МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,
управління та адміністрування
Кафедра інформаційних
технологій

Кваліфікаційна робота бакалавра

на тему: Розробка технології автоматизації розвідки по відкритих джерелах інформації

Виконав студент групи К-19
спеціальності 122 Комп'ютерні науки
Ісламов Нурали

Керівник д. техн. наук, професор
Казакова Надія Феліксівна

Рецензент Т.в.о. директора КП
"Обласний інформаційно-
аналітичний центр" Попов В.Л.

Одеса 2023

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

OSINT – розвідка на основі відкритих джерел.

POI – points of Interest.

SaaS – програмне забезпечення як послуга.

MAU – місячні активні користувачі, термін, що означає кількість унікальних клієнтів, які взаємодіяли з продуктами або послугами компанії протягом місяця.

QR code – двовимірний штрих-код, який надає інформацію.

CVE – common vulnerabilities and exposures, база даних вразливостей.

CVSS – common vulnerability scoring system, числовий рейтинг вразливостей.

ІП – пошук інформації.

SQL – структурована мова запитів.

KBPS – одиниця вимірювання швидкості передачі інформації (кілобіт на секунду).

IDD – intelligence driven defence.

API – інтерфейс прикладного програмування, набір підпрограм, комунікаційних протоколів та інструментів для створення програмного забезпечення.

XML – розширювана мова розмітки, стандарт для створення мови розмітки ієрархічних даних, якими обмінюються різні програми.

JSON – javascript object notation, текстовий формат для обміну даними між комп'ютерами.

PGP – pretty good privacy, бібліотека функцій, які можуть виконувати операції шифрування та цифрового підпису комп'ютерних програм, а також повідомлень, файлів та іншої інформації.

ВСТУП

Щоб захиститися від сучасних загроз, потрібно бути на крок попереду. Про активне розуміння вашої організації та її цифрового сліду є частиною процесу, який називається "Інформаційна розвідка для захисту інформації"(Information Defence Intelligence, IDD). Знання того, яка інформація доступна з точки зору зловмисника, може допомогти вам зрозуміти, де знаходяться потенційні слабкі місця, і підготуватися до того, що реальний зловмисник спробує їх використати.

OSINT є швидкозростаючою, багатогранною інформаційною технологією, і все більше організацій за межами фінансових фірм, федеральних агентств і правоохоронних органів інвестують в інструменти, які можуть полегшити роботу аналітиків і скоротити час на вирішення питань, пов'язаних з автоматизацією методів OSINT.

Актуальність теми кваліфікаційної роботи бакалавра пов'язана з поширенням явища витоку даних та необхідністю розробки автоматизованих систем збору інформації для скорочення часу, що витрачається аналітиками з кібербезпеки.

Предметом дослідження є безпека корпоративних та персональних даних.

Об'єктом дослідження є OSINT-методика інструменти для визначення стану захищеності корпоративних та персональних даних.

Завдання дослідження:

- проаналізувати способи та методи пошуку інформації з використанням існуючих інструментів та технічних рішень;
- дослідити принципи роботи автоматизованих систем пошуку інформації;
- пояснити явище витоку корпоративної та персональної інформації та ситуацію загрози в сучасному кіберпросторі;

- визначити ефективність автоматизації процесу пошуку інформації;
- розробити інструментарій для автоматизації інформаційного пошуку шляхом інтеграції існуючих інструментів.

Мета дипломної роботи на тему "Розробка технології автоматизації розвідки по відкритих джерелах інформації" полягає у створенні ефективної технології, яка дозволить автоматизувати процес збору та аналізу інформації з відкритих джерел. Конкретні цілі дослідження включають:

- розробку системи збору інформації;
- розвиток алгоритмів обробки даних;
- впровадження інтелектуального аналізу;
- розробку інструментарію.

Метою дипломної роботи є розробка ефективної технології, яка допоможе підвищити швидкість, точність та надійність розвідки по відкритих джерелах інформації, сприяючи прийняттю інформованих рішень та покращенню процесів відстеження трендів, конкурентної інтелігенції, аналізу громадської думки тощо.

Дипломна робота містить в собі 58 сторінок, 35 рисунків та 15 посилань.

1 АНАЛІЗ КОНЦЕПЦІЙ І МЕТОДІВ OSINT

1.1 OSINT як частина системи кіберзахисту

Сьогодні OSINT, яка є дослідницькою галуззю кібербезпеки, є надзвичайно важливим інструментом. OSINT охоплює пошук, відбір і збір розвідувальної інформації з відкритих джерел, а також її аналіз. Зазвичай, цей процес включає моніторинг, аналіз та розслідування доступної інформації з Інтернету.

Розвідка з відкритими джерелами забезпечує всі необхідні методи і техніки для зберігання, аналізу та поширення інформації. Застосування OSINT в міжнародному співтоваристві стає все поширенішим для вирішення різноманітних проблем. Цей підхід має велику цінність завдяки своїм перевагам, таким як оперативність, доступність великої кількості інформації, наочність інформаційних потоків, а також простота використання та вартість [1].

На процес планування та підготовки до проведення OSINT-операцій впливають такі фактори:

- Ефективна інформаційна підтримка – значна частина необхідного довідкового матеріалу на тему використання інформації зібрана з відкритих джерел. Ця база даних формується шляхом збору інформації із засобів масової інформації. Накопичення даних з відкритих джерел є важливою особливістю OSINT.
- Релевантність – доступність, глибина і масштабність опублікованої інформації дозволяє людям знаходити потрібну інформацію без залучення спеціалізованих людських або технічних інформаційних інструментів.
- Спрощення процесу збору даних – OSINT надає необхідну інформацію і не потребує залучення зайвих технічних і людських методів розвідки.

- Глибший аналіз даних – в рамках процесу розвідки менеджери можуть детально аналізувати загальнодоступну інформацію і приймати відповідні рішення.
- Ефективність – значно скорочуваний час, необхідний для доступу до інформації в Інтернеті. Скорочення людино-годин, витрачених на пошук інформації на основі відкритих джерел, людей та їхніх стосунків. Швидкий доступ до цінної та релевантної інформації.
- Обсяг – можливість моніторити великі обсяги конкретних джерел інформації для пошуку необхідного контенту, людей і подій. Досвід показує, що фрагменти інформації, вміло зібрані з відкритих джерел, в цілому є такими ж або навіть більш важливими, ніж експертний розвідувальний звіт.
- Якість – інформація з відкритих джерел є більш надійною у порівнянні зі звітами спеціальних агентів.
- Ясність – у випадку використання OSINT надійність відкритих джерел або очевидна, або невідома, тоді як у випадку з даними, отриманими таємно, їх надійність завжди залишається під сумнівом.
- Доступність – будь-які секрети повинні бути захищені такими бар'єрами, як "засекреченість", допуск і обмеження доступу; у випадку з OSINT-даними, вони можуть бути легко передані будь-якій зацікавленій організації. На основі даних з Інтернету також можна проводити комплексні дослідження.
- Вартість – вартість збору даних за допомогою OSINT мінімальна і визначається виключно ціною використовуваних послуг.

Ситуація, що склалася, вимагає розробки спеціальних методів та інструментів для пошуку та аналізу мережевої інформації. Однак поки що не знайдено задовільних рішень для швидкої аналітичної обробки інформації, пошуку необхідних фактичних даних, виявлення та прогнозування тенденцій у суміжних галузях. Проблеми кількості та динаміки багатомовних інформаційних

ресурсів у глобальних мережах вимагають фундаментальних досліджень у таких галузях, як математика (теорія графів, комплексні моди), розпізнавання образів (класифікація, кластерний аналіз, нейронні мережі), комп'ютерна лінгвістика, цифрова обробка сигналів та нелінійний аналіз.

Обсяг доступних інформаційних ресурсів в глобальних мережах на сьогодні перевищує кілька сотень трильйонів документів, приміром, Facebook щодня генерує понад чотири петабайти даних [2].

Соціальні мережі містять великі обсяги цінних інформаційних даних, а акаунти в соціальних мережах часто опиняються в результаті скомпрометованими і зіпсованими даними. Нижче наведено статистику щодо кількості активних користувачів найпопулярніших соціальних мереж:

– Facebook (2022) – 3 мільярди щомісячних активних користувачів (MAU)

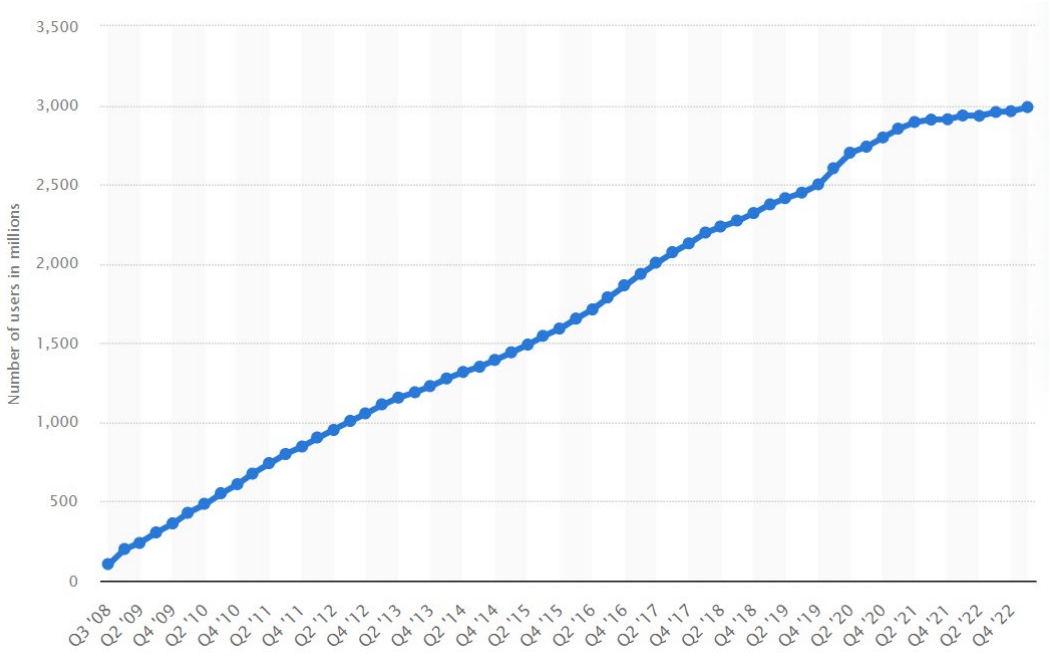


Рисунок 1.1 – Кількість активних користувачів у Facebook за даними statista.com.

– Instagram (2021) – 2 мільярди MAU.

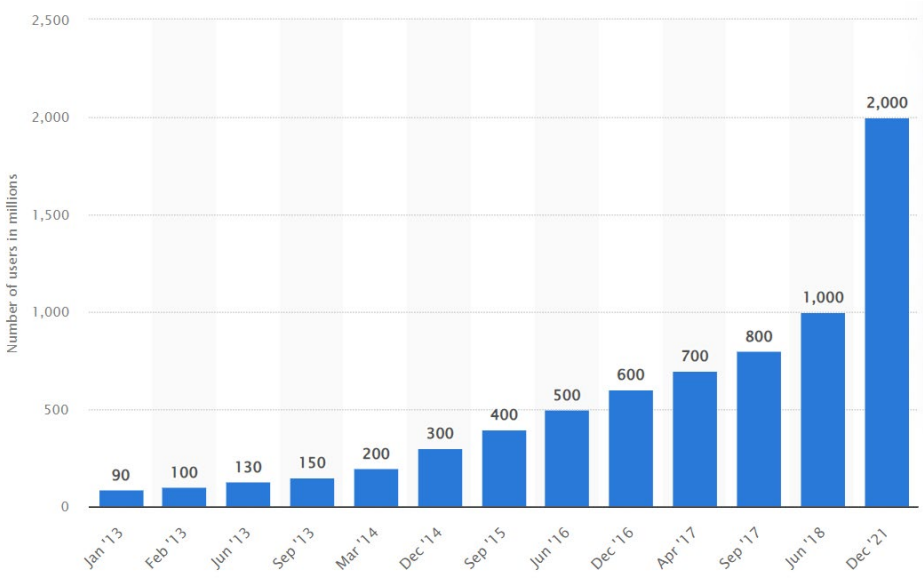


Рисунок 1.2 – Кількість активних користувачів в Instagram за даними statista.com

– Telegram (2022) – 700 млн MAUs.

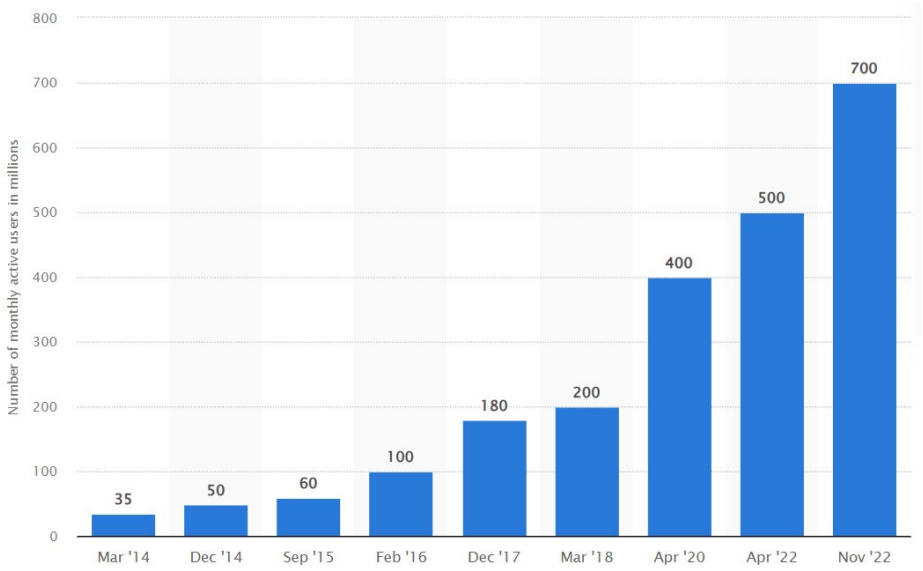


Рисунок 1.3 – Кількість активних користувачів у Telegram за даними statista.com.

