ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Система аналізу данних з соціальних мереж за допомогою методів OSINT»

на здобуття освітнього ступеня бакалавра (магістра)
зі спеціальності 126, інформаційні системи та технології
(код, найменування спеціальності)
освітньо-професійної програми інформаційні системи та технології

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Владислав ТИЛЬВАНЧУК
(підпис) *Ім'я, ПРІЗВИЩЕ здобувача*

Виконав: здобувач(ка) вищої освіти гр.ІСД-43
Владислав ТИЛЬВАНЧУК
Ім'я, ПРІЗВИЩЕ
Керівник: к.е.н, доцент, Сергій СОЛОМАХА
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ

Рецензент: _____
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Інформаційних систем та технологій

Ступінь вищої освіти бакалавр

Спеціальність інформаційні системи та технології

Освітньо-професійна програма інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедрою Інф. систем та технологій

Каміла СТОРЧАК Ім'я, ПРІЗВИЩЕ

«____» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Тильванчук Владислав Володимирович
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи:" Система аналізу данних з соціальних мереж за допомогою методів OSINT"
керівник кваліфікаційної роботи Сергій СОЛОМАХА, кандидат економічних наук, доцент
(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій
від «24» березня 2025 р. №56

2. Строк подання кваліфікаційної роботи «30» травня_____ 2025 р.

3. Вихідні дані до кваліфікаційної роботи: Науково технічна література з теми бакалаврської роботи, Аналіз існуючих систем аналізу інформації по відкритим джерелам _____
_____

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Теоретичні основи використання методів OSINT _____
2. Розробка системи аналізу даних з соціальних мереж_____
3. Реалізація та тестування системи_____

5. Перелік ілюстративного матеріалу: *презентація*

6. Дата видачі завдання «24» лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Вступ	14.03.25 – 26.03.25	Завершено
2	Розділ 1. Теоретичні основи використання методів OSINT	27.03.25 – 29.03.25	Завершено
3	Розробка скелету додатку	30.03.25 – 09.04.25	Завершено
4	Розділ 2. Розробка системи аналізу даних з соціальних мереж	10.04.25 – 21.04.25	Завершено
5	Тестування та оптимізація додатку	22.04.25 – 25.04.25	Завершено
6	Розділ 3. Реалізація та тестування системи	26.04.25 – 29.04.25	Завершено
7	Висновки	30.04.25 – 02.05.25	Завершено
8	Оформлення дипломної бакалаврської роботи (чистовий варіант)	04.05.25	Завершено

Здобувач(ка) вищої освіти _____ Владислав ТИЛЬВАНЧУК
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник
кваліфікаційної роботи _____ Сергій СОЛОМАХА
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 53 стор., 14 рис., 24 джерела.

У цій роботі представлено результати дослідження, метою якого було створення системи аналізу даних із соціальних мереж з використанням методів OSINT (Open Source Intelligence). Актуальність роботи обумовлена зростанням значення відкритих даних у соціальних мережах для кібербезпеки, правоохоронної діяльності, маркетингових досліджень та соціологічних аналізів.

На тлі стрімкого розвитку цифрових технологій традиційні методи збору інформації стають недостатньо ефективними. Це зумовлює потребу в автоматизованих рішеннях, які дозволяють швидко збирати, обробляти та аналізувати великі обсяги даних із відкритих джерел. У рамках цієї роботи розроблено програмний засіб для аналізу публічних профілів користувачів месенджера Telegram з використанням офіційного API та бібліотеки Telethon.

Система реалізована на мові програмування Python і забезпечує збір ключової інформації про користувача, включаючи ім’я, прізвище, username, ID, номер телефону, статус активності, фото профілю та спільні чати. Для підвищення ефективності OSINT-аналізу реалізовано генерацію пошукових запитів (дорків) для популярних пошукових систем та соціальних мереж, що дозволяє знаходити додаткові дані про користувача в інших джерелах.

Особливу увагу приділено забезпеченню зручності використання системи. Розроблено графічний інтерфейс, який дозволяє користувачам без технічних навичок легко взаємодіяти з програмою. Також реалізовано автоматичне створення сесійного файлу для підключення до Telegram API, що спрощує процес налаштування.

У процесі роботи були розглянуті сучасні методи та інструменти OSINT, проаналізовано їхні переваги та недоліки. На основі цього розроблено алгоритми збору даних, виявлення зв’язків між користувачами та аналізу їхньої діяльності в мережі. Система тестувалася на реальних даних, що підтвердило її працездатність і ефективність.

Важливим елементом роботи є дотримання етичних та правових аспектів використання OSINT. Система збирає лише публічно доступні дані, не порушуючи конфіденційність користувачів.

Запропоноване рішення може бути використане правоохоронними органами, службами безпеки, маркетологами та журналістами для проведення OSINT-досліджень. У майбутньому систему можна вдосконалити шляхом додавання підтримки інших соціальних мереж, інтеграції зі штучним інтелектом

для аналізу текстів та зображень, а також розширення функціоналу для роботи з великими обсягами даних.

Ключові слова: OSINT, TELEGRAM, СОЦІАЛЬНІ МЕРЕЖІ, АНАЛІЗ ДАНИХ, КІБЕРБЕЗПЕКА, PYTHON, TELETHON, РОЗВІДКА, ВІДКРИТІ ДЖЕРЕЛА, АВТОМАТИЗАЦІЯ, ЗБІР ІНФОРМАЦІЇ, ЦИФРОВА РОЗВІДКА, КІБЕРРОЗВІДКА, АНАЛІТИКА, ІНФОРМАЦІЙНА БЕЗПЕКА.

.

ЗМІСТ

ВСТУП

У сучасному цифровому середовищі соціальні платформи перетворилися на вагоме джерело інформації, що містить великий обсяг відкритих даних про користувачів, події та взаємодії. Аналіз цих даних є дуже важливим для певних галузей, це наприклад, для кібербезпеки, правоохоронних структур, маркетингових планів та соціологічних вивчень.Методи OSINT дозволяють результативно збирати, обробляти та досліджувати наявну інішення.

Мета цієї праці – розробка системи аналізу даних із соцмереж з використанням методів OSINT, що забезпечить ефективний пошук та впорядкування здобутої інформації.

У дослідженні ми розглянемо ключові методи збору відкритих даних, способи опрацювання великих обсягів інформації та практичне застосування OSINT у сфері безпеки, прогнозування ризиків та відстеження подій.

Зростання присутності користувачів в інтернеті зумовлює акумуляцію величезних обсягів відкритих даних, здатних містити інформацію, що становить цінність для різноманітних галузей. Автоматизовані OSINT-системи відкривають можливості не лише для аналізу цих даних, але й для їх використання з метою виявлення можливих загроз, моніторингу подій, проведення розслідувань та прогнозування розвитку ситуацій.

Створення та вдосконалення ефективної системи аналізу даних з соціальних мереж із застосуванням OSINT є важливим чинником розвитку сучасних технологій обробки інформації, що має значну практичну цінність як для державних та приватних установ, так і для незалежних дослідників.

1.ТЕОРЕТИЧНІ ОСНОВИ ВИКОРИСТАННЯ МЕТОДІВ OSINT

1.1Поняття та історія розвитку OSINT

1.1.1Поняття OSINT

OSINT (Open Source Intelligence) – це процес збирання та вивчення даних, отриманих з відкритих джерел. Зустрічається також назва "розвідка по відкритих джерелах", це відповідає сутності OSINT, враховуючи сучасні умови, де основним джерелом інформації є інтернет.

OSINT широко застосовується у сферах національної безпеки, правоохоронних органах та бізнес-розвідці. Крім того, він корисний для аналітиків, які використовують відкриті розвідувальні дані для вирішення секретних, незасекречених або власних розвідувальних питань, що стосуються попередніх розвідувальних дисциплін.

Джерела OSINT поділяються на кілька основних категорій інформаційних потоків:

Медіа, друковані журнали та газети, телебачення і радіо у різних країнах та між ними.

Інтернет, публікації в мережі, блоги, дискусійні форуми, суспільні медіа (зокрема, користувацькі відео з мобільних пристроїв), YouTube, Facebook, Twitter, Instagram та інші. Дане джерело інформації також випереджає чимало інших, завдяки швидкості появи відомостей та зручності користування.

Фахові та наукові публікації, відомості, що надходять з фахових видань, дисертацій, симпозіумів, наукових робіт, тез та конференцій .

Комерційні відомості, фінансові й виробничі експертизи та бази даних.

Сіра література: технічні звіти, патенти. Також до сірої літератури відносять ділові документи, робочу документацію, неопубліковані наукові праці, та інформаційні бюлетені.

OSINT відрізняється від звичайного дослідження тим, що використовує розвідувальні методи для отримання спеціалізованої інформації. Ця інформація спрямована на підтримку прийняття рішень, які здійснює конкретна особа або ж

група осіб.

У Сполучених Штатах Америки визначення OSINT закріплено в Законі 109-163. Цей акт є ключовим нормативним документом, на який посилаються як Директор Національної розвідки США, так і Міністерство оборони США (МО), як розвіддані, "отримані з загальнодоступної інформації, яка збирається, використовується і своєчасно поширюється серед відповідної аудиторії з метою задоволення конкретних розвідувальних потреб".

За визначенням НАТО, OSINT - це розвіддані, "отримані з загальнодоступної інформації, а також іншої несекретної інформації, яка має обмежений публічний доступ".

Потужність OSINT – цифровий відбиток. Інтернет – це цифрове дзеркало нашого існування. Користувачі заводять профілі в соціальних мережах, пишуть коментарі, тиснуть "лайки", роблять перепости, реєструються на різноманітних веб-ресурсах. Ці дані можна акумулювати та використовувати, наприклад, у процесі підбору персоналу.

Як застосовується OSINT у рекрутингу:

Пошук резюме – аналіз загальнодоступних резюме на спеціалізованих платформах, в соціальних мережах (LinkedIn, GitHub, Xing) та професійних онлайн-спільнотах.

Виявлення потенційних кандидатів – моніторинг активності майбутніх співробітників в мережі, їх участі в конференціях, публікацій та дискусій у професійних групах.

Збір контактів – знаходження email-адрес, телефонних номерів та інших контактних відомостей за допомогою спеціалізованих OSINT-інструментів (Hunter.io, Snov.io, Pipl).

Оцінка інформації про кандидата – аналіз профілів в соціальних мережах, професійної діяльності, відкритих баз даних та публікацій з метою оцінки репутації та кваліфікації.

Пошук переліків кандидатів – виявлення вже готових списків потенційних претендентів, які можуть бути розміщені на форумах, у звітах конференцій або

відкритих базах даних.

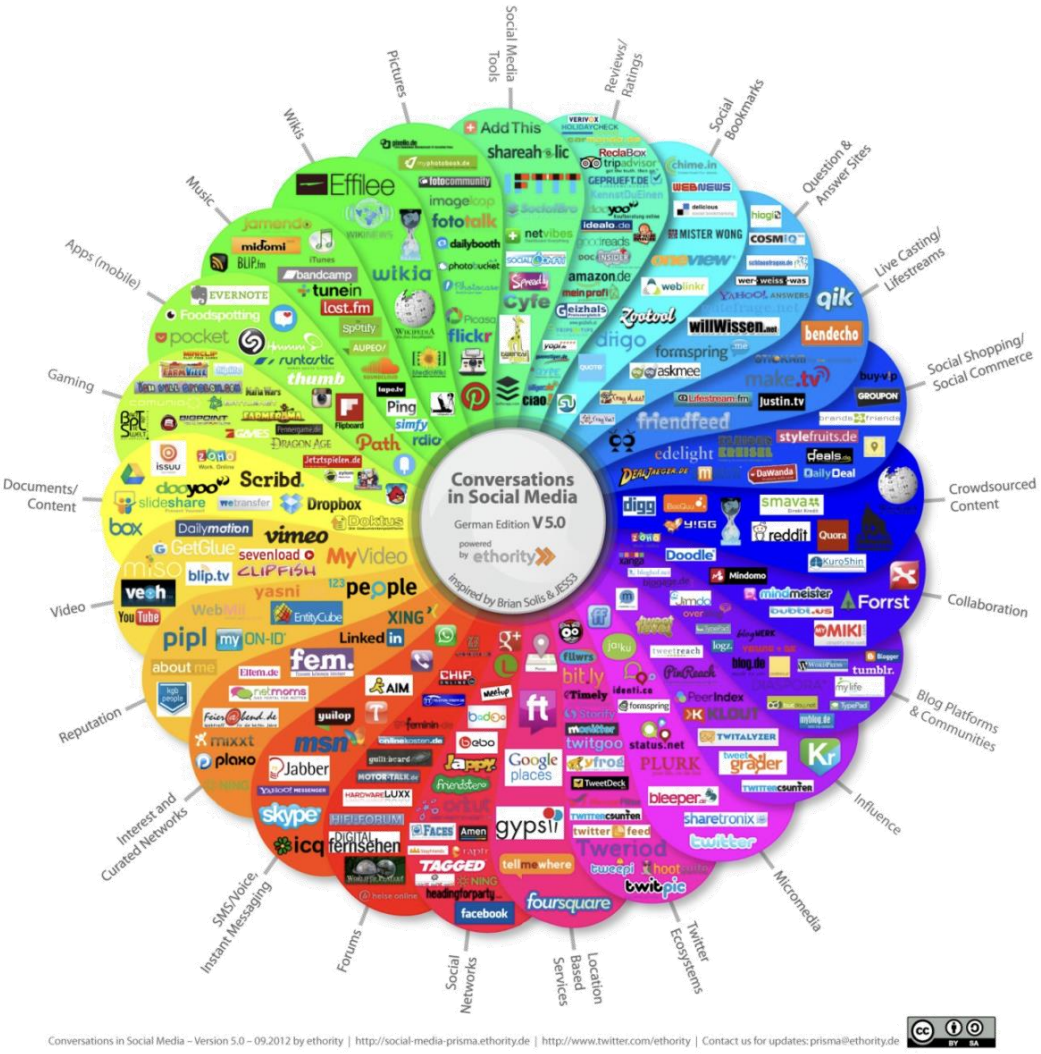


Рисунок 1.1 — Джерела інформації

1.1.2Історія розвитку OSINT

Витоки OSINT сягають давнини, коли урядовці та воєначальники застосовували відкриті джерела для отримання цінних відомостей про супротивника. Проте сучасна еволюція OSINT бере початок у XX столітті, коли поступ засобів масової інформації та технологій значно збільшив можливості збирання інформації з відкритих джерел.

