuster, DomainTools, Sublist3r.

Техніка **Social_Media_Analysis** фокусується на зборі інформації з соціальних мереж. Вона охоплює сабтехніки Profile_Analysis (Maltego, LinkedIn, Instagram, SpiderFoot), що має процедуру Relationship_Mapping де ми встановлюємо зв’язки типу “хтось знає когось”, та Content_Analysis в рамках якої аналізують контент, з яким взаємодіє ціль. До цієї сабтехніки входить процедура Hashtag_Tracking з інструментами Hootsuite і Social_Blade.

Техніка **Exploring_Public_Databases** використовується для дослідження відкритих баз даних. Сабтехніки Code_Repositories (GitHub, PasteBin), Business_Records (Crunchbase, Corporation_wiki), Goverment_Archives, Telephone_Numbers (SpyDialer, PhoneFinder), підкреслюють, які саме джерела можна розглядати в якості відкритих баз даних.

Техніка **Website_Analysis** полягає у аналізі веб-ресурсів за допомогою сабтехнік Web_Scraping (Scrapy, BeautifulSoup) – часто автоматизованого вилучення інформації з веб-сторінок і Web_Archives (web.archive.org, cachedview.com) - перегляду інтернет архівів.

Техніка **Dark_Web_Monitoring** передбачає моніторинг ресурсів у мережі Tor, які часто містять витоки інформації і дуже специфічні обговорення. Єдина сабтехніка Discovery використовує інструменти OnionScan і TorBot.

Техніка **Geolocation_Tracking** спрямована на визначення географічного розташування об’єктів. Притаманна їй, сабтехніка IP_Geolocation (IP2Location, MaxMind_GeoIP) полягає у використанні IP-адрес для визначення геолокації, а Satellite_Images (Google_Earth і Sentinel_Hub) – у використанні супутникових зображень. Це реалізується через процедури Image_Georeferencing і Object_Identification.

Техніка **Metadata_Analysis** полягає у дослідженні метаданих файлів для збору додаткової інформації. Сабтехніки Image_Metadata (ExifTool, PhotoME), Document_Metadata (FOCA, ApacheTika), та Video_Metadata (FFmpeg, ExifTool) описують метадані якого типу файлу будуть досліджуватися.

Техніка **Data_Breach_Analysis** орієнтована на виявлення витоків чутливої інформації в зламаних або відкритих базах. Сабтехніка Credential_Leak (Dehashed, Have_I_been_pwned) стосується, здебільшого, логінів, паролів та ключів, а Email_Addresses_Leak (MailScrap, MailboxValidator), яка має процедуру Email_Verification – пошт.

3.3 HUMINT метод

Перейдемо до HUMINT методу. Це найстаріша форма збору даних, що ґрунтується на прямому спілкуванні чи спостеріганні людської поведінки. І навіть попри розвиток технологій, метод не втрачає своєї актуальності. Він дедалі частіше використовується в кіберпросторі — через соціальні мережі, месенджери, форуми та інші канали комунікації. Дерево для цього методу виглядає наступним чином (Рисунок 3.2)

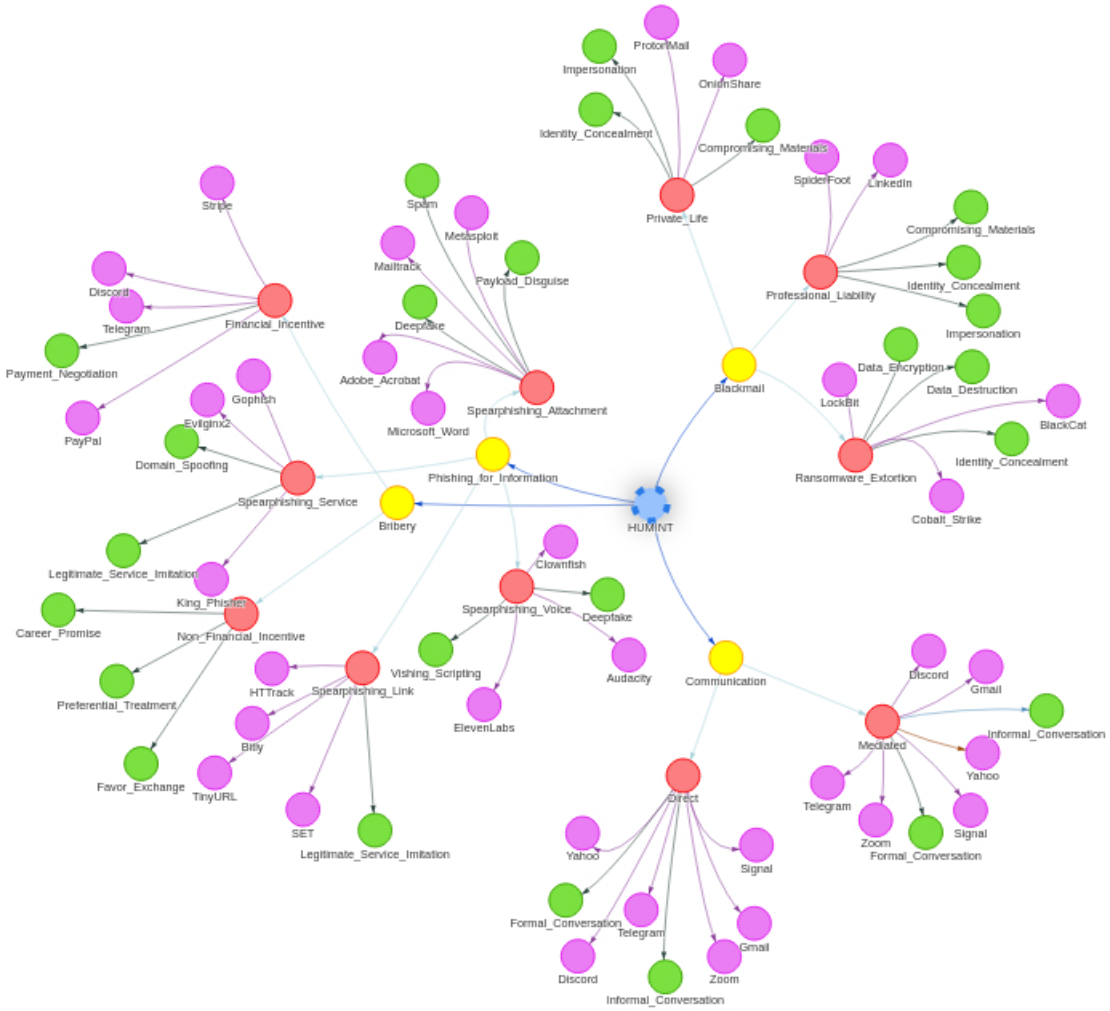

Рисунок 3.2 – Дерево рішень для HUMINT методу

Для наповнення цього дерева було виокремлено наступні техніки, сабтехніки, інструменти і процедури:

Техніка **Phishing_for_Information** передбачає отримання чутливої інформації шляхом обману або соціальної інженерії. До техніки входить сабтехніка Spearphishing_Service (Gophish, Evilginx2, King_Phisher) якій притаманні процедури Domain_Spoofing та Legitimate_Service_Imitation. Сабтехніка Spearphishing_Attachment (Microsoft_Word, Adobe_Acrobat, Metasploit, Mailtrack) полягає у розповсюдженні фішингових файлів: від згенерованих штучним інтелектом картинок до підроблених документів. В її рамках відбуваються процедури Spam, Payload_Disguise і Deepfake. Сабтехніка Spearphishing_Link фокусується на обманних гіперпосиланнях із використанням інструментів HTTrack, SET, Bitly, TinyURL. Також присутня сабтехніка

Spearphishing_Voice, яка охоплює використання синтезу мови та зміну голосу за допомогою інструментів ElevenLabs, Clownfish, Audacity і процедур Deepfake та Vishing_Scripting.

Техніка **Bribery** передбачає надання будь-яких благ з метою отримання доступу до інформації або впливу на поведінку людини. Сабтехніка Financial_Incentive включає процедуру Payment_Negotiation і використовує інструменти для комунікації (Telegram, Discord) і передачі коштів (Stripe, PayPal). Сабтехніка Non_Financial_Incentive базується на нематеріальній мотивації та включає процедури Career_Promise, Preferential_Treatment і Favor_Exchange.

