Всеукраїнський конкурс на кращу студентську наукову роботу
2020/2021 навчального року

Шифр: «Розвідка»

Тема роботи: «Розробка та програмна реалізація методики цифрової розвідки на основі відкритих джерел»

Секція: «**Кібербезпека**»

АНОТАЦІЯ

наукової роботи під шифром "Розвідка".

Наукова робота: 36 сторінок, 9 рисунків, 1 таблиця, 29 джерел.

Наводиться аналіз існуючих методів розвідки за відкритими джерелами, зокрема з використанням сучасних джерел збору та агрегації даних, що є актуальним питанням у сфері інформаційної безпеки.

Метою роботи *є розробка та реалізація методики цифрової розвідки на основі агрегації інформації з відкритих джерел, що дозволяє зробити персоналізацію користувача в Інтернет-мережі*.

Наукова новизна *роботи полягає у тому, що розроблено засіб для проведення ідентифікації користувачів в Інтернет-мережі, запропонована методика вивчення Інтернет-активності*.

Практична цінність *результатів роботи полягає у тому, що розроблено програмне забезпечення, яке дозволяє на основі попередньо відомої інформації щодо користувача, зробити відповідну агрегацію даних з доступних Інтернет-джерел з метою подальшої персоналізації користувача Інтернет-мережі*.

Робота виконана в рамках кафедральної НДР № 04518 «Дослідження та аналіз методів і засобів кібербезпеки».

Основні положення й результати роботи доповідалися й обговорювалися на наукових конференціях.

РОЗВІДКА ЗА ВІДКРИТИМИ ДЖЕРЕЛАМИ, BITTORRENT, АКТИВНІСТЬ, ПЕРСОНАЛЬНЕ ІМ’Я, OSINT, NICKNAME, IP-АДРЕСА

ЗМІСТ

ВСТУП

Інформація у XXI столітті виступає найбільш ціннім ресурсом з усіх наявних. Використовуючись в економіці, політиці, військовій сфері, вона є важливим чинником для прийняття стратегічних та тактичних рішень. Отже, дане дослідження є актуальним, адже порушує насущну проблему добування та агрегації різноцільової інформації законним шляхом.

Таким чином, метою запропонованої наукової роботи є розробка та реалізація методики цифрової розвідки на основі агрегації інформації з відкритих джерел, що дозволяє зробити персоналізацію користувача в Інтернет-мережі.

Для реалізації цієї мети автором ставляться наступні завдання:

1) визначення місця концепції OSINT – розвідки за відкритими джерелами – у процесі добування та аналізу інформації.
2) використання принципів Data Scraping та Data-Crawling за допомогою високорівневих мов програмування з метою автоматизації процесу збору та дослідження інформації.
3) вивчення принципів функціонування протоколу BitTorrent, peer-to-peer мереж з метою ідентифікації користувачів.
4) розробка клієнт-серверного програмного забезпечення (ПЗ) з метою ідентифікації користувачів різними методами:
   a) виведення інформації про поточного користувача мережі;
   b) можливість пошуку та агрегації соціальних мереж, форумів, тематичних сайтів за нікнеймом користувача;
   c) можливість пошуку та агрегації інформації torrent-файлів, які скачувались з поточної IP-адреси або такої, що запитувалася.

У результаті розробки клієнт-серверного ПЗ був отриманий сервер, який реалізує мету дослідження, а саме надає можливості пошуку та агрегації інформації щодо «нікнейму» користувача та щодо завантажень torrent-файлів.

Для виконання перелічених завдань використовувалися емпіричні методи технічно-програмної реалізації, що передбачали використання мови програмування JavaScript для розробки серверної та клієнтської частини застосунку, веб-сервера на платформі Node JS тощо.

Отже, об’єктом цього дослідження є інформаційні потоки, яку можливо зібрати з відкритих джерел.

Предметом наукової роботи виступає інформація як ресурс, що дозволяє аналізувати та в подальшому ідентифікувати конкретного користувача мережі Інтернет.

З огляду на зазначене вище, дана робота має прикладну та теоретичну цінність. У роботі окреслена роль концепції OSINT, як загальнодоступного методу пошуку та агрегації даних різноцільового спрямування без порушення умов національного законодавства. Програмне забезпечення, розроблене задля реалізації ідеї наукової роботи, може використовуватися під час проведення аудиту використання комп’ютерних мереж з недоцільною метою на підприємствах; ідентифікації користувача у мережі Інтернет; інформування про потенційне порушення кримінального законодавства у сфері інтелектуальних прав.

1. ПОНЯТТЯ, ПРИЗНАЧЕННЯ ТА ЗАСТОСУВАННЯ РОЗВІДКИ ЗА ВІДКРИТИМИ ДЖЕРЕЛАМИ

1.1. Поняття OSINT. Історія застосування концепції

Відомий значний вплив інформації, але зростання кількості користувачів соціальних мереж, форумів або того, що можливо характеризувати як джерело вільної та відкритої інформації, спричинило відкриття нової методології інформаційної аналітики - розвідки за відкритими джерелами - OSINT.

Знайти загальноприйнятне визначення цього типу розвідки не можливо і сьогодні. Найбільш детальне визначення концепції OSINT міститься у програмному документі Північноатлантичного альянсу (НАТО) за 2001 рік – «Посібник НАТО з розвідки з використанням відкритих джерел» (англ. NATO Open Source Intelligence Handbook)[1]. У цьому посібнику перелічені поради і інструкції для аналітиків всіх ланок альянсу, що працюють з публічною інформацією. Зокрема, зазначене питання інформаційної гігієни, коректної агрегації, зберігання публічних даних. Але найбільш вичерпне визначення OSINT можна знайти у звіті Офісу Директора Національної Розвідки США за 2011 рік: у цьому документі зпзначено, що розвідка за відкритими джерелами – це розвідувальна діяльність, яка послуговується інформацією з загальнодоступних джерел, яка збирається, використовується та своєчасно надається аудиторії з метою задоволення потреб аналітичної служби [2].

Зростання впливу Інтернету та соціальних медіа зробили концепцію OSINT більш комплексним явищем, додавши до військової тематики ще сфери політики, економіки, кібербезпеки. Користувачі мережі залишають велику кількість інформації про себе самостійно: публікація фотографій з місця проживання, подій та явищ які пов’язані з цим місцем; висвітлення власних думок та вподобань у мікро-блогах; формування власної мережі онлайн-друзів у соціальних мережах тощо. Накопичення цього масиву інформації паралельно зі зростанням можливостей комп’ютерного аналізу розширили можливості для аналітиків щодо збору та агрегації даних. Усі ці зміни спровокували потребу у

формуванні нової концепції та методології збору інформації з відкритих джерел, яка допоможе систематизувати відкриті данні та прив'язати результати аналізу до мети дослідження.

Згідно з новим та більш загальним визначення, розвідка за відкритими джерелами (англ. - OSINT) – це концепція, методологія і технологічний принцип добування і агрегації військової, політичної, економічної та іншої інформації з відкритих джерел, без порушення законів.

Джерелами подібної інформації можуть бути [3]:

- ЗМІ: друковані газети, журнали, радіо та телебачення з різних країн.
- Інтернет: онлайн-публікації, блоги, відео з мобільних телефонів, вікі-довідники, соціальні медіа.
- державні дані: публічні урядові звіти, бюджети, слухання, телефонні довідники, прес-конференції.
- комерційні дані: фінансові та промислові оцінки, бази даних.
- сіра література – це інформація отримана від некомерційних організацій та інститутів. Зокрема: технічні звіти, патенти, робочі документи, ділові документи, неопубліковані роботи та інформаційні бюлетені.

Історія OSINT як концепції почалась з заснування 26 лютого 1941 року з відкриття Агентства з моніторингу іноземних трансляцій (FBMS) з метою агрегації та аналізу новин держав-членів Осі[1] та їх сателітів, використовуючи отримані газети, журнали, радіопередачі [4]. Завдяки опрацюванню великої кількості невпорядкованих даних, аналітики знаходили окремі факти (наприклад: інформація про результати бомбардувань, що базується на аналізі цін на продукти харчування; аналіз пропагандистських матеріалів щодо реакції населення на прихід військ до міста; матеріали про економічному становищі країн-супротивників) і використовували їх з метою верифікації інших джерел інформації щодо супротивника або як основне джерело. Згідно звіту директора

[1] Країни Осі — об'єднання країн в часи Другої світової війни. Три головні країни-елементи осі: націонал-соціалістичний Третій Рейх, фашистська Італія та Японська імперія

FBMS до президента Рузвельта, 95 відсотків інформації щодо економічного, психологічного становища Японії йшло через його агентство [5]. Згідно з доповіді голови відділу ЗМІ у FBMS: «Ми слухаємо що люди говорять своїм родичам з закордону, нейтральним країнам, світу у цілому…. …наші агенти, це люди які знайомі з психологією, мовою, культурою, економікою, традиціями ворожої країни. Через сенси ворожої пропаганди до своїх громадян, ми отримуємо тренди ворожої дипломатії або військових операцій»[6].