Нещодавно великий злам, який призвів до компрометації даних мільйонів людей, став вагомою новиною. В даний час порушення, що торкаються

сотень мільйонів або навіть мільярдів людей, стали поширеним явищем. Порушення даних, на відміну від витоку даних, відноситься до несанкціонованої передачі інформації з організації до зовнішнього отримувача. Цей термін також застосовується, коли дані передаються як у фізичному, так і в цифровому форматі. У багатьох випадках цей тип даних знаходиться в відкритих джерелах і є складовою частиною розвідувальних операцій.

Станом на 2021 рік до найбільш серйозних витоків даних належать

1. "Компіляція витоків Менні" (Compilation of Manny Breaches, COMB)

Дата: 2 лютого 2021 року.

Кількість записів: 3,2 млрд.

Незаконний доступ був отриманий до таких даних: адреси електронної пошти, паролі та облікові дані [3].

2. Витік даних Facebook

Дата: 3 квітня 2021 року.

Кількість записів: 533 мільйони.

Скомпрометовано такі дані: номери телефонів, адреси електронної пошти, повні імена, адреси та дати народження.

3. Інцидент з витоком даних LinkedIn

Дата: 6 квітня 2021 року

Кількість записів: 500 мільйонів

Скомпрометовані дані: номери телефонів, адреси електронної пошти, повні імена, посилання на інші профілі в соціальних мережах та інші персональні дані [4].

Факт компрометації персональних даних або даних компанії є дуже важливим при аналізі вразливості системи, і на основі наявних даних можна розробити план для запобігання можливим сценаріям інцидентів.

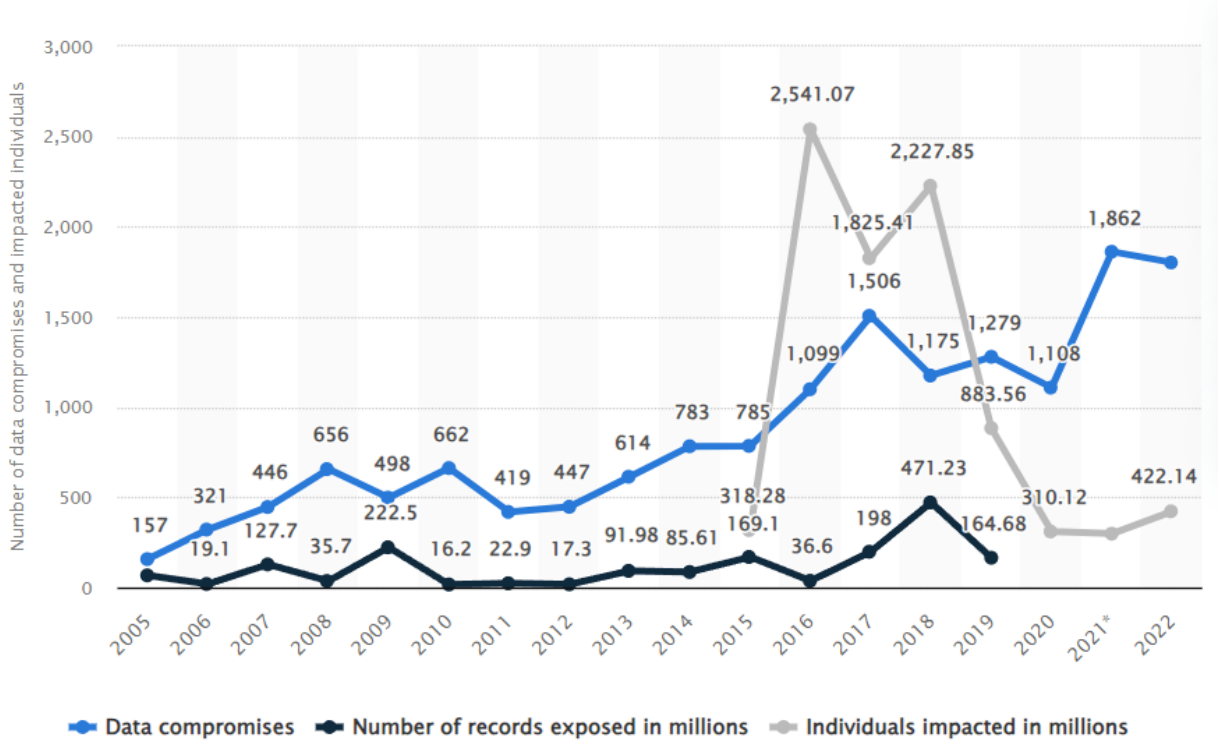


Рисунок 1.4 – Щорічна кількість зламаних даних і постраждалих осіб у Сполучених Штатах з 2005 по 2022 рік за даними statista.com.

У 2022 році, за даними tdwi.org, у США відбудеться загалом 1802 витоки даних. Тим часом, протягом того ж року понад 172,4 мільйона людей зазнали витоку даних, тобто випадкового розголошення конфіденційної інформації через неналежний рівень інформаційної безпеки.

1.2 Інструменти та методи OSINT

Основною вимогою до реалізації OSINT є пошук тільки на основі відкритих джерел, а це означає, що цей метод розвідки буде включати тільки пасивне сканування і загальнодоступну інформацію [5].

Пасивне сканування – це метод виявлення вразливостей, який ґрунтується на отриманні інформації з мережевих даних цільового комп'ютера без

прямої взаємодії. Для пасивного сканування часто використовується програмне забезпечення для перехоплення пакетів, яке дозволяє виявляти таку інформацію, як операційні системи, відомі протоколи, що працюють на нестандартних портах, а також активні мережеві програми з відомими вразливостями. Пасивне сканування може бути використане мережевими адміністраторами для перевірки безпеки або зловмисниками як попередній крок до активної атаки.

Однак, пасивне сканування має свої обмеження. Воно не таке детальне, як активне сканування вразливостей, і не може виявити програми, які не генерують мережевий трафік у даний момент. Крім того, цей тип сканування не може надійно розрізнити неправдиву інформацію, яка може бути надана з метою заплутування.

У сучасній глобальній економіці, що швидко розвивається, одноразової оцінки недостатньо. Натомість потрібен постійний моніторинг для виявлення ризиків для репутації (наприклад, постачальників, причетних до злочинів або корумпованого управління) або виробництва (наприклад, постачальників, які не платять працівникам, екологічних проблем, які можуть зашкодити наданню послуг). Технологія OSINT, заснована на семантичній технології, допомагає аналітикам збирати і відстежувати тисячі даних, що містяться в конкретних інформаційних потоках, пов'язаних з відкритим кодом, соціальними мережами і ланцюгами поставок.

Дослідницьке охоплення даних компанії за допомогою OSINT можна розділити на три категорії:

- люди;
- організації;
- домени.

Техніка OSINT для дослідження людей починається з визначення їхніх справжніх імен. Якщо справжнє ім'я та прізвище людини вже відоме, пошук можна розпочати, використовуючи їх як ключові слова. Однак слід зазначити,

що пошук за справжнім ім'ям людини часто призводить до помилкових спрацьовувань. Тому поєднання справжнього імені з додатковою інформацією, наприклад, місцезнаходженням, може зменшити кількість нерелевантних результатів, які не збігаються.

Наступний крок – визначення нікнеймів. Найпростіший спосіб визначити ім'я користувача – здійснити пошук за інформацією, необхідною для його пошуку. Це може бути комбінація імені та прізвища або доменне ім'я, що належить людині. Тип імені користувача, який використовує людина, може змінюватися залежно від того, наскільки легко вона хоче, щоб її ідентифікували.

Як і у випадку зі справжніми іменами, пошук за іменами користувачів може призвести до низки хибних спрацьовувань. Рисунок 1.5 ілюструє основні методи, що використовуються в робочому процесі OSINT [6]. Як видно з рисунка, перший етап передбачає збір даних. На цьому етапі збирається будь-яка інформація, яка може мати відношення до предмета дослідження. Наступним кроком є аналіз даних, який може включати в себе поділ інформації на особисту, корпоративну та мережеву. Під час аналізу перевіряється достовірність даних, виявляються взаємозв'язки та отримуються більш глибокі деталі. На завершальному етапі з наявної інформації витягуються точні факти, формулюється оцінка ризиків, визначаються слабкі місця об'єкта розслідування та розробляється стратегічне рішення, яке запобігає потенційному використанню конфіденційних даних проти їхнього власника.

Пошуки є більш успішними, якщо доступна адреса електронної пошти, за якою можна здійснити пошук. На відміну від справжнього імені або імені користувача, адреса електронної пошти є унікальною для цієї особи і не схильна до помилкових спрацьовувань. Маючи адресу електронної пошти, ви з більшою ймовірністю отримаєте потрібну вам інформацію. Як і у випадку з іменами користувачів, особа може мати більше однієї адреси електронної пошти.

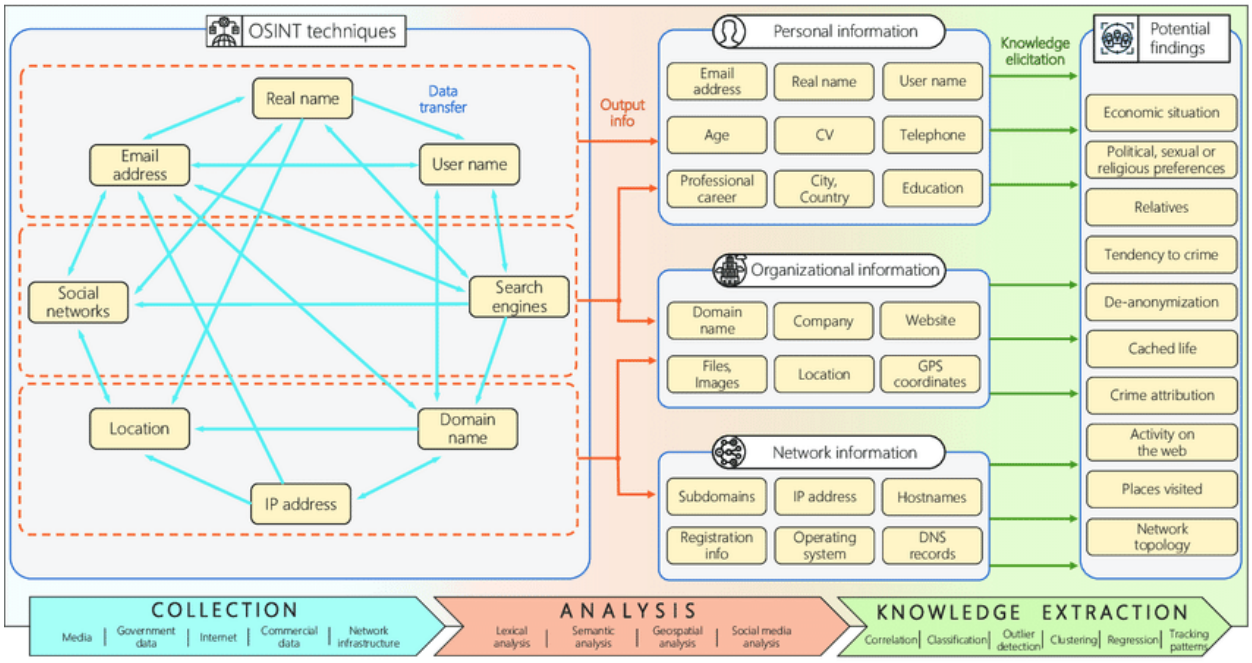


Рисунок 1.5 – Схема застосування технології OSINT.

Існує безліч легальних і публічних джерел даних, доступних для збору відкритих даних, і пошук потрібних даних з потрібного джерела займає багато часу і здебільшого здійснюється вручну. Поширені пошукові системи можна використовувати для початку дослідження або виявлення даних і нових джерел. Найпопулярнішою з них є Google; розширений пошук Google називається "Google Dork" і використовується для пошуку точних результатів в ім'я збору даних; база даних хакерської діяльності Google містить цілу базу даних варіантів і використовується для пошуку OSINT у сфері безпеки та хакерської діяльності.

Наприклад, нижче наведено пошук захищених паролем конфігураційних файлів. Файли конфігурації не повинні бути загальнодоступними, і файл .ENV є хорошим прикладом. Шукаючи файл .ENV, що містить рядок паролів до бази даних, ви можете миттєво знайти пароль до виявленої бази даних.