На відміну від Bribery, техніка **Blackmail** вже не передбачає ніяких бенефітів для цілі. Вона полягає в шантажі осіб, використовуючи компромат на них або створюючи загрозу їхньому “цифровому майну”. Сабтехніка Private_Life фокусується на інформаціїї про особисте життя цілі, з використанням інструментів ProtonMail і OnionShare для збереження анонімності та процедур Compromising_Materials, Identity_Concealment, Impersonation. Сабтехніка Professional_Liability спрямована на загрози, пов’язані з кар’єрою, з тими ж процедурами, а також із використанням LinkedIn і SpiderFoot для збору інформації. Сабтехніка Ransomware_Extortion реалізується через інструменти LockBit, BlackCat, Cobalt_Strike, що використовуються для процедур Data_Encryption, Data_Destruction та Identity_Concealment.

Оскільки інформацію, в рамках HUMINT методу, можна добувати і без суперечностей з моральними нормами, було виокремлено техніку **Communication**, яка охоплює як прямі, так і опосередковані способи встановлення контактів з метою збору інформації або впливу за допомогою сабтехнік Direct та Mediated відповідно. Сабтехніка Direct включає процедури Formal_Conversation та Informal_Conversation, що реалізуються через інструменти Zoom, Signal, Telegram, Discord, Gmail, Yahoo. Сабтехніка Mediated використовує ті ж процедури та платформи.

3.4 SIGINT метод

Цей метод охоплює широкий спектр технік, спрямованих на прехоплення і аналіз сигналів різної природи. Його також можна поділити на два “підметоди”: ELINT (Electronic Intelligence) – розвідка зосереджена на електронних сигналах і COMINT (Communications Intelligence) – розвідка комунікацій. В запропонованому дереві (Рисунок 3.3) COMINT висвітлено за допомогою методів перехоплення та аналізу мережевих, голосових, таких як VoIP (Voice over IP) та мобільних комунікацій. ELINT представлений за допомогою методів, пов'язаних з радіоконтролем, супутниковими сигналами та аналізом зашифрованого трафіку.

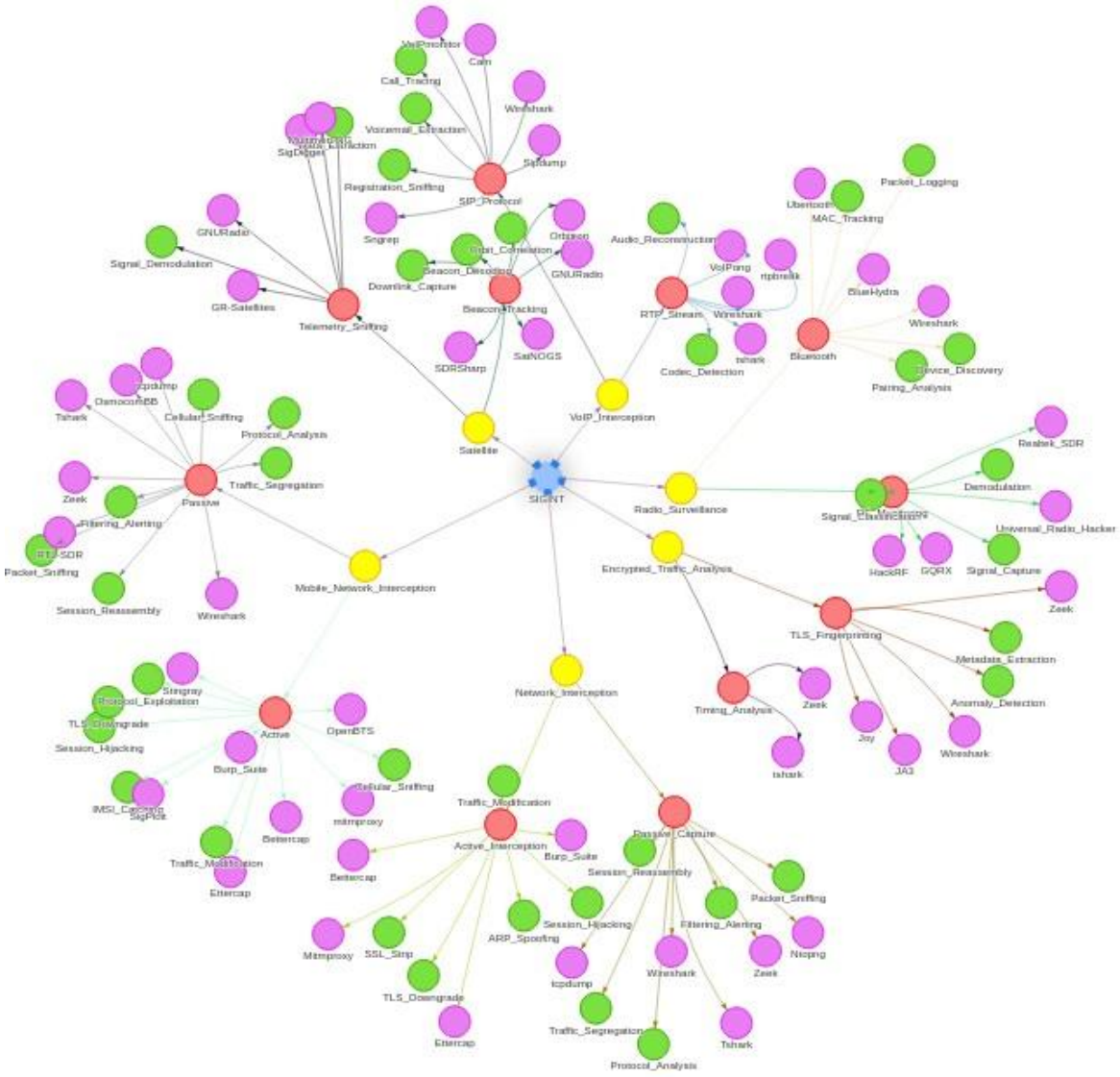

Рисунок 3.3 - Дерево рішень для SIGINT методу

Техніка **Network_Interception** полягає в перехопленні мережевого трафіку. До техніки належить сабтехніка Passive_Capture, яка реалізується за допомогою інструментів Wireshark, tcpdump, Zeek, Tshark, Ntopng, що дозволяють виконувати процедури Packet_Sniffing, Session_Reassembly, Protocol_Analysis, Filtering_Alerting та Traffic_Segregation. У свою чергу, сабтехніка Active_Interception охоплює активні атаки на мережу із використанням Ettercap, Bettercap, Mitmproxy, Burp_Suite і процедур ARP_Spoofing, SSL_Strip, Traffic_Modification, TLS_Downgrade і Session_Hijacking.

Так само як і **Network_Interception**, техніка **Mobile_Network_Interception** зосереджена на перехопленні трафіку, але тільки мобільного. Так само має сабтехніки Passive і Active. Passive використовує інструменти Wireshark, tcpdump, Zeek, Tshark, RTL-SDR, OsmocomBB для процедур Cellular_Sniffing, Packet_Sniffing, Session_Reassembly, Protocol_Analysis, Filtering_Alerting і Traffic_Segregation. Сабтехніка Active базується на використанні інструментів активного втручання, зокрема Stingray, OpenBTS, SigPloit, Ettercap, Bettercap, mitmproxy і Burp Suite, для реалізації процедур Cellular_Sniffing, IMSI_Catching, Traffic_Modification, TLS_Downgrade, Session_Hijacking та Protocol_Exploitation.

Техніка **Radio_Surveillance** охоплює моніторинг радіосигналів для виявлення активності в ефірі. В рамках сабтехніки RF_Monitoring можуть використовуватися Realtek_SDR, GQRX, HackRF і Universal_Radio_Hacker для процедур Signal_Capture, Signal_Classification і Demodulation. Сабтехніка Bluetooth реалізується через Ubertooth, Wireshark і BlueHydra, що дозволяють проводити Device_Discovery, MAC_Tracking, Packet_Logging та Pairing_Analysis.