Під час Холодної війни завдяки роботі FBMS американське керівництво отримувало інформацію щодо радянських атомних ракет на Кубі, інформацію щодо радянської участі у конфлікті в Афганістані, інформацію щодо кризи в Угорщини та Чехословаччині. Загалом, 80 відсотків інформації про становище СРСР перед розпадом було отримано з загальнодоступних джерел [7].

Таким чином, концепція OSINT сформувалась як важлива ланка у системі розвідки США та країн НАТО.

1.2. Відмінності концепції OSINT від інших типів розвідки

У структурі розвідки країн НАТО, існує шість загальноприйнятих типів розвідувальної діяльності: аеророзвідка (IMINIT), агентурна розвідка (HUMINT), радіоелектронна розвідка (SIGINT), розвідка на основі фізичних полів (MASINT), геопросторова розвідка(GEOINT), розвідка на основі відкритих джерел (OSINT) [8].

Усі принципи є повністю незалежними один від одного і можуть лише комбінуватись у різних ситуаціях. Одночасно OSINT знаходиться поміж іншими типами та дає комплексну відповідь при використанні інших типів розвідки. Наприклад, GEOINT може бути частиною розвідки за відкритими джерелами, якщо користуватись послугами супутників комерційних установ: Google, Bing тощо та доповнювати інформацію з військових супутників; розвідка за відкритими джерелами у соціальних мережах може бути доповнена агентурною розвідкою або радіорозвідкою тощо.

На рисунку 1 [8] зображена візуальна репрезентація можливої конвергенції розвідувально-аналітичних дисциплін та наявні розмиті межі між ними.

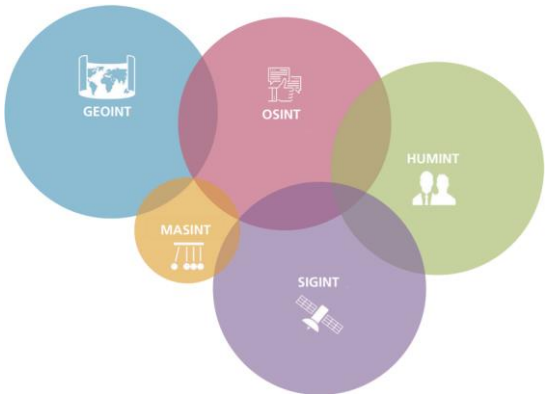

Рисунок 1 – Схема конвергенції різних типів розвідки

Таким чином, можна зробити висновок, що OSINT є важливим джерелом інформації, який доповнює вже отримані данні з інших джерел.

1.3. Сфери застосування концепції OSINT

Як було зазначено вище, методологія OSINT, будучи комплексним явищем, знайшла своє місце у багатьох сферах нашого життя, де потрібно встановлювати зв’язки між людьми, подіями, організаціями. Яскравим прикладом застосування OSINT окрім військової сфери може бути правоохоронна діяльність, кібербезпека, економічна безпека, сфера найму працівників, суспільними діячами [9].

У правоохоронній сфері OSINT використовується для запобігання, розслідування та переслідування злочинів пов’язаних з Інтернетом. Особливо це стосується протидії терористичним організаціям, незаконному відмиванню та легалізації грошей отриманих злочинним шляхом, боротьбі з розповсюдження наркотичних речовин, зброї тощо. Пошук у соціальних мережах потенційно небезпечних груп та індивідуумів є важливою частиною роботи міжнародних правоохоронних організаціях таких як Europol чи Interpol [10].

У сфері кібербезпеки, OSINT використовується при реверсивному аналізі систем безпеки з метою пошуку вразливостей, наприклад: аналіз відомої інформацію про компанію у інтернеті, перевірка працівників на дотримання службової таємниці у соціальних-мережах.

Служби безпеки комерційних установ також застосовують інструменти OSINT. Вони проводять індивідуальні перевірки власних співробітників, контрагентів. Зокрема, особливо важливою для служб безпеки є інформація про офшорні компанії, справжніх власників, залучення компанії-партнера до тіньових схем. Знання подібної інформації може мати вирішальне значення перед узгодженням будь-якої великої угоди.

OSINT використовується у страховому бізнесі. Це стосується аналізу персональних даних компанії та бізнес-аналітики. Наприклад, компанія зазначає, що в одному регіоні зросли виплати за окремий страховий продукт. Перевірка афілійованих осіб у соціальних мережах співробітників відділення компанії показала, що один із менеджерів страхував своїх друзів та родину, щоб потім реєструвати страхові випадки та виплати. Наявність такої інформації порушує питання доцільності внутрішнього розслідування.

Журналістська спілка також використовує засоби OSINT. Зокрема, це стосується більшості розслідувань корупційної діяльності. Журналісти використовуючи загальнодоступні декларації доходів, соціальні мережі порівнюють реальні статки з задекларованими. Також яскравим прикладом журналістської діяльності з використанням OSINT можуть бути праці спільноти Bellingcat створеної Еліотом Хіггінсом. Зокрема, волонтери проекту стали відомими після розслідувань збиття літака MH17 Російською Федерацією, оприлюднення фактів збройної агресії проти України з боку РФ (це буде детально описано у розділі 1.4), інформації щодо діяльність терористичної Ісламської Держави у Сирії. Наприклад, відомим є матеріал щодо визначення місцезнаходження одного з глав терористичної гілки ІДІЛ[2] у Сирії за фотографіями у соціальних мережах та за допомогою Google Maps[11];

[2] Ісламська держава Іраку та Леванту

матеріали щодо визначення позицій ракет під час бомбардування Дамаску 21 серпня 2013 року ракетами з зарином [12] тощо.

1.4. Приклади використання OSINT в Україні

До початку збройної агресії проти України практики OSINT публічно використовувались лише під час антикорупційних розслідувань журналістів. Але початок військових дій на теренах України сформував спілку OSINT розслідувачів, які займаються аналізом військової інформації. Такими спілками є InformNapalm, центр «Миротворець».

Однією з перших спілок, які почали працювати з використанням методології OSINT, є InformNapalm [13]. За час існування цієї спілки було проведено багато ґрунтовних розслідувань стосовно поповнення списку військовослужбовців РФ, які приймали участь у війни проти України, розслідувань стосовно збиття боїнгу МН-17. У своїй роботі спілка послуговувалася аналізом облікових записів у соціальних мережах місцевих жителів Донбасу та військовослужбовців РФ. Приклад подібних розслідувань: установлення даних 3 військових частин приймаючих участь у агресії проти України [14]; докази перегрупувань танків з РФ до України [15].

Не менш відомою організацією є центр «Миротворець» [16]. У своїй діяльності вони публікують особисті дані бойовиків незаконних збройних формувань та військовослужбовців РФ. Зокрема, велика кількість інформації було отримано з соціальних мереж та психологічних навичок волонтерів, які спілкувались з військовими. Приклад подібних розслідувань: знаходження особистих даних терориста за допомогою соціальних мереж [17]; матеріали щодо коректного пошуку персональних даних за допомогою соціальних мереж і державних відкритих баз даних[18].

Також у своїх дослідженнях методами OSINT послуговується зазначена вище спільнота Bellingcat [19]. Створений окремий розділ, у якому волонтери займаються дослідженням катастрофи МН17. Більша частина цих даних була отримана з облікових сторінок військових РФ у соціальних мережах

«ВКонтакте» та «Однокласники» та підтверджені відеозаписами з публічних відеокамер, Google Maps тощо [20]. Отримана інформація використовується міжнародною слідчою групую у справі розслідувань збиття МН-17. Також завдяки сервісу Google Earth Pro та супутниковими знімками були підтверджені факти обстрілів Українських військ з території Гуково (РФ) [21].

Таким чином, в Україні практика OSINT реалізована в багатьох сферах - особливо у сфері національної безпеки та оборони. У більшості випадків волонтери з метою аналізу військових злочинів послуговуються соціальними мережами (аналізують фотографії, інформацію з особистих сторінок, використовують методи соціальної інженерії), відкритими картографічними сервісами (Google Maps, Bing Map, Yandex Map). Наявність такої кількості компрометуючої інформації щодо пересувань, особистих даних військових у загальнодоступних мережах порушує питання коректної організації інформаційної безпеки. Загальна доступність інформації та «цифрові сліди» у соціальних мережах підтверджують неможливості приховування актів військових злочинів, тероризму та замовчування фактажу у ЗМІ.