Північ штату Нью-Йорк, 1883 рік. Саме тут з'явилася на світ одна з найважливіших постатей в історії OSINT. Вільям Донован, народжений у родині працьовитих ірландських іммігрантів, що сповідували релігію, демонстрував

чудові успіхи в навчанні, як у школі, так і в університеті.

Донован мріяв стати першим католиком на посаді президента США. Він наблизився до своєї мети у 1905 році, коли вирішив вивчати юриспруденцію в Колумбійському університеті – Франклін Делано Рузвельт був його однокурсником.

Після служби на Першій світовій війні, Донован зробив кар'єру міжнародного юриста, трохи не дотягнувши до посади генерального прокурора США у 1925 році. Протягом воєнного періоду Донован здійснював подорожі світом як правник, зустрічався з впливовими закордонними діячами та, виконуючи роль неофіційного посла, робив звіти для уряду США.

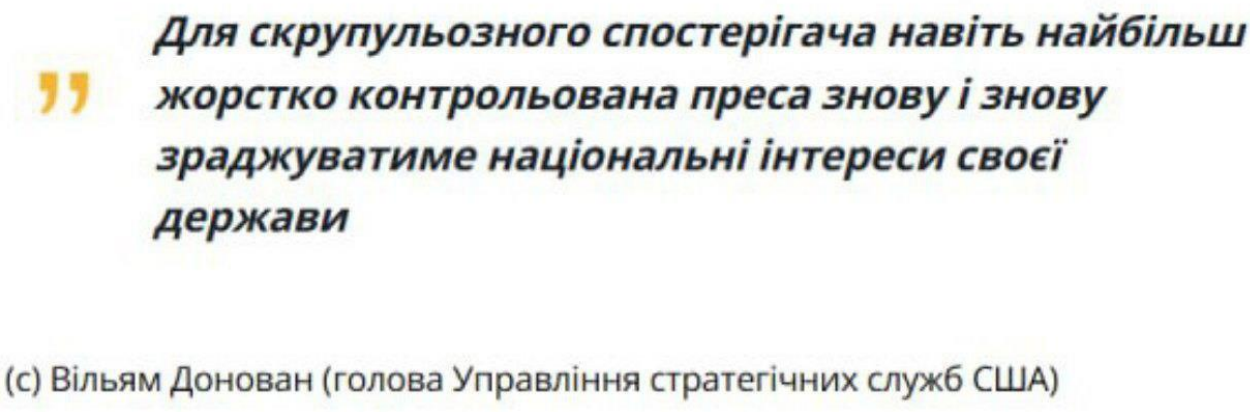


Рисунок 1.2 — Цитата В.Донована

Саме завдяки близьким стосункам Вільяма Донована з президентом Франкліном Делано Рузвельтом у США почав формуватися перший розвідувальний орган. До цього моменту розвідка в американському суспільстві сприймалася як щось сумнівне й мало серйозне. Доновану вдалося переконати Рузвельта надати офіційного статусу його діяльності, яка до того часу носила неформальний характер. У результаті, 11 липня 1941 року президент призначив його на новостворену посаду «Координатора інформації».

Після атаки на Перл-Гарбор потреба у створенні ефективної розвідки стала очевидною, і структура, очолювана Донованом, була реорганізована в Управління стратегічних служб (УСС) — установу, що стала попередницею ЦРУ. Подібно до британського Управління спеціальних операцій, УСС займалося широким

спектром завдань: від вербування агентів і шпигунства до проведення саботажів і таємних операцій.

Цікавим фактом, який може здивувати сучасного читача, є те, що в межах УСС існував підрозділ, який займався саме аналізом відкритих джерел (OSINT). Аналітики цього відділу збирали та опрацьовували величезні обсяги друкованих матеріалів — газет, журналів, вирізок, а також радіопередач, — з метою виявлення інформації, яка могла мати стратегічну цінність для розвідки.

УСС вели спостереження за всіма некрологами в німецьких виданнях, шукаючи відомості про впливових нацистів. Знімки нових лінкорів, авіації та вирв від бомбардувань ретельно систематизувались та вивчались, даючи УСС можливість оцінювати становище супротивника. Дивовижно, наскільки схожа робота УСС на діяльність сучасних OSINT-дослідників, хоча тоді не було комп'ютерів.

З огляду на дії УСС та УСО, підкреслимо, що витоки розвідки на основі відкритих джерел простягаються в минуле століття. Справді, слова Донована лишаються актуальними; серед мільярдів дописів, завантажень, шерів та лайків, наполегливий спостерігач знову і знову зможе виявити приховані інтереси.

Після Другої світової війни дисципліна OSINT перемістилася на периферію державних та військових установ, де трудилися здебільшого бібліотекарі й науковці. З відходом від використання слова "бібліотека" у середині 2000-х, небагато фахівців з розвідки виявляли зацікавленість у цій сфері. Тоді більшість обирала більш привабливий, таємничий світ агентурної та радіоелектронної розвідки.

Протягом багатьох десятиліть розвідка з відкритих джерел (у сфері державних структур) перебувала у глибокому сні, який не порушили ні Холодна війна, ні події 11 вересня.

Аж раптом, у 2009 році, ситуація зазнала змін. Іран був на порозі демократичної “Зеленої революції”, де численні громадяни висловлювали протест проти режиму. Мільйони молодих іранців використовували інтернет для координації своїх дій, поширення вірусного контенту та закликів долучитися до кампанії. Вперше інтернет-простір наповнився інформацією від громадян про ключову політичну подію, значною мірою завдяки смартфонам, вільному доступу до інтернету та соціальним медіа. Протягом першого тижня протестів, 60% всіх посилань, розміщених у блогах на Twitter, стосувалися іранської політики.

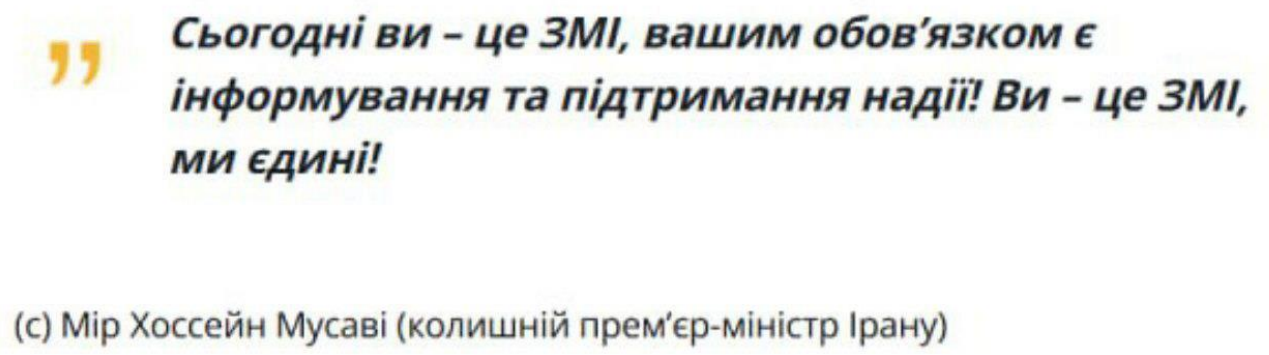


Рисунок 1.3 — Цитата Міра Хоссейна Мусаві

Кількість користувачі інтернету в Ірані зросло у 2008 році із 34 відсодків до 48відсодків у 2009, кількістьмобільних телеефонів збільшилась із 59відсодків до 72відсодків населення. У той час BBC опублікували статтю яка мала назву “Інтернет повертає Іран до життя”, в ній йшлося про процвітання нової форми журналістики.

У світовій практиці з'явилась можливість досліджувати соціальні медіа для збору розвідувальних даних, опрацьовувати їх та створювати статті, передбачення для поглибленого розвідувального аналізу. Незважаючи на те, що протести в Ірані не змогли досягти поставленої мети, а іранський режим зміг оперативно відновити контроль над мережею, зараз ми можемо окинути поглядом минуле та визнати провальну "Зелену революцію" ключовою подією історії розвідки на основі відкритих джерел.

Наразі не дивовижно що саме в Ірані виник OSINT; в Ірані налічується більше користувачів інтернету, ніж у Саудівській Аравії , Йорданії, Ізраїлі, Сирії, Бахрейні,

Ємені та ОАЕ узятих разом. Під час революції Метью Вівер з The Guardian висловив своє здивування щодо реалій розвідки по відкритим джерелам:
“Те, що люди говорять у певний момент часу, через чотири-п’ять годин підтверджується більш традиційними джерелами інформації”.

Не дивно, що громадські журналісти та незалежні дослідники, а не державні органи, стали лідерами в розвитку OSINT . Ця сфера діяльності стрімко розвивається, постійно з'являються нові інструменти та методи розвідки по відкритим джерелам. Світи наукової спільноти та урядової структури лише починають усвідомлювати потенціал цих даних. Люди,які не обтяжені слабким знанням IT, можуть стати кваліфікованими OSINT-спеціалістами, генерувати аналітичні звіти та виявляти колись невідомі зв'язки.

Минуло небагато часу після "Зеленої революції", як арабський світ охопила революція, що живилася соціальними медіа. Поєднання гніву громадян, смартфонів та медіа призвело до стрімких падіннь диктаторських режимів у Північній Африці та на Близькому Сході. Директор OSINT-центру ЦРУ одного разу зазначив, що агентство не змогло передбачити швидке зростання соціальної активності в інтернет-просторі арабського світу. Причиною цієї прорахунку стало те, що розвідувальні служби були зосереджені переважно на елітах і представниках влади, нехтуючи великою кількістю доступної інформації з відкритих джерел.

Останніми роками Сполучені Штати, Велика Британія та низка інших держав значно активізували свою діяльність у сфері OSINT. Зокрема, у США минулого року військові ліквідували об’єкт ІДІЛ з виробництва вибухівки вже за 23 години після того, як один з бойовиків опублікував селфі, де був помітний дах відповідної будівлі. Цей випадок демонструє, як ефективно можуть бути застосовані відкриті джерела в рамках військових операцій.

У приватному секторі США та Великої Британії активно розвивається ринок OSINT-послуг: десятки компаній вже надають аналітичну підтримку як державним структурам, так і комерційним клієнтам. Згідно з публікацією у журналі WIRED за квітень 2016 року, державні органи дедалі частіше звертаються до аналізу відкритих джерел, усвідомлюючи їхню значущість у критично важливих операціях.

Короткий огляд розвитку OSINT дозволяє стверджувати, що сучасна відкрита розвідка є продуктом технологічного зближення кількох ключових процесів. Цей зсув почався приблизно у 2009 році, коли відбулися три важливі зміни. По-перше, значна частина населення вже мала у своєму розпорядженні смартфони з доступом до мереж 3G. По-друге, громадяни активно використовували обмежену кількість додатків для поширення великого обсягу контенту про події у своїх країнах. І по-третє, цей контент був відкритим, вільно доступним і безкоштовним для всього світу.

З того часу сфера OSINT продовжує динамічно розвиватися, змінюючись під впливом нових технологій. Сьогодні зростає інтерес до передачі інформації у режимі реального часу, що створює серйозні виклики для органів безпеки, правоохоронних структур та OSINT-аналітиків — від США до Близького Сходу. У найближчі роки очікується, що штучний інтелект, машинне навчання та віртуальна реальність ще більше вплинуть на способи збору та аналізу відкритих даних.

OSINT впевнено перетворюється на перспективну сферу для інвестицій, особливо в контексті кібербезпеки. Вона також дедалі більше привертає увагу медіа та широкої громадськості. Хоча щодня в інтернеті з’являються мільярди повідомлень, відео, трансляцій, зображень і документів, лише небагато людей усвідомлюють, які безпрецедентні можливості для аналітики й пошуку важливої інформації вони приховують.

1.2Методи та інструменти OSINT

1.2.1Методи OSINT

Методи OSINT дають можливість досліджувати текстову, графічну, аудіо та відеоінформацію, добуту з відкритих джерел, серед яких соціальні мережі, сайти, форуми, бази даних, загальнодоступні реєстри та інші. Ці методи дозволяють виявляти взаємозв'язки між суб'єктами, викривати схеми шахрайства, відстежувати кіберзагрози і навіть передбачати події, спираючись на великі обсяги даних.

Сучасні методи OSINT включають як ручні, так і автоматизовані способи збору даних. Це може бути простий пошук у пошукових системах, дослідження

профілів у соціальних мережах або застосування спеціальних інструментів для виявлення прихованої інформації, такої як IP-адреси, домени чи витоки даних.