Техніка **Satellite** фокусується на перехопленні супутникових сигналів. Сабтехніка Beacon_Tracking, яка полягає у відслідковуванні супутникових маяків, реалізується інструментами GNURadio, SatNOGS, SDRSharp і Orbitron та має процедури Beacon_Decoding, Orbit_Correlation і Downlink_Capture. Сабтехніка Telemetry_Sniffing передбачає використання GNURadio, GR-Satellites, Multimon-NG і SigDigger у процедурах Signal_Demodulation і Data_Extraction.

Техніка **Encrypted_Traffic_Analysis** передбачає вивчення зашифрованого трафіку з метою виявлення патернів або метаданих або будь-якої іншої корисної інформації. Сабтехніка TLS_Fingerprinting включає процедури Anomaly_Detection і Metadata_Extraction, що виконуються через Joy, Zeek, Wireshark і JA3. Сабтехніка Timing_Analysis реалізується за допомогою tshark і Zeek безпосередньо через аналіз часових характеристик трафіку.

Техніка **VoIP_Interception** полягає у перехопленні голосового трафіку, що передається через IP-протоколи. В рамках сабтехніки SIP_Protocol розглядається Session Initiation Protocol, який забезпечує безперервну роботу IP телефонії. Використовуються Wireshark, Sipdump, Sngrep, VoIPmonitor і Cain для процедур Registration_Sniffing, Voicemail_Extraction і Call_Tracing. Сабтехніка RTP_Stream фокусується на протоколі Real-time Transport Protocol, який є основою для передачі мультимедійних даних у реальному часі. Вона має процедури Audio_Reconstruction та Codec_Detection за допомогою інструментів Wireshark, tshark, rtpbreak і VoIPong.

3.5 Приклад застосування

В якості наочного прикладу, змоделюємо ситуацію коли ми визначилися, яким методом проводити кіберрозвідку, нехай це буде HUMINT. Подивимося як працювати з деревом для HUMINT методу. Для початку виведемо всі колекції для того, щоб побачити вміст дерева.

Запит (рисунок 3.4)**:**

```
1  FOR coll IN COLLECTIONS()
2    RETURN coll.name
3
```

Рисунок 3.4 – AQL запит для виводу наявних колекцій

Результат виконання запиту (рисунок 3.5):

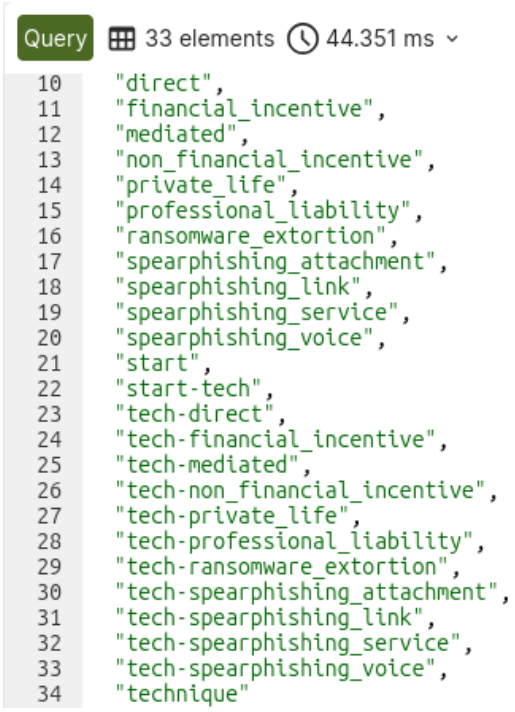

Рисунок 3.5 – Наявні колекції в дереві

Виведемо всі техніки, які притаманні HUMINT методу. Можна відобразити як у JSON форматі (рисунок 3.7) так і у вигляді графу (рисунок 3.9).

Запит (рисунок 3.6):

```
FOR rel IN start-tech
  FILTER rel._from == "start/HUMINT" AND rel.relation == "hasTechnique"
  RETURN rel._to
```

Рисунок 3.6 – AQL запит для виводу технік HUMINT у форматі JSON

Результат виконання запиту:

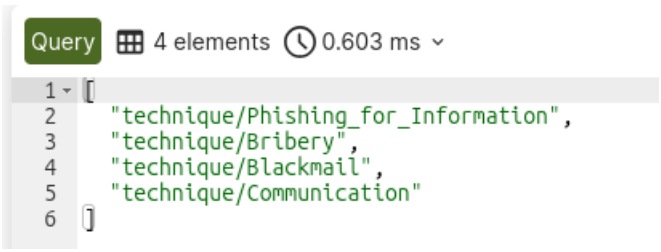

Рисунок 3.7 – Техніки притаманні HUMINT методу у форматі JSON

Запит (рисунок 3.8):

```
FOR e IN `start-tech`
  FILTER e._from == "start/HUMINT" AND e.relation == "hasTechnique"
  LET target = DOCUMENT(e._to)
  RETURN {
    _from: e._from,
    _to: e._to,
    _id: CONCAT("start-tech/", e._key),
    label: target.name
  }
```

Рисунок 3.8 – AQL запит для виводу технік HUMINT у вигляді графу

Результат виконання запиту:

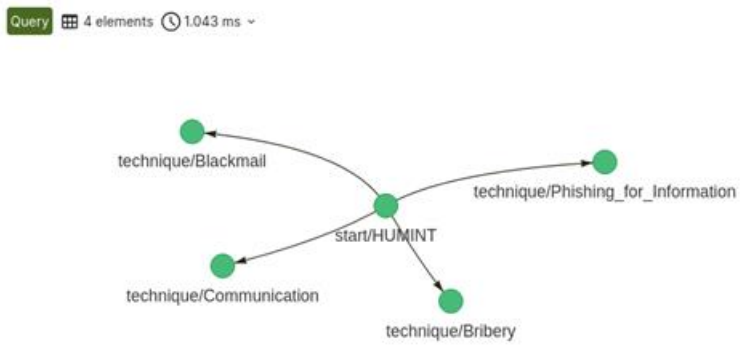

Рисунок 3.9 – Техніки притаманні HUMINT методу у вигляді графу

Тепер необхідно зрозуміти пріоритетність технік. Оскільки вони мають вагу, ми можемо вивести техніку з найбільшою вагою і подивитися які сабтехніки їй притаманні. Спочатку відсортуємо техніки за спаданням і виберемо першу. По графу знайдемо всі вихідні зв’язки типу “hasSubTechnique” для обраної техніки і виведемо сабтехніки.

Запит (рисунок 3.10):

```
LET topTechnique = FIRST(
  FOR t IN technique
    SORT t.weight DESC
    LIMIT 1
    RETURN t
)
LET graphData = (
  FOR v, e, p IN 1..1 OUTBOUND topTechnique GRAPH "humint_graph"
    FILTER e.relation == "hasSubTechnique"
    RETURN {
      vertices: [
        { _id: topTechnique._id, _key: topTechnique._key, weight: topTechnique.weight, type: "technique" },
        { _id: v._id, _key: v._key, weight: v.weight, type: "subtechnique" }
      ],
      edges: [
        { _id: e._id, _from: e._from, _to: e._to, relation: e.relation }
      ]
    }
)
RETURN {
  vertices: UNIQUE(FLATTEN(graphData[*].vertices)),
  edges: UNIQUE(FLATTEN(graphData[*].edges))
}
```

Рисунок 3.10 – AQL запит для виводу техніки з найбільшою вагою і її сабтехнік

Результат виконання запиту (рисунок 3.11):

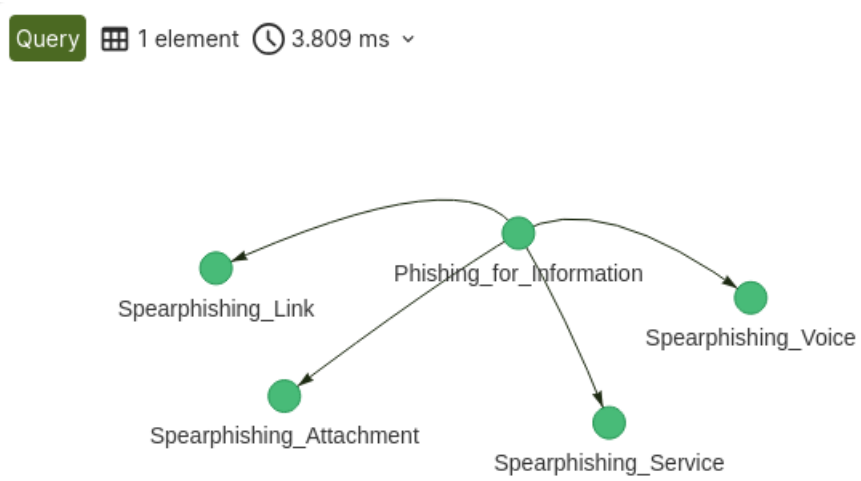

Рисунок 3.11 – Сабтехніки притаманні техніці з найбільшою вагою

Трохи модифікуємо попередній запит і спробуємо вивести інструменти лише тих сабтехнік, які мають вагу в діапазоні від 4.0 до 4.5.