2. ПОНЯТТЯ ВЕБ-СКРАПІНГУ. ВИКОРИСТАННЯ ВЕБ-СКРАПІНГУ У КОНЦЕПЦІЇ OSINT

Концепція OSINT потребує формування підходу збору та агрегації даних, який дозволить автоматично збирати та упідпорядковувати данні. Як було зазначено у 1 розділі, OSINT-методологія орієнтована також на опрацювання інформації з соціальних мереж, блогів тощо. Зазвичай для аналітика потрібно, щоб веб-сторінка людини у соціальній мережі перетворилась на структурований набір таблиць або баз даних. Схематичний приклад подібного процесу зображений на рисунку 2.

id214ksg423klaf42134	
Им'я/ Призвіще	Марк Цукерберг
Місце роботи	Chan Zuckerberg Initiative
Освіта	Вища; Harvard University
Фото	https://facebook.com/photo....
Улюблена музика	LCD Sounsystem American Dr..
Улюблені книги	Платон
Улюблені фільми	Pixar
Улюблена спортивна команда	Yankees
.....	.....

Рисунок 2 – Схема переходу від HTML-сторінки до табличної форми виводу даних

Процес подібного перетворення даних називається веб-скрапінгом. Він грунтується на перетворенні «людино-сприйнятної» форми відображення інформації на веб-платформі (HTML або XHTML сторінка) до машиносчитуємої (XML, JSON, CSV). Подібний процес виконується спеціальними скриптовими програмами, які імітують поведінку користувача сайту. Веб-скрапінг починається з завантаження HTML-коду сторінки через HTTP протокол, вилучення з отриманого коду текстових даних та подальша обробка або фільтрація інформації.

Завдяки подібному принципу агрегації даних працюють багато пошукових систем, погодних сайтів, інтернет-магазинів тощо.

3. ІДЕНТИФІКАЦІЯ КОРИСТУВАЧА У МЕРЕЖІ BIT-TORRENT

3.1. Однорангова мережа. Протокол Bit-Torrent

Основним архітектурним типом побудови комп'ютерних мереж є так звана клієнт-серверна, приклад якої можна побачити на рисунку 3.

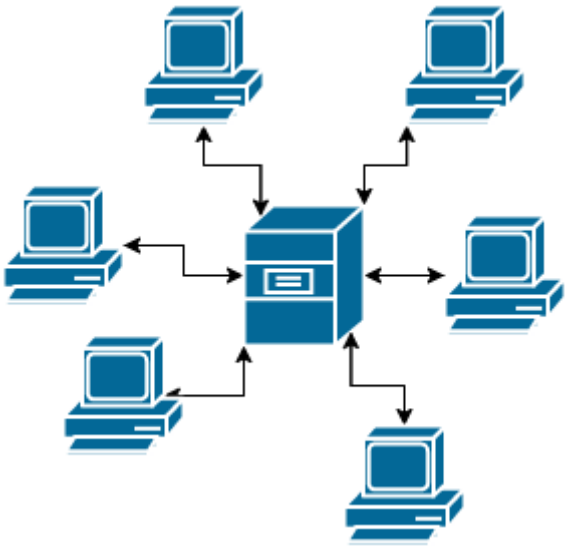

Рисунок 3 – Схема «клієнт-серверної» архітектури

Архітектура цього плану є дуже простою у реалізації через наявність одного «достовірного» джерела даних, з яким і буде працювати «клієнт». Під серверною частиною у цій архітектурі розуміється об'єднання різного виду серверів (поштового, веб, термінального) та бази даних. Однак серверна частина потребує правильного обслуговування, постійного масштабування (додавання інших серверних компонентів, заміни обладнання), наявності штату системних адміністраторів для обслуговування. Особливо помітними ці проблеми постають під час зростання кількості користувачів мережі та кількості інформації створеної ними. Таким чином, клієнт-серверна мережа може стати занадто дорогою у обслуговуванні.

Вирішити цю проблему можливо з переходом або часткової заміни «повільних» ланок мережі на однорангову архітектуру. Приклад схеми подібної мережі зображений на рисунку 4.

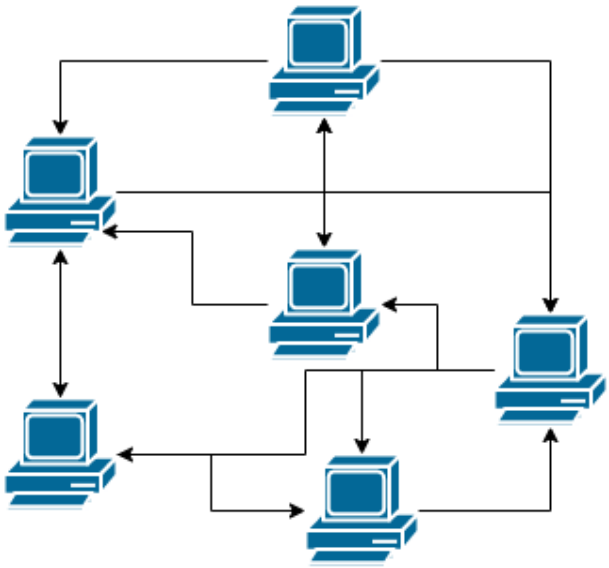

Рисунок 4 – Схема «однорангової» архітектури

Головна відмінність від звичної клієнт-серверної архітектури у тому, що центральний сервер відсутній або його роль мінімізується. Клієнти подібної мережі можуть обмінюються інформацією одне з одним, виконуючи одночасно роль як сервер і як клієнт. Кожен член подібної мережі не гарантує своєї присутності на постійній основі: він має можливість як з’являтись, так і зникати у будь-який час, тому серверна машина повинна ще виконувати адміністративні функції та підтримувати актуальність інформації о клієнтах. Подібна організація дозволяє зберегти повну працездатність мережі за будь-якою кількістю вузлів та різною апаратною конфігурацією.

Питання прикладного дослідження якості однорангових мереж порівняно з клієнт-серверними опрацьовано у дослідженні «Ghareeb, M., Rouibia, S., Parrein, B., Raad, M., & Thareau, C. (2013). P2PWeb: A Client/Server and P2P hybrid architecture for content delivery over internet. 2013 Third International Conference on Communications and Information Technology (ICCIT)» [22]. З метою практичної перевірки дослідниками були створені прототипи файлового серверу «клієнт-серверної» та «однорангової» архітектури з пропускним каналом 2 Мбіт/с у яких замірювались швидкість передачі файлів до клієнтів, кількість яких поступово мала зрости до 23. Одноранговий сервер мав віддавати файли першим вісьмом клієнтам, а далі його функції були лише у

передачі списку поточних користувачів мережі з якими можливо обмінюватись даними, а пропускна швидкість зменшилась до 300 Кбіт/с. На Рисунку 5 [22] можна побачити, що у клієнт-серверної архітектури під час передачі файлів є пряма залежність затримки від кількості одночасно завантажуючих файл клієнтів: чим більша кількість клієнтів – тим більша затримка.

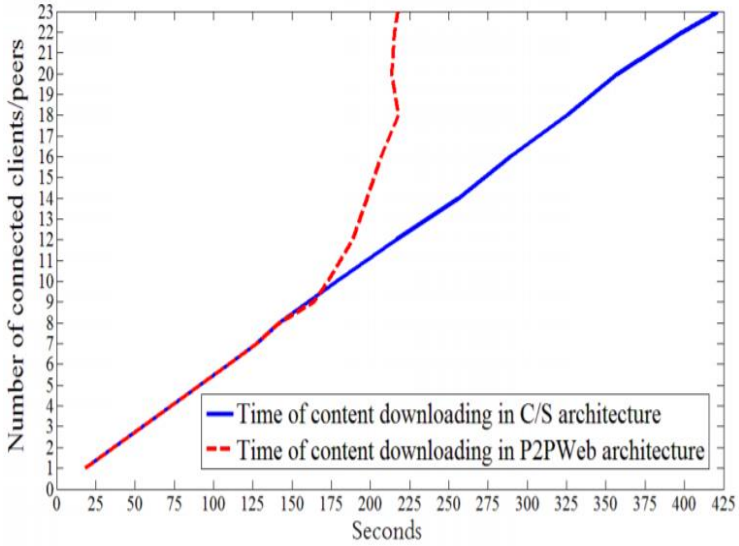

Рисунок 5 – Порівняння «клієнт-серверної» та «однорангової» архітектури за часом передачі інформації. Синій графік– час завантаження контенту з серверу з «клієнт-серверною архітектурою». Червоний графік – час завантаження контенту з серверу з «однорангового» серверу

Маючи такий незалежний від сервера підхід у передачі інформації, однорангова мережа потребує першочергової «підтримки» від сервера першим користувачам. Наступні клієнти мережі вже мали змогу обмінюватись інформацією поміж собою, таким чином швидкість передачі даних вже не залежала від пропускної здатності сервера.

Основними сферами застосування однорангових мереж є сервіси потокової передачі медіа-контенту, платіжні системи, файлообмінні мережі.