Хоча інформація, здобута за допомогою OSINT, є загальнодоступною, її використання мусить відповідати етичним та правовим нормам. Неправильне тлумачення або поширення такої інформації може мати серйозні наслідки. В цьому розділі ми розглянемо основні методи OSINT, які допомагають ефективно збирати, перевіряти та аналізувати відкриті дані.

Пошук в Інтернеті: Використання пошукових систем, таких як Google або Bing, з метою виявлення інформації, що стосується певного предмету, зокрема інформації про можливі уразливості, наявні загрози та способи атак.

Основні пошуковики:

Google – найбільш потужний інструмент із розширеними операторами пошуку.

Bing – корисний для пошуку сторінок, які можуть бути недоступні в Google.

Yandex – має потужний пошук за зображеннями.

DuckDuckGo – підходить для анонімного пошуку без відстеження.

Google dorks “Дорки” або “Гугл Дорки” – це метод, який використовується для виявлення вразливостей на сайтах, використовуючи спеціальні запити в пошукових системах, зокрема Google. Ці запити сконструйовані таким чином, щоб витягати інформацію, яка зазвичай недоступна через звичайний пошук або навіть прихована від більшості звичайних інтернет-користувачів.

Google Dorks, дозволяє зібрати досить великі обсяги інформації:

Пошук людей (імена та прізвища, нікнейми);

Пошук інформації у соцмережах;

Метадані і log-файли;

Електронні адреси;

Файли і документи (pdf, doc, txt, sql, mp3, jpeg, avi та ін.);

Веб-сторінки (URL);

Веб-служби та сервери;

Мережеві пристрої (IoT);

Адміністративні частини електронних ресурсів і додатків (адмін-панелі);

Вразливості в електронних ресурсах та системах;

Інші витоки інформації.

Метод Google Dorking можна ефективно використовувати не лише у Google, а й у таких пошукових системах, як Bing, Yahoo та DuckDuckGo. У цьому процесі застосовуються спеціальні оператори — ключові слова або конструкції, які мають певне значення для пошукової машини. До найпоширеніших належать: inurl, intext, site, feed, language. Після оператора зазвичай ставиться двокрапка, після якої вводиться потрібне слово або фраза для пошуку.

Використання таких операторів дозволяє значно звузити коло пошуку і знаходити конкретні типи інформації — наприклад, тексти на певних сторінках чи документи, доступні за конкретними веб-адресами. Google Dorking, серед іншого, допомагає виявляти сторінки входу, що не були належно приховані, системні повідомлення про помилки, які можуть містити конфіденційну інформацію, а також файли, що випадково опинилися у відкритому доступі. Часто причина такої доступності — недогляд веб-адміністратора, який не встановив належні обмеження доступу.

Однією з корисних функцій Google є можливість перегляду кешованих версій сторінок, навіть якщо вони були видалені або тимчасово недоступні. Для цього використовується оператор cache:, який дозволяє відкрити збережену копію веб-сторінки з архіву Google.

Оператори для пошуку інформації про користувачів:

Оператор «@» дозволяє шукати профілі користувачів у соціальних мережах, таких як Twitter, Facebook, Instagram.

Оператор «related:» відображає список веб-сайтів, схожих на заданий. Це схожість ґрунтується на функціональних зв’язках, а не на змісті сторінок.

Оператор «define:» корисний для наукових досліджень і дозволяє отримувати визначення слів з енциклопедій та онлайн-словників.

«AROUND(n)» — шукає два слова, розташовані на певній відстані одне від одного в тексті.

Тильда («~») — дозволяє шукати схожі слова або синоніми.

Хештеги (#) — використовуються для групування інформації, особливо у соціальних мережах (Instagram, VK, Facebook, Tumblr, TikTok). Google може шукати хештеги одночасно в кількох соцмережах.

Dorking може допомогти знайти інформацію, яку власники сайтів залишили у відкритому доступі випадково, наприклад, структуру веб-сайту або навіть файли з паролями.

Синтаксис операторів Google Dorks має структуру:

operator:пошуковий_запит’Використовуються такі символи:

(+) — додати ще один запит;

(-) — використати стоп-слово;

(“ ”) — шукати в точній відповідності;

(.) — розділювач;

(*) — перелік рандомних значень;

(|) —булеановий запит ‘OR’ (АБО).

OSINT по соціальним мережам – це збір та дослідження даних з медіаплатформ, на кшталт Twitter, LinkedIn та Facebook. OSINT, що стосується соціальних мереж, застосовують для виявлення ймовірних загроз, відстеження настроїв у суспільстві та отримання даних про окремих людей чи організації.

DarkNet OSINT (Open-Source Intelligence) – це процес збирання та вивчення суспільно доступної інформації з DarkNet з метою здобуття важливих розвідувальних даних. У такому контексті, інструменти, які доступні на GitHub, можуть бути корисними для вирішення різноманітних задач DarkNet OSINT.

GitHub – це веб-платформа для зберігання, спільної роботи над розробкою та управління проєктами програмного забезпечення. Вона відома великою кількістю відкритих кодів та репозиторіїв, які дозволяють завантажувати, змінювати та співпрацювати з програмістами з усього світу. Для DarkNet OSINT, інструменти на GitHub можуть бути задіяні для таких задач.

Dark Web – найбільш закрита складова глибокої мережі. Це, безсумнівно,

найтемніша частина глобальної інформаційної системи, куди вхід відкритий кожному, але вона таїть в собі чимало секретів та неприємних несподіванок.

Даркнет використовують злодії, шахраї, ділки наркотиками і зброєю, терористи та багато інших правопорушників, які наживаються на наданій їм анонімності. Саме анонімність та повна відсутність будь-яких правил, що стосуються цього куточка Всесвітньої павутини, є його основною привабливістю.

І хоча в основному ми асоціюємо це явище зі злочинними діями, слід зауважити, що Даркнет також є місцем обходу цензури "звичайного" інтернету в деяких країнах або каналом для безпечного зв'язку та обміну даними між журналістами та їх джерелами. Доступ до темної мережі може отримати майже будь-хто.

Фізичний OSINT передбачає збирання даних через безпосереднє спостереження та фізичне розвідування, наприклад, відвідування конкретних місць, фотофіксацію та використання інформації з загальнодоступних ресурсів. Застосування фізичного OSINT дозволяє оцінювати рівень фізичної безпеки, виявляти ймовірні загрози та збирати відомості про конкретних людей або юридичні особи.

Юридичний OSINT передбачає збирання та ретельний аналіз інформації з повністю легальних джерел, як-от судові рішення, офіційні урядові звіти та публічно доступні документи. Правовий OSINT може бути використаний для одержання розвідувальних даних про конкретних осіб або організації, ідентифікації можливих юридичних ризиків і здійснення контролю за дотриманням вимог законодавства.

Приєднання до спільнот обміну інформацією, як ISAC (Центри аналізу та обміну інформацією), з метою обміну відомостями та розвідданими про ймовірні небезпеки та вразливості.

Застосування автоматизованих інструментів, на зразок пошукових систем та засобів інтелектуального аналізу даних, для накопичення та дослідження великих масивів даних з різних джерел.

1.2.2Інструменти OSINT

OSINT інструменти – це зібрання ресурсів, аплікацій та програмного забезпечення, котрі дають змогу аналітику оперативно добувати потрібні дані з відкритих джерел. Усі OSINT інструменти класифікуються на дві категорії за поширеністю: публічні та кастомні.

Публічні OSINT інструменти - це програми або додатки, доступні для кожного з інтернет-користувачів без виключень. Наприклад, певний сервіс для пошуку людини за її нікнеймом.

Кастомні OSINT інструменти - це спеціалізовані програми, розроблені досвідченими людьми для особистого використання. Кастомні софти найчастіше оперують такими самими відкритими джерелами, але часто мають специфічний функціонал. Приклад: аналітики створили інструмент для пошуку інформації про особу за прізвищем серед резюме на сайтах для пошуку роботи.

За статистикою, переважна частина людей має доволі обмежені здібності до збору інформації з відкритих джерел. Розповсюдження знань про OSINT інструменти підвищує шанси на успішний пошук.

Ось деякі з найпопулярніших інструментів розвідки з відкритим кодом на ринку:

Maltego — це багатогранна розвідувальна платформа з відкритим вихідним кодом, що здатна спростити та пришвидшити слідчі дії. Вона відкриває доступ до 58 джерел інформації та дозволяє завантажувати дані вручну, включаючи бази даних, що містять до 1 мільйона об'єктів, для покращення вашого аналізу. Потужний інструментарій візуалізації дозволяє обирати з різноманітних макетів, як-от блокові, ієрархічні або кругові графіки з позначеннями ваг та нотаток, що сприятиме подальшому удосконаленню.

Використовуючи Maltego, команди довіри та безпеки, правоохоронці та спеціалісти з кібербезпеки можуть отримувати результати своїх розслідувань одним кліком, з легкодоступною та зрозумілою інформацією.

OSINT може бути незамінним у різних сферах, починаючи з правоохоронних органів і закінчуючи фінансовими установами. Саме тому компанія також інвестує значні кошти в надання чудових ресурсів щодо інструментів та методів OSINT.

Вони не лише різноманітні та всеохопні, а й відібрані командою експертів, щоб клієнти отримували максимум від продукту. Компанія також пропонує онлайн-курс Maltego Foundation, який можна придбати.

Spiderfoot – це OSINT-інструмент з відкритим кодом, що надає доволі широкий набір можливостей. Він здатний збирати та аналізувати різноманітні типи даних, включаючи IP-адреси, CIDR-діапазони, домени та субдомени, інформацію про ASN, електронні адреси, телефонні номери, імена та нікнейми, а також BTC-адреси та багато іншого.Доступний як у вигляді інтерфейсу командного рядка, так і з вбудованим веб-сервером та інтуїтивно зрозумілим графічним інтерфейсом користувача (GUI), розміщеним на GitHub, Spiderfoot має у своєму арсеналі понад 200 модулів. Ці модулі використовуються для здійснення всебічних дій та виявлення важливих відомостей про будь-який цільовий об'єкт.Крім того, інструмент може бути корисним для оцінки наявності у публічному доступі даних, які потенційно можуть становити загрозу безпеці організації. Загалом, це потужний інструмент для кіберрозвідки, що дозволяє отримати цінне уявлення про потенційно шкідливі онлайн-сутності.

OSINT Framework — чудовий ресурс для збору розвідувальних відомостей, доступний без обмежень. Тут зібрано все необхідне: від різних джерел інформації до корисних посилань на дієві інструменти. Це значно полегшує роботу, замість того, щоб самостійно вивчати кожну програму та засіб окремо.

Каталог надає можливості для операційних систем, які не обмежуються лише Linux, пропонуючи варіанти для будь-яких потреб. Єдиною складністю може бути розробка оптимальної стратегії пошуку, що допоможе звузити результати, наприклад, під час пошуку інформації про реєстрацію автомобіля або електронні адреси. Проте з такими систематизованими ресурсами це перетворюється на значну перевагу.

OSINT Framework стрімко набирає популярності як один з найефективніших інструментів для збору даних, пошуку інформації та впорядкування результатів.

У сучасному цифровому світі, верифікація особистості через соціальні мережі та онлайн-акаунти, використовуючи їх як дані, стає все більш поширеною

практикою. Компанія SEON стоїть на передовій цього процесу цифрової перевірки ідентичності.

Використовуючи інформацію про електронну пошту та телефонні номери, ваш бізнес може отримати доступ до понад 50 різних соціальних сигналів, що дозволяє скласти повне уявлення про ризики. Ці сигнали не тільки підтверджують актуальність адреси електронної пошти або номера телефону клієнта, але й надають глибшу інформацію про його цифрову історію. Крім того, SEON пропонує бізнесам гнучкість у використанні: запити можна робити вручну, через API або навіть за допомогою розширення для Google Chrome. Це робить систему простою у використанні та максимально доступною.

Lampyre — це платне програмне забезпечення, спеціально створене для OSINT, що надає потужний інструмент для перевірки фактів, кібербезпеки, розслідування злочинів та фінансового аналізу. Цей зручний інструмент просто встановити на комп'ютер одним кліком миші, або ж запустити його онлайн, не докладаючи зусиль. Ввівши вихідну точку даних, наприклад, реєстраційний номер підприємства, повне ім'я особи або номер телефону, Lampyre автоматично опрацює понад сотню джерел інформації, що регулярно оновлюються, для виявлення цінних відомостей. У разі необхідності доступ до даних можливий через програмне забезпечення для комп'ютера або виклики API. Для підприємств, що прагнуть до комплексної платформи моніторингу ризиків та дослідження різноманітних загроз, SaaS-пропозиція Lampyre під назвою Lighthouse дозволяє клієнтам оплачувати використання API.

Shodan – це потужний інструмент пошуку, який надає можливість користувачам швидко знаходити та отримувати дані про технології, що використовуються різними компаніями. Шляхом введення назви організації можна отримати вичерпну інформацію про її IoT пристрої, включаючи розташування, деталі налаштувань та слабкі місця, згруповані за мережею або IP-адресою. Крім того, Shodan дає змогу роботодавцям проводити глибинний аналіз застосованих операційних систем, відкритих портів, типу веб-сервера та мови дизайну з високою точністю, забезпеченою передовими наборами програмних інструментів.