Запит (рисунок 3.12):

```
LET topTechnique = FIRST(
  FOR t IN technique
    SORT t.weight DESC
    LIMIT 1
    RETURN t
)
LET subTechniqueData = (
  FOR v, e, p IN 1..1 OUTBOUND topTechnique._id GRAPH "humint_graph"
    FILTER e.relation == "hasSubTechnique" AND v.weight >= 4.0 AND v.weight <= 4.5
    RETURN {
      vertex: v,
      edge: e
    }
)
LET graphData = (
  FOR sub IN subTechniqueData
    LET subVertex = sub.vertex
    LET subEdge = sub.edge
    LET toolData = (
      FOR v, e, p IN 1..1 OUTBOUND subVertex._id GRAPH "humint_graph"
        FILTER e.relation == "hasTool" AND v.type == "tool"
        RETURN {
          toolVertex: v,
          toolEdge: e
        }
    )
    FOR tool IN toolData
      RETURN {
        vertices: [
          { _id: topTechnique._id, _key: topTechnique._key, weight: topTechnique.weight, type: "Technique" },
          { _id: subVertex._id, _key: subVertex._key, weight: subVertex.weight, type: "SubTechnique" },
          { _id: tool.toolVertex._id, _key: tool.toolVertex._key, weight: tool.toolVertex.weight, type: "tool" }
        ],
        edges: [
          { _id: subEdge._id, _from: subEdge._from, _to: subEdge._to, relation: subEdge.relation },
          { _id: tool.toolEdge._id, _from: tool.toolEdge._from, _to: tool.toolEdge._to, relation: tool.toolEdge.relation }
        ]
      }
)
RETURN {
  vertices: UNIQUE(FLATTEN(graphData[*].vertices)),
  edges: UNIQUE(FLATTEN(graphData[*].edges))
}
```

Рисунок 3.12 – Модифікований AQL запит

Результат виконання запиту (рисунок 3.13):

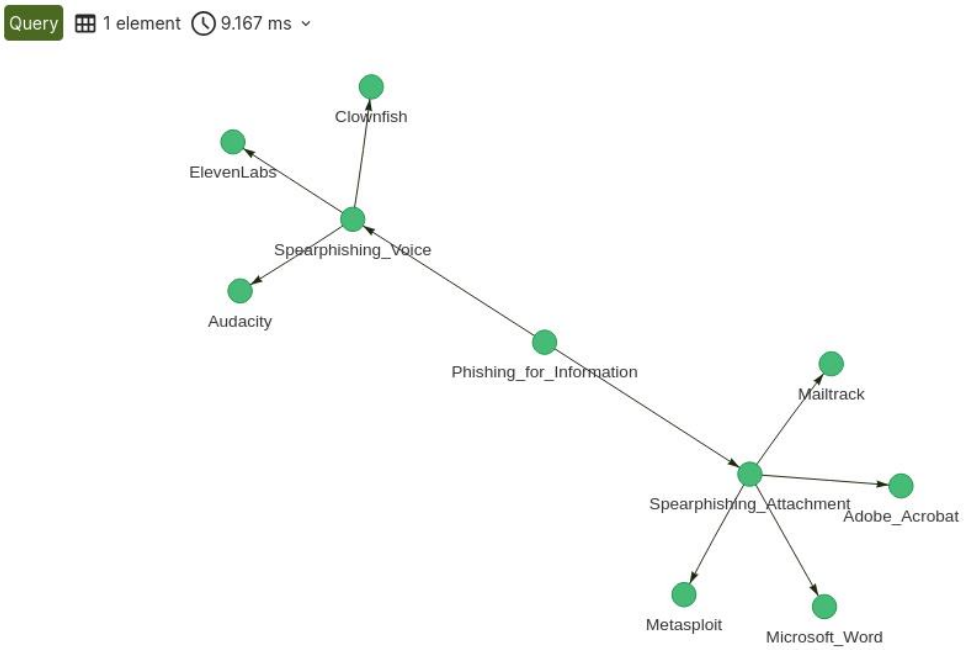

Рисунок 3.13 – Інструменти сабтехнік з певною вагою

Тобто, за допомогою графового представлення і комплексних запитів мовою AQL, можна приймати більш взважені рішення стосовно підбору підходів проведення процедури кіберрозвідки.

3.6 Приклад використання у ситуації з витоком даних конкурента

Нехай є дві компанії: компанія А і компанія Б, нам навіть не потрібно знати сферу їх діяльності, головне те, що вони між собою конкуренти. Уявімо ситуацію, коли в інфополі з’явилася новина про витік даних клієнтів компанії Б. Про це компанія А дізналася з якогось ЗМІ і вирішила провести розслідування. Оскільки компанія А має справу з інформацією з відкритих джерел, спершу, вона вирішує просто пошукати дані у відкритому доступі, тобто провести OSINT. Виведемо техніки, які притаманні OSINT методу.

Запит (рисунок 3.14):

```
FOR e IN `start-tech`
 FILTER e._from == "start/OSINT" AND e.relation == "hasTechnique"
 LET target = DOCUMENT(e._to)
 RETURN {
 _from: e._from,
 _to: e._to,
 _id: CONCAT("start-tech/", e._key),
 label: target.name
 }
```

Рисунок 3.14 – AQL запит для виводу технік OSINT

Результат виконання запиту (рисунок 3.15):

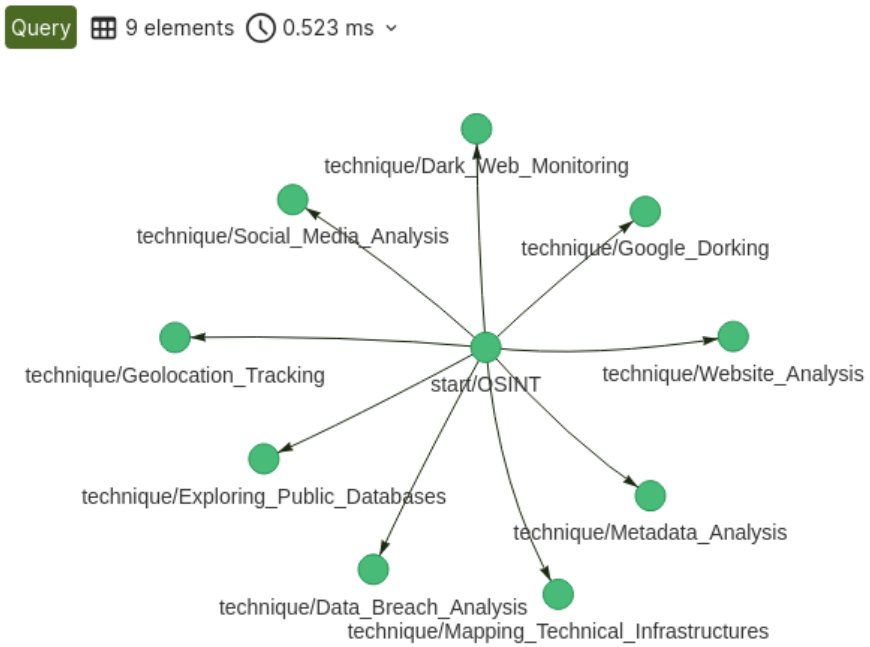

Рисунок 3.15 - Техніки притаманні OSINT методу у вигляді графу

За допомогою техніки Google_Dorking, яка передбачає використання розширених пошукових запитів, і її інструментів і процедур, компанія А шукає витоки даних у відкритих джерелах (рисунок 3.17).