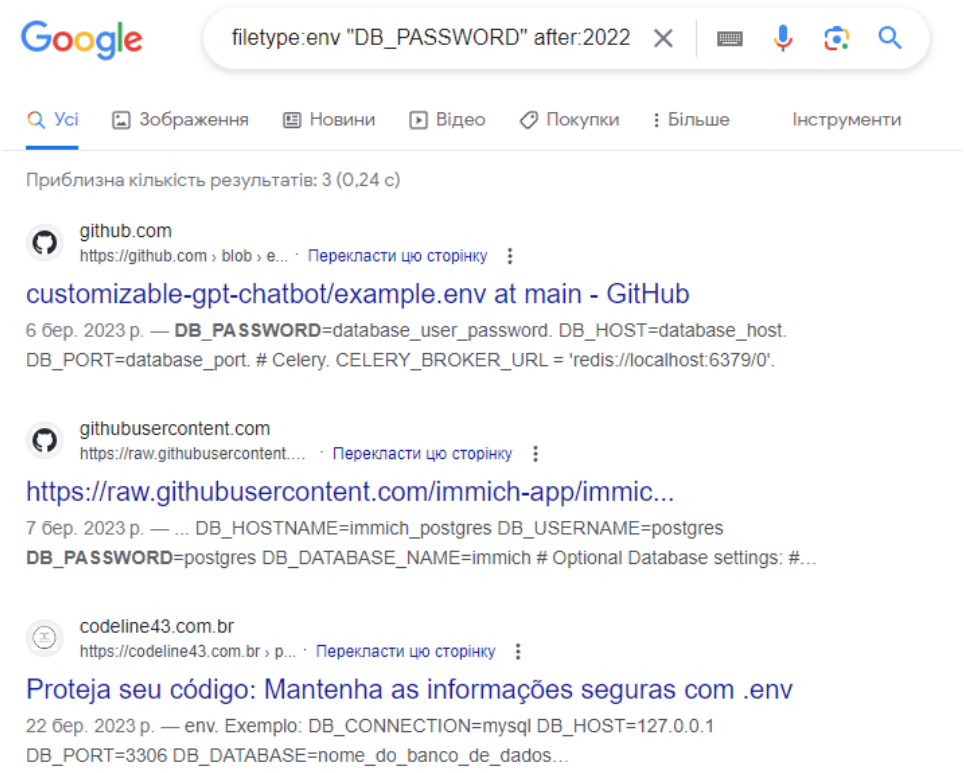


Рисунок 1.6 – Результати пошуку файлів .env

Списки адрес електронної пошти є корисним інструментом для знаходження адрес електронної пошти та отримання інформації про компанії. Ці списки зазвичай створюються компаніями або школами для ведення списків розсилок для своїх членів.

Існує кілька способів знайти такі адреси. Наприклад, можна переглянути електронні таблиці, такі як файли у форматі .XLS, шукаючи рядок "email.xls" в URL-адресі.

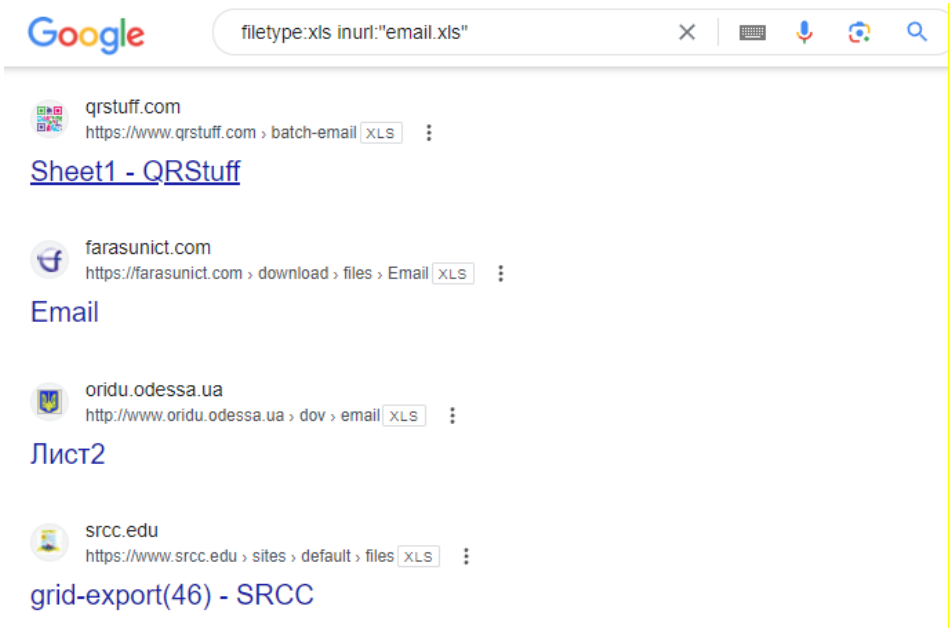


Рисунок 1.7 – Результати пошуку файлів, що містять адреси електронної пошти

Результати пошуку файлів, що містять адреси електронної пошти, можна знайти за допомогою пошукових систем, таких як Google або DuckDuckGo. Крім того, загальнодоступна інформація, така як витоки даних, може бути джерелом інформації про адреси електронної пошти. Сервіси, як от haveibeenpwned.com, надають інформацію про витоки даних, пов'язані з електронною поштою, що може бути корисним для аналітиків і слідчих з кібербезпеки.

Важливо враховувати, що цифровий простір містить багато інформації про людей, і загальнодоступні дані можуть бути використані для пасивного розслідування. Автоматизовані інструменти можуть допомогти встановити зв'язки між даними та знайти цінну інформацію. При розслідуванні осіб важливо аналізувати офіційну інформацію про бренд, людський фактор та технологічний слід, який охоплює різні аспекти, такі як веб-сайти, прес-релізи, активи бренду, діяльність співробітників та технологічні вразливості.

Інтернет – це скарбниця інформації про людей. Однак загальнодоступні інтернет-дані – це найпростіший спосіб пасивного розслідування та перевірки

третіх осіб, новобранців і шахраїв. Залежно від мети розслідування, не вся інформація може бути особливо цінною. Як правило, підсумковий звіт повинен містити особисту інформацію, таку як адреса електронної пошти суб'єкта, справжнє ім'я, ім'я користувача, вік, походження, номер телефону, місцезнаходження, освіту та досвід роботи.

Як було вказано у цьому розділі, ручний запис цифрових слідів є часоємним і не завжди ефективним. Автоматизовані інструменти можуть сприяти встановленню зв'язків між даними, які часто пропускаються. Згідно з даними від mediasonar.com, існують певні аспекти, які викликають інтерес при дослідженні осіб інтересу в організації:

- Офіційна інформація про бренд: Це охоплює офіційні активи бренду, такі як веб-сайти та офіційні матеріали, наприклад, звіти для інвесторів. Сюди також входять прес-релізи та новини про організацію, опубліковані у ЗМІ.
- Людський фактор: Людський фактор є складним для визначення, оскільки він непередбачуваний. Він охоплює діяльність агентів організації, таких як співробітники та керівники. Людський фактор вразливий до помилок, оскільки люди часто діють за своїм розсудом і не завжди керуються найкращими інтересами організації.
- Технологічний слід: Це місце, де розвідка відкритих джерел перетинається з кібербезпекою. Можна дослідити відкриті порти, DNS, IP-адреси, мережеві служби, можливості віддаленого доступу та вразливості.

У цифровому світі, де кожна дія може залишити слід, моніторинг активності в соціальних мережах також може бути корисним для розкриття діяльності та перспектив окремих осіб і компаній [7].

Важливо враховувати етичні аспекти при зборі і використанні інформації, забезпечуючи, щоб розслідування відбувалося в межах закону та відповідало приватності індивідів.

OSINT використовується для отримання інформації з розвідувальних джерел про домен, що допомагає визначити стратегії кібербезпеки, розуміти інциденти і виявляти джерела загроз. Цей підхід також використовується для оцінки доцільності співпраці та інтеграції зі сторонніми організаціями, а також для аналізу, що здійснюється фінансовими установами в рамках програми "Знай свого клієнта" (KYC – Know Your Customer).

На зображенні нижче показано інформацію про тип порушення даних на https://odeku.edu.ua/.

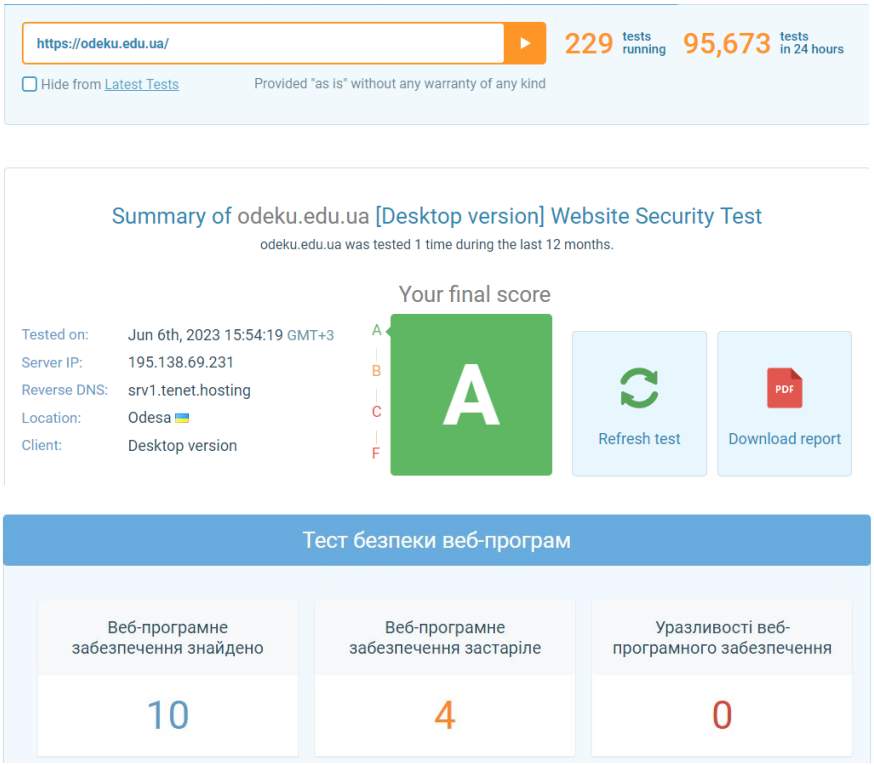


Рисунок 1.8 – Інформація про витік даних на домені https://odeku.edu.ua/

OSINT Framework є ресурсом для дослідників OSINT, який надає інструменти та концепції для їх досліджень. Він був створений Джастіном Нордіном

з метою допомогти початківцям та стати основою для багатьох людей і організацій. Ця структура надає корисний огляд при дослідженні домену.

На початку дослідження важливо встановити дату та реєстраційні дані домену. Для отримання основної інформації про реєстрацію домену можна скористатися сервісами, такими як Who.is або ICANN Lookup. Проте треба мати на увазі, що ця інформація може бути захищена або недостовірна.

Інструменти пошуку, такі як Hunter.io, дозволяють знайти домени та відповідні електронні адреси, що допомагає отримати повну картину про осіб, пов'язаних з доменом або веб-сайтом. Це відкриває нові напрямки досліджень та забезпечує більше інформації про профіль компанії.

Під час подальших досліджень важливо знати, які типи технологій використовуються на домені. Один з відомих та простих у використанні інструментів для цього – BuiltWith. Цей сервіс надає корисні вказівки про домен і дає доступ до безкоштовного пошукового інтерфейсу для отримання інформації про постачальників технологій. Певні домени можуть використовувати різноманітні популярні інструменти, такі як Google Analytics, платформи автоматизації маркетингу, системи управління контентом тощо. Проте, деякі з цих сервісів можуть створювати вразливості для організації або, що ще гірше, виявити, що домен використовується незаконно або недозволено. При розгляді потенційних партнерів та постачальників, особливо якщо домен пов'язаний з онлайн-сервісом або платформою, цей крок є важливим, оскільки він може надати додаткову інформацію.

Для отримання цифрового сліду домену потрібно вийти за межі кореневого домену. Існують різні способи скласти карту всього домену, але одним з найкращих є активне сканування. Також існують сканери піддоменів та інструменти, що полегшують цей процес. Компанії часто мають портали для співробітників, внутрішні піддомени, тестові сервери та інші складові, до яких вони не хочуть надавати загальний доступ. Сервіси, які працюють на цих субдоменах, можуть бути вразливими, що свідчить про проблеми з провайдером.

Підозрілі домени можуть підтвердити ці підозри або допомогти зрозуміти масштаб проблеми.

В аналізі безпеки часто використовується візуальний аналіз для відображення структури та зв'язків між точками даних. Методи візуалізації були використовувані задовго до появи комп'ютерів і застосовуються в різних сценаріях. Вони є більш корисними, ніж карти сайту або списки сторінок, коли ми намагаємося зрозуміти архітектуру веб-сайту.

1.3 Аналіз відомих технічних рішень і програмних продуктів

Збір даних є першою фазою дослідницького процесу і включає різні етапи, такі як пошук, зберігання, обробка, аналіз та розповсюдження даних. Автоматизація цих процесів означає, що вони виконуються автоматично з мінімальним втручанням людини.

Проте, такий підхід має свої проблеми, особливо щодо збору даних для досліджень, оскільки це може призвести до інформаційного перенавантаження. Неорганізований підхід, коли збираються всі доступні дані, не завжди є ефективним. Аналітикам не хочеться витрачати час на просіювання та фільтрацію великого обсягу даних, щоб уникнути кіберзагроз або швидко знайти сутність проблеми. Тому автоматизація збору даних фактично вимагає активної участі людини, принаймні на початкових етапах процесу.