Програмне забезпечення для збору даних з відкритих джерел перетворюється на незамінний інструмент для збирання загальнодоступної інформації з різноманітних джерел, таких як системи для пошуку, соціальні мережі та офіційні документи. Використовуючи передові алгоритми, інструменти OSINT здатні перехресно аналізувати дані, щоб генерувати точну інформацію та виявляти приховані зв'язки. Ця технологія має надзвичайну цінність для фахівців з кібербезпеки, правоохоронців та підприємств, які прагнуть всебічного розуміння особистостей чи організацій.

1.3Етичні та правові аспекти використання OSINT

1.3.1Етичні аспекти використання OSINT

Поважайте приватне життя: Збір відкритих даних (OSINT) повинен здійснюватися з дотриманням конфіденційності як окремих осіб, так і організацій. Це означає, що слід утримуватися від збору приватної інформації або даних, доступ до яких обмежено. Зокрема, неетичним вважається використання обману — таких методів як фішинг чи соціальна інженерія — для отримання інформації, яка не є відкритою.

При роботі з OSINT важливо уникати будь-яких дій, які можуть призвести до шкоди — як фізичної, так і репутаційної — для людей чи компаній. Сюди входить неприпустимість розповсюдження конфіденційних або чутливих матеріалів, а також утримання від дій, що можуть викликати юридичні наслідки або інші негативні наслідки для сторін, до яких стосується інформація.

Дотримуйтеся законодавства: Збір і використання OSINT-даних має відповідати чинному законодавству. Недопустимо використовувати незаконні методи, такі як злом, несанкціонований доступ або порушення правил платформ і сервісів.

Перевіряйте достовірність інформації: Використання неперевірених даних може призвести до поганих наслідків, таких як поширення неправдивої інформації або звинувачення невинних осіб. Завжди перевіряйте отриману інформацію з декількох незалежних джерел.

Відповідальне розповсюдження: Навіть якщо інформація є загальнодоступною, перед її публікацією слід оцінити потенційні наслідки. Деякі дані можуть використовуватися для маніпуляцій, переслідувань або інших шкідливих дій.

Прозорість та доброчесність: Уникайте спотворення фактів, маніпуляції або використання отриманої інформації у спосіб, який може ввести в оману або створити неправдиве уявлення про події, людей чи організації.

Обмежуйте використання чутливої інформації: Якщо OSINT-аналіз виявив інформацію, яка може порушувати права на конфіденційність або завдати шкоди особам чи компаніям, слід утриматися від її використання або поширення без вагомої підстави.

Дотримання цих етичних принципів допоможе використовувати OSINT відповідально, не порушуючи прав та свобод інших людей.

1.3.2Правовіаспекти використання OSINT

OSINT-підрозділи зобов'язані гарантувати, що їхня робота відповідає усім релевантним законам та постановам. Це включає в себе відповідність законам про захист даних та конфіденційність, законодавству щодо інтелектуальної власності, а також іншим чинним правовим нормам.

Варто пам'ятати, що чимало онлайн-платформ мають власні умови використання, які, зокрема, можуть обмежувати або взагалі забороняти застосування автоматизованих інструментів та скрапінгу інформації. OSINT-спеціалісти повинні переконатися, що їхні дії не суперечать цим правилам, адже це може спричинити юридичні проблеми або блокування доступу до потрібних ресурсів.

Перш ніж збирати інформацію, яка не є відкритою, або використовувати персональні дані, організації повинні заручитися згодою осіб або компаній, до яких належить ця інформація. Ігнорування цього принципу може призвести до порушення законів про конфіденційність та інших правових наслідків.

OSINT-робота повинна проводитися у повній відповідності з законами тієї держави чи регіону, де виконується збір даних. Оскільки законодавчі акти про

збереження та захист персональних відомостей, а також про кібербезпеку можуть бути досить відмінними між собою, потрібно брати до уваги можливі правові ризики у кожній конкретній юрисдикції.

Застосування методів, які можуть трактуватися як несанкціонований доступ до систем, наприклад, хакінг чи обхід механізмів безпеки, є протизаконним і може спричинити кримінальну відповідальність.

Якщо OSINT-операції потребують збору та зберігання персональних даних, необхідно дотримуватися відповідних правил захисту даних, як, наприклад, GDPR (ЄС), CCPA (США) та інші регіональні нормативи.

У випадку збору інформації, що може мати чутливий характер, організації повинні забезпечити її відповідне зберігання та обмежити доступ до неї, щоб не допустити витоків або неправомірного використання.

OSINT може надати безцінні дані про загрози, що виникають, та слабкі місця, що дає змогу організаціям активно виявляти й знешкоджувати можливі ризики для безпеки.

OSINT сприяє глибшому розумінню організаціями ландшафту безпеки, зокрема тактик і методів, які використовуються суб'єктами загроз, що допомагає у формуванні стратегій безпеки та планів реагування.

Інструменти та методи OSINT можуть бути порівняно недорогими у порівнянні з іншими способами отримання розвідувальних даних, наприклад, збір даних з людських джерел (HUMINT) або радіоелектронна розвідка (SIGINT).

OSINT може використовуватися для збору розвідувальної інформації про велику кількість об'єктів, включаючи фізичних осіб, організації та системи, і може надати розуміння багатьох аспектів їхніх операцій і діяльності.

OSINT може бути джерелом цінної інформації в процесі реагування на інциденти, наприклад, для визначення джерела атаки або відстеження дій зловмисника.

Розвідувальні дані з відкритих джерел (OSINT) стрімко набирають вагу для забезпечення національної безпеки, особливо в епоху інформаційної війни.

OSINT здатний генерувати цінну інформацію про потенційні загрози та слабкі місця, що дає змогу організаціям національної безпеки проактивно виявляти та нівелювати відповідні ризики.

Застосування OSINT допомагає організаціям краще орієнтуватися в ландшафті безпеки, зокрема в тактиках та методах, які використовують ворожі суб'єкти, що є ключем до розробки ефективних стратегій захисту та планів реагування.

Методи OSINT успішно застосовуються для збору розвідувальних даних щодо широкого спектру цілей, серед яких іноземні уряди, терористичні угруповання та інші джерела загроз. Ці дані можуть бути використані для підтримки дипломатичних, військових та розвідувальних операцій.

Інструментарій та методології OSINT характеризуються відносно низькою вартістю в порівнянні з іншими методами отримання розвідувальних даних, наприклад, збору інформації за допомогою людей (HUMINT) або перехоплення сигналів (SIGINT).

OSINT може надати важливі розвідувальні дані для правоохоронних органів, допомагаючи виявляти джерела злочинної діяльності або відстежувати переміщення підозрюваних.

Інструменти OSINT (відкритої розвідки) дозволяють отримувати критично важливу інформацію про потенційні загрози та слабкі місця в системах, що дає можливість компаніям проактивно виявляти ризики та оперативно зменшувати їхній вплив.

Методи OSINT можуть бути ефективними для збору даних про діяльність конкурентів – від характеристик їхніх продуктів і сервісів до маркетингових підходів. Отримана інформація сприяє формуванню ефективної стратегії розвитку бізнесу та прийняттю обґрунтованих рішень.

Використання відкритих джерел інформації дає змогу відстежувати згадки про компанію в інтернеті та соціальних мережах. Це дозволяє вчасно реагувати на негатив, що виникає, і знижувати репутаційні ризики.

OSINT також застосовується для перевірки потенційних ділових партнерів, клієнтів або майбутніх працівників. Такий підхід допомагає організаціям приймати зважені рішення щодо співпраці або найму, базуючись на перевірених фактах.

OSINT можна використовувати для моніторингу онлайн-каналів на предмет несанкціонованого використання назви компанії, товарних знаків або інтелектуальної власності, допомагаючи захистити бренд і репутацію компанії.

OSINT здатен оперативно постачати інформацію про актуальні загрози та слабкі місця, що значно прискорює процес виявлення інцидентів безпеки та відповідного реагування організацій.

OSINT дозволяє організаціям глибше зрозуміти актуальну картину загроз, а також техніки та підходи, що використовуються зловмисниками. Ці дані допомагають формувати стратегії реагування та приймати обґрунтовані рішення.

OSINT може бути надзвичайно цінним джерелом інформації в цифрових кримінальних розслідуваннях, включаючи ідентифікацію джерела атаки та відслідковування активності зловмисника.

OSINT можна використовувати для моніторингу онлайн-платформ з метою виявлення негативних коментарів або оцінок про організацію чи її продукти, що дозволяє управляти репутаційними ризиками як під час, так і після інциденту.

OSINT може сприяти покращенню комунікації та співпраці між командами реагування на інциденти, полегшуючи ефективний обмін розвідувальною інформацією та координацію дій.

OSINT здатен оперативно постачати дані про нові загрози та слабкі місця, що виникають. Завдяки цьому, компанії отримують можливість швидше розпізнавати загрози для безпеки та реагувати на них.

За допомогою OSINT організації можуть отримати глибше розуміння контексту безпеки, враховуючи тактику і прийоми, які використовують зловмисники. Ця інформація критично важлива для розробки стратегій аналізу загроз і прийняття обґрунтованих рішень.

OSINT використовується для виявлення потенційних загроз і вразливостей до їх практичного використання. Це дозволяє організаціям вживати превентивні

заходи, щоб уникнути атак.

OSINT може надати важливу інформацію для визначення джерела кібератаки чи суб'єкта загрози. Ці дані можуть бути застосовані для підтримки слідчих дій правоохоронних органів та розробки відповідних планів реагування.

OSINT надає цінний контекст для аналізу загроз, дозволяючи організаціям розібратися в мотивах та методах учасників загроз, а також оцінити можливі наслідки загроз безпеки.

Пошук в інтернеті та інтелектуальний аналіз даних передбачає автоматизоване вилучення інформації з веб-сайтів і онлайн-джерел, таких як соціальні мережі, для збору даних про конкретних осіб чи групи.

Розслідування у соціальних мережах полягає у зборі даних з соціальних мереж таких як Facebook, Twitter, Instagram, LinkedIn та інших, задля аналізу активності особи, яка викликає підозру, її кола спілкування та захоплень в онлайн-просторі.

Глибокий веб охоплює усі розділи інтернету, недоступні для стандартних пошукових систем. Темний веб – це частина глибокого вебу, навмисно прихована і доступна тільки через спеціалізоване програмне забезпечення чи авторизацію. OSINT методи можуть бути застосовані для збору інформації з цих ділянок з метою підтримки розслідувань кіберзлочинів.

Технічна розвідка включає в себе збір відомостей про інфраструктуру підозрюваного, чи то організація, зокрема доменні імена, IP-адреси, сервери, системи та програмне забезпечення, яке вони застосовують.

2.РОЗРОБКА СИСТЕМИ АНАЛІЗУ ДАННИХ З СОЦІАЛЬНИХ МЕРЕЖ

2.1Аналіз потреб та вимог до системи

2.1.1Аналіз потреб до системи

У сучасному інформаційному середовищі, де цифрова ідентичність має не менше значення, ніж фізична присутність, виникає постійна потреба у системах, що дозволяють здійснювати аналітичне опрацювання відкритих даних користувачів, зокрема — у таких платформах, як Telegram. Кількість користувачів месенджерів зростає, і водночас зростає зацікавленість у способах збору, обробки та структурування публічної інформації.

Одна з основних потреб, яка лежить в основі розробки подібних систем, — це необхідність оперативного та повного збору даних про користувача на основі обмеженого первинного ідентифікатора: це може бути username, ID або номер телефону. Для користувача або фахівця, що здійснює OSINT-дослідження (Open-Source Intelligence), важливо швидко отримати максимум доступної інформації, не заходячи вручну на десятки сервісів.

Telegram дозволяє отримувати значну кількість метаінформації про користувача навіть без його підтвердження особистістю. Проте, у звичайному інтерфейсі Telegram ці дані розпорошені. Отже, виникає потреба в інструменті, який агрегує доступну інформацію в одному місці: ім’я, прізвище, username, наявність номера телефону, дата останньої активності, фото профілю, спільні чати тощо. Користувач системи повинен мати змогу швидко приймати рішення, базуючись на зібраній інформації.

Наступною ключовою потребою є автоматизація аналізу цифрового сліду користувача в мережі. Після отримання username або номера телефону, виникає потреба перевірити їх присутність в інших джерелах — таких як форуми злитих даних, загальнодоступні пошукові системи тощо. Ручний пошук цих даних займає багато часу і потребує навичок. Тому система повинна формувати готові пошукові запити, які можна одразу використати.

У деяких сферах, наприклад, під час роботи журналістів-розслідувачів, служб безпеки, або корпоративного аудитингу, критично важливо мати журнал попередніх звернень: хто перевірявся, коли, які результати були отримані. Це дозволяє не тільки зберігати безперервність аналізу, а й використовувати попередні дані як основу для подальших дій. Збереження історії та результатів у файли — ще одна очевидна потреба, що випливає з контексту.

Фото профілю є важливою частиною цифрового профілю людини. Часто саме воно допомагає ідентифікувати особу серед кількох схожих акаунтів. Відповідно, виникає потреба не тільки зберігати ці зображення, а й надавати зручні способи для подальшого пошуку за ними — наприклад, через сервіси зворотного пошуку зображень (Google Images, TinEye, Bing Visual Search тощо). Автоматичне створення таких посилань — це суттєве підвищення зручності використання системи.