Запит (рисунок 3.16):

```
LET paths = (
  FOR vertex, edge, path IN 1..2 OUTBOUND 'technique/Google_Dorking' GRAPH 'osint_graph'
    FILTER (LENGTH(path.edges) == 1 AND path.edges[0].relation == 'hasSubTechnique') OR
           (LENGTH(path.edges) == 2 AND path.edges[0].relation == 'hasSubTechnique' AND
            path.edges[1].relation IN ['hasProcedure', 'hasTool'])
    RETURN path
)

LET all_vertices = UNIQUE(FLATTEN(
  FOR p IN paths
    FOR v IN p.vertices
      RETURN {
        _id: v._id,
        _key: v._key,
        label: v.name || v._key,
        type: v.type,
        weight: v.weight
      }
))

LET all_edges = UNIQUE(FLATTEN(
  FOR p IN paths
    FOR e IN p.edges
      RETURN {
        _id: e._id,
        _from: e._from,
        _to: e._to,
        relation: e.relation
      }
))

RETURN {
  vertices: all_vertices,
  edges: all_edges
}

```

Рисунок 3.16 – AQL запит для виводу інструментів і процедур певної техніки

Результат виконання запиту (рисунок 3.17):

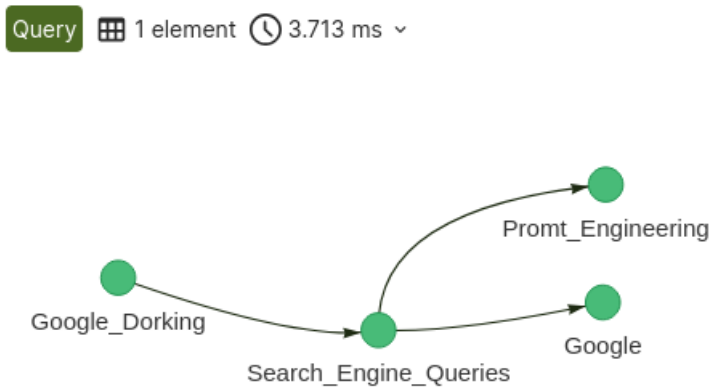

Рисунок 3.17 – інструменти і процедури техніки Google_Dorking

Компанії А вдалося знайти деякі витоки даних і наступним кроком розслідування вона обрала використання техніки Data_Breach_Analysis, яка передбачає процес аналізу даних, отриманих з витоків інформації з метою отримання корисної розвідувальної інформації. Задля ефективності застосування техніки, команда

розвідки, задає певний параметр: вага інструментів і процедур певних сабтехнік повинна перевищувати 3.

Запит (рисунок 3.18):

```
LET technique = DOCUMENT("technique/Data_Breach_Analysis")

LET graphData = (
  FOR v, e, p IN 1..2 OUTBOUND technique._id GRAPH "osint_graph"
    FILTER (
      (LENGTH(p.edges) == 1 AND e.relation == "hasSubTechnique") OR
      (LENGTH(p.edges) == 2 AND p.edges[0].relation == "hasSubTechnique" AND
       p.edges[1].relation IN ["hasProcedure", "hasTool"] AND v.weight > 3)
    )
    RETURN {
      vertices: [
        { _id: technique._id, _key: technique._key, weight: technique.weight, type: "Technique" },
        { _id: v._id, _key: v._key, weight: v.weight,
          type: v.type || (e.relation == "hasSubTechnique" ? "SubTechnique" : e.relation == "hasProcedure" ? "Procedure" :
            "Tool") }
      ],
      edges: [
        { _id: e._id, _from: e._from, _to: e._to, relation: e.relation }
      ]
    }
)

RETURN {
  vertices: UNIQUE(FLATTEN(graphData[*].vertices)),
  edges: UNIQUE(FLATTEN(graphData[*].edges))
}
```

Рисунок 3.18 – AQL запит для виводу інструментів і процедур певної техніки з вагою більше 3

Результат виконання запиту (рисунок 3.19):

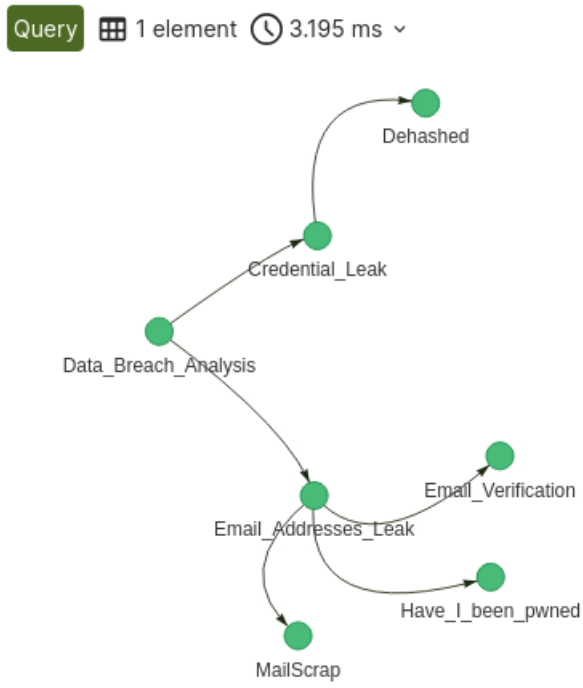

Рисунок 3.19 – Сабтехніки, інструменти і процедури притаманні аналізу витоків даних

Компанії А пощастило, вона змогла зібрати актуальні дані клієнтів компанії-конкурента Б і в подальшому зможе використовувати ці дані у свої цілях.

Отже, такий приклад демонструє, що рамках побудованого дерева рішень аналітики мали змогу швидко орієнтуватися у варіантах подальших дій, використовуючи вагові коефіцієнти для вибору найбільш доцільних технік, сабтехнік, інструментів і процедур.

Висновки до розділу 3

У розділі було здійснено практичну реалізацію моделі дерева рішень для процедур кіберрозвідки. На основі обраної методології були побудовані окремі графові структури для трьох методів – OSINT, HUMINT та SIGINT, з урахуванням технік, сабтехнік, інструментів та процедур.

Модель реалізована в середовищі ArangoDB, що забезпечило можливість зберігання даних у вигляді документів і візуалізації графа. В процесі побудови було створено зв’язки між ними відповідно до обраної структури. Для кожного елемента моделі було призначено вагові коефіцієнти, що дозволяють оцінювати ефективність та доцільність використання окремих елементів у розвідці.

На прикладі HUMINT і OSINT методів було підтверджено практичну придатність дерева рішень, як інструменту формалізації та оптимізації процесів кіберрозвідки.

ВИСНОВКИ

У ході виконання дипломної роботи було розглянуто процес кіберрозвідки в контексті кібербезпеки. Описано його основні аспекти, такі як життєвий цикл, вектори застосування і класифікацію.

Було описано теоретичні аспекти моделі дерева рішень і запропоновано її використання в якості інструменту для структуризації і формалізації процесу кіберрозвідки. Також було запропоновано модель ідентифікації сутностей і зв’язків між ними, яка була універсальною для методів кіберрозвідки і грала ключову роль у побудові дерев.

Дослідивши можливості графових баз даних, було обрано мультимодельну графову бази даних ArangoDB в якості інструменту для практичної реалізації ідеї. Практичний розділ роботи завершився демонстрацією прикладу застосування такого підходу: за допомогою запитів AQL було продемонстровано приклад роботи з моделлю. Приклад полягав у знаходженні техніки з найвищою вагою і виводу пов’язаних з нею, маючих вагу в певному діапазоні, сабтехнік й інструментів. Це дає змогу аналітику в будь-який момент, спираючись на ваги вузлів, оцінювати якість проведення процедури розвідки та розглядати альтернативні шляхи.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

[1] УРЯДОВІ КІБЕРФАХІВЦІ ЗА 2024 РІК ОПРАЦЮВАЛИ 4315 КІБЕРІНЦИДЕНТІВ [Електронний ресурс] // MediaSapiens. – 2025. – Режим доступу до ресурсу: https://ms.detector.media/kiberbezpeka/post/37156/2025-01-08-uryadovi-kiberfakhivtsi-za-2024-rik-opratsyuvaly-4315-kiberintsydentiv

[2] MITRE ATT&CK: Groups [Електронний ресурс] // MITRE. – Режим доступу до ресурсу: https://attack.mitre.org/groups/