Найбільш поширеною серед користувачів мережі-Інтернет є файлообмінні сервіси, побудовані завдяки протоколу BitTorrent. Зокрема, у звіті компанії SandVine під назвою «The Global Internet Phenomena Report September»[23] надається факт, що більше 27.58 відсотків завантажень і 2.46 відсотків скачувань файлів у мережі належать BitTorrent.

BitTorrent є повністю відкритим протоколом обміну інформації. Протокол розроблявся з метою облегшити обмін між частинам мережі файлів великого розміру. Основним є принцип, за яким клієнт який вже отримав перший «пакет» файлу, що запитується, одразу починає сам розповсюджувати отримані данні у мережу навіть не маючи цілого файлу.

Після підключення до BitTorrent мережі, клієнт (у BitTorrent-мережах має назву «пір») має отримати список клієнтів поточної мережі з наявними у них пакетами. Для подібної ініціалізації обміну інформації у мережі є так звані трекери. Трекер — це спеціалізований сервер, який працює за допомогою HTTP протоколу. На трекері зберігаються IP-адреси клієнтів, вхідні порти клієнтів та хеш-суми, які унікальним чином ідентифікують об'єкти, що беруть участь у закачуваннях. Також трекер виконує роль збирача статистичної інформації, метаданих, опису файлів тощо.

Перед початком завантаження клієнт має ініціалізуватись у трекера за допомогою власної IP-адреси та хеш-суми запитуємих файлів. Цей процес називається «анонс». У відповідь клієнт отримає список з IP-адресою інших учасників мережі (у BitTorrent-мережах має назву «рій»), які мають цей файл. Після ініціалізації клієнти мають змогу з’єднатись один з одним та обмінятись інформацією про наявні у них сегменти файлу. Після того, як клієнт оголосить про свою зацікавленість у сегменті даних і якщо інший клієнт у якого сегмент є відповість готовністю до передачі, почнеться процес завантаження. Після цього проводиться порівняння контрольних сум отриманого сегменту з контрольною сумою потрібного сегменту з трекеру та проводиться повідомлення інших учасників мережі про успішність або помилку під час передачі. Кожний клієнт зберігає інформацію о отриманих і неотриманих сегментах від інших клієнтів і формує власний «пріоритетний» список IP-адрес, які віддають більше якісної і «неушкодженої» інформації. При цьому пріоритет на обмін мають ті сегменти, які є у цей момент найрідкіснішими у мережі, що підвищує якість передачі даних.

3.2. Використання DHT-протоколу з метою ідентифікації користувача

Як було зазначено вище, протокол BitTorrent використовує трекер для процесу «анонсу» у мережі. Така залежність від центрального серверу робить торрент-мережу вразливою до зміни пропускної здатності Інтернет-каналу трекера. Не менш важливою проблемою постає потенційний вплив третьої сторони у коректність функціонування мережі: DDoS-атака на трекер, закриття трекеру його власником тощо. Таким чином, наступним кроком у розвитку BitTorrent-мереж є поступова відмова від трекеру.

Починаючи з версії 4.2.0 офіційного BitTorrent-клієнт з’явилась реалізація безтрекенгової роботи на основі протоколу Kademlia(Kad). У новому типі мережі трекер стає повністю децентралізованим та його функції переходять на клієнтів-учасників мережі, через яких і реалізуються пошукові запити, передача інформації тощо. Сам принцип лежить на використанні розподіленої хеш-таблиці (Distributed hash-table, DHT) та її розширення для спілкування між клієнтами - PEX (Peer exchange) [24].

Механізм комунікації завдяки DHT-таблиці побудований на основі UDP протоколу. Клієнти повинні слухати порт, працюючий з UDP, який вони використовують для вхідних з'єднань через TCP.

У DHT мережі кожен пір є повністю окремим вузлом зі своїм унікальним ID (ідентифікатором), випадково обираним з 160-бітного поля хешу торента.

Кожен вузол зберігає таблицю маршрутизації, яка містить контактну інформацію про «найближчих» до себе вузлів. Параметр «близькості» залежить від схожості ID і ніяк не корелюється з місцеположенням піру.

Коли вузол хоче знайти BitTorrent-мережу, він має порівняти хеш роздачі з ID відомих йому вузлів які зберігаються у DHT-таблиці. Після завершення процесу порівняння, пір знаходить вузол у якого ID найбільше схожий до шуканого і надсилає йому запит. Запитуваний вузол при отриманні подібного запиту від іншого піру порівнює отриманий ID з наявними у власній DHT-таблиці і повертає іншого піра, який ще більше схожий до необхідного хешу.

Тоді вузол-шукач посилає вже новий запит знайденому вузлу, і отримує від нього адресу наступного вузла, ID якого ще більш схожий на хеш торента.

Виходячи з цього алгоритму, клієнт-шукач поступово буде отримувати вузли, хеш яких найбільше схожий до шуканого торенту. Кожний з вузлів цього ланцюга запитів записує у власну DHT-таблицю інформацію про автора запита і його результат, поступово заповнюючи її більш актуальною інформацією та розширюючи власний список «контактів».

Проте викладений алгоритм на основі DHT має суттєвий недолік зі сфери інформаційної безпеки: кожний клієнт при підключенні до рою віддає усій мережі власну IP-адресу та свій унікальний хеш який ідентифікує його у рою. Кожний клієнт, у цей же час, може бути учасником інших мереж, у яких він так само надає власний хеш та IP-адресу іншим учасникам мережі.

Таким чином, подібна проблема відкритості даних надає великі можливості для експлуатації цієї вразливості. Для аналітика необхідно розробити власний торент-клієнт основний функціонал якого збір «анонс»-запитів та їх подальший аналіз. Прикладом подібного сервіса може бути: програмне забезпечення, яке допомагає правовласникам відшуковувати порушників авторського права; розробка статистичної платформи для аналізу інтересів користувачів BitTorrent протоколу; розробка моніторингової системи, яка дозволяє аналізувати активність у потенційно «небезпечних» мережах через які розповсюджується заборонений контент.

На даний момент існують вже готові програмні рішення з експлуатацією вразливостей DHT-таблиці. Переважна кількість з них працює як онлайн-пошуковець за IP-адресою, який надає користувачу можливість вивести інформацію про відому за цією адресою активність у мережі, а саме: назву файлу, його вміст, час та дату завантаження, тип файлу (аудіо чи відео контент) тощо. Подобними сервісами користується велика кількість спеціалістів з різних сфер під завдання яких адаптовані ці пошукові сервіси. Наприклад, спеціалісти з правоохоронних органів можуть користуватись «cps.gridcop»[25], розробленою Wyoming ICAC; маркетологи, рекламні агенти можуть

користуватись «iknowwhatyoudownload.com» у якому наведена статистика окремої IP-адреси (список завантажених файлів через BitTorrent) або самого хешу (інформація про кількість завантаження по країнам).

Для розвідника за відкритими джерелами інформація про завантажені торент-файли стає потенційною «точкою зачіпки» для вивчення можливих інтересів людини і при сумісному аналізі інших джерел може доповнити, підтвердити або спростувати наявну інформацію.

4. РОЗРОБЛЕННЯ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ OSINT-ПЛАТФОРМИ

4.1. Мета та потенційне застосування програмної реалізації

Аналізуючи отриману під час дослідження інформацію, автор поставив мету розробки програмного веб-застосунку, який дозволить практично продемонструвати та автоматизувати процес пошуку інформації у мережі Інтернет завдяки методологіїї «розвідки за відкритими джерелами» . Особливо корисним цей застосунок має бути на етапі початкового збору інформаціїї про аналізуємого користувача мережі.

Для демонстрації використання методу «веб-скрапінгу» при реалізації автоматизації OSINT є створення сервісу з пошуку у мережі Інтернет популярними веб-сервісами, форумами, соціальними мережами особистого імені користувача або «нікнейму». Треба пам’ятати, що переважна кількість користувачів інтернет мережі використовує одне особисте ім’я для багатьох веб-застосунків, тому реалізація подібного пошуку може пришвидшити аналіз та агрегацію інформації про користувача та допомогти сформувати структурну схему з різних профілів і ассоціювати отримані дані з окремою людиною.

Згідно з отриманою у третій частині роботи інформацією, BitTorrent-мережа має архітектурний недолік при експлуатації якого можливо визначити деякі персональні данні (зокрема, IP-адресу, інформацію про завантаженні торент-файли(вміст, час їх завантаження,тип файлу тощо)). Таким чином, використовуючи вже готові програмні рішення(зокрема веб-сервіс iknowwhatyoudownload.com) та методику веб-скрапінгу, маємо можливість розробки відкритого для дослідників веб-інтерфейсу системи ідентифікації користувача у BitTorrent-мережі.