Користувачі подібних систем часто мають різні потреби залежно від специфіки своєї діяльності. Для одних важливо тільки отримати базові дані, як-от username та дату останньої активності, для інших — потрібен повноцінний аналіз зображень, перевірка за номером телефону, виявлення спільних груп та пов’язаної активності. Тому система повинна будуватись таким чином, щоб її функції можна було адаптувати — як шляхом налаштувань, так і шляхом підключення чи відключення окремих модулів. Це дозволяє зробити інструмент універсальним як для простого користувача, так і для професіонала.

З часом, особливо в умовах корпоративного чи дослідницького використання, виникає потреба в обробці десятків або навіть сотень запитів на добу. Система повинна забезпечити масштабованість — як технічну (ефективна обробка запитів без зависань і тайм-аутів), так і логічну (можливість зберігати, сортувати, архівувати та вивантажувати результати у зручному вигляді — CSV, JSON тощо). Таким чином, система має потенціал для використання як у невеликих особистих дослідженнях, так і у великих аналітичних кампаніях.

Ще однією важливою потребою є доступ до системи з різних платформ. Сучасні

користувачі не обмежуються лише однією операційною системою — вони працюють з Windows, Linux, macOS, а також мобільними платформами. Тому базова реалізація має бути незалежною від ОС, наприклад, реалізована через Python-скрипт, який може запускатися з терміналу. У майбутньому, якщо система розвивається, — можливе створення веб-інтерфейсу або мобільного додатку. Це забезпечить зручність і гнучкість при роботі в польових умовах, під час подорожей або в різних часових зонах.

У реальному житті аналіз одного джерела рідко буває достатнім. Тому виникає потреба в інтеграції з іншими сервісами й інструментами для OSINT. Наприклад, можливість автоматично передавати отриману інформацію в такі системи, як Maltego, Spiderfoot або інтегрувати з власними базами даних організацій. Інтерфейс API або генерація структури результатів у зручному форматі для подальшого парсингу — усе це підвищує корисність системи в аналітичному середовищі.

Залежно від контексту використання, вкрай важливо, щоб збір даних не залишав цифрових слідів, які можуть ідентифікувати користувача системи. Це особливо актуально у випадках журналістських розслідувань, моніторингу зловмисної активності чи аналітики, пов’язаної з безпекою. Telegram-клієнт має бути налаштований так, щоб не взаємодіяти з користувачем напряму — не писати йому, не додавати в контакти, не переглядати історію повідомлень, а лише витягати публічно доступні метадані. Таким чином, потреба в безпечному й непримітному зборі інформації є критично важливою.

З кожним запитом система накопичує унікальний набір даних, які можуть стати в пригоді в майбутньому. Тому існує потреба у збереженні результатів запитів у стабільному та доступному вигляді — окремими файлами або у вигляді єдиної бази. Важливо, щоб ці файли були структуровані зрозуміло: мали чітку розмітку (наприклад, ім’я, час запиту, ID користувача, список знайдених посилань), дозволяли швидкий пошук і, за потреби, могли бути повторно завантажені в систему для подальшої обробки. Це дозволяє працювати з історією без дублювання зусиль і зберігати контекст запитів.

Багато користувачів не мають технічного бекграунду або глибоких знань у програмуванні. Відповідно, виникає потреба у максимально простому запуску системи, з мінімальною кількістю налаштувань. З точки зору інтерфейсу це може бути запуск через один скрипт із коментарями або навіть автоматичне створення сесії, як у поточній реалізації. Це знижує бар'єр входу, робить інструмент доступним для журналістів, активістів, юристів або просто зацікавлених користувачів, які не працюють у сфері IT.

Система повинна мати потенціал до розвитку. З плином часу з'являються нові джерела інформації, змінюються алгоритми Telegram API, змінюється і сама поведінка користувачів у мережі. Відповідно, потреба — це не тільки вирішення задачі «тут і зараз», а й здатність адаптуватися: додавати нові модулі пошуку, вбудовувати штучний інтелект для аналізу текстів або фото, зчитувати активність у групах, інтегрувати інші джерела (наприклад, Twitter, Discord, WhatsApp Web, якщо такі API з'являться). Гнучкість коду та модульна архітектура — це відповідь на потребу в довгостроковому існуванні рішення.

Зростає потреба в тому, щоб подібні інструменти використовувались не лише в професійній діяльності, а й у навчальних цілях. Наприклад, у курсах з кібербезпеки, цифрової гігієни, OSINT-дослідження. Навчальні заклади, громадські організації або курси для журналістів могли б використовувати таку систему як приклад практичного інструменту для вивчення відкритих джерел. Отже, система має бути достатньо прозорою, добре документованою та безпечною, аби її можна було демонструвати студентам та новачкам.

Нарешті, важливою складовою є потреба у вбудованих етичних запобіжниках або рекомендаціях для користувача. Оскільки система працює з особистими даними, нехай і публічними, вона повинна стимулювати відповідальне використання. Це може включати інформування про юридичні обмеження, посилання на закони щодо обробки персональних даних, або навіть логи з попередженням про межі використання інструменту. Етичне використання — така ж потреба, як і технічна ефективність.

2.1.2Аналіз вимог до системи

Вимоги до системи є визначальними для її правильного функціонування та ефективної роботи. Оскільки перед нами стоїть завдання створення програми для збору даних з Telegram, аналіз вимог охоплює різноманітні аспекти, що стосуються як функціональних, так і нефункціональних характеристик системи. Нижче детально розглянемо ключові вимоги до розробки такої системи, починаючи з обрання мови програмування та інструментів, які використовуються для реалізації коду, а також вимог до архітектури та компонентів програми.

Для реалізації цієї програми я обрав мову програмування Python. Python є однією з найпопулярніших мов для розробки програм, що взаємодіють з API та здійснюють обробку даних.

Вибір Python зумовлений рядом причин:

Python має чистий та зрозумілий синтаксис, що дозволяє швидко розробляти програмне забезпечення.

Для роботи з Telegram API використовується бібліотека Telethon, що є зручною та ефективною для отримання даних з Telegram.

Python підтримує асинхронне програмування через бібліотеку asyncio, що дозволяє ефективно обробляти запити та взаємодіяти з Telegram API без блокування головного потоку програми.

Для взаємодії з Telegram API використовуємо бібліотеку Telethon, яка дозволяє отримувати дані про користувачів, їх фото, статуси та іншу інформацію.

Модуль OS для роботи з файловою системою, що дозволяє зберігати дані в текстових файлах і керувати директоріями.

Модуль Datetime для роботи з датами та часом, що є важливим для збереження історії запитів і запису часу отримання даних.

Для взаємодії з Telegram API необхідно створити сесію, яка має бути автентифікована за допомогою API ID і hash, наданих Telegram. Автоматичне створення сесії: Якщо сесія ще не існує, система повинна створити нову сесію та зберегти її локально для подальших підключень.

Система повинна мати можливість отримати повну інформацію про

користувача за його ID, username або номером телефону, включаючи ім’я, прізвище, username, телефонний номер та статус. Обробка інформації: Дані про користувача мають бути оброблені в форматі, зручному для подальшого використання, зокрема виведення посилань на пошук користувача в Google та інших сервісах. Збір фото профілю: Якщо користувач має фотографії профілю, система повинна завантажити їх на локальний диск, зберігаючи їх в окремій директорії.

Система повинна генерувати автоматично посилання для пошуку користувача за його username та номером телефону в таких пошукових системах, як Google, DuckDuckGo, Facebook, Instagram тощо. Додаткові можливості пошуку: Для username і номеру телефону має бути реалізована можливість пошуку на специфічних ресурсах, таких як Pastebin або Scylla, якщо дані є доступними.

Зібрана інформація про користувача повинна зберігатися в текстових файлах (наприклад, user_info.txt), що дозволяє зберігати дані для подальшого аналізу або використання.Історія запитів: Для кожного запиту до системи має бути збережена історія в окремому файлі (history.txt), щоб можна було відслідковувати всі запити та їх результати.

Система повинна мати можливість отримувати інформацію про спільні чати з іншими користувачами, що дозволяє зрозуміти, з ким ще користувач може бути взаємодіє в Telegram.

Система повинна швидко обробляти запити користувачів та надавати результат у межах кількох секунд. Масштабованість: У разі необхідності система повинна бути здатна обробляти одночасно кілька запитів або працювати з великими обсягами даних, якщо це потрібно для розширених функцій.

Для забезпечення безпеки збереження та передачі даних необхідно застосовувати шифрування при зберіганні чутливих даних, таких як номери телефонів, користувацькі ID та іншу особисту інформацію. Захист від несанкціонованого доступу: Для доступу до системи та даних повинна бути передбачена система автентифікації, щоб захистити систему від потенційних зловмисників.

Система повинна мати механізми для збереження результатів навіть при несподіваному збою, щоб уникнути втрат даних. Відновлення після помилки: У випадку виникнення помилок система повинна забезпечувати можливість відновлення попереднього стану без значних втрат інформації.

Інтерфейс системи повинен бути максимально легким в користуванні, щоб навіть ті, хто не має досвіду в програмуванні, могли ефективно працювати з програмою. Це включає зрозуміле виведення результатів та легкість у налаштуванні параметрів пошуку та збору даних.

Користувачі повинні отримати доступ до журналів системи, що містять інформацію про всі дії програми та можливість перевірки історії запитів, щоб забезпечити прозорість роботи системи.

Система повинна бути зроблена так, щоб нові функції або зміни до наявних могли бути легко інтегровані без порушення основної логіки роботи.

Враховуючи можливість масштабування системи для роботи з більшими обсягами даних або додаванням нових джерел, система повинна бути здатна адаптуватися до нових вимог без серйозних змін у архітектурі.

Загалом, вимоги до системи повинні забезпечити надійність, безпеку, ефективність та зручність у використанні, а також можливість для її подальшого розвитку та адаптації до нових задач і технічних змін.

2.2Розробка алгоритмів збору даних з соціальних мереж

Сучасні соціальні мережі та месенджери містять велику кількість публічно доступної інформації, яка може бути корисною для аналізу, розвідки з відкритих джерел (OSINT), верифікації особистостей або моніторингу онлайн-активності. Для ефективного збору таких даних необхідно розробити чіткі та адаптивні алгоритми, які дозволяють автоматизувати процес взаємодії з платформами без порушення політик конфіденційності та безпеки.

У межах даної дипломної роботи розроблено програмний засіб, орієнтований на аналіз публічних профілів користувачів месенджера Telegram. Основу роботи системи становить використання офіційного Telegram API через асинхронну

бібліотеку Telethon, яка надає прямий доступ до більшості функцій Telegram-клієнта.

Основні етапи збору даних реалізовано наступними алгоритмами:

1. Ідентифікація цільового облікового запису

Алгоритм приймає на вхід один з трьох варіантів: username, ID або номер телефону. Залежно від формату вхідного значення, обирається відповідний метод виклику Telegram API:

Якщо введення починається з «+», система інтерпретує його як номер телефону.

Якщо це цифрове значення — як ID.

У всіх інших випадках — як username.

Цей алгоритм забезпечує гнучкість у роботі з різними джерелами вхідних даних.

2. Отримання загальнодоступної інформації

Після успішної ідентифікації користувача система запитує наступні поля:

Ім’я та прізвище (якщо вказані);

Username;

Telegram ID;

Номер телефону (якщо не приховано);

Статус користувача (online, offline, дата останньої активності).

Цей блок реалізовано через метод get_entity() та аналіз об'єкта типу User, отриманого через API.

3. Завантаження фото профілю

Для користувачів, які мають фотографії, система викликає метод get_profile_photos() і зберігає кожне зображення у локальну директорію. Далі ці зображення можуть бути використані для ручного зворотного пошуку по картинках (reverse image search).

Кожне зображення супроводжується посиланнями на сервіси:

Google Images;

Bing Visual Search;

TinEye.

Цей підхід дозволяє дослідникові перевірити, чи використовувалась така сама

фотографія на інших платформах або акаунтах.

4. Визначення спільних груп

Алгоритм звертається до методу messages.GetCommonChats, який дозволяє виявити кількість чатів, у яких одночасно перебувають бот та цільовий користувач. Ця функція особливо корисна для встановлення потенційних зв’язків або спільнот, до яких належить користувач.

5. Генерація дорків (інтелектуальних пошукових запитів)

Важливою частиною системи є генерація посилань для подальшого OSINT-аналізу. Для цього реалізовано автоматичну побудову URL-запитів до популярних пошукових систем, зокрема для номеру телефону:

Пошук у Google, DuckDuckGo;

Дорки в pastebin.com, leakforums.com, raidforums.com;

Соціальні мережі (Facebook, Instagram, Twitter, LinkedIn);

Спеціалізовані бази типу Scylla.

Для username аналогічні запити, що дозволяють перевірити, чи використовувався такий самий нікнейм на інших платформах

Ці посилання не виконуються автоматично, але їхня наявність у звіті дозволяє дослідникові швидко перейти до ручного аналізу, мінімізуючи час на формулювання запитів.

6. Формування структурованого текстового звіту

Після завершення збору даних система формує файл user_info.txt, який містить усю зібрану інформацію у форматі, зручному для перегляду, копіювання та архівації. Кожен блок (основні дані, дорки, фотографії, спільні чати) чітко відділений і позначений.

Розроблені алгоритми забезпечують швидкий та структурований спосіб взаємодії з Telegram API, що дозволяє отримувати цінну інформацію про користувача без порушення його приватності. Використання дорків і генерація посилань на зворотний пошук значно розширює можливості системи, дозволяючи проводити багаторівневий OSINT-аналіз у напівавтоматичному режимі.