[3] MITRE ATT&CK: Groups (v3) [Електронний ресурс] // MITRE. – Режим доступу до ресурсу: https://attack.mitre.org/versions/v3/groups/

[4] РОЗВІДКА КІБЕРЗАГРОЗ [Електронний ресурс] // IITD. – Режим доступу до ресурсу: https://iitd.ua/ru/razvedka-kiberugroz/

[5] WHAT IS CYBERTHREAT INTELLIGENCE (CTI) [Електронний ресурс] // Palo Alto Networks. – Режим доступу до ресурсу: https://www.paloaltonetworks.com/cyberpedia/what-is-cyberthreat-intelligence-cti

[6] CTI METHODS & METHODOLOGY [Електронний ресурс] // FIRST. – Режим доступу до ресурсу: https://www.first.org/global/sigs/cti/curriculum/methods-methodology

[7] Зубок В.Ю., Ланде Д.В. Корпоративна мережа з позиції противника. Рекогностування навпаки // Телеком. – 2009. – №11. – С. 42–47. – [Електронний ресурс]. – Режим доступу до ресурсу: http://dwl.kiev.ua/art/recogn/recogn.pdf

[8] WHAT IS CYBER RECONNAISSANCE? [Електронний ресурс] // SentinelOne. – Режим доступу до ресурсу: https://www.sentinelone.com/cybersecurity-101/threat-intelligence/what-is-cyber-reconnaissance/

[9] CYBERSECURITY 101: THREAT INTELLIGENCE [Електронний ресурс] // CrowdStrike. – Режим доступу до ресурсу: https://www.crowdstrike.com/en-us/cybersecurity-101/threat-intelligence/

[10] Wilhoit, Kyle, and Joseph Opacki. Operationalizing Threat Intelligence. Packt Publishing, 2022.

[11] A COMPREHENSIVE OVERVIEW OF INTELLIGENCE GATHERING METHODS [Електронний ресурс] // The Kootneeti. – 2022. – Режим доступу до ресурсу: https://thekootneeti.in/2022/08/03/a-comprehensive-overview-of-intelligence-gathering-methods-humint-sigint-imint-and-osint/

[12] Garner, Godfrey. The Five Disciplines of Intelligence Collection. – Lanham: Rowman & Littlefield, 2016. – [Електронний ресурс]. – Режим доступу до ресурсу: https://books.google.com.ua/books?id=rdI5DQAAQBAJ

[13] Jean Nestor M. Dahj. Mastering Cyber Intelligence. Packt Publishing, 2022.

[14] WHAT IS THREAT HUNTING [Електронний ресурс] // IBM. – Режим доступу до ресурсу: https://www.ibm.com/think/topics/threat-hunting

[15] ДЕРЕВО РІШЕНЬ: ПРИНЦИП РОБОТИ [Електронний ресурс] // UA5.org. – Режим доступу до ресурсу: https://ua5.org/algorithm/1976-derevo-rishen.html

[16] Кушлик-Дивульська, О.І., та Кушлик, Б.Р. Основи теорії прийняття рішень. Київ, 2014. [Електронний ресурс]. – Режим доступу до ресурсу: https://ela.kpi.ua/server/api/core/bitstreams/f9d59169-7c45-4661-8165-6931d004ca62/content

[17] Pashynska N., Snytyuk V., Putrenko V., Musienko A. A decision tree in a classification of fire hazard factors // Eastern-European Journal of Enterprise Technologies. – 2016. – №5. – С. 32–37. – DOI: 10.15587/1729-4061.2016.79868. – Режим доступу: https://www.researchgate.net/publication/309874046_A_decision_tree_in_a_classification_of_fire_hazard_factors

[18] THE MATHEMATICS BEHIND DECISION TREES [Електронний ресурс] // Fritz.ai. – Режим доступу до ресурсу: https://fritz.ai/the-mathematics-behind-decision-trees/

[19] WHAT IS RANDOM FOREST [Електронний ресурс] // IBM. – Режим доступу до ресурсу: https://www.ibm.com/think/topics/random-forest

[20] Що таке графова база даних [Електронний ресурс] // Amazon Web Services. – Режим доступу до ресурсу: https://aws.amazon.com/ru/nosql/graph/#topic-0

[21] RDF N-Triples [Електронний ресурс] // World Wide Web Consortium (W3C). – Режим доступу до ресурсу: https://www.w3.org/TR/rdf12-n-triples/

[22] ArangoDB Official Documentation [Електронний ресурс] // ArangoDB. – Режим доступу до ресурсу: https://docs.arangodb.com/stable/about-arangodb/