Отримане програмне забезпечення має бути пристосованне для використання різноманітною авдиторією. Таким чином, програмний сервіс окрім функціоналу пошуку інформації про персональне ім’я чи завантаженні торрент-файли за окремою IP-адресою збирає наявну з інтернету інформацію

про контент, який потенційно міг завантажити користувач. Прикладом збираємої інформації може бути тип завантаженої інформації (фільм, книга, комп’ютерна гра, розмір файла, опис тощо).

4.2. Засоби реалізації

Під час архітектурного планування програмної реалізації було обрано рішення розробити веб-застосунок, який має здатність до швидкого та легкого розширення. Рішення про створення веб-застосунку пов’язано з бажанням зробити застосунок мультиплатформеним, при цьому не використовуючи різні мови програмування для різних типів устройств. Веб-застосунок повинен ефективано працювати за умовах багатьох одночасних підключень та використання алгоритмів «веб-скрапінгу» з мережі Інтернет, при цьому забезпечуючи мінімізацію використання серверного часу.

Для реалізації алгоритму «веб-скрапінгу» було прийняте рішення використати мову программування з неблокуючим виводом та асинхронним методом виконання. Зокрема, керуючись сучасними архітектурними підходами до створення програмного забезпечення, вирішено почати розробку на основі програмного комплексу MERN. Основна особливість цього комплексу - це підтримка однієї мови программування для розробки серверної і клієнтської частини додатку – JavaScript. MERN об’єднує у собі чотири структурні елементи веб застоунку: базу даних, серверна платформа, клієнтський інтерфейс, інтерператор мови програмування. За зберігання даних у цьому стеці відповідає СУБД MongoDB з використанням документоорієнтованого підходу, який спрощує розробку розподілених додатків; основна платформа виконання коду на боці сервера – це інтерператор Node.js, поверх якого використовується «каркас» у вигляді Express.js, а розробка інтерфейса заснована на платформі React JS зі створенням швидких односторінкових застосунків.

Маючи різні за побудовою функціональні частини у застосунку, логіка програми була розбита на окремі програмні модулі, які мають «з’єднуватись» з сервером і формувати його функціональну частину. Таким чином,

підтримується парадигма «компонентної» розробки і полегшується подальше масштабування застосунку. Приклад подібних функцій-модулів наведений у таблиці 1.

Таблиця 1 – Приклади розроблених функцій-модулів

Ім'я функції	Передані значення	Отриманні значення	Опис функції і лістінг програмного коду реалізації
getTorrents	IP-адреса користувача; Формат виводу даних: 1) вивести всіх; 2) вивести тільки останній	Об’єкт, вмістом якого є: 1) масив завантажених торентів; 2) індикація про потенційну наявність забороненого контенту	Функція виконує запит до веб-сайту з метою початку процедури Data Scraping; лістінг А.1
ipLocate	IP-адреса користувача;	Об'єкт, у якому наведена інформація щодо країни походження адреси, місто походження, географічні координати	Функція розширює інформацію про IP-адресу з BitTorrent мережі; лістінг А.2
parseData	HTML-код вивчаємої сторінки	Об’єкт з вмістом HTML-сторінки	Функція виконує процедуру перетворення HTML-стркутури сторінки до JSON-формату; лістінг А.3
nickname GetInformationAbout Services	Особисте ім’я(нікнейм) користувача	Масив сервісів з інформацією про наявність чи відсутність запитаного нікнейму на сервісі	Функція необхідна для виконання AJAX-запитів до «функціонального» веб-сервера і збереження отриманої інформації у пам'ять клієнтського застосунку; лістінг А.4
home	----------	HTML-код головної сторінки	Функція необхідна для формування логіки та відображення головної сторінки та початкової ініціалізації AJAX-запитів до «функціонального» веб-сервера; лістінг А.5

Під час розробки веб-застосунку особливу увагу приділялось налаштуванню систем захисту. З метою попередження потенційних «атак на відмову в обслуговуванні» прийнято рішення створити два окремих веб-сервери, які мають різні ролі і з’єднанні між собою через AJAX-запити, де

перший – це «функціональний» сервер-обробник інформації отриманої шляхом Data Scraping що знаходится за адресою scrambler-project-backend2-0.herokuapp.com [26], а другий – це сервер зберігаючий React JS застосунок з можливістю перенаправлення запитів до першого сереверу за адресою scrambler-project2-0.herokuapp.com [27]. Таким чином подібна схема робить невідомою адресу «функціонального» серверу, що ускладнює пошук шляхів «впливу» на нього. Схема перенаправлення запитів між серверами зображена на рисунку 6.

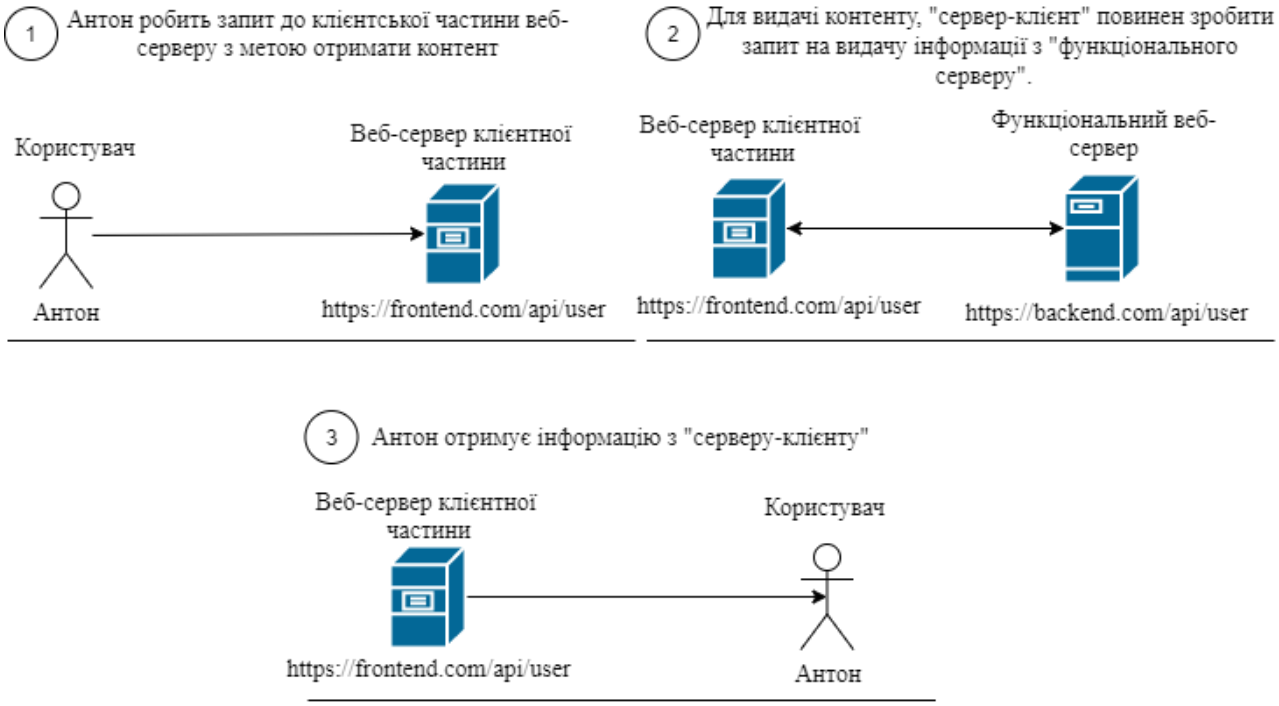

Рисунок 6 – Схема перенаправлення запитів між серверами

Ознайомитись з практичною реалізацією веб-платформи можливо за посиланням: scrambler-project2-0.herokuapp.com [27], а побачити програмний код – на git-репозіторіїї проектів серверу[28] і клієнту [29]. Таким чином, був створений клієнт-серверний застосунок, який має головну сторінку на рисунку 7.