2.3Розробка алгоритмів виявлення зв'язків між користувачами та їх діяльністю в мережі

Розроблено алгоритм для ініціалізації користувача за одним із трьох типів ідентифікаторів: username, числовий ID або номер телефону. Це дозволяє гнучко знаходити цільову особу в Telegram незалежно від формату вхідних даних. Після ініціалізації з профілю користувача витягується його ім’я, прізвище, username, ID, а також статус активності й номер телефону (якщо доступний). Цей крок є базовим для побудови подальших алгоритмів, спрямованих на виявлення мережевих зв’язків.

Для користувачів із наявним username створено набір динамічних URL для автоматизованого ручного пошуку в загальнодоступних джерелах. Згенеровані запити охоплюють такі платформи як Google, DuckDuckGo, Facebook, Instagram, Twitter, LinkedIn і Telegram. Цей алгоритм дозволяє оцінити цифрову активність користувача за межами Telegram і виявити можливі перехресні профілі в інших соціальних мережах.

Аналогічно до пошуку за username, якщо в профілі користувача присутній номер телефону, система генерує URL-запити для перевірки його на наявність у відкритих джерелах. Додатково передбачено запити на спеціалізованих ресурсах (напр., Scylla, Pastebin, Raidforums), що дозволяє виявляти сліди користувача у базах даних злитої інформації. Це може бути корисним у випадках розслідувань або OSINT-аналізу.

Реалізовано алгоритм, який аналізує статус користувача (онлайн/офлайн) і у разі наявності інформації — фіксує точний час останнього входу в мережу або очікуваний час виходу з онлайну. Це дозволяє досліджувати часові закономірності активності, що може бути корисно при спостереженні або моделюванні поведінки.

Система завантажує наявні фотографії профілю користувача, зберігає їх локально та пропонує перелік сервісів для подальшого зворотного пошуку зображень (Google Images, Bing Visual Search, TinEye). Це дозволяє перевірити повторне використання аватарок або виявити інші акаунти, які можуть належати цій же особі або бути фейковими копіями.

Реалізовано алгоритм, який виконує запит до API Telegram і отримує список

спільних чатів між досліджуваним користувачем та обліковим записом, під яким запущено скрипт. Кількість спільних груп дає змогу виявити потенційні зв’язки або спільноти, де відбувається взаємодія. У перспективі це може стати основою для побудови графа зв’язків між користувачами.

Кожен успішний запит до користувача фіксується у файлі історії з позначкою часу, ID користувача та його username. Це забезпечує збереження сліду досліджень, дозволяє відстежувати повторні звернення та може бути використано для статистичного аналізу.

Кожен із цих алгоритмів виконує окрему функцію, але в сукупності вони формують комплексну систему первинного OSINT-аналізу користувача в Telegram. Її можна масштабувати для автоматичного виявлення зв’язків, побудови соціальних графів і навіть виявлення аномальної активності.

У цьому розділі були розроблені та сформульовані алгоритми функціонування першої версії програмного забезпечення, яке реалізує базову архітектуру системи. Описані алгоритми визначають основну логіку роботи ключових модулів програми, забезпечуючи початкову функціональність та взаємодію між її компонентами.

3. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ

3.1 Реалізація функціонального скелету системи

3.1.1 Загальний опис функціонального скелету системи

У межах цього проєкту було реалізовано систему збору відкритої інформації про користувачів Telegram з використанням офіційного API та асинхронної Python-бібліотеки Telethon. Система призначена для обробки одного користувача за його ідентифікатором (ID, username або номер телефону), автоматичного збору доступних персональних даних, формування зручного текстового звіту, завантаження фотографій профілю та створення пошукових посилань для ручного OSINT-аналізу.

Додатково, у системі реалізовано збереження історії всіх запитів у окремий лог-файл. Таким чином, система виконує роль автоматизованого клієнта для OSINT-розвідки в Telegram, що може бути використано як окремо, так і інтегровано до більших аналітичних рішень.

Основні можливості системи:

Підключення до Telegram API з використанням сесійного механізму;

Підтримка різних форматів введення ідентифікатора користувача;

Збір персональної інформації (ім’я, прізвище, username, ID, телефон)

Генерація дорок (пошукових запитів) для username і телефону;

Визначення статусу користувача: онлайн/останній візит;

Завантаження фотографій профілю користувача;

Генерація посилань на сервіси пошуку зображень (Google, TinEye, Bing);

Отримання інформації про спільні чати з цим користувачем;

Збереження повного звіту у файл user_info.txt;

Запис кожного звернення до окремого файлу history.txt з міткою часу.

Усі дії виконуються в межах асинхронного середовища, що дозволяє ефективно масштабувати систему або використовувати її для обробки кількох запитів у майбутньому.

3.1.2 Підключення до Telegram API та авторизація

Функціонування системи базується на використанні офіційного Telegram API, що надає розробникам повний набір інструментів для створення клієнтських застосунків, ботів або програм збору даних. Для доступу до API необхідно зареєструвати власний застосунок на порталі my.telegram.org, де користувач отримує два ключових параметри: api_id та api_hash.

У даній реалізації система використовує Python-бібліотеку Telethon, яка є асинхронним Telegram-клієнтом. Вона дозволяє підключатися до акаунта

Telegram як повноцінний користувач (а не бот) і виконувати всі дії, доступні звичайному клієнту: перегляд контактів, обробка повідомлень, перегляд профілів, завантаження медіа тощо.

```
1   import os
2   from datetime import datetime
3   from telethon import TelegramClient, functions
4
5   # Данные для подключения
6   api_id = '
7   api_hash =
8   session_name =  my_session
9
10  client = TelegramClient(session_name, api_id, api_hash)
11
```

Рисунок 3.1 - Реалізація підключення

Змінна session_name визначає ім’я локального файлу .session, у якому зберігаються дані сесії користувача. Це дозволяє уникати повторної авторизації при наступних запусках програми. У разі першого запуску система запитає код авторизації, надісланий Telegram на зареєстрований номер телефону.

Оскільки Telethon працює на базі асинхронного підходу, усі операції з API (наприклад, отримання профілю, завантаження фото, пошук груп) повинні виконуватися всередині асинхронних функцій. Тому основна логіка збору даних реалізована у функції get_user_data(), яка викликається з головної функції main() за допомогою await.

Таким чином, блок авторизації забезпечує безпечне з'єднання з Telegram API,

повторне використання сесії, а також створює фундамент для роботи всієї системи.

3.1.3 Отримання та обробка ідентифікатора користувача

На цьому етапі система приймає вхідні дані, які ідентифікують конкретного користувача Telegram. Ідентифікатор може мати одну з трьох форм:

1. Нікнейм (username) — у форматі рядка без символу @,
2. Номер телефону — починається з плюса (+),
3. Цифровий ID користувача Telegram — наприклад: 123456789

Ця універсальність дозволяє системі бути гнучкою та працювати з будь-яким відомим параметром користувача, що спрощує використання.

Перед обробкою система визначає тип вхідного значення за допомогою умовної логіки. Якщо значення починається з плюса, це означає, що передано номер телефону, і обробка відбувається відповідно. Якщо передано лише цифри — система спробує перетворити рядок на ціле число та обробити як Telegram ID. В іншому випадку обробка відбувається як username.

Особливості реалізації:

Спроба перетворення значення на ціле число (int(identifier)) дозволяє точно визначити, чи має користувач ID.

Якщо перетворення не вдалося (наприклад, передано текстовий username), система обробляє значення як текстовий псевдонім.

Передбачено обробку помилок через конструкцію try-except, щоб уникнути аварійного завершення роботи програми в разі неправильного формату даних.

Цей етап є дуже важливим, тому що від правдивого визначення типу ідентифікатора залежить успішність подальших операцій. Якщо Telegram API не зможе розпізнати користувача, всі наступні етапи не будуть виконані.

3.1.4. Збір базової інформації про користувача

Після того, як система успішно отримала ідентифікатор користувача (будь то username, телефон чи ID), наступним кроком є збір основної інформації про цього користувача. Інформація, яку ми отримуємо, включає основні дані профілю, такі як:

1. Ім'я користувача (first_name)
2. Прізвище користувача (last_name) — якщо воно зазначено в профілі.
3. Username — якщо користувач має публічний нікнейм.
4. ID — унікальний ідентифікатор користувача в системі Telegram.
5. Номер телефону — якщо він вказаний у профілі користувача.
6. Статус користувача — час останнього виходу в мережу або наявність активного статусу.
7. Фото профілю — завантаження та збереження фото профілю користувача, якщо воно доступне.

Після отримання об'єкта користувача через функцію get_entity, система перевіряє наявність поля first_name та last_name, щоб отримати їх значення. Якщо вони відсутні, система виводить відповідне повідомлення про те, що ці дані не були надані користувачем.

Якщо користувач вказав публічний нікнейм (username), він буде доступний через атрибут username об'єкта користувача. Система також генерує посилання на профіль користувача в Telegram за допомогою цього нікнейма, яке має вигляд https://t.me/username. Якщо нікнейм відсутній, виводиться повідомлення про його відсутність.

Ідентифікатор користувача (user.id) завжди присутній в об'єкті користувача і є унікальним для кожного акаунта в Telegram. Це важлива інформація, яка використовується для подальшої роботи з даним користувачем у Telegram API.

Якщо користувач вказав свій номер телефону в профілі, він буде доступний через атрибут phone. Якщо номер не вказано, виводиться відповідне повідомлення. Враховуючи, що номер телефону є чутливою інформацією, система також генерує пошукові посилання для перевірки номеру через пошукові системи.

Статус користувача в Telegram можна визначити через атрибут status. Він може бути різного типу:

Був в мережі (was_online): показує, коли користувач востаннє був онлайн.

Онлайн на даний момент (expires): визначає, коли користувач планує вийти з мережі.

Статус "Неактивний" або інший тип статусу.

Якщо користувач не має статусу, система виводить відповідне повідомлення.

Після збору цієї інформації ми формуємо рядок user_info, який містить всі отримані дані про користувача. Ці дані будуть використані для подальшого збереження в файл або аналізу.

3.1.5. Формування пошукових дорок по username та телефону

Одним із важливих аспектів збору інформації є створення пошукових дорок, які дозволяють користувачеві знайти додаткові відомості про користувача в мережі, використовуючи публічні дані, такі як його username або номер телефону. Ці дорки генеруються для кількох популярних пошукових систем та платформ, де можна знайти додаткові ресурси чи профілі цього користувача.

1. Пошукові дорки по username

Якщо користувач має публічний username (нікнейм), система формує пошукові дорки для наступних платформ та сайтів:

Google: загальний пошук по username.

DuckDuckGo: альтернатива Google, що зберігає анонімність.

Google (дорки): спеціальні пошукові запити для виявлення додаткових ресурсів користувача на таких сайтах, як Pastebin, RaidForums, LeakForums.

Facebook, Instagram, Twitter, LinkedIn: пошук на основних соціальних мережах для виявлення профілю користувача, якщо він є.

Telegram: пряме посилання на Telegram-аккаунт користувача, якщо такий є.

Ці посилання надають можливість перевірити, чи є у користувача інші акаунти чи публічні дані, доступні для пошуку в інтернеті.

2. Пошукові дорки по телефону

Якщо у користувача вказаний номер телефону, система також формує пошукові дорки, щоб допомогти виявити додаткову інформацію або акаунти користувача в мережі:

Google: пошук по номеру телефону для виявлення публічної інформації.

DuckDuckGo: аналогічно, для збереження анонімності.

Google (дорки): спеціальні запити для пошуку номеру телефону на сайтах Pastebin, RaidForums, LeakForums.

Scylla: інструмент для пошуку номерів телефонів через спеціалізовану базу даних.

Facebook, Instagram, Twitter, LinkedIn: пошук по номеру телефону на основних соціальних мережах.

Формування таких пошукових дорок дає можливість знайти додаткові акаунти або іншу публічну інформацію, пов’язану з конкретним номером телефону.

```
    if user.username:
        uname = user.username
        user_info += f"Поиск по username:\n"
        user_info += f"- Google: https://www.google.com/search?q=\"{uname}\"\n"
        user_info += f"- DuckDuckGo: https://duckduckgo.com/?q=\"{uname}\"\n"
        user_info += f"- Google (дорки): https://www.google.com/search?q=\"{uname}\"+site:pastebin.com+OR+site:raidforums.com+OR+site:leakforums.com\n"
        user_info += f"- Facebook: https://www.google.com/search?q=\"{uname}\"+site:facebook.com\n"
        user_info += f"- Instagram: https://www.google.com/search?q=\"{uname}\"+site:instagram.com\n"
        user_info += f"- Twitter: https://www.google.com/search?q=\"{uname}\"+site:twitter.com\n"
        user_info += f"- LinkedIn: https://www.google.com/search?q=\"{uname}\"+site:linkedin.com\n"
        user_info += f"- Telegram: https://t.me/{uname}\n"
    else:
        user_info += "Поиск по username: Пропущено (username отсутствует)\n"

    user_info += f"ID: {user.id}\n"

    if user.phone:
        user_info += f"Номер телефона: {user.phone}\n"
        phone = user.phone
        user_info += f"Поиск по номеру телефона:\n"
        user_info += f"- Google: https://www.google.com/search?q=\"{phone}\"\n"
        user_info += f"- DuckDuckGo: https://duckduckgo.com/?q=\"{phone}\"\n"
        user_info += f"- Google (дорки): https://www.google.com/search?q=\"{phone}\"+site:pastebin.com+OR+site:raidforums.com+OR+site:leakforums.com\n"
        user_info += f"- Scylla (если открыт): https://scylla.sh/search?q={phone}\n"
        user_info += f"- Facebook: https://www.google.com/search?q=\"{phone}\"+site:facebook.com\n"
        user_info += f"- Instagram: https://www.google.com/search?q=\"{phone}\"+site:instagram.com\n"
        user_info += f"- Twitter: https://www.google.com/search?q=\"{phone}\"+site:twitter.com\n"
        user_info += f"- LinkedIn: https://www.google.com/search?q=\"{phone}\"+site:linkedin.com\n"
    else:
        user_info += "Номер телефона: Не указан\n"
        user_info += "Поиск по номеру телефона: Пропущено\n"

    # Статус
```

Рисунок 3.2 - Дорки (код)

1. Пошук по username:

   Якщо у користувача є публічний username, система формує рядок з різними пошуковими запитами на популярні платформи, що дозволяють знайти додаткову інформацію про цього користувача.