Існують технічні рішення, які можуть виконувати функції пошуку в відкритих джерелах, такі як:

- theHarvester: це інструмент, розроблений на мові Python, який дозволяє збирати інформацію, таку як електронні адреси, субдомени, хости, імена співробітників та відкриті порти, з різних загальнодоступних джерел, включаючи пошукові системи, сервери ключів PGP і базу даних SHODAN.

– Maltego: це інструмент OSINT і комп'ютерної криміналістики, який надає інтерактивний аналіз даних з розширеною візуалізацією, що дозволяє ефективно аналізувати зв'язки між даними.

Це програмне забезпечення дозволяє досліджувати онлайн-зв'язки між даними з різних джерел в Інтернеті і виявляти зв'язки між людьми та компаніями, а також знаходити загальнодоступну інформацію.

На наведеному нижче зображенні показано робоче вікно програми і приклад відображення інформації у вигляді графіка:

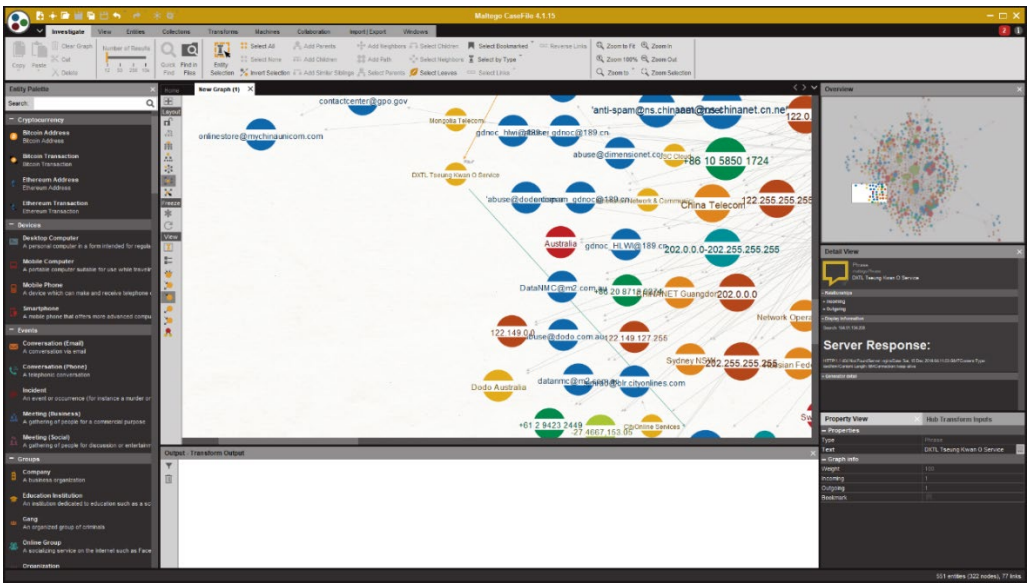


Рисунок 1.9 – Приклад Maltego

SpiderFoot – це відкритий інструмент розвідки з дактилоскопічним підходом, який володіє найширшою колекцією відкритих джерел інформації (OSINT) [8].

Цей інструмент може автоматично запитувати понад 100 відкритих джерел і збирати різноманітну інформацію, таку як IP-адреси, доменні імена, вебсервери та адреси електронної пошти. Він реалізований на мові програмування Python.

Нижче наведена візуалізація результатів, отриманих для домену https://odeku.edu.ua/:

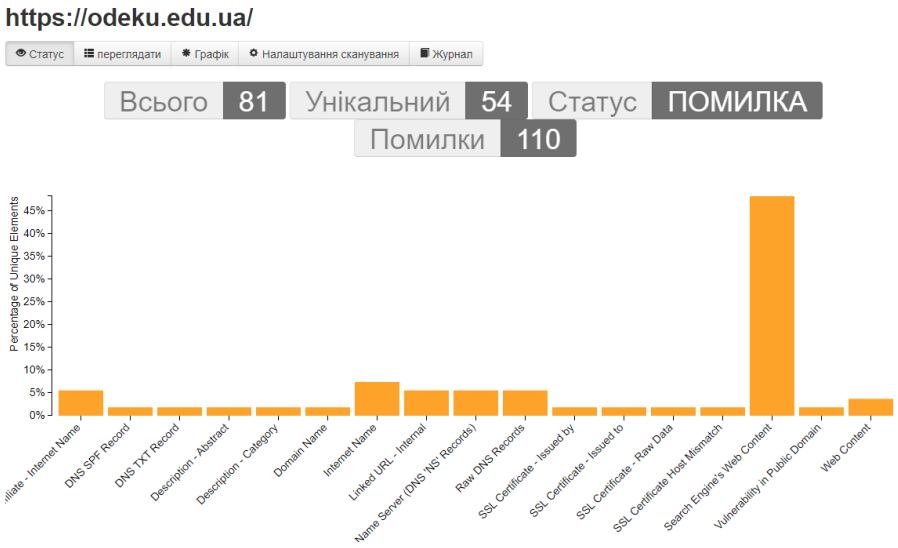


Рисунок 1.10 – Візуалізація даних в SpiderFoot

1.4 Розробка та аналіз функціональних і нефункціональних вимог

Функціональні вимоги визначають очікувану поведінку продукту, його можливості та функції. Вони описують, як система має взаємодіяти з користувачами або зовнішніми факторами, щоб досягти певних цілей.

Нефункціональні вимоги встановлюють специфікації, що впливають на функціональність системи, включаючи обмеження та можливості, що покращують його роботу. Ці вимоги можуть включати такі аспекти, як швидкість, безпека, надійність та інші.

На рисунку 1.11 наведено основні відмінності між функціональними та нефункціональними вимогами:

Рисунок 1.11 – Відмінності між функціональними та нефункціональними вимогами

У цьому випадку проект автоматизації OSINT складається з трьох основних завдань, які містять ряд функцій, що мають бути виконані. Для того, щоб ці функції працювали правильно, необхідно розробити вимоги для досягнення бажаних результатів. На рисунку 1.12 показано, як буде використовуватися продукт. Опис вимог для кожного варіанту використання, які можуть бути включені в діаграму:

- Отримання облікових даних:
  - Розпізнавання формату вхідних даних;
  - Пошук факту причетності до витоку даних з використанням відкритих джерел;
  - Відображення доступної інформації у форматі "email@example.com:password".
- Пошук інформації за доменом:
  - Розпізнавання формату вхідних даних;
  - Сканування об'єктів дослідження на основі відкритих джерел.

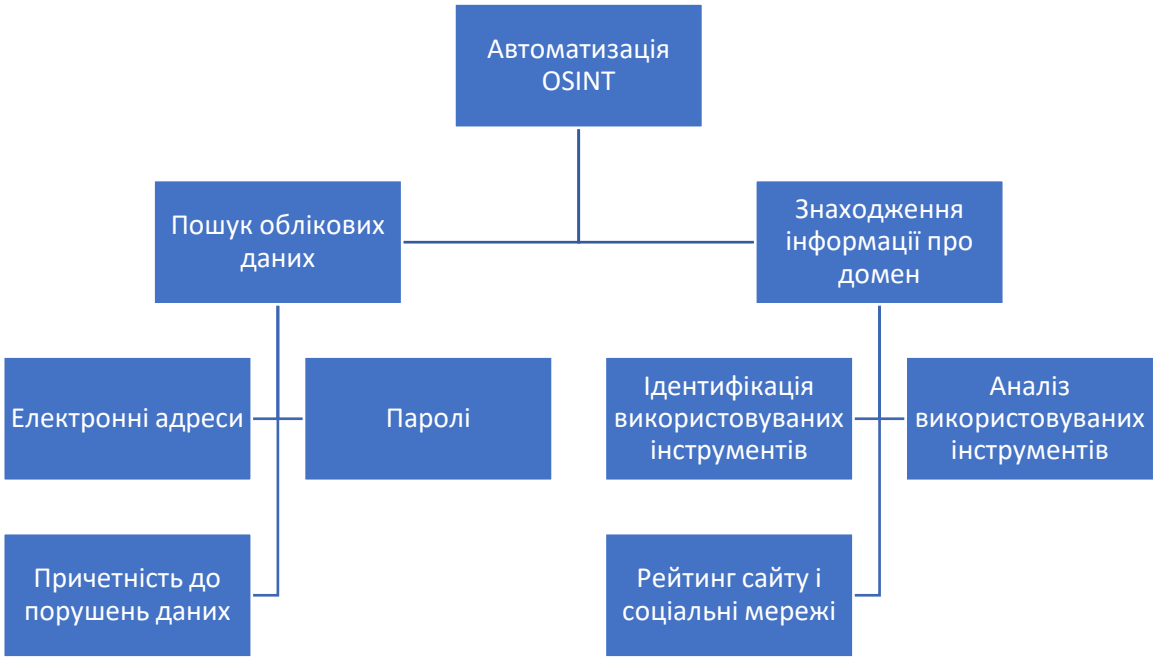


Рисунок 1.12 – Візуалізація варіантів використання

Корисність можна розглядати як найважливішу нефункціональну вимогу. Основна увага при розробці ботів приділяється створенню гарного, зручного інтерфейсу та вивченню того, як найкраще реалізувати дизайн інтерфейсу та взаємодію в конкретному середовищі. Крім того, необхідно вирішити деякі з наступних завдань [9]:

- реалізувати способи подання інформації у зручному для читання форматі;
- створити чіткий модуль для роботи з помилками у вхідних даних та сповіщення про них користувача;
- забезпечити безперервне використання сервісу.

2 МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ТА АВТОМАТИЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ OSINT

2.1 Моделювання математичних рішень

Пошук інформації є процесом знаходження ресурсів інформаційної системи, які задовольняють інформаційну потребу, з великого набору таких ресурсів. Цей пошук може базуватись на повнотекстових або інших індексах, що враховують зміст. Науковець, що займається пошуком інформації, досліджує пошук інформації в документах, самі документи та метадані, які описують текст, зображення, аудіо та бази даних.

Автоматизовані системи пошуку інформації використовуються для подолання проблеми інформаційного перевантаження. Ці системи надають доступ до документів, таких як книги і журнали, і забезпечують їх збереження та керування. Пошукові системи в Інтернеті є найбільш відомими прикладами таких систем.

Процес пошуку інформації включає кілька етапів. Спочатку визначаються і уточнюються інформаційні потреби та формулюються інформаційні вимоги. Далі визначаються потенційні джерела інформації, які можуть задовольнити ці потреби. Потім здійснюється вилучення інформації з вибраних джерел, а отриману інформацію оцінюють і проводять її аналіз.

Процес пошуку інформації починається зі введення користувачем запиту до системи. Запит є формальним описом інформаційної потреби, наприклад, пошуковий рядок у веб-пошуковій системі. Під час пошуку інформації один запит може відповідати кільком об'єктам з колекції, і результати пошуку зазвичай ранжуються за ступенем релевантності.

Класичне завдання систем інформаційного пошуку полягає в пошуку документів, що задовольняють певний запит, у статичній колекції документів. Однак сфера завдань систем інформаційного пошуку постійно розширюється, і включає в себе інші важливі аспекти:

- проблеми моделювання;
- класифікація документів;
- фільтрація документів;
- кластеризація документів;
- архітектура пошукової системи та дизайн інтерфейсу користувача;
- видобування інформації (включаючи анотацію та абстрагування документів);
- мови запитів.

Крім того, існують завдання, які можуть бути покладені на інструменти ІП, такі як морфологічний аналіз і лексичне розмежування.

Більшість систем ІП видають числове значення того, наскільки кожен об'єкт у базі даних відповідає запиту, і ранжують об'єкти відповідно до цього значення. Потім користувачеві показується об'єкт з найвищим рейтингом. Цей процес можна повторити пізніше, якщо користувач хоче уточнити запит.

Для ефективного пошуку релевантних документів за допомогою стратегій ІП документи зазвичай перетворюються у відповідне подання. Кожна пошукова стратегія містить певну модель представлення документів для своїх цілей. Наведена нижче схема ілюструє взаємозв'язок між деякими поширеними моделями. На рисунку 2.1 моделі класифіковано за двома параметрами: математичною основою та властивостями моделі [10].