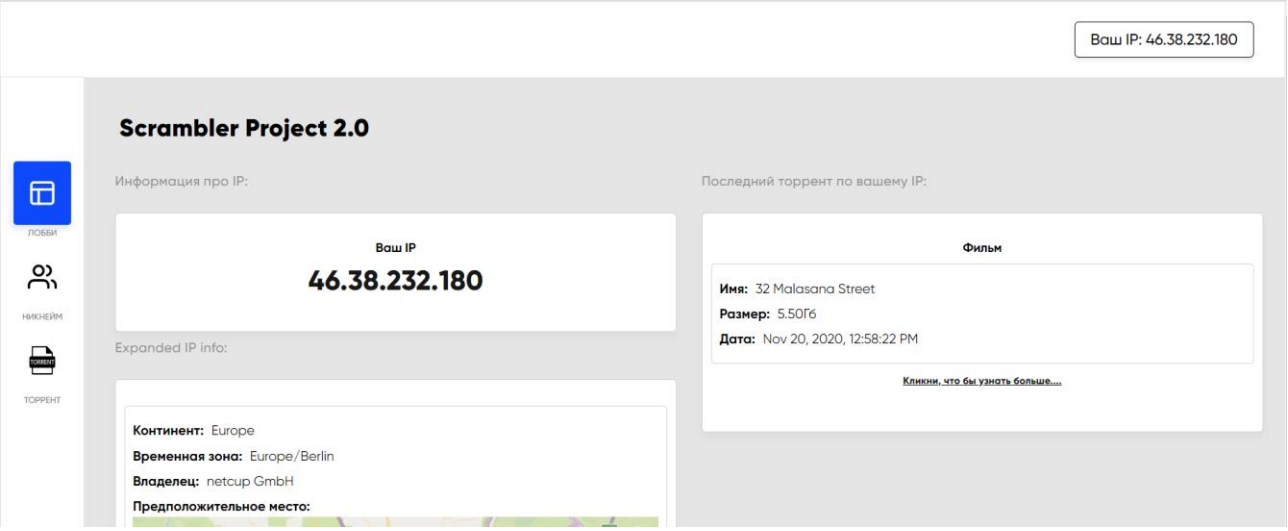

Рисунок 7 – Зображення головної сторінки інтерфейсу

4.3. Пошук людини за псевдонімом

Як було зазначено у пункті 4.1, люди переважно використовують один псевдонім для різних типів сервісів. Механічний перегляд та аналіз наявності певного користувача на певному форумі, соціальній мережі може бути часозатратним. Цей процес можливо автоматизувати за допомогою програмних засобів. Таким чином, для пришвидшення процесу аналізу необхідно сформувати список аналізуємих веб-додатків на яких буде шукатись певний псевдонім. Також потрібно пам’ятати що різні сервіси мають різні підходи до формування власного інтерфейсу, що може ускладнити процес автоматизації.

Перед почаком написання програмного коду-реалізації автоматизованого пошуку методами OSINT необхідно вивчити структуру HTML сторінки, проаналізувати запити від інтерфейсу до серверу щоб сформувати картину передачі інформації між аналізуємим клієнтом та сервером.

У результаті сформовано список 134 веб-серверів на основі яких буде виконуватись процес пошуку. Для того, щоб виконати пошук по цим джерелам у програмній розробці необхідно перейти на вкладку «Никнейм». На цій сторінці є веб-форма запиту до сервера, у якій користувач може ввести шуканий нікнейм та побачити візуальний результат у вигляді посилань на

профілі користувача у веб-застосунку. Приклад роботи з програмою зображений на рисунку 8.

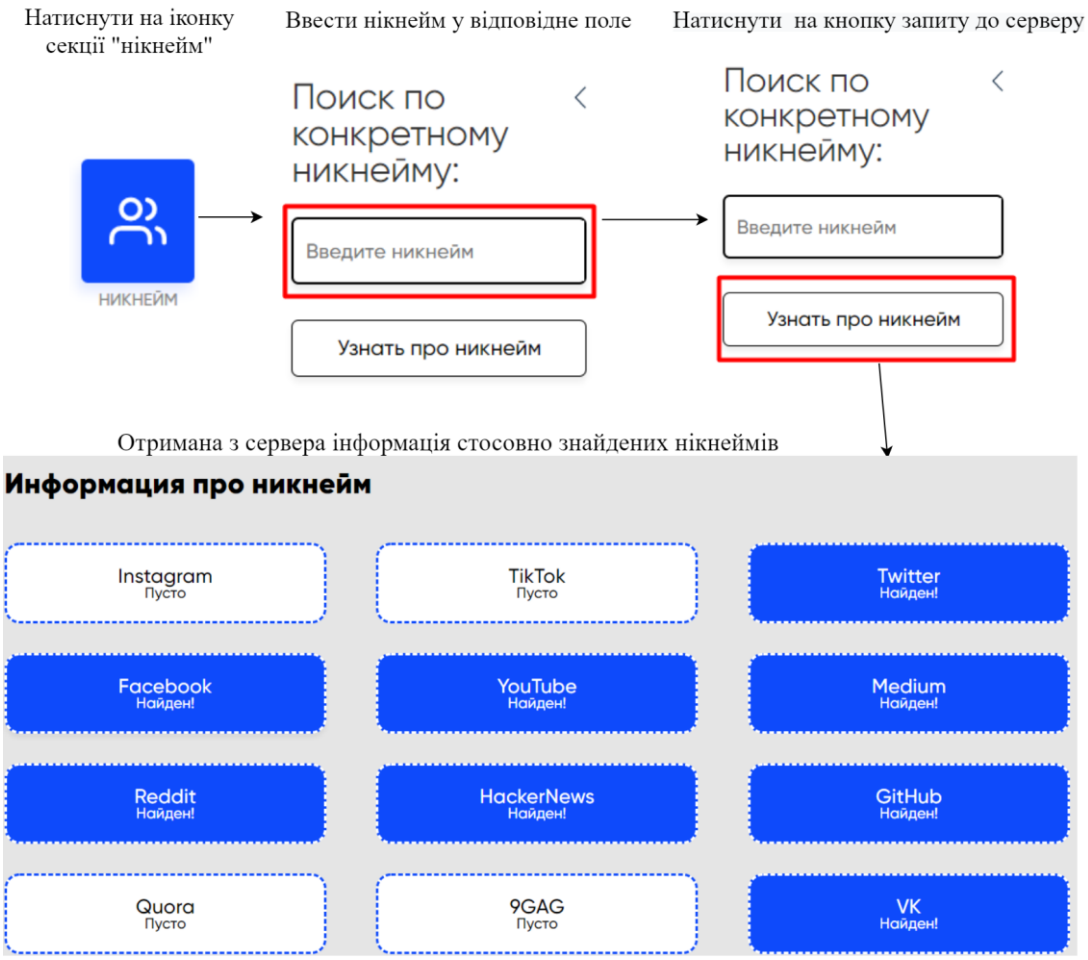

Рисунок 8 – Інтерфейс пошуку за нікнеймом

На основі отриманої інформації можливо продовжити подальший аналіз профілей користувача у Інтернет-мережі.

4.4. Пошук інформації про користувача BitTorrent мережі

У пункті 3.2 наведен принцип, за яким виконується пошук інформації о користувачі у мережі BitTorrent. Особливу роль у ньому відводиться використанню DHT-таблиць, у яких міститься інформація щодо учасників BitTorrent-мережі.

Таким чином, якщо дослідник має хеш-ідентифікатори великої кількості торрентів, то йому необхідно написати власного «торрент-клієнта», який буде

постійно моніторити мережу завдяки «анонс» запитам, що дозволить отримати інформацію щодо всіх учасників мережі . Масштабуючи процес моніторінгу, можна зустріти однакові IP-адреси у різних «роях», що дозволить робити вибірку вже не за наявними у «клієнта-спостерігача» мережами, а саме за IP-адресами і контентом, який поширюють у мережі BitTorrent.

Пошук та агрегацію інформації щодо учасників мережі подібним шляхом виконує низка відкритих та приватних сервісів. Таким чином, використовуючи парадігму OSINT, можливо використати ці сервіси задля більшої персоналізації людини-володільця IP-адреси.

Метою використання розробленого програмного забезпечення є автоматизація збору інформації про учасників BitTorrent мережі з доступних джерел. Для цього у веб-застосунок влаштований Node JS веб-парсер Cheerio, який аналізує структуру HTML-коду відповідної веб-платформи та збирає інформацію про контент завантажений з конкретної IP адреси. Для деяких типів файлів, зокрема фільмів, можливий пошук по відкритим даним. У результаті розробки ПЗ та з метою поліпшення процесу парсінгу утворено власний програмний інтерфейс, який є відкритим для інших розробників та інформація з якого виводиться на веб-сайт.

Для того, щоб почати пошук інформації про торенти, ассоційовані з окремим IP, потрібно вибрати вкладку «Торрент». Через форму запиту, яка знаходится у меню сторінки, користувач має можливість ввести шукану IP-адресу та отримати таблицю зі спиком завантажених через протокол BitTorrent файлів. Приклад роботи з програмою зображений на рисунку 9.