   Пошукові дорки по username значно розширюють можливості для аналізу публічних даних користувача через різні канали.

2. Пошук по телефону:

   Якщо номер телефону вказаний, створюються дорки для пошуку інформації про телефон через основні пошукові системи та соціальні мережі.

   Враховується важливість таких пошукових інструментів, як Scylla, що може допомогти знайти додаткові ресурси, пов’язані з номером телефону.

Цей етап додає значну цінність до процесу збору інформації, оскільки дозволяє провести ретельнішу перевірку публічних даних користувача через

пошукові системи та соціальні платформи. Це важливо, якщо необхідно отримати більше відомостей про користувача на основі публічних профілів або контактних даних.

3.1.6. Отримання статусу користувача

Статус користувача в Telegram є важливою інформацією, оскільки він дозволяє визначити, коли користувач востаннє був в мережі або чи є він наразі онлайн. Ця інформація може бути корисною для аналізу активності користувача, а також для отримання загальної картини про його поведінку в Telegram.

В Telethon API статус користувача доступний через атрибут status об'єкта користувача. У різних випадках статус може містити різну інформацію в залежності від того, чи був користувач онлайн або коли він востаннє був активний.

```
# Статус
if user.status:
    if hasattr(user.status, 'was_online'):
        was_online = user.status.was_online.strftime('%d.%m.%Y %H:%M UTC')
        user_info += f"Статус: Был в сети {was_online}\n"
    elif hasattr(user.status, 'expires'):
        expires = user.status.expires.strftime('%d.%m.%Y %H:%M UTC')
        user_info += f"Статус: Онлайн (до {expires})\n"
    else:
        user_info += f"Статус: {type(user.status).__name__}\n"
else:
    user_info += "Статус: Не указан\n"
```

Рисунок 3.3 — Код який збирає коли користувач був онлайн

Знання про те, коли користувач був в мережі, або чи буде він доступний в майбутньому, допомагає розуміти його поведінку в Telegram. Це може бути корисним для тих, хто взаємодіє з користувачем в рамках важливих процесів або комунікацій.

Якщо користувач має обмежений доступ до своєї активності (наприклад, встановлений статус offline або hidden), це також відображається в його статусі.

Отримання статусу користувача є важливою частиною збору інформації, оскільки воно дозволяє визначити рівень активності користувача в Telegram та його доступність для комунікації. Така інформація може допомогти у визначенні того, чи

є сенс намагатися зв'язатися з користувачем або коли саме він буде доступний для взаємодії.

3.1.7. Завантаження фото профілю та створення посилань на зворотний пошук

Завантаження фото профілю користувача — важлива частина цього процесу, оскільки це дозволяє отримати додаткову візуальну інформацію про користувача, що може бути корисним для подальших операцій, таких як зворотний пошук зображень.

Зворотний пошук зображень дозволяє знаходити інші місця в Інтернеті, де з’являються ці ж фотографії або схожі зображення. Це може допомогти у випадках, коли необхідно дізнатися більше про користувача або знайти його профілі в інших соціальних мережах чи платформах.

Для завантаження фото профілю використовуємо метод get_profile_photos(), який дозволяє отримати список фотографій профілю користувача. Якщо такі фотографії є, вони завантажуються на ваш комп'ютер для подальшої обробки або використання.

Алгоритм роботи:

Перевірка наявності фотографій: Спочатку перевіряємо, чи є у користувача фото профілю.

Завантаження фотографій: Якщо фото є, завантажуємо їх в зазначену директорію на диску.

Створення посилань для зворотного пошуку: Для кожного завантаженого фото створюються посилання на популярні сервіси зворотного пошуку зображень, такі як Google, Bing, TinEye тощо.

```
    # Фото профиля
    os.makedirs(save_dir, exist_ok=True)
    photos = await client.get_profile_photos(user)
    photo_count = len(photos)

    if photo_count > 0:
        user_info += f"Фото профиля: Есть ({photo_count})\n"
        for i, photo in enumerate(photos, start=1):
            photo_filename = os.path.join(save_dir, f"{user.id}_{i}.jpg")
            await client.download_media(photo, file=photo_filename)

            # Добавляем ссылки для ручного поиска по изображению
            user_info += f"\n[Фото {i}] Ссылки для поиска по изображению:\n"
            user_info += f"- Google: https://images.google.com\n"
            user_info += f"- Bing: https://www.bing.com/visualsearch\n"
            user_info += f"- TinEye: https://tineye.com\n"
            user_info += f"  Локальный файл: {photo_filename}\n"
    else:
        user_info += "Фото профиля: Нет\n"

    # Общие группы
    try:
        common_chats = await client(functions.messages.GetCommonChatsRequest(
```

Рисунок 3.4 — Пошук по картинкам

Для кожного користувача ми отримуємо список його фотографій через метод get_profile_photos(). Якщо фотографії відсутні, виводиться повідомлення про їх відсутність.

Якщо фото є, ми використовуємо client.download_media(), щоб завантажити кожну фотографію у вказану директорію save_dir. Для кожної фотографії генерується унікальна назва файлу, що містить ID користувача та номер фото (наприклад, user_id_1.jpg).

Після завантаження фото, для кожного зображення створюються посилання на популярні сервіси для зворотного пошуку зображень, такі як Google, Bing, і TinEye. Такі посилання дозволяють користувачам шукати це фото в Інтернеті, що може призвести до знаходження інших профілів або контекстів, де це зображення використовується.

Крім зовнішніх посилань, ми також надаємо шлях до локального файлу, щоб користувач міг використовувати завантажене зображення для подальших маніпуляцій.

3.1.8. Отримання списку спільних груп

Отримання списку спільних груп користувачів є важливою частиною процесу збору інформації про користувача. Спільні групи можуть багато чого розповісти про людину: з ким вона спілкується, які інтереси має, в яких спільнотах бере участь, і навіть які можуть бути пов'язані акаунти.

Спочатку відправляється запит до Telegram API для отримання списку спільних груп, в яких знаходиться як поточний користувач, так і запитуваний користувач. Цей процес займає деякий час, оскільки Telegram API обробляє великий обсяг даних.

Після отримання списку спільних груп обчислюється їхня кількість. Якщо групи є, вони додаються до звіту про користувача.

Інформація про кількість спільних груп додається до файлу звіту user_info.txt. Також у разі необхідності можна додати конкретні назви груп або додаткову інформацію.

```
# Общие группы
try:
    common_chats = await client(functions.messages.GetCommonChatsRequest(
        user_id=await client.get_input_entity(user),
        max_id=0,
        limit=100
    ))
    group_count = len(common_chats.chats)
    user_info += f"\nОбщие группы: {group_count}\n"
except Exception as e:
    user_info += f"\nОбщие группы: Не удалось определить ({e})\n"

# Сохраняем информацию в файл
with open(file_path, "w", encoding="utf-8") as file:
```

Рисунок 3.5 — Отримання списку спільних групп

Отримання списку спільних груп допомагає розширити інформацію про користувача, надаючи додатковий контекст для аналізу його соціальних зв'язків та активності в Telegram. Це дає можливість зрозуміти, з ким користувач взаємодіє в мережі і на яких платформах його діяльність зосереджена.

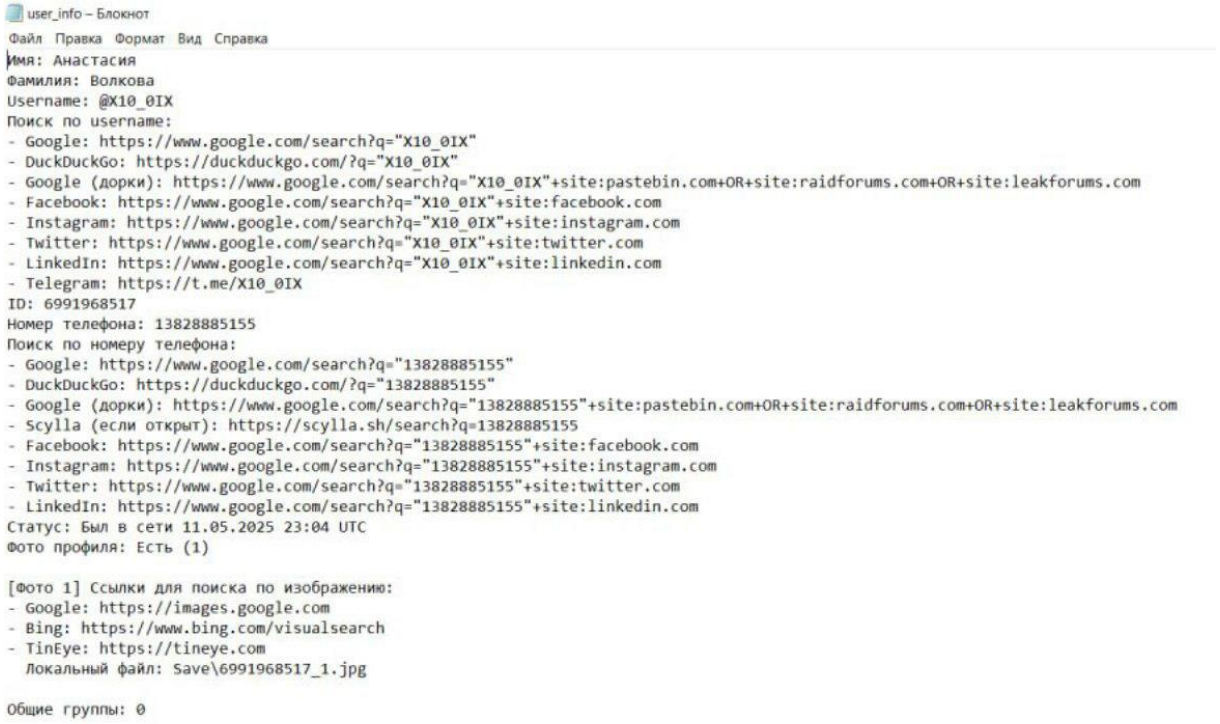
3.1.9. Збереження детального звіту у файл user_info.txt

Після того, як потрібні дані про користувача зібрані, важливо правильно їх зберегти в файл для подальшого аналізу або використання. Один з головних етапів процесу — це формування детального звіту, який включає всю зібрану інформацію, що стосується користувача.

На основі зібраних даних (ім'я, прізвище, username, телефон, статус, фото профілю, спільні групи тощо) створюється текстовий звіт. Цей звіт включає все, що було зібрано за час виконання скрипта. Він може бути розширений або налаштований для певних потреб користувача.

Після того, як звіт сформовано, його потрібно зберегти в текстовий файл, щоб можна було зберігати історію змін, зберігати дані для подальшого використання або просто для архівування інформації.

Файл user_info.txt може бути перезаписаний щоразу при запуску скрипта (якщо потрібно створювати новий звіт) або оновлений за допомогою додавання нових даних (якщо потрібно зберігати інформацію в одному файлі без стирання попередніх даних).


user_info – Блокнот

Файл Правка Формат Вид Справка

```
Имя: Анастасия
Фамилия: Волкова
Username: @X10_0IX
Поиск по username:
- Google: https://www.google.com/search?q="X10_0IX"
- DuckDuckGo: https://duckduckgo.com/?q="X10_0IX"
- Google (дорки): https://www.google.com/search?q="X10_0IX"+site:pastebin.com+OR+site:raidforums.com+OR+site:leakforums.com
- Facebook: https://www.google.com/search?q="X10_0IX"+site:facebook.com
- Instagram: https://www.google.com/search?q="X10_0IX"+site:instagram.com
- Twitter: https://www.google.com/search?q="X10_0IX"+site:twitter.com
- LinkedIn: https://www.google.com/search?q="X10_0IX"+site:linkedin.com
- Telegram: https://t.me/X10_0IX
ID: 6991968517
Номер телефона: 13828885155
Поиск по номеру телефона:
- Google: https://www.google.com/search?q="13828885155"
- DuckDuckGo: https://duckduckgo.com/?q="13828885155"
- Google (дорки): https://www.google.com/search?q="13828885155"+site:pastebin.com+OR+site:raidforums.com+OR+site:leakforums.com
- Scylla (если открыт): https://scylla.sh/search?q=13828885155
- Facebook: https://www.google.com/search?q="13828885155"+site:facebook.com
- Instagram: https://www.google.com/search?q="13828885155"+site:instagram.com
- Twitter: https://www.google.com/search?q="13828885155"+site:twitter.com
- LinkedIn: https://www.google.com/search?q="13828885155"+site:linkedin.com
Статус: Был в сети 11.05.2025 23:04 UTC
Фото профиля: Есть (1)

[Фото 1] Ссылки для поиска по изображению:
- Google: https://images.google.com
- Bing: https://www.bing.com/visualsearch
- TinEye: https://tineye.com
  Локальный файл: Save\6991968517_1.jpg

Общие группы: 0
```

Рисунок 3.6 — Звіт

Архівування даних дозволяє зберігати всю зібрану інформацію в одному місці. Архівування даних необхідне для подальшого використання або для створення звітів. Це також може бути корисно для збереження історії перевірок користувачів.