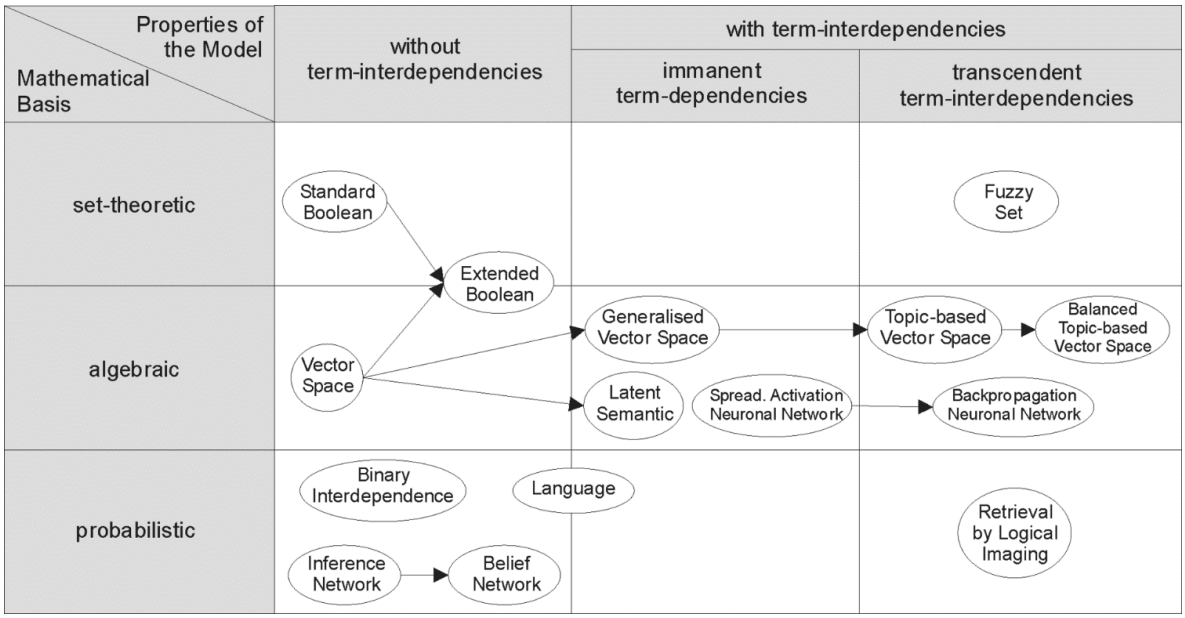


Рисунок 2.1 – Класифікація моделей ІП

Математичні основи пошукових моделей включають в себе:

1. Теоретичні моделі множин представляють документи як набори слів або фраз і використовують теоретико-множинні операції для виведення подібності між ними. До загальних моделей відносяться:

– стандартна булева модель, яка оперує булевими операціями (AND, OR, NOT) над множинами слів або фраз;

– розширена булева модель, яка дозволяє використовувати більш складні умови для пошуку, включаючи вагові коефіцієнти;

– нечіткий пошук, який використовує нечіткі множини та розмиті операції для оцінки подібності.

2. Алгебраїчні моделі представляють документи і запити у вигляді векторів, матриць або кортежів. Подібність між вектором запиту і вектором документа виражається скалярним значенням. До алгебраїчних моделей належать:

– векторна просторова модель, де кожен документ та запит представляються як вектор у просторі термінів, і подібність обчислюється за допомогою косинусної схожості;

- узагальнені моделі векторного простору, які враховують додаткові аспекти, такі як вагові коефіцієнти для термінів або динамічні зміни векторів;
- тематичні моделі векторного простору, що використовують статистичні методи для виявлення тематичної структури в документах та запитах;
- розширена булева модель, яка поєднує алгебраїчні та булеві операції для пошуку.

3. Імовірнісні моделі розглядають процес пошуку документів як імовірнісний висновок. Подібність обчислюється як ймовірність того, що документ є релевантним для запиту. До імовірнісних моделей належать:

- модель біноміальної незалежності, яка базується на припущенні незалежності входження термінів у документи;
- невизначений висновок, де релевантність документа визначається на основі ймовірнісного висновку, враховуючи статистичні параметри та додаткову інформацію;
- мовні моделі, що використовують статистичні методи для моделювання ймовірності появи термінів у документах та запитах;
- розбіжність з випадковими моделями, які базуються на випадкових процесах для моделювання подібності та релевантності;
- прихований відбір Діріхле, який використовує приховані станові моделі для моделювання подібності між термінами та документами.

4. Моделі пошуку на основі функцій розглядають документи як вектори значень функцій і намагаються знайти оптимальний спосіб поєднати ці функції в оцінку релевантності, часто за допомогою методів навчання ранжування. Функції ознак можуть бути будь-якими функціями документів і запитів і можуть включати в себе різні моделі пошуку як спрощені функції.

Моделі з взаємозалежністю термінів враховують взаємозалежність між термінами або словами. Рівень взаємозалежності зазвичай визначається самою моделлю, і може бути виражений через спільне використання термінів у всьому наборі документів.

Моделі з трансцендентною взаємозалежністю термінів використовують зовнішні джерела або складні алгоритми для визначення ступеня взаємозалежності між термінами.

Оцінка ступеня релевантності документа, знайденого упорядником результатів пошуку, є суб'єктивним поняттям, і залежить від оцінювача, який оцінює результати запиту [11].

Точність

Визначається як кількість релевантних документів, знайдених ІП, у відсотках від загальної кількості знайдених документів:

$$Precision = \frac{|D_{rel} \cap D_{retr}|}{|D_{retr}|} \quad (2.1)$$

де D_{rel} множина релевантних документів у базі даних, а D_{retr} – це множина документів, знайдених системою.

Повнота

Відношення чистої кількості знайдених релевантних документів до загальної кількості релевантних документів у базі даних:

$$Recall = \frac{|D_{rel} \cap D_{retr}|}{|D_{rel}|} \quad (2.2)$$

де D_{rel} – множина релевантних документів у базі даних, а D_{retr} – множина документів, знайдених системою.

Випадіння

Випадіння – це ймовірність знайти нерелевантний ресурс і визначається як відношення кількості знайдених нерелевантних документів до загальної кількості нерелевантних документів у базі даних:

$$Fall - out = \frac{|D_{nrel} \cap D_{retr}|}{|D_{nrel}|} \quad (2.3)$$

де D_{nrel} – множина нерелевантних документів у базі даних, а D_{retr} – цемножина документів, знайдених системою.

Міра Ван Різбергена

Іноді буває корисно об'єднати точність і повноту в одне середнє значення. Для цього середні арифметичні значення не підходять. Наприклад, пошукова система може повернути всі документи з точністю, близькою до нуля, але повнотою, що дорівнює одиниці, так що середнє арифметичне значення точності та повноти дорівнює принаймні половині. Середнє гармонійне не має цього недоліку, оскільки воно наближається до мінімуму при великій різниці у значеннях.

Тому придатною мірою для спільного оцінювання точності та повноти є F-міра, яка визначається як зважене середнє гармонійне значення точності P та повноти R:

$$F = \frac{1}{a\frac{1}{P}+(1-a)\frac{1}{R}}, a \in [0,1] \qquad (2.4)$$

Зазвичай, F-міра записується у вигляді

$$F = \frac{(\beta^2+1)PR}{\beta^2 P+R}, \beta^2 = \frac{(1-a)}{a}, \beta^2 \in [0,\infty] \qquad (2.5)$$

Альтернативно, a = ½ чи β = 1 F-міра надає рівну вагу точності та повноті і називається збалансованою або F_1-мірою (нижня міра зазвичай позначається значенням β), формула спрощується:

$$F_1 = \frac{2PR}{P+R} \qquad (2.6)$$

Використання збалансованої F-міри не є обов'язковим, 0 < β < 1 точності надається пріоритет, а β > 1 більша перевага. [13]

Центральне завдання ІП – допомогти користувачам задовольнити їхні інформаційні потреби. Описати інформаційні потреби користувача технічно непросто, тому вони формулюються у вигляді запиту, який являє собою набір ключових слів, що описують те, що користувач шукає.

2.2 Прогноз загрози

У цьому підрозділі наводяться джерела даних, що використовуються для прогнозування загроз, а також описується архітектура прогнозування загроз. Вказується на інформацію, яка доступна у відкритих базах даних вразливостей, таких як ExploitDB, VulnDB та CVE, а також в онлайн-джерелах, включаючи соціальні мережі, форуми та блоги.

На рисунку 2.2 демонструється, як зловмисники можуть використовувати вразливості в інфраструктурі організації для здійснення атак. Прогнозування загроз також показує, як можна використовувати наявні анотовані джерела даних, такі як VulnDB і ExploitDB, для навчання моделей прогнозування загроз і як ці знання можуть бути використані для виявлення загроз, які знайдені в різних джерелах OSINT і DarkWeb.

Прогностичні моделі можуть бути розроблені з використанням статистичних методів для передбачення результатів і проміжних результатів різних проектів. Моделі продуктивності процесу використовують ймовірнісні концепції. Для подальшого дослідження можуть бути використані симуляції. Результати аналізуються у формі діапазонів. Залежно від передбачуваних результатів можуть бути надані рекомендації щодо коригувань. Моделі можуть бути побудовані для прогнозування кінцевих результатів на основі запропонованих коригувань. Таким чином, ця активна модель допомагає технічним аналітикам аналізувати дані і прогнозувати результати. Аналітики можуть змінювати дані, проводити аналіз, записувати ці сценарії і вибирати найкращий варіант. Модель допомагає аналітикам вирішити, які параметри слід налаштувати для досягнення кінцевої мети проекту [12].

Зазначається, що в поточній системі виправлення вразливостей не враховується потенційний вплив вразливості. Крім того, відбувається перебільшення уваги, що приділяється виправленню всіх вразливостей через обмеження часу та ресурсів. Проактивна оцінка ризиків не може бути проведена

перед випуском IT-додатків. Вплив вразливостей визначається з використанням CVSS-калькулятора після аналізу пройденої атаки.

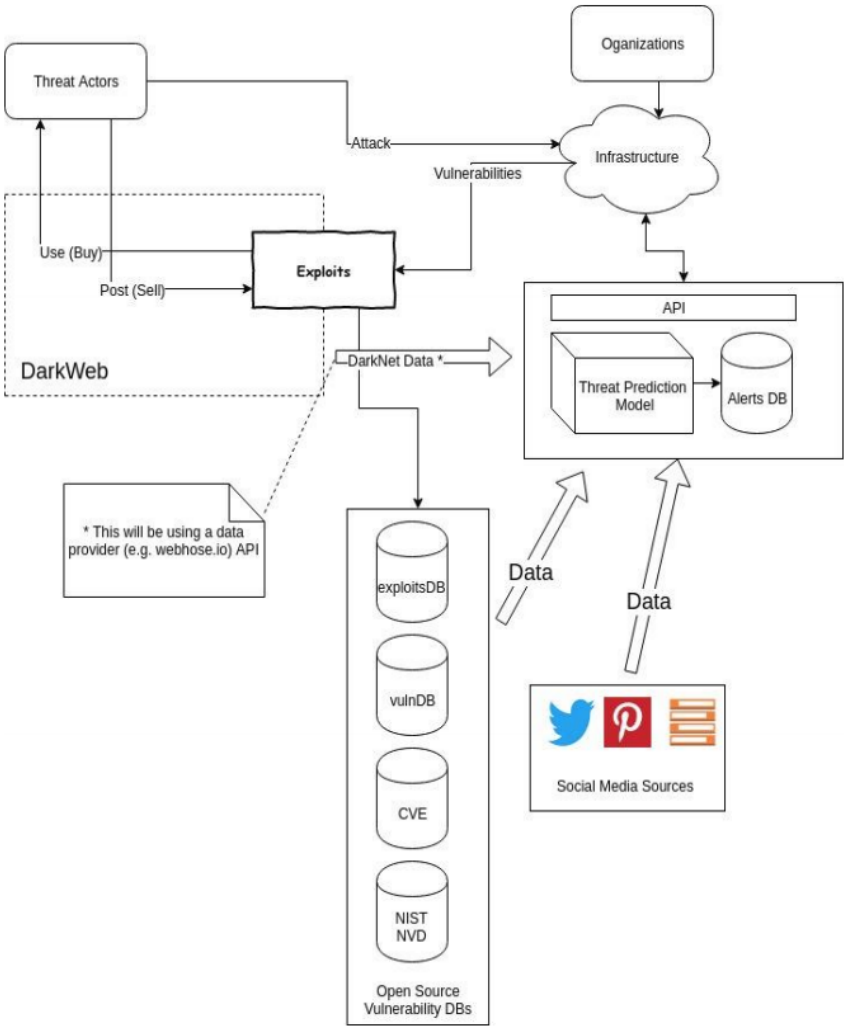


Рисунок 2.2 – Огляд суб’єктів та їхніх взаємозв'язків, а також роль прогнозувальника загроз

У запропонованій моделі, яка також застосовується у прикладних проектних організаціях, передбачений прогноз потенційного впливу вразливостей задовго до введення IT-додатка в експлуатацію. Отримуючи таку інформацію про вплив, можна виділити необхідні витрати та ресурси для виправлення вразливості, тим самим зменшуючи її наслідки.

Для прогнозування наслідків атаки використовуються методи множинної регресії. Множинна регресія базується на певних припущеннях, таких як лінійність, відсутність мультиколінеарності, однорідність і нормальність.

Нижче наведено робоче визначення розглянутих кібербезпекових показників:

- Y – залежний фактор, загальна оцінка CVSS; CVSS прогнозується на основі характеристик середовища та системи цільової програми;
- X1 – кількість вразливостей, тобто загальна кількість виявлених вразливостей за допомогою статичних та динамічних інструментів виявлення вразливостей для цільового додатка. Інструменти, встановлені та запущені проти цільової програми, можуть виявляти численні вразливості за допомогою таких алгоритмів, як тестування на проникнення;
- X2 – середній отриманий мережевий трафік (KBPS), зафіксований для програми протягом тижня атаки.

Уразливості, про які повідомляє інструмент, можна класифікувати за наступними категоріями:

- зловживання API;
- уразливості аутентифікації;
- уразливості авторизації;
- уразливості доступності;
- уразливості дозволу на виконання коду;
- якісні уразливості в коді;
- конфігураційні уразливості;
- криптографічні уразливості;
- уразливості при обробці помилок;
- поширені логічні помилки;
- вхідні валідаційні уразливості;
- уразливості в логуванні та аудиті;

– уразливості в управлінні паролями;
– стежкові уразливості;
– протокольні помилки;
– обсяг та тип помилок.