ДОДАТОК А ЗАПИТИ ДО БАЗИ ДАНИХ

```
FOR toolName IN ["Gophish", "Evilginx2", "King_Phisher"]
  INSERT { _key: toolName, type: "tool" } INTO spearphishing_service

FOR procName IN ["Domain_Spoofing", "Legitimate_Service_Imitation"]
  INSERT { _key: procName, type: "procedure" } INTO spearphishing_service

FOR toolName IN ["Microsoft_Word", "Adobe_Acrobat", "Metasploit", "Mailtrack"]
  INSERT { _key: toolName, type: "tool" } INTO spearphishing_attachment

FOR procName IN ["Spam", "Payload_Disguise", "Deepfake"]
  INSERT { _key: procName, type: "procedure" } INTO spearphishing_attachment

FOR toolName IN ["HTTrack", "SET", "Bitly", "TinyURL"]
  INSERT { _key: toolName, type: "tool" } INTO spearphishing_link

FOR procName IN ["Legitimate_Service_Imitation"]
  INSERT { _key: procName, type: "procedure" } INTO spearphishing_link

FOR toolName IN ["ElevenLabs", "Clownfish", "Audacity"]
  INSERT { _key: toolName, type: "tool" } INTO spearphishing_voice

FOR procName IN ["Deepfake", "Vishing_Scripting"]
  INSERT { _key: procName, type: "procedure" } INTO spearphishing_voice

FOR toolName IN ["Telegram", "Discord", "Stripe", "PayPal"]
  INSERT { _key: toolName, type: "tool" } INTO financial_incentive

FOR procName IN ["Payment_Negotiation"]
  INSERT { _key: procName, type: "procedure" } INTO financial_incentive

FOR procName IN ["Career_Promise", "Preferential_Treatment", "Favor_Exchange"]
  INSERT { _key: procName, type: "procedure" } INTO non_financial_incentive

FOR toolName IN ["ProtonMail", "OnionShare"]
  INSERT { _key: toolName, type: "tool" } INTO private_life

FOR procName IN ["Compromising_Materials", "Identity_Concealment", "Impersonation"]
  INSERT { _key: procName, type: "procedure" } INTO private_life
```

```
FOR toolName IN ["SpiderFoot", "LinkedIn"]
  INSERT { _key: toolName, type: "tool" } INTO professional_liability

FOR procName IN ["Compromising_Materials", "Identity_Concealment", "Impersonation"]
  INSERT { _key: procName, type: "procedure" } INTO professional_liability

FOR toolName IN ["LockBit", "BlackCat", "Cobalt_Strike"]
  INSERT { _key: toolName, type: "tool" } INTO ransomware_extortion

FOR procName IN ["Data_Encryption", "Data_Destruction", "Identity_Concealment"]
  INSERT { _key: procName, type: "procedure" } INTO ransomware_extortion

FOR procName IN ["Formal_Conversation", "Informal_Conversation"]
  INSERT { _key: procName, type: "procedure" } INTO direct

FOR toolName IN ["Zoom", "Signal", "Telegram", "Discord", "Gmail", "Yahoo"]
  INSERT { _key: toolName, type: "tool" } INTO direct

FOR procName IN ["Formal_Conversation", "Informal_Conversation"]
  INSERT { _key: procName, type: "procedure" } INTO mediated

FOR toolName IN ["Zoom", "Signal", "Telegram", "Discord", "Gmail", "Yahoo"]
  INSERT { _key: toolName, type: "tool" } INTO mediated
```

```
FOR tool IN ["Gophish", "Evilginx2", "King_Phisher"]
  INSERT {
    _from: "spearphishing_service/Spearphishing_Service",
    _to: CONCAT("spearphishing_service/", tool),
    relation: "hasTool"
  } IN `tech-spearphishing_service`

FOR proc IN ["Domain_Spoofing", "Legitimate_Service_Imitation"]
  INSERT {
    _from: "spearphishing_service/Spearphishing_Service",
    _to: CONCAT("spearphishing_service/", proc),
    relation: "hasProcedure"
  } IN `tech-spearphishing_service`

FOR tool IN ["Microsoft_Word", "Adobe_Acrobat", "Metasploit", "Mailtrack"]
  INSERT {
    _from: "spearphishing_attachment/Spearphishing_Attachment",
    _to: CONCAT("spearphishing_attachment/", tool),
    relation: "hasTool"
  } IN `tech-spearphishing_attachment`

FOR proc IN ["Spam", "Payload_Disguise", "Deepfake"]
  INSERT {
    _from: "spearphishing_attachment/Spearphishing_Attachment",
    _to: CONCAT("spearphishing_attachment/", proc),
    relation: "hasProcedure"
  } IN `tech-spearphishing_attachment`

FOR tool IN ["HTTrack", "SET", "Bitly", "TinyURL"]
  INSERT {
    _from: "spearphishing_link/Spearphishing_Link",
    _to: CONCAT("spearphishing_link/", tool),
    relation: "hasTool"
  } IN `tech-spearphishing_link`
```

```
FOR proc IN ["Legitimate_Service_Imitation"]
  INSERT {
    _from: "spearphishing_link/Spearphishing_Link",
    _to: CONCAT("spearphishing_link/", proc),
    relation: "hasProcedure"
  } IN `tech-spearphishing_link`

FOR tool IN ["ElevenLabs", "Clownfish", "Audacity"]
  INSERT {
    _from: "spearphishing_voice/Spearphishing_Voice",
    _to: CONCAT("spearphishing_voice/", tool),
    relation: "hasTool"
  } IN `tech-spearphishing_voice`

FOR proc IN ["Deepfake", "Vishing_Scripting"]
  INSERT {
    _from: "spearphishing_voice/Spearphishing_Voice",
    _to: CONCAT("spearphishing_voice/", proc),
    relation: "hasProcedure"
  } IN `tech-spearphishing_voice`

FOR tool IN ["Telegram", "Discord", "Stripe", "PayPal"]
  INSERT {
    _from: "financial_incentive/Financial_Incentive",
    _to: CONCAT("financial_incentive/", tool),
    relation: "hasTool"
  } IN `tech-financial_incentive`

FOR proc IN ["Payment_Negotiation"]
  INSERT {
    _from: "financial_incentive/Financial_Incentive",
    _to: CONCAT("financial_incentive/", proc),
    relation: "hasProcedure"
  } IN `tech-financial_incentive`
```

```
FOR proc IN ["Career_Promise", "Preferential_Treatment", "Favor_Exchange"]
  INSERT {
    _from: "non_financial_incentive/Non_Financial_Incentive",
    _to: CONCAT("non_financial_incentive/", proc),
    relation: "hasProcedure"
  } IN `tech-non_financial_incentive`

FOR tool IN ["ProtonMail", "OnionShare"]
  INSERT {
    _from: "private_life/Private_Life",
    _to: CONCAT("private_life/", tool),
    relation: "hasTool"
  } IN `tech-private_life`

FOR proc IN ["Compromising_Materials", "Identity_Concealment", "Impersonation"]
  INSERT {
    _from: "private_life/Private_Life",
    _to: CONCAT("private_life/", proc),
    relation: "hasProcedure"
  } IN `tech-private_life`

FOR tool IN ["SpiderFoot", "LinkedIn"]
  INSERT {
    _from: "professional_liability/Professional_Liability",
    _to: CONCAT("professional_liability/", tool),
    relation: "hasTool"
  } IN `tech-professional_liability`

FOR proc IN ["Compromising_Materials", "Identity_Concealment", "Impersonation"]
  INSERT {
    _from: "professional_liability/Professional_Liability",
    _to: CONCAT("professional_liability/", proc),
    relation: "hasProcedure"
  } IN `tech-professional_liability`
```

```
FOR tool IN ["LockBit", "BlackCat", "Cobalt_Strike"]
  INSERT {
    _from: "ransomware_extortion/Ransomware_Extortion",
    _to: CONCAT("ransomware_extortion/", tool),
    relation: "hasTool"
  } IN `tech-ransomware_extortion`

FOR proc IN ["Data_Encryption", "Data_Destruction", "Identity_Concealment"]
  INSERT {
    _from: "ransomware_extortion/Ransomware_Extortion",
    _to: CONCAT("ransomware_extortion/", proc),
    relation: "hasProcedure"
  } IN `tech-ransomware_extortion`

FOR tool IN ["Zoom", "Signal", "Telegram", "Discord", "Gmail", "Yahoo"]
  INSERT {
    _from: "direct/Direct",
    _to: CONCAT("direct/", tool),
    relation: "hasTool"
  } IN `tech-direct`

FOR proc IN ["Formal_Conversation", "Informal_Conversation"]
  INSERT {
    _from: "direct/Direct",
    _to: CONCAT("direct/", proc),
    relation: "hasProcedure"
  } IN `tech-direct`

FOR tool IN ["Zoom", "Signal", "Telegram", "Discord", "Gmail", "Yahoo"]
  INSERT {
    _from: "mediated/Mediated",
    _to: CONCAT("mediated/", tool),
    relation: "hasTool"
  } IN `tech-mediated`
```

```
FOR proc IN ["Formal_Conversation", "Informal_Conversation"]
  INSERT {
    _from: "mediated/Mediated",
    _to: CONCAT("mediated/", proc),
    relation: "hasProcedure"
  } IN `tech-mediated`
```

```
FOR toolName IN ["Wireshark", "tcpdump", "Zeek", "Tshark", "Ntopng"]
  INSERT { _key: toolName, type: "tool" } INTO passive_capture

FOR procName IN ["Packet_Sniffing", "Session_Reassembly", "Protocol_Analysis", "Filtering_Alerting",
"Traffic_Segregation"]
  INSERT { _key: procName, type: "procedure" } INTO passive_capture

FOR toolName IN ["Ettercap", "Bettercap", "Mitmproxy", "Burp_Suite"]