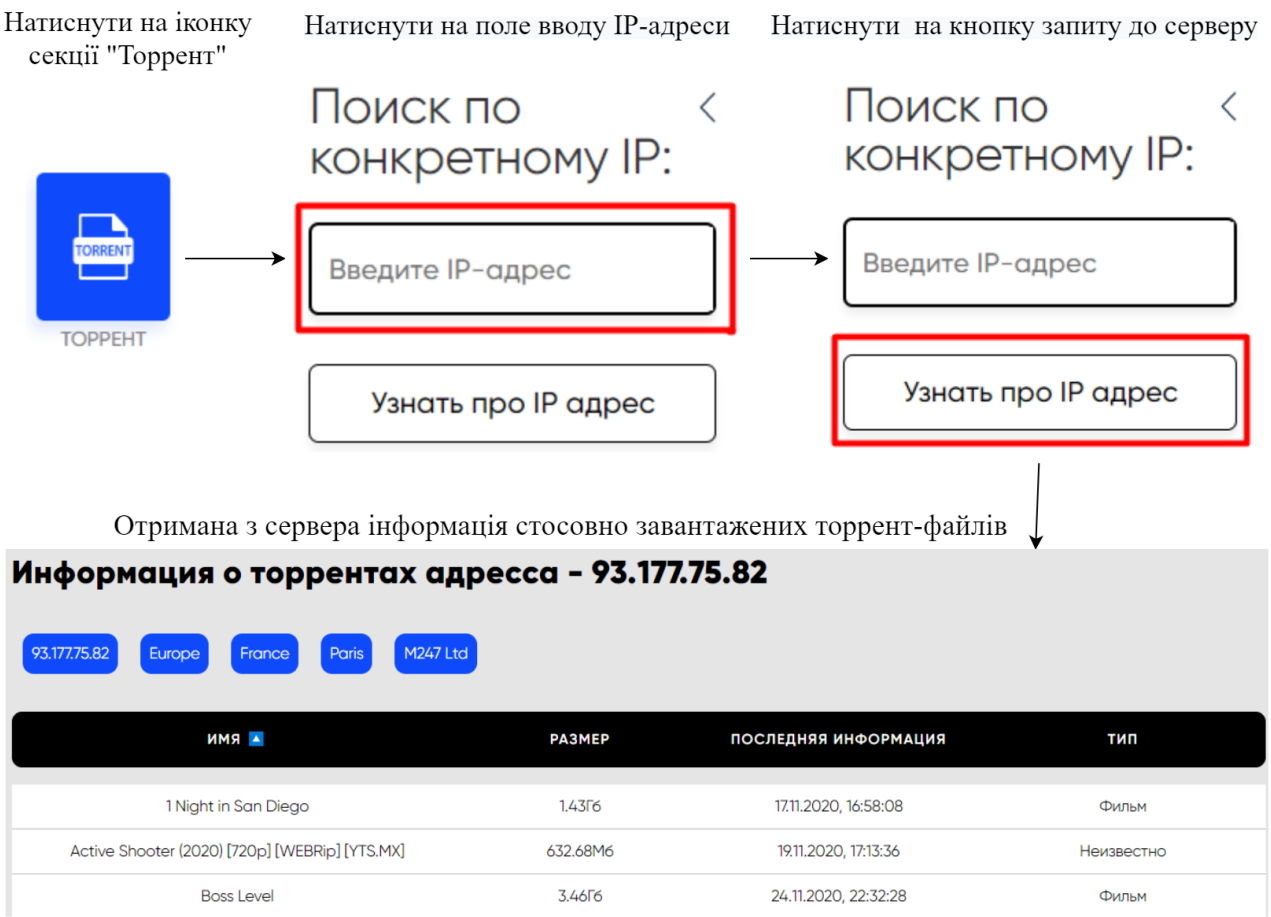

Рисунок 9 – Інтерфейс пошуку завантажених IP-адресою торентів

Таким чином, отримана інформація з відкритих джерел може стати частиною процесу аналізу інформації щодо користувача. Наприклад, подібне джерело інформації може бути використане правоохоронними органами з метою аналізу порушень у сфері інтелектуальних прав або попередження завантаження забороненого контенту.

ВИСНОВКИ

Під час виконання цієї наукової роботи була зроблена програмна реалізація методики цифрової розвідки на основі агрегації інформації з відкритих джерел, що дозволяє зробити персоналізацію користувача в Інтернет-мережі.

Розвідка за відкритими джерелами – тип аналізу інформації на основі мережі Інтернет. Ця концепція обусловлює правила за якими проводиться пошук необхідної для дослідження інформації з роздільних веб-платформ.

З метою автоматизації процесу аналізу інформації з відкритих джерел використовують програмні застосунки, які називаються «парсерами». Ці парсери працюють за технологією Data Scraping-у та виконують збір «людино-зрозумілої» інформації з різного роду сервісів(соціальних мереж, форумів, сайтів зі звітами тощо) та роблять перетворення цієї інформації у машинозчитувальну форму(XML, JSON).

Простіром для застосування концепції OSINT у науковій роботі є протокол BitTorrent. Займаючи високий рівень популярності серед користувачів та маючи лідерськи позиції за кількістю переданої через протокол інформації, BitTorrent має архітектурні вразливості, що можуть призвести до подальшої деанонімізації користувача. Таким чином, можливою є розробка програмного забезпечення, яке зчитує інформацію про завантаження з BitTorrent мережі окремою IP-адресою.

На даний момент, існує декілька платформ які реалізують пошук у мережі BitTorrent. За допомогою концепції OSINT був розроблений веб-застосунок з функцією веб-парсеру, який збирає інформацію з цих сервісів з метою формування у користувача повної картини завантажень файлів з мережі BitTorrent за запитаною адресою.

Напрямом подальшого розвитку за викладеною темою бачимо перехід від системи на основі веб-парсингу готових рішень, до створення власної системи мережевих ботів, які аналізують BitTorrent мережу.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. NATO Open Source Intelligence Handbook [Електронний ресурс]. Режим доступу до ресурсу: http://www.oss.net/dynamaster/file_archive/030201/ca5fb66734f540fbb4f8f6ef759b258c/NATO%20OSINT%20Handbook%20v1.2%20-%20Jan%202002.pdf – Назва з екрана
2. U.S. National Intelligence: An Overview 2011 //2011. — 54 с. [Електронний ресурс]. – Режим доступу до ресурсу: https://www.hsdl.org/?view&did=697740. – Назва з екрана.
3. Richard A. Best «Open Source Intelligence (OSINT): Issues for Congress, Congressional Research Service» // 2005. — 9 с. [Електронний ресурс]. – Режим доступу до ресурсу: https://fas.org/sgp/crs/intel/RL34270.pdf . – Назва з екрана.
4. Joseph E. Roop «Foreign Broadcast Information Service History. Part I: 1941-1947» //1969. — 7 с. [Електронний ресурс]. – Режим доступу до ресурсу: https://www.cia.gov/library/center-for-the-study-of-intelligence/csi-publications/books-and-monographs/foreign-broadcast-information-service/1-FBIS-Early-Beginnings.pdf. – Назва з екрана.
5. Joseph E. Roop «Foreign Broadcast Information Service History. Part I: 1941-1947 Chapter 2 - Impact of Pearl Harbor» //1969. — 9 с. [Електронний ресурс]. – Режим доступу до ресурсу: https://www.cia.gov/library/center-for-the-study-of-intelligence/csi-publications/books-and-monographs/foreign-broadcast-information-service/2-FBIS-Pearl-Harbor.pdf. – Назва з екрана.
6. Joseph E. Roop «Foreign Broadcast Information Service History. Part I: 1941-1947 Chapter 2 - Impact of Pearl Harbor» //1969. — 11 с. [Електронний ресурс]. – Режим доступу до ресурсу: https://www.cia.gov/library/center-for-the-study-of-intelligence/csi-publications/books-and-monographs/foreign-broadcast-information-service/2-FBIS-Pearl-Harbor.pdf. – Назва з екрана.
7. Admiral William Studeman, «Teaching the Giant to Dance: Contradictions and Opportunities in Open Source Within the Intelligence Community» ,//1993. — 1-2 с. [Електронний ресурс]. – Режим доступу до ресурсу: http://www.oss.net/dynamaster/file_archive/090716/f571532e0af491b3aefe870fe9f454f0/AIJ%2092%20011-018%20Studeman.pdf . – Назва з екрана.
8. Williams, H., & Blum, I. (2018). Defining Second Generation Open Source Intelligence (OSINT) for the Defense Enterprise. Defining Second Generation Open Source Intelligence (Osint) for the Defense Enterprise. https://doi.org/10.7249/rr1964 – Назва з екрана.
9. Where to use OSINT in your Business? [Електронний ресурс]. – Режим доступу до ресурсу: https://medium.com/@roman_41036/where-to-use-osint-in-your-business-4e9a1b45c19f. – Назва з екрана.
10. CYBER INTELLIGENCE by Europol [Електронний ресурс]. – Режим доступу до ресурсу: https://www.europol.europa.eu/activities-