Для кожної атаки фіксуються дані CVSS. Результати роботи інструменту реєструються для кожної цільової програми. Також записуються дані про мережевий трафік для визначеного діапазону атаки. Під час кожної атаки ці дані з трьох джерел агрегуються, як показано на Рисунку 2.3. Визначення патернів здійснюється технічним аналітиком. В даній регресійній моделі оцінка CVSS (Y) прогнозується за допомогою двох змінних: кількістю вразливостей (X1) та мережевим трафіком (X2). Нульова гіпотеза, яка розглядається, полягає в тому, що X1 і X2 не впливають на Y. Це означає, що вразливість та мережевий трафік не мають впливу на оцінку CVSS [13].

Y	*X*1	*X*2
CVSS Score	Vulnerability	Network Traffic
2.1	20	324
5.3	53	623
1.0	15	235
8.0	85	932
2.9	28	438
3.0	25	498
3.8	38	391
1.0	18	132
1.2	16	177
5.9	63	823
4.3	39	579
2.8	30	455
1.1	14	231
4.2	35	725
5.4	51	740
1.9	21	345
2.0	25	432
4.1	37	467
6.2	58	845
1.1	15	111
2.3	22	191
1.2	16	182
2.8	30	292
6.9	68	952
4.8	55	600

Рисунок 2.3 – Точки даних проекту

На Рисунку 2.3 показані точки даних для кожної метрики для всіх випадків атаки.

Графік нормального розподілу, показаний на Рисунку 2.4, має приблизно лінійну форму. З нього видно, що припущення про нормальність помилок не порушується.

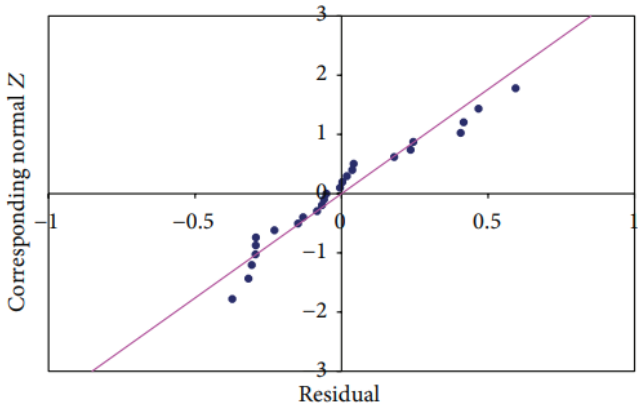


Рисунок 2.4 – Графік нормальної ймовірності

Як показано на Рисунку 2.5, вразливість має позитивний вплив на оцінку CVSS. Вища вразливість означає вищу оцінку CVSS і більший вплив на IT-активи. Вплив мережевого трафіку на CVSS також є позитивним. Це означає, що якщо вплив мережевого трафіку та вразливості є високим, оцінка CVSS також буде високою.

Intercept	Vulnerability	Network Traffic
−0.2983	0.07174	0.0025

Рисунок 2.5 – Рівняння регресії

Таким чином, як вразливість, так і мережевий трафік позитивно впливають на оцінку CVSS.

Прогнозована оцінка CVSS = -0,2893 + 0,07174 * Кількість вразливостей IT-додатків, про які повідомляє інструмент + 0,0025 * Розрахунковий середній вхідний мережевий трафік на тиждень для додатку, виміряний за допомогою KBPS.

Як показано на Рисунку 2.6, b – це коефіцієнт, який дає оцінку за методом найменших квадратів, b – стандартна похибка оцінки за методом найменших квадратів змінної x, а t – розрахована статистика t. Цей коефіцієнт, поділений на стандартну похибку, використовується для перевірки гіпотези. P-значення визначається для перевірки гіпотези; VIF кількісно оцінює ступінь мультиколінеарності у звичайному регресійному аналізі за методом найменших квадратів. У цих даних VIF становить 6,41.

	Intercept	Vulnerability	Network Traffic
b	−0.296	0.0706	0.002
s(*b*)	0.121	0.007	0.0005
t	−2.442	9.359	4.545
P	0.0231	0.0000	0.0002

Рисунок 2.6 – Результати множинної регресії

Як показано на Рисунку 2.7, SS – це сума квадратів, отриманих в результаті регресії. Це міра загальної кількості варіації Y, яку можна пояснити регресією зі змінною X. Df – це ступені свободи; MS – це міра суми квадратів. Середньоквадратична регресія (MSR) та середньоквадратична помилка (MSE) – це дві змінні, які визначають F: F = MSR / MSE. Ця статистика F використовується для перевірки того, чи пов'язані між собою змінні Y та X.

Source	SS	Df	MS	*F*	*P*
Regression.	96.22	2	48.15	597	0.000
Error	1.77	22	0.080		
Total	98	24			

Рисунок 2.7 – Дисперсійний аналіз

У цих даних MSR становить 48, а MSE – 0,08. P-значення в цьому випадку дорівнює 0, що підтверджує наявність лінійного зв'язку між CVSS та двома змінними – мережевим трафіком та вразливістю. Коефіцієнт детермінації R2 надає інформацію про відповідність моделі. У рівнянні регресії

коефіцієнт детермінації R2 визначає, наскільки точно лінія регресії апроксимує фактичні точки даних. Налаштований R2 – це модифікована версія, яка враховує кількість предикторів у моделі. У цих даних $R2$ становить 0,9819, а скоригований $R2$ – 0,9803.

З отриманих даних випливає, що на загальну оцінку CVSS позитивно впливають як вразливості, так і мережевий трафік. Команда, відповідальна за інфраструктуру організації, регулярно надає команді з якості необроблені дані про ці змінні. Для кожного мережевого процесу можна визначити логічні групи та порогові значення на основі типу мережі та хост-додатку. Технічний аналітик може посилатись на базові організаційні дані при проектуванні мережі.

Під час проектування мережі можна використовувати ці еталонні значення для встановлення верхніх та нижніх меж специфікацій. Ці значення доступні для кожного підпроцесу. Технічні аналітики можуть визначити та проаналізувати, які вразливості потрібно контролювати, і на основі цього встановити порогові значення.

На основі обраних порогових значень виконується аналіз "що, якщо". Розглядаються різні сценарії, в яких беруться значення вразливостей і мережевого трафіку, і вони слугують вхідними даними для моделі. Прогнозовані результати порівнюються з пороговими значеннями. Важливо відзначити, що при зміні параметрів технічні аналітики повинні усвідомлювати реалістичні наслідки. Це означає, що їхня математична модель має бути не лише теоретичною, але й практично застосовуваною [14].

Таким чином, цей підхід дозволяє технічним аналітикам визначати вразливості, які необхідно контролювати, і налаштовувати порогові значення на основі обраної стратегії. Аналізуються можливі сценарії та використовуються модельні дані для прогнозування результатів. Застосовуються порогові значення для оцінки, наскільки результати відповідають заданим специфікаціям. При цьому важливо брати до уваги практичну реалізованість моделі при зміні параметрів.

2.3 Архітектура API

API (Application Programming Interface) – це набір правил та протоколів, які визначають, як різні програми або компоненти програмного забезпечення можуть взаємодіяти один з одним. API визначає, які функції та процедури доступні для використання, які параметри потрібно передавати та які результати можна отримати під час взаємодії з програмою.

API виступає як проміжний шар між різними програмами або компонентами, що дозволяє їм обмінюватися даними та виконувати певні функції без прямого доступу до внутрішньої реалізації. Він визначає як взаємодіяти з системою або службою, приховуючи деталі реалізації та надаючи стандартизований спосіб комунікації.

API може бути реалізований у вигляді набору функцій, класів, методів, протоколів або навіть веб-служб, залежно від контексту використання. Він дозволяє розробникам використовувати функціональність або дані іншої програми або служби, не розглядаючи всю деталізацію реалізації, а лише викликаючи відповідні методи або запити до API.

API використовується для створення розширень, інтеграції програм та служб, обміну даними між різними системами, розробки додатків та багато інших сценаріїв взаємодії між програмами. Він є важливою складовою для створення розподіленого та модульного програмного забезпечення.

Нижче перераховані типи API [15]:

Приватні API (Private API) – (також відомі як внутрішні або внутрішньоорганізаційні API) використовуються в середині конкретної організації або компанії. Ці API не доступні публічно та не призначені для використання зовнішніми сторонами. Вони зазвичай використовуються для внутрішньої автоматизації процесів, обміну даними між внутрішніми системами або побудови внутрішніх інструментів.

Партнерські API (Partner API) – використовуються для співпраці між різними організаціями або партнерами. Ці API дозволяють певним сторонам або партнерам отримувати доступ до функціональності або даних системи або платформи. Зазвичай, партнерські API використовуються для розширення функціональності, інтеграції зі сторонніми сервісами або спільної роботи з партнерами.

Ці два типи API зазвичай мають обмежений доступ і вимагають аутентифікації або домовленостей між сторонами щодо використання та обміну даними. Вони забезпечують більш контрольовану та обмежену взаємодію з іншими сторонами, ніж загальнодоступні публічні API.

Відкриті API – ці API, також відомі як зовнішні API, доступні для будь-якого стороннього розробника. Програма відкритих API, якщо її правильно реалізувати, може підвищити впізнаваність бренду і принести додатковий дохід.

Рисунок 2.8 ілюструє потік даних через архітектуру API: шлюз API (пристрій або сервіс) діє як центральний вузол, де різні клієнти можуть отримувати інформацію від різних сервісів. Типи клієнтів можуть варіюватися від мобільних пристроїв, серверів, дослідників, веб-додатків тощо. Сервіси надаються розробниками OSU та сторонніми додатками. Така архітектура дозволяє відокремити клієнта від сервісу і зосередити зусилля щодо забезпечення відмовостійкості на шлюзі API та сервісі.

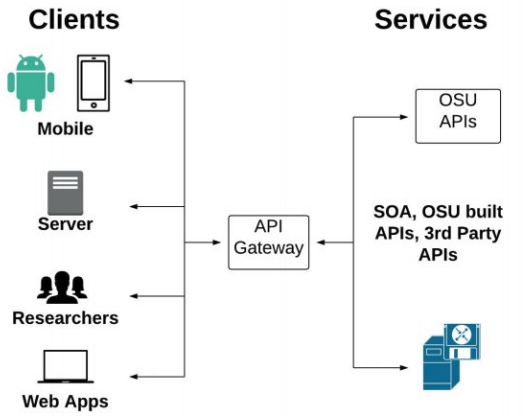


Рисунок 2.8 – Потік даних через API

Всі API містять і реалізуються за допомогою викликів функцій (операторів мови, які вимагають від програмного забезпечення виконати певну дію або послугу). Виклик функції – це фраза, що складається з дієслова та іменника, наприклад

- почати або завершити сеанс;
- відновити або отримати об'єкт з сервера.

Роль API ще більша, якщо розглядати їх з точки зору розробки програмного забезпечення, а також з точки зору узгодження бізнесу: у звіті "Стан інтеграції API за 2018 рік" понад 60% респондентів погодилися, що інтеграція API важлива для їхньої бізнес-стратегії. Опитування також показує, що понад 50% усіх підприємств співпрацюють через API.

У зв'язку з цим двома основними викликами для осіб, які приймають рішення, і розробників є вибір API, які відповідають їхнім конкретним бізнес-потребам, і розуміння того, як їх ефективно використовувати.

У проекті буде використана інтеграція наступних сервісів

- IntelligenceX;
- BuildWith.

3 АВТОМАТИЗАЦІЯ РОЗРОБКИ OSINT-СИСТЕМ

3.1 Реалізація та моделювання проектних рішень

Система автоматизації реалізована за допомогою бота у месенджері Telegram. Боти є сторонніми додатками, які працюють у месенджері Telegram. Користувачі можуть взаємодіяти з ботом, надсилаючи йому повідомлення, команди та вбудовані запити. Керування ботами здійснюється за допомогою HTTPS-запитів до API бота.

Багато галузей переносять обслуговування клієнтів на системи чат-ботів через їх низьку вартість, надійність та постійну доступність. Чат-боти забезпечують певний рівень клієнтської підтримки без значних додаткових витрат.

Незалежно від складності чат-бота, структура програмного забезпечення зазвичай є однаковою. Чат-боти можуть ставати складнішими з додаванням додаткових компонентів для забезпечення більш природної взаємодії. Рисунок 3.1 ілюструє архітектуру чат-бота.

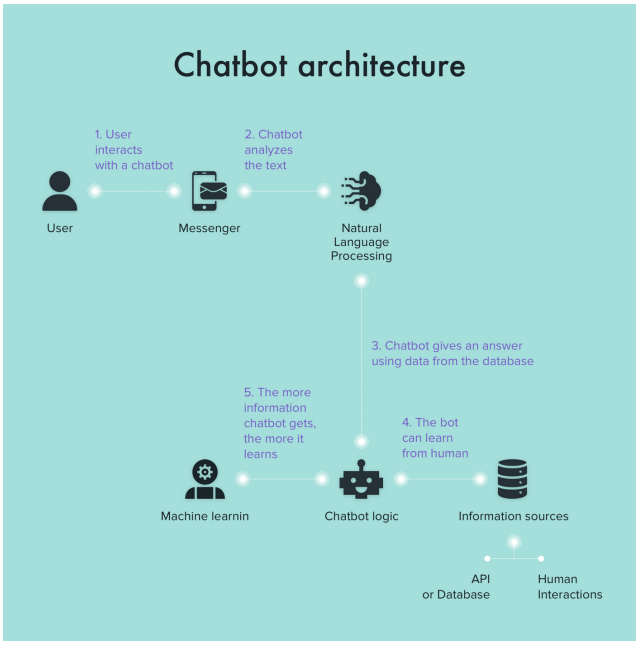


Рисунок 3.1 – Архітектура чат-бота

Telegram є однією з найпопулярніших платформ для обміну миттєвими повідомленнями. Він забезпечує збереження повідомлень як у хмарі, так і на пристрої, і підтримує різні операційні системи, такі як Android, iOS, Windows та інші веб-версії. Це дозволяє використовувати Telegram на різних платформах і виділяє його своєю хорошою крос-платформенною підтримкою.