  INSERT { _key: toolName, type: "tool" } INTO active_interception

FOR procName IN ["ARP_Spoofing", "SSL_Strip", "Traffic_Modification", "TLS_Downgrade",
"Session_Hijacking"]
  INSERT { _key: procName, type: "procedure" } INTO active_interception

FOR toolName IN ["Wireshark", "tcpdump", "Zeek", "Tshark", "RTL-SDR", "OsmocomBB"]
  INSERT { _key: toolName, type: "tool" } INTO passive_capture

FOR procName IN ["Cellular_Sniffing", "Packet_Sniffing", "Session_Reassembly", "Protocol_Analysis",
"Filtering_Alerting", "Traffic_Segregation"]
  INSERT { _key: procName, type: "procedure" } INTO passive_capture

FOR toolName IN ["Ettercap", "Bettercap", "mitmproxy", "Burp_Suite", "Stingray", "OpenBTS", "SigPloit"]
  INSERT { _key: toolName, type: "tool" } INTO active_interception

FOR procName IN ["Cellular_Sniffing", "IMSI_Catching", "Traffic_Modification", "TLS_Downgrade",
"Session_Hijacking", "Protocol_Exploitation"]
  INSERT { _key: procName, type: "procedure" } INTO active_interception

FOR toolName IN ["Realtek_SDR", "GQRX", "HackRF", "Universal_Radio_Hacker"]
  INSERT { _key: toolName, type: "tool" } INTO rf_monitoring

FOR procName IN ["Signal_Capture", "Signal_Classification", "Demodulation"]
  INSERT { _key: procName, type: "procedure" } INTO rf_monitoring

FOR toolName IN ["Ubertooth", "Wireshark", "BlueHydra"]
  INSERT { _key: toolName, type: "tool" } INTO bluetooth

FOR procName IN ["Device_Discovery", "MAC_Tracking", "Packet_Logging", "Pairing_Analysis"]
  INSERT { _key: procName, type: "procedure" } INTO bluetooth

FOR toolName IN ["GNURadio", "SatNOGS", "SDRSharp", "Orbitron"]
  INSERT { _key: toolName, type: "tool" } INTO beacon_tracking

FOR procName IN ["Beacon_Decoding", "Orbit_Correlation", "Downlink_Capture"]
  INSERT { _key: procName, type: "procedure" } INTO beacon_tracking

FOR toolName IN ["GNURadio", "GR-Satellites", "Multimon-NG", "SigDigger"]
  INSERT { _key: toolName, type: "tool" } INTO telemetry_sniffing

FOR procName IN ["Signal_Demodulation", "Data_Extraction"]
  INSERT { _key: procName, type: "procedure" } INTO telemetry_sniffing

FOR toolName IN ["Joy", "Zeek", "Wireshark", "JA3"]
  INSERT { _key: toolName, type: "tool" } INTO tls_fingerprinting

FOR procName IN ["Anomaly_Detection", "Metadata_Extraction"]
  INSERT { _key: procName, type: "procedure" } INTO tls_fingerprinting

FOR toolName IN ["tshark", "Zeek"]
  INSERT { _key: toolName, type: "tool" } INTO timing_analysis

FOR toolName IN ["Wireshark", "Sipdump", "Sngrep", "VoIPmonitor", "Cain"]
  INSERT { _key: toolName, type: "tool" } INTO sip_protocol_analysis

FOR procName IN ["Registration_Sniffing", "Voicemail_Extraction", "Call_Tracing"]
  INSERT { _key: procName, type: "procedure" } INTO sip_protocol_analysis

FOR toolName IN ["Wireshark", "tshark", "rtpbreak", "VoIPong"]
  INSERT { _key: toolName, type: "tool" } INTO rtp_stream_capture

FOR procName IN ["Audio_Reconstruction", "Codec_Detection"]
  INSERT { _key: procName, type: "procedure" } INTO rtp_stream_capture
```

```
FOR name IN ["Ettercap", "Bettercap", "Mitmproxy", "Burp_Suite"]
  INSERT {
    _from: "active_interception/Active_Interception",
    _to: CONCAT("active_interception/", name),
    relation: "hasTool"
  } IN `tech-active_interception`

FOR name IN ["ARP_Spoofing", "SSL_Strip", "Traffic_Modification", "TLS_Downgrade", "Session_Hijacking"]
  INSERT {
    _from: "active_interception/Active_Interception",
    _to: CONCAT("active_interception/", name),
    relation: "hasProcedure"
  } IN `tech-active_interception`

FOR name IN ["Wireshark", "tcpdump", "Zeek", "Tshark", "RTL-SDR", "OsmocomBB"]
  INSERT {
    _from: "passive_capture/Passive_Capture",
    _to: CONCAT("passive_capture/", name),
    relation: "hasTool"
  } IN `tech-passive_capture`

FOR name IN ["Cellular_Sniffing", "Packet_Sniffing", "Session_Reassembly", "Protocol_Analysis",
"Filtering_Alerting", "Traffic_Segregation"]
  INSERT {
    _from: "passive_capture/Passive_Capture",
    _to: CONCAT("passive_capture/", name),
    relation: "hasProcedure"
  } IN `tech-passive_capture`

FOR name IN ["Ettercap", "Bettercap", "mitmproxy", "Burp_Suite", "Stingray", "OpenBTS", "SigPloit"]
  INSERT {
    _from: "active_interception/Active_Interception",
    _to: CONCAT("active_interception/", name),
    relation: "hasTool"
  } IN `tech-active_interception`

FOR name IN ["Cellular_Sniffing", "IMSI_Catching", "Traffic_Modification", "TLS_Downgrade",
"Session_Hijacking", "Protocol_Exploitation"]
  INSERT {
    _from: "active_interception/Active_Interception",
    _to: CONCAT("active_interception/", name),
    relation: "hasProcedure"
  } IN `tech-active_interception`
```

```
FOR name IN ["Realtek_SDR", "GQRX", "HackRF", "Universal_Radio_Hacker"]
  INSERT {
    _from: "rf_monitoring/RF_Monitoring",
    _to: CONCAT("rf_monitoring/", name),
    relation: "hasTool"
  } IN `tech-rf_monitoring`

FOR name IN ["Signal_Capture", "Signal_Classification", "Demodulation"]
  INSERT {
    _from: "rf_monitoring/RF_Monitoring",
    _to: CONCAT("rf_monitoring/", name),
    relation: "hasProcedure"
  } IN `tech-rf_monitoring`

FOR name IN ["Ubertooth", "Wireshark", "BlueHydra"]
  INSERT {
    _from: "bluetooth/Bluetooth",
    _to: CONCAT("bluetooth/", name),
    relation: "hasTool"
  } IN `tech-bluetooth`

FOR name IN ["Device_Discovery", "MAC_Tracking", "Packet_Logging", "Pairing_Analysis"]
  INSERT {
    _from: "bluetooth/Bluetooth",
    _to: CONCAT("bluetooth/", name),
    relation: "hasProcedure"
  } IN `tech-bluetooth`

FOR name IN ["GNURadio", "SatNOGS", "SDRSharp", "Orbitron"]
  INSERT {
    _from: "beacon_tracking/Beacon_Tracking",
    _to: CONCAT("beacon_tracking/", name),
    relation: "hasTool"
  } IN `tech-beacon_tracking`

FOR name IN ["Beacon_Decoding", "Orbit_Correlation", "Downlink_Capture"]
  INSERT {
    _from: "beacon_tracking/Beacon_Tracking",
    _to: CONCAT("beacon_tracking/", name),
    relation: "hasProcedure"
  } IN `tech-beacon_tracking`
```

```
FOR name IN ["GNURadio", "GR-Satellites", "Multimon-NG", "SigDigger"]
  INSERT {
    _from: "telemetry_sniffing/Telemetry_Sniffing",
    _to: CONCAT("telemetry_sniffing/", name),
    relation: "hasTool"
  } IN `tech-telemetry_sniffing`

FOR name IN ["Signal_Demodulation", "Data_Extraction"]
  INSERT {
    _from: "telemetry_sniffing/Telemetry_Sniffing",
    _to: CONCAT("telemetry_sniffing/", name),
    relation: "hasProcedure"
  } IN `tech-telemetry_sniffing`

FOR name IN ["Joy", "Zeek", "Wireshark", "JA3"]
  INSERT {
    _from: "tls_fingerprinting/TLS_Fingerprinting",
    _to: CONCAT("tls_fingerprinting/", name),
    relation: "hasTool"
  } IN `tech-tls_fingerprinting`

FOR name IN ["Anomaly_Detection", "Metadata_Extraction"]
  INSERT {
    _from: "tls_fingerprinting/TLS_Fingerprinting",
    _to: CONCAT("tls_fingerprinting/", name),
    relation: "hasProcedure"
  } IN `tech-tls_fingerprinting`

FOR name IN ["tshark", "Zeek"]
  INSERT {
    _from: "timing_analysis/Timing_Analysis",
    _to: CONCAT("timing_analysis/", name),
    relation: "hasTool"
  } IN `tech-timing_analysis`

FOR name IN ["Wireshark", "Sipdump", "Sngrep", "VoIPmonitor", "Cain"]
  INSERT {
    _from: "sip_protocol_analysis/SIP_Protocol_Analysis",
    _to: CONCAT("sip_protocol_analysis/", name),
    relation: "hasTool"
  } IN `tech-sip_protocol_analysis`
```

```
FOR name IN ["Registration_Sniffing", "Voicemail_Extraction", "Call_Tracing"]
  INSERT {
    _from: "sip_protocol_analysis/SIP_Protocol_Analysis",
    _to: CONCAT("sip_protocol_analysis/", name),
    relation: "hasProcedure"
  } IN `tech-sip_protocol_analysis`

FOR name IN ["Wireshark", "tshark", "rtpbreak", "VoIPong"]
  INSERT {
    _from: "rtp_stream_capture/RTP_Stream_Capture",
    _to: CONCAT("rtp_stream_capture/", name),
    relation: "hasTool"
  } IN `tech-rtp_stream_capture`

FOR name IN ["Audio_Reconstruction", "Codec_Detection"]
  INSERT {
    _from: "rtp_stream_capture/RTP_Stream_Capture",
    _to: CONCAT("rtp_stream_capture/", name),
    relation: "hasProcedure"
  } IN `tech-rtp_stream_capture`
```