services/services-support/intelligence-analysis/cyber-intelligence. – Назва з екрана.
11. Geolocating Tunisian Jihadists in Raqqa [Електронний ресурс]. – Режим доступу до ресурсу: https://www.bellingcat.com/resources/case-studies/2014/12/19/geolocating-tunisian-jihadists-in-raqqa . – Назва з екрана.
12. Locating the Rockets Used During the August 21st Sarin Attacks in Damascus [Електронний ресурс]. – Режим доступу до ресурсу: https://www.bellingcat.com/resources/case-studies/2014/08/10/locating-the-rockets-used-during-the-august-21st-sarin-attacks-in-damascus . – Назва з екрана.
13. InformNapalm [Електронний ресурс]. – Режим доступу до ресурсу: https://informnapalm.org . – Назва з екрана.
14. Установлены новые данные относительно 3 воинских частей России, участвовавших в агрессии против Украины [Електронний ресурс]. – Режим доступу до ресурсу: https://informnapalm.org/49281-novye-dannye-3-v-ch-okkupantov . – Назва з екрана.
15. Танки для войны на Донбасс поставляли солдаты 76-го РВБ России (ФОТОДОКАЗАТЕЛЬСТВА) [Електронний ресурс]. – Режим доступу до ресурсу: https://informnapalm.org/46493-tanki-dlya-vojny-na-donbass-postavlyali . – Назва з екрана.
16. Центр «Миротворець» [Електронний ресурс]. – Режим доступу до ресурсу: https://myrotvorets.center . – Назва з екрана.
17. Как домашние животные помогают Центру «Миротворец» выявлять российских наемников-убийц [Електронний ресурс]. – Режим доступу до ресурсу: https://myrotvorets.center/882955-kak-domashnie-zhivotnye-pomogayut-centru-mirotvorec-vyyavlyat-rossijskix-naemnikov . – Назва з екрана.
18. Начинающему волонтеру Центра Миротворец. Пример поиска и анализа информации [Електронний ресурс]. – Режим доступу до ресурсу: https://myrotvorets.center/875989-primer-dlya-zhelayushhix-pomoch . – Назва з екрана.
19. Bellingcat [Електронний ресурс]. – Режим доступу до ресурсу: https://www.bellingcat.com . – Назва з екрана.
20. Что известно о крушении Боинга спустя три года [Електронний ресурс]. – Режим доступу до ресурсу: https://www.bellingcat.com/app/uploads/2017/07/mh17-3rd-anniversary-report-ru.pdf . – Назва з екрана.
21. Bellingcat Report - Origin of Artillery Attacks on Ukrainian Military Positions in Eastern Ukraine Between 14 July 2014 and 8 August 2014 [Електронний ресурс]. – Режим доступу до ресурсу: https://www.bellingcat.com/news/uk-and-europe/2015/02/17/origin-of-artillery-attacks. – Назва з екрана.
22. Ghareeb, M., Rouibia, S., Parrein, B., Raad, M., & Thareau, C. (2013). P2PWeb: A Client/Server and P2P hybrid architecture for content delivery over internet. 2013 Third International Conference on Communications and

Information Technology (ICCIT). – Режим доступу до ресурсу: https://doi.org/10.1109/iccitechnology.2013.6579542.
23. The Global Internet Phenomena Report September 2019 [Електронний ресурс]. – Режим доступу до ресурсу: https://www.sandvine.com/hubfs/Sandvine_Redesign_2019/Downloads/Internet%20Phenomena/Internet%20Phenomena%20Report%20Q32019%2020190910.pdf . – Назва з екрана.
24. BitTorrent/DHT [Електронний ресурс]. – Режим доступу до ресурсу: https://ru.wikibooks.org/wiki/BitTorrent/DHT
25. Chad M.S. Steel «Digital Child Pornography: A Practical Guide for Investigators» [Електронний ресурс]. – Режим доступу до ресурсу: https://books.google.com.ua/books?id=FOyLAwAAQBAJ&pg=PA26&dq=cps+gridcops+bittorent&hl=ru&sa=X&ved=2ahUKEwil1srahpDtAhWHtYsKHS6gCEsQuwUwAHoECAAQCQ#v=onepage&q=cps%20gridcops%20bittorent&f=false . – Назва з екрана.
26. Scrambler Project 2.0 Backend [Електронний ресурс]. – Режим доступу до ресурсу: https://scrambler-project-backend2-0.herokuapp.com . – Назва з екрана.
27. Scrambler Project 2.0 [Електронний ресурс]. – Режим доступу до ресурсу: https://scrambler-project2-0.herokuapp.com . – Назва з екрана
28. scrambler-project 2.0 [Електронний ресурс]. – Режим доступу до ресурсу: https://gitlab.com/1revolman1/scrambler-project-2.0 . – Назва з екрана.
29. scrambler-project 2.0-frontend [Електронний ресурс]. – Режим доступу до ресурсу: https://gitlab.com/1revolman1/scrambler-project-2.0-frontend . – Назва з екрана.

ДОДАТОК А

ПЕРЕЛІК ЛІСТИНГІВ ПРОГРАМНОГО КОДУ ОКРЕМИХ ФУНКЦІЙ

Лістінг А.1 – Код функції getTorrents

```
const axios = require("axios");
const getTorrents = (ip, type = "ALL") =>
 new Promise(async (resolve, reject) => {
 let response = await axios.get(
 `https://iknowwhatyoudownload.com/ru/peer/?ip=${ip}`
 );
 if (response.status === 200) {
 let torrent_info = await parseData(response.data),
 information;
 if (type === "ALL") {
 information = {
  hasAdultContent: false,
  hasDangerContent: false,
  content: torrent_info,
 };
 } else {
 information = {
  hasAdultContent: false,
  hasDangerContent: false,
  content: torrent_info[0],
 };
 }
 torrent_info.forEach(({ type }) => {
 if (type === "1")
  information = { ...information, hasAdultContent: true };
 if (type === "2")
  information = { ...information, hasDangerContent: true };
 });
 resolve(information);
 } else {
 reject(response.status);
 }
 }).then((data) => data);
```

Лістінг А.2 – Код функції ipLocate

```
const iplocate = require("node-iplocate");
```

```
const ipLocate = (ip) =>
 new Promise(async (resolve, reject) => {
 let results = await iplocate(ip);
 if (results !== null) {
 resolve(results);
 } else {
 reject(results);
 }
 });
```

Лістінг А.3 – Код функції parseData

```
const cheerio = require("cheerio");
const parseData = async function (html) {
 data = [];
 try {
 const $ = cheerio.load(html);
 $(".table tbody .torrent_files").each((index, element) => {
 let last = $(`.table tbody .date-column`)
 .eq(index + index + 1)
 .text();

 data.push({
 name: $(`.table tbody .torrent_files`)
  .eq(index)
  .text()
  .replace(/\s+/g, " "),
 size: $(`.table tbody .size-column`).eq(index).text(),
 lastData: last,
 type:
  $(`.table tbody .category-column`).eq(index).text() &&
  $(`.table tbody .category-column`).eq(index).text().length > 0
  ? $(`.table tbody .category-column`).eq(index).text()
  : "Неизвестно",
 id: $(`.table tbody .torrent_files`)
  .eq(index)
  .find("a")
  .attr("href")
  .split("=")[1],
 });
 });
 return data;
 } catch (msg) {
 return msg;
```

```
 }
};
```

Лістінг А.4 – Код функції nicknameGetInformationAboutServices

```
export const nicknameGetInformationAboutServices = () => async (dispatch) => {
 dispatch(getInformationAboutServicesLoading());
 let response = await fetch(nickname, {
 method: "GET",
 headers: {
 "Content-Type": "application/json;charset=utf-8",
 },
 credentials: "same-origin",
 });
 if (response.status === 200) {
 let { data } = await response.json();
 dispatch(getInformationAboutServices({ data }));
 }
};
```

Лістінг А.5 – Код функції Home

```
import React, { useEffect } from "react";
import {
 StyledNewWrapper,
 StyledContainer,
 StyledBlockContainer,
 Loader,
} from "./styled";
import IpContainer from "./components/IpContainer";
import LastKnownTorrent from "./components/LastKnownTorrent";
import IpContainerExpanded from "./components/IpContainerExpanded";
import { useDispatch, useSelector } from "react-redux";
import { homeGetHomeData } from "../../redux/actions/Home";
import { home_data, isHomeDataLoading } from "../../redux/selector/Home";

function Home() {
 //REDUX
 const dispatch = useDispatch();
 const homeData = useSelector(home_data);
 const isLoading = useSelector(isHomeDataLoading);
 //REDUX
 useEffect(() => {
 dispatch(homeGetHomeData());
```

```
}, []);
useEffect(() => {
console.log(homeData);
}, [homeData]);

if (isLoading)
return <Loader type="Bars" color="#00BFFF" height={100} width={100} />;
else
return (
<StyledNewWrapper>
<h1>Scrambler Project 2.0</h1>
<StyledBlockContainer>
 {homeData && homeData.ipdata && (
 <StyledContainer className="first">
 {homeData.ipdata.ip && <IpContainer ip={homeData.ipdata.ip} />}
 <IpContainerExpanded ipdata={homeData.ipdata} />
 </StyledContainer>
 )}
 {homeData && homeData.torrent && (
 <StyledContainer className="second">
 <LastKnownTorrent data={homeData.torrent} />
 </StyledContainer>
 )}
</StyledBlockContainer>
</StyledNewWrapper>
);
}
```