Telegram використовує власний протокол шифрування під назвою MTProto API MTProto (також званий Telegram API) – це API для додатків Telegram для зв'язку з сервером Telegram API є повністю відкритим тому будь-який розробник може написати власний клієнт месенджера.

Нові боти створюються за допомогою одного і того ж бота.

Ви можете знайти його, набравши в пошуку ім'я BotFather, як показано на рисунку 3.2.

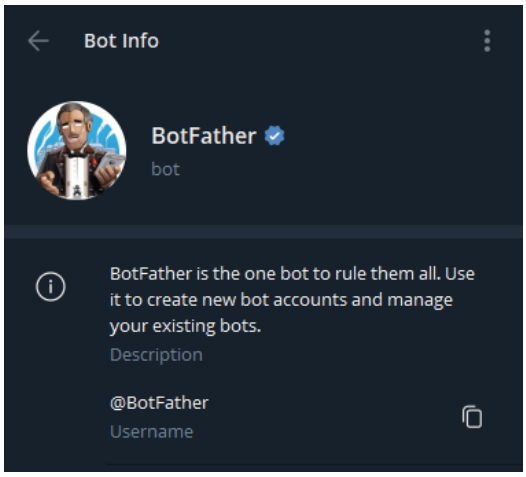


Рисунок 3.2 – Пошук BotFather.

Для створення нового бота використайте команду /newbot, як показано на рисунку 3.2. Після введення цієї команди, BotFather запитає вас про ім'я та ім'я користувача бота і створить маркер автентифікації для вашого нового бота.

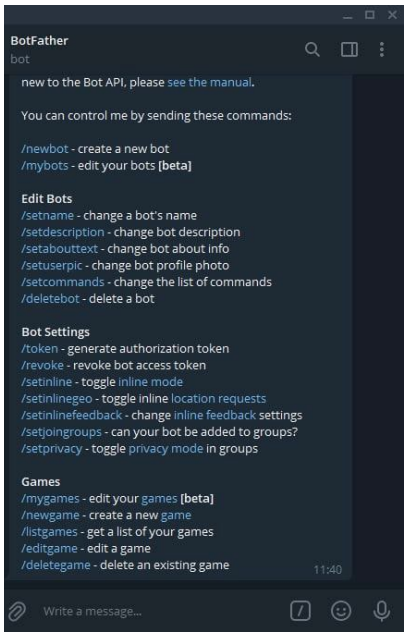


Рисунок 3.3 – Створення нового бота.

Ім'я бота з'явиться, наприклад, в Контактах. Імена користувачів – це короткі імена, які використовуються у згадках та посиланнях на t.me. Імена користувачів мають довжину від 5 до 32 символів і не залежать від регістру, але можуть містити лише латинські літери, цифри та знаки підкреслення. Ім'я користувача бота має закінчуватися на "bot", наприклад, "tetris_bot" або "TetrisBot". Рисунок 3.3 ілюструє процес створення імені бота, який у цьому випадку називатиметься OSINT Automation.

Після того, як бот створений, батько бота надає токен. Токен виглядає так: 110201543: AAHdqTcvCH1vGWJxfSeofSAs0K5PALDsaw. Цей токен використовується для керування ботом.

Дизайн бота задається в BotFather: меню / мої боти → Редагувати бота, там же його можна змінити:

– назва бота;
– description – текст під заголовком "Що вміє цей бот", який з'являється на початку взаємодії з ботом;
– about – текст, який з'являється в профілі бота;

– аватар – аватар бота не може бути анімованим, на відміну від аватара користувача або чату. Можливі лише зображення;
– команда – тут мається на увазі командний рядок бота. Детальніше про команди див. нижче;
– вбудований заповнювач.

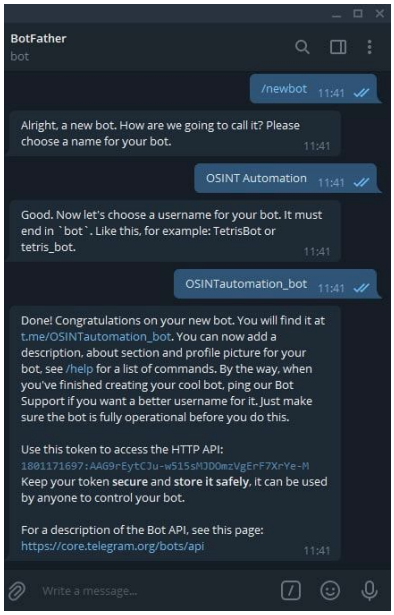


Рисунок 3.4 – Процес створення імені бота

Коли користувач відкриває бота вперше, відображається кнопка "Пуск" або "Старт" (залежно від платформи користувача). Натискання цієї кнопки призведе до відправки команди /start.

Існує два основних способи роботи з Telegram у Python: надсилаючи HTTPS-запити або використовуючи веб-хук. Проект складається з трьох частин: комп'ютер з Python, сервер Telegram і клієнт Telegram.

На комп'ютері працює інтерпретатор Python, а всередині інтерпретатора – програма Python; програма Python відповідає за весь контент і містить всі текстові шаблони, всю логіку і поведінку.

Усередині Python-програми є бібліотека, яка відповідає за зв'язок із сервером Telegram; у бібліотеку інтегрований секретний ключ, щоб сервер Telegram міг зрозуміти, що додаток пов'язаний із певним ботом.

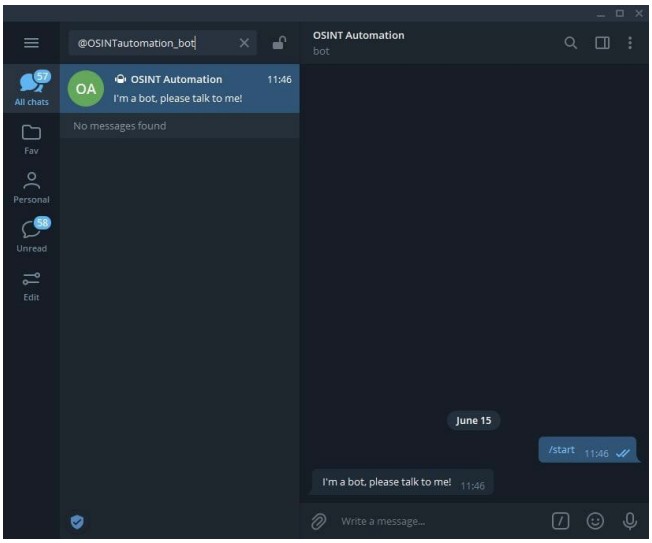


Рисунок 3.5 – Запуск першої команди /start

Коли користувач Telegram запитує інформацію у бота, його запит надсилається на сервер. Сервер обробляє цей запит за допомогою програми, написаної на мові Python, і відправляє відповідь назад на сервер Telegram, який передає відповідь клієнту.

Робота бота залежить від того, чи працює комп'ютер і запущена програма Python. Якщо комп'ютер вимкнений, відсутній доступ до Інтернету або програма Python закрита, то бот перестає працювати: запити продовжують надходити, але ніхто не може на них відповідати.

Логіка роботи бота реалізована за допомогою бібліотеки python-telegram-bot. Ця бібліотека дозволяє швидко і просто створювати ботів, надаючи набір методів для взаємодії з Telegram.

Субмодуль telegram.ext, який побудований на основі API Telegram, надає простий у використанні інтерфейс. Він складається з кількох класів, два з яких є ключовими: telegram.ext.Updater і telegram.ext.Dispatcher.

Клас Updater постійно отримує нові оновлення від Telegram і передає їх класу Dispatcher. Диспетчер відповідає за сортування інформації про оновлення та передачу її відповідним зареєстрованим функціям зворотного виклику відповідно до зареєстрованих обробників.

Кожен обробник є екземпляром підкласу класу telegram.ext.Handler. Бібліотека надає різні класи обробників для різних сценаріїв використання.

При створенні екземпляра класу Updater використовується токен доступу, який був отриманий при створенні бота. Нижче наведений приклад створення екземплярів класів Updater і Dispatcher, а також додавання обробників для команд і повідомлень від кінцевого користувача. Деталі можна побачити на рисунку 3.6.

```
from telegram.ext import CommandHandler, Filters, MessageHandler, Updater
from config import BOT_API_KEY
from telegram_bot_handlers import TelegramBotHandlers
updater = Updater(token=BOT_API_KEY, use_context=True)
dispatcher = updater.dispatcher
start_handler = CommandHandler('start', TelegramBotHandlers.wellcome_message)
help_handler = CommandHandler('help', TelegramBotHandlers.wellcome_message)
unknown_message_handler = MessageHandler(Filters.text & (~Filters.command),
TelegramBotHandlers.unknown_message)
scan_email_handler = CommandHandler('scan_email',
TelegramBotHandlers.scan_email)
scan_domain_handler = CommandHandler('scan_domain',
TelegramBotHadlers.scan_domain)
dispatcher.add_handler(start_handler)
dispatcher.add_handler(help_handler)
dispatcher.add_handler(unknown_message_handler)
dispatcher.add_handler(scan_email_handler)
dispatcher.add_handler(scan_domain_handler)
updater.start_polling()
```

Рисунок 3.6 – Ініціалізація боту

В рамках даної роботи був реалізований клас TelegramBotHandlers, який включає функціонал для обробки подій і здійснення наступних дій:

– метод wellcome_message: Вітає користувача та надає інформацію про доступні функції бота та їх опис;

- метод unknown_message: Виводить повідомлення, що інформує користувача про некоректні введені дані, та надає підказку для команди /help, де доступна інформація про всі функції бота;
- метод scan_email: Команда для сканування електронних адрес. Цей метод звертається до API сервісу IntelX для отримання відповідних даних про порушення даних електронних адрес. При успішному пошуку, він повертає інформацію про загальну кількість порушень даних та, якщо є, аутентифікаційну інформацію, таку як паролі. Реалізацію функції сканування електронної пошти можна побачити на Рисунку 3.7;
- метод scan_domain: Команда для сканування доменів. Цей метод також використовує API сервісу IntelX для отримання відповідних даних про порушення даних доменів.

Ці методи дозволяють взаємодіяти з ботом та отримувати інформацію про порушення даних для вказаних електронних адрес і доменів.

```
def search_email(self, email: str):
"""Method used to search email in intelx database and return results
Args:
mail (str): String formated email [example@domain.com]
"""
result_str = f"No information found about {email}!"
record_count, search = self.__search_email(email)
if record_count == 0:
return result_str
result_str = f"Information found about {email}:\n\n"
stats_str = self.__parse_email_stats(search=search)
result_str = result_str + stats_str
file_name = self.__download_first_file(search)
email_data = self.__parse_downloaded_file(file_id=file_name, email=email)
result_str = result_str + "\n\n" + email_data
os.remove(f"downloads/intelx/{file_name}")
return result_str
```

Рисунок 3.7 – Функція сканування електронної адреси

Функція сканування доменів, з іншого боку, використовує сканер BuildWith. Він надає додаткову інформацію про сервіси та інструменти, що використовуються доменом, наприклад, його версію та опис, рейтинг сайту тощо, а також посилання на соціальні мережі (якщо такі є).

3.2 Функціональний тест

Тестування програмного продукту можна розділити на два класи з точки зору класифікації за програмними цілями:

- функціональне тестування;
- нефункціональне тестування.

Функціональне тестування – це перевірка відповідності програмного продукту функціональним вимогам, зазначеним в умовах створення цього продукту.

Нефункціональне тестування оцінює якість програмного продукту, наприклад, ергономіку та продуктивність.

Під час навантажувального тестування було виявлено, що запити до серверів Telegram обмежені, про що зазначено в Bots FAQ на сайті Telegram:

- до одного повідомлення в секунду в чаті;
- загалом не більше 30 повідомлень на секунду;
- не більше 20 повідомлень на хвилину на групу.

Ці ліміти не є суворими, але приблизними. Для більших ботів ліміти можуть бути збільшені через службу підтримки.

На рисунку 3.8 показано вихідні дані програми, включаючи інформацію про дані, введені користувачем, і час введення. На рисунку 3.9 показано вивід повідомлення від користувача.