Оскільки інформація зберігається в текстовому файлі, її можна легко прочитати або імпортувати в інші системи для подальшого аналізу. Такі файли можуть використовуватись як база даних для обробки іншими скриптами або системами.

Файли, в яких зберігається інформація про користувачів, можуть бути використані для аудиту або контролю. Вони дають змогу відстежувати зміни та фіксувати всю діяльність.

Якщо потрібно обробляти велику кількість користувачів, можна створити систему, яка автоматично збирає інформацію про всіх користувачів і зберігає її у різних файлах чи базах даних.

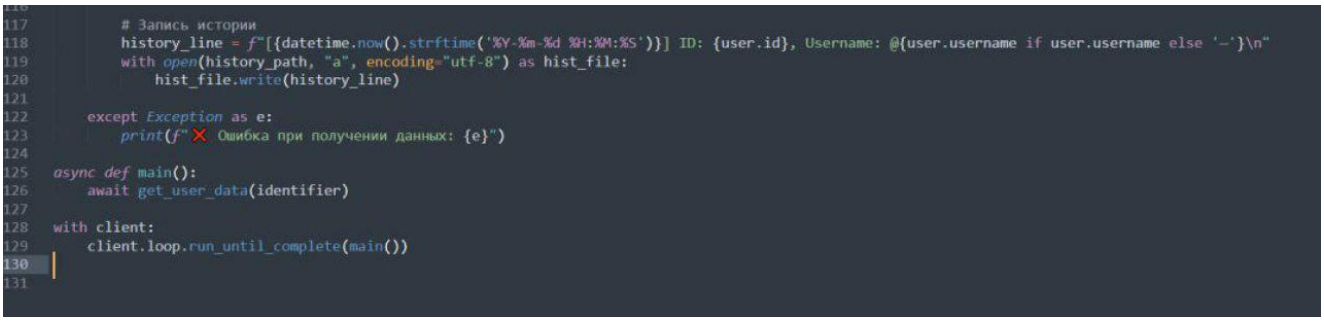
3.1.10 Формування історії звернень у файл history.txt

Історія звернень є важливим компонентом будь-якої системи, яка займається збором та обробкою даних. В нашому випадку, після кожного виконання скрипта, до файлу history.txt додається новий запис, який містить інформацію про час звернення та ID або username користувача, для якого була зібрана інформація. Це дозволяє створити хронологічний запис всіх запитів, що було корисно для аналізу або моніторингу.

Кожного разу, коли скрипт виконується, ми додаємо новий запис до файлу history.txt. Запис містить дату і час виконання запиту, ID або username користувача та інші важливі деталі.

Запис у файл здійснюється у зручному для користувача форматі, який включає час звернення, ID або username користувача. Це допомагає зберігати історію всіх звернень до системи.

Усі записи зберігаються в history.txt, і кожен новий запис додається в кінець файлу. Це дозволяє створити хронологічний список всіх операцій.



```
        # Запись истории
        history_line = f"[{datetime.now().strftime('%Y-%m-%d %H:%M:%S')}] ID: {user.id}, Username: @{user.username if user.username else '—'}\n"
        with open(history_path, "a", encoding="utf-8") as hist_file:
            hist_file.write(history_line)

    except Exception as e:
        print(f"❌ Ошибка при получении данных: {e}")

async def main():
    await get_user_data(identifier)

with client:
    client.loop.run_until_complete(main())
```

Рисунок 3.7 — Запис історії

3.2Тестування системи на реальних даних

3.2.1Тестування системи

З метою перевірки працездатності розробленої системи було проведено тестування на реальних даних із використанням відкритого API Telegram. Для цього було обрано кілька тестових користувачів, інформація про яких частково доступна в загальнодоступних джерелах (акаунти з публічними username та фото профілю).

Перед запуском тестування було:

Зареєстровано Telegram API ID та Hash для авторизації.

Створено сесію клієнта my_session з використанням бібліотеки Telethon.

Забезпечено наявність каталогу Save для збереження результатів.

У тестовому прикладі використовувався ідентифікатор , який представляє реального користувача Telegram з публічним профілем. Також у ході тестування перевірялися варіанти з числовим ID та номером телефону.

Система успішно:

Ідентифікувала користувача.

Витягла доступні персональні дані (ім’я, прізвище, username, ID).

Згенерувала URL-запити для пошуку інформації про користувача за username і номером телефону.

Отримала інформацію про статус онлайн/офлайн користувача.

Завантажила та зберегла профільні фотографії.

Визначила кількість спільних груп між користувачем та авторизованим акаунтом.

Отримана інформація була збережена у текстовому файлі user_info.txt, а також зафіксована у журналі history.txt. Перевірка вмісту файлу підтвердила коректність:
Відображення основних атрибутів користувача.
Формування пошукових посилань.
Завантаження зображень профілю з правильними назвами файлів
Логування дати та часу кожного звернення.

Під час тестування спеціально перевірялися сценарії з відсутністю username або телефону, а також непублічними акаунтами. Усі помилки були успішно оброблені через try-except блоки, без критичних збоїв виконання.

Результати тестування підтверджують працездатність системи, її здатність працювати з реальними користувачами Telegram, адаптивність до різних типів ідентифікаторів та стійкість до виняткових ситуацій. Зібрані дані є коректними та готовими для подальшого аналізу. Система може бути використана як інструмент початкового OSINT-аналізу.

3.2.2Недоліки системи

У ході реалізації першої версії системи збору відкритої інформації (OSINT) з платформи Telegram було виявлено низку обмежень, які впливають на зручність користування та повноту отримуваної інформації. Наведені нижче недоліки не є критичними, проте їх усунення є бажаним для підвищення ефективності системи та її масштабованості в межах ширшого проєкту, присвяченого OSINT-аналізу користувачів соціальних мереж.

Одним із критичних недоліків першої версії системи є відсутність алгоритму для автоматичного створення сесійного файлу Telegram при першому запуску програми. На поточному етапі реалізації система функціонує лише в тому випадку, якщо сесійний файл (.session) вже існує в директорії проєкту, тобто був попередньо згенерований вручну через авторизацію у Telegram Client.
Це суттєво обмежує зручність та універсальність програми:

Складність запуску для нових користувачів. При першому використанні

необхідно попередньо вручну запустити скрипт авторизації, ввести код підтвердження з Telegram і лише після цього запускати основний функціонал системи.

Відсутність захисту від помилки. У разі відсутності сесійного файлу користувач не отримає зрозуміле повідомлення, а система просто не зможе під’єднатися до Telegram API.

Неможливість масштабування. Для обробки кількох облікових записів (наприклад, у версії для аналітичних відділів або команд OSINT) потрібно створювати всі сесії вручну, що значно уповільнює процес.

Ще одним важливим недоліком поточної реалізації є повна відсутність графічного інтерфейсу для взаємодії з користувачем. Система повністю залежить від консольного вводу/виводу, що суттєво ускладнює використання програми для нетехнічних користувачів або в умовах, де зручність та наочність мають вирішальне значення.

Основні проблеми, пов’язані з цим:

Низька доступність для звичайних користувачів. Люди без технічного досвіду не зможуть легко взаємодіяти з системою — їм складно орієнтуватися в консольному режимі, вводити параметри, чекати на відповіді без візуальних підказок.

Відсутність візуального контролю. Користувач не бачить статус підключення до Telegram, прогресу виконання задач, помилок або результатів у зручному вигляді. Уся інформація подається «сухим» текстом у консолі, що знижує юзабіліті.

Складність масштабування під багатокористувацький сценарій. У разі використання програми в команді або організації (наприклад, OSINT-відділ), консольна взаємодія є непридатною — GUI значно полегшив би процес роботи з декількома обліковими записами, сесіями та результатами.

Ускладнення інтеграції з іншими компонентами. Графічний інтерфейс дозволяє реалізувати централізовану панель управління, звіти, кнопки для запуску задач тощо — чого консольна версія не передбачає.

Отримані результати зберігаються у простому .txt-файлі, без форматування

або структурування. Це ускладнює швидке ознайомлення з інформацією, особливо у випадку великого обсягу даних.

3.3Оптимізація та вдосконалення системи

Після створення першої версії системи OSINT-аналізу було виявлено низку недоліків і функціональних обмежень, які стримують її повноцінне використання в реальних умовах. На цьому етапі особливо важливо не лише усунути технічні недоліки, а й розширити можливості системи, адаптувавши її до більш широкого кола задач та підвищивши зручність для кінцевого користувача. Оптимізація та вдосконалення системи є ключовим етапом у процесі її еволюції — від базового інструмента збору даних до повноцінної платформи багатоканального аналізу цифрової присутності користувачів у соціальних мережах.

Цей розділ присвячений вдосконаленню недоліків, що були виявлені на попередніх етапах розробки, та опису можливих шляхів їх усунення для підвищення ефективності та зручності системи. Розглядатимуться аспекти, що стосуються як технічної реалізації (вдосконалення алгоритмів, інтеграція нових технологій), так і підвищення зручності користувацького інтерфейсу та розширення функціональних можливостей.

У результаті вдосконалення система зможе перетворити базовий скелет чи скрипт на повноцінну програму з функціональним інтерфейсом та додатковими можливостями для автоматизації процесу збору даних та їх аналізу.

3.3.1Додавання алгоритму створення сесійного файлу Telegram

Наразі система працює лише з уже існуючим сесійним файлом, що створює обмеження в її гнучкості та масштабованості.

Для вирішення цього питання необхідно додати алгоритм, який автоматично генеруватиме сесійний файл під час першого запуску програми. Це забезпечить користувачів більш простим процесом налаштування та уникне необхідності ручного втручання для підключення до Telegram.

Реалізація цього алгоритму покращить зручність використання програми, а

також автоматизує процес налаштування, що є критично важливим для користувачів, які не мають технічного досвіду.

```python
# Функція для перевірки наявності сесії та автоматичного її створення
def create_session():
    if not os.path.exists(f"{session_name}.session"):
        print("Сесія не знайдена, створюю нову...")
        return TelegramClient(session_name, api_id, api_hash)
    else:
        print("Сесія знайдена, підключаюсь...")
        return TelegramClient(session_name, api_id, api_hash)

client = create_session()
```

Рисунок 3.8 — Функція створення сессії

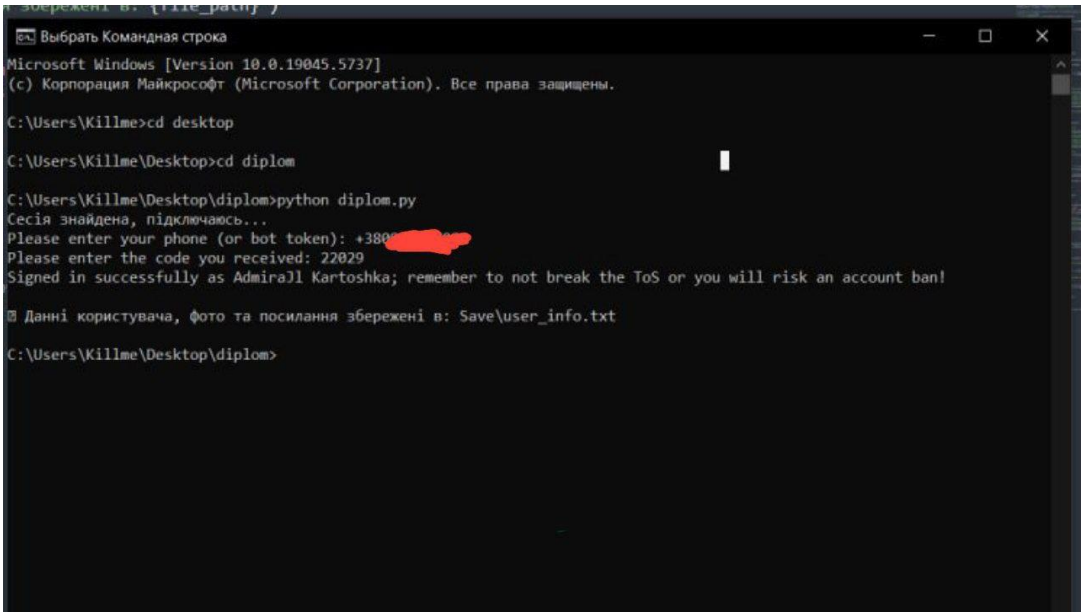


Рисунок 3.9 — Функція створення сессії

Було усунуто одне з ключових обмежень системи — залежність від попередньо створеного сесійного файлу. Нова функціональність дозволяє масштабувати рішення, розгортати його на нових інстанціях без необхідності ручної взаємодії з Telegram API під час першого підключення.

3.3.2 Додавання графічного інтерфейсу (GUI)

У відповідь на виявлену проблему відсутності зручного способу взаємодії з користувачем, у новій версії системи реалізовано графічний інтерфейс**,** який значно покращує юзабіліті програми та розширює її функціональні можливості.

Рисунок 3.10 — Графічний