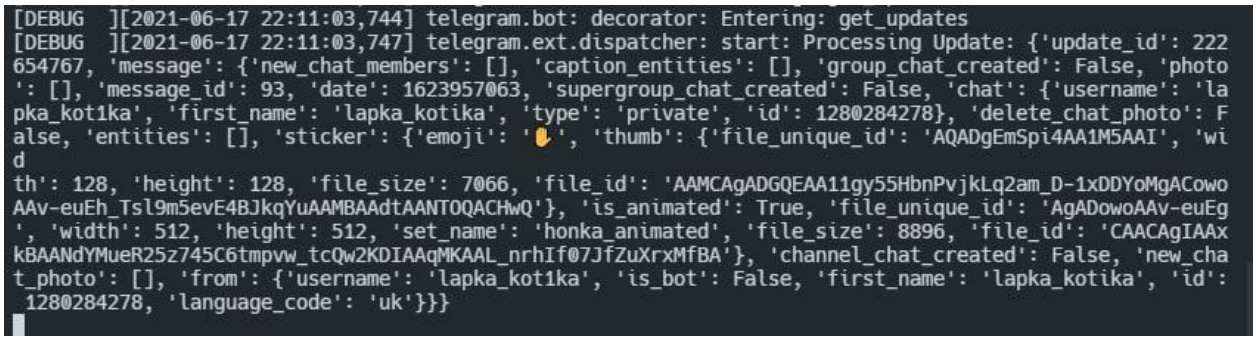
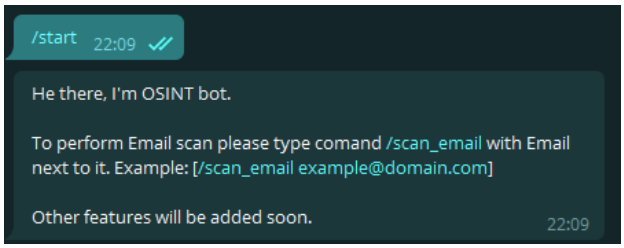


Рисунок 3.8 – Виведення повідомлення користувача lapka_kot1ka



3.9 – Введення користувачем привітального повідомлення

Для демонстрації успішного виявлення витоку адреси електронної пошти було обрано зразок landry.todd@gmail.com, який відповідає критеріям. Це особиста електронна адреса віце-президента компанії JMA Wireless, світового лідера в галузі мобільного бездротового зв'язку, включаючи зовнішні та внутрішні розподілені антенні системи. На рисунках 3.10 та 3.11 показано результати виведення інформації.

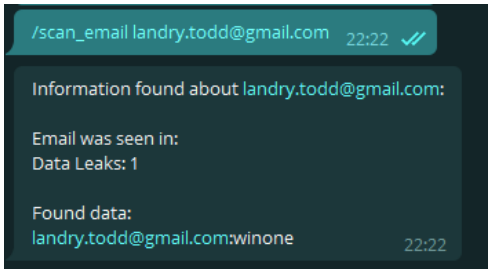


Рисунок 3.10 – Перевірка електронної адреси на витік даних, позитивні результати

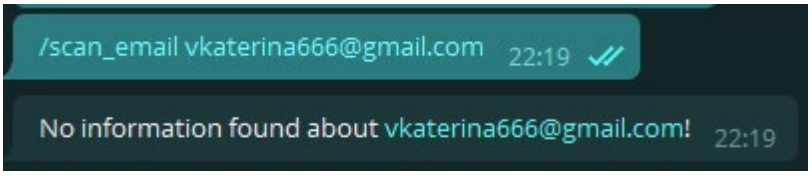


Рисунок 3.11 – Перевірка електронної адреси на витік даних, негативний результат

Бот шукає інформацію на основі домену за доменом і за допомогою цієї інтеграції може отримати детальні дані, такі як використовувані інструменти, рейтинг домену, сторінки в соціальних мережах та сторінки в соціальних мережах, серед інших детальних даних.

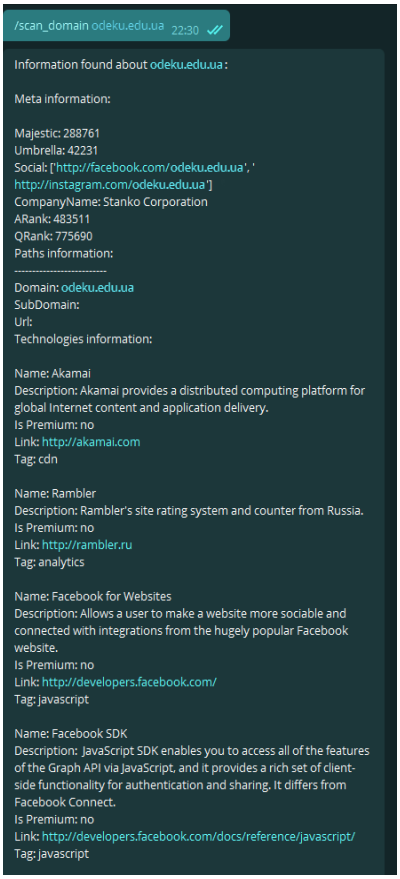


Рисунок 3.12 – Інформація про домен odeku.edu.ua

3.3 Конфігурація системи автоматизації

По-перше, отримайте ключ API, необхідний для інтеграції сервісу: BuiltWith Domain API надає доступ до технічної інформації на веб-сайті у форматах XML та JSON. Ця технічна інформація включає всю технічну інформацію, знайдену за допомогою розширеного пошуку на веб-сайті builtwith.com, а також додаткові метадані, якщо такі є.

Щоб отримати ключ API, ви повинні увійти до свого особистого кабінету або зареєструватися на сайті builtwith.com.

В особистому кабінеті IntelligenceX ви можете отримати ключ API, а також інформацію про кількість запитів, доступних у безкоштовній версії, як показано на рисунках 3.13 та 3.14.

Рис. 3.13 – Інформація про API

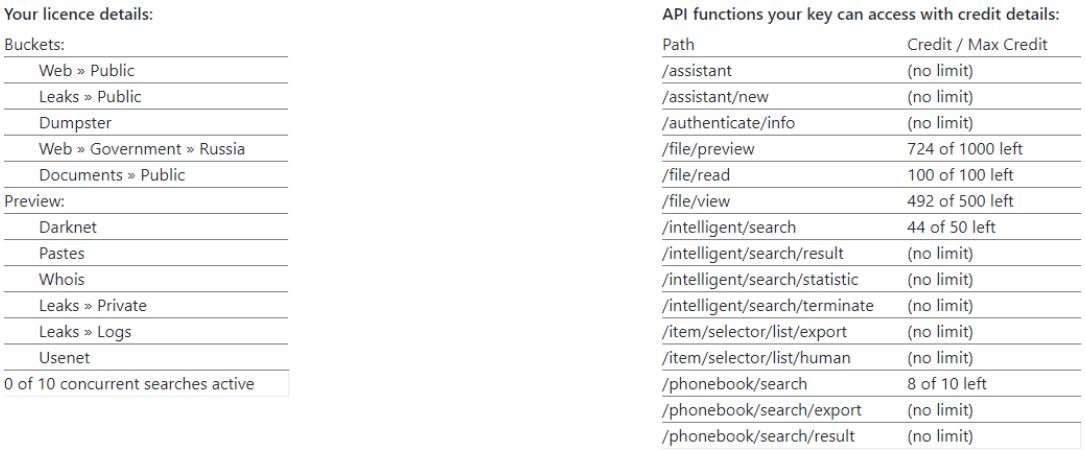


Рис. 3.14 – Кількість доступних запитів

Всі використовувані ключі API доступні в конфігураційному файлі і показані на рисунку 3.15.

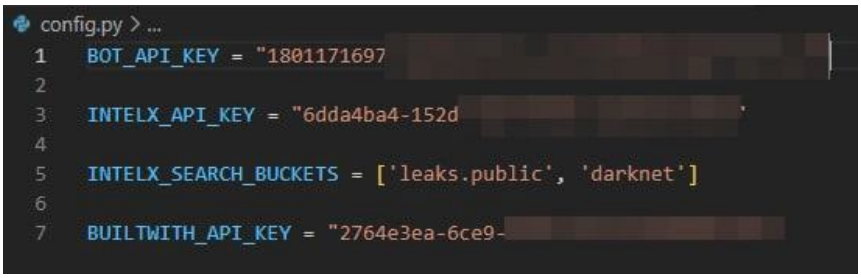


Рис. 3.15 – Ключі API

Процес конфігурації продовжується встановленням бібліотеки, для чого необхідно встановити бібліотеку за допомогою наступної команди: pip3 install python-telegram-bot.

Ця бібліотека необхідна, оскільки вона не входить до стандартного набору пакунків Python. Ця бібліотека надає чистий інтерфейс Python для API Telegram Bot і сумісна з версією Python 3.6.2 і вище; PTB також можна запускати на PyPy, але PyPy офіційно не підтримується.

На додаток до чистої реалізації API, бібліотека надає ряд високорівневих класів, які роблять розробку ботів простою і зрозумілою. Ці класи містяться в підмодулі telegram.ext.

ВИСНОВКИ

Через витоки даних автоматизовані системи пошуку корпоративної та особистої інформації пропонують можливість швидкого доступу до конфіденційної інформації. Встановивши таку систему, кожен користувач має можливість:

- визначити факти порушення електронної пошти;
- отримати облікові дані користувачів, які постраждали від порушення даних;
- отримати інформацію про інструменти, які використовуються для аналізу домену;
- отримати домен оцінка безпеки.

При використанні різних розвідувальних ресурсів для інтеграції були обрані IntelligenceX і BuildWith, оскільки ці сервіси пропонують можливість користуватися їх послугами безкоштовно, хоча і з обмеженою кількістю спроб. Більшість інструментів із цінною інформацією вимагають фінансових інвестицій, що є проблемою під час дослідження домену.

Ринок інтерактивних-цифрових помічників розширюється, і месенджер Telegram є гарною можливістю для розвитку подібних проектів. Кількість активних користувачів Telegram перевищила 1,5 мільярди, що свідчить про популярність програми серед користувачів і її здатність охопити широку аудиторію своїх продуктів.

За період роботи виконано основні завдання, а саме:

- отримав досвід використання OSINT-методів і методів пошуку корпоративної та особистої інформації;
- проаналізував стан ринку автоматизованих систем OSINT;
- описав явище витоку корпоративної та особистої інформації;
- розробив інструмент автоматизації пошуку інформації за допомогою інтеграції готових засобів;

– виконано випробування готової продукції та функціональну оцінку.

Для досягнення цих цілей у боті Telegram використовується метод інтеграції сервісу за допомогою ключів API, а вся програмна реалізація здійснюється на мові програмування Python.

Рішення використовувати Telegram-ботів для автоматичного збору інформації дало ряд переваг:

– зручно та швидко отримувати інформацію;
– простий інтерфейс, все в одній програмі Telegram;
– відсутні вимоги завершення captcha;
– не потрібно використовувати Tor і DarkWeb;
– відсутність реєстрації на спеціалізованих сайтах.

У найближчому майбутньому OSINT-технології стануть більш досконалими та зможуть збирати більше даних про фізичних осіб та підприємства. Можливості для бізнесу збільшуються, а ті, хто втрачає їх, втрачають цінну інформацію. Забігаючи вперед, необхідно запитати, які ресурси з відкритим кодом є можливість використовувати для збору даних. Відповідь буде така – майже будь-які. З новітніми технологіями, які з’являються в мережі щодня, ви ніколи не знаєте, де OSINT з’явиться після 2023 року. Очікується, що до 2026 року глобальний ринок аналітичної інформації з відкритим кодом досягне майже двадцяти мільярдів умовних одиниць, зростаючи на 17% у середньому. Збільшення доступності інформації та посилення загроз безпеці збільшать попит на OSINT. Тому на даний час дуже важливо працювати над удосконаленням технології та запровадженням новітніх методів пошуку достовірної інформації, щоб забезпечити ефективну оцінку замовникам, які хочуть знати справжній статус безпеки своєї особистої інформації.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Libor Benes. OSINT, New Technologies, Education: Expanding Opportunities and Threats. A New Paradigm Vol. 6, No. 3. 2013. 17 с.
2. Fahimeh Tabatabaei. Douglas Wells. OSINT in the Context of Cyber-Security. 2017. 231 с.
3. Seonghyeon Gong. Jaeik Cho. Changhoon Lee. A Reliability Comparison Method for OSINT Validity Analysis. 2018. 5435 с.
4. Gašper Hribar. Iztok Podbregar. Teodora Ivanuša. OSINT: A “Grey Zone”? 2014. 549 с.
5. Javier Pastor-Galindo. Pantaleone Nespoli. Félix Gómez MárB. Jansen. The Not Yet Exploited Goldmine of OSINT: Opportunities, Open Challenges and Future Trends. 2020. 10304 с.
6. Williams. Heather J. Blum. Ilana. Defining Second Generation Open Source Intelligence (OSINT) for the Defense Enterprise. 2018. 62 с.
7. Michael Glassman. Min Ju Kang J. Pastor-Calindo. P. Nespoli. Intelligence in the internet age: The emergence and evolution of Open Source Intelligence (OSINT). 2011. 235 с.
8. João Rafael Gonçalves Evangelista. Renato José Sassi. Márcio Romero. Domingos Napolitano. Systematic Literature Review to Investigate the Application of Open Source Intelligence (OSINT) with Artificial Intelligence. 2020. 369 с.
9. Dmytro V. Lande. Ellina V. Shnurko-Tabakova. OSINT as a part of cyber defense system. 2019. 108 с.
10. Dodonov, Lande, Putyatin, Computer networks and analytical research. IIR of NAS of Ukraine, 2014.
11. Astafieva, “Wavelet analysis: bases of the theory and examples of application,” Achievements of physical sciences, no. 11, pp. 1145–1170, 1996.

12. Davydov, “Wavelet analysis of social processes,” Sociological researches, no. 11, pp. 97–103, 2003.
13. Feder, Fractals. Plenum, 1988.
14. Lande, I. Balagura, and V. Andrushchenko, “The detection of actual research topics using co-word networks,” Open Semantic Technologies for Intelligent Systems (OSTIS-2018): Proceedings of the international scientific and technical conference, 2018.
15. Axelrod, Structure of decision: The cognitive maps of political elites. Princeton University Press, 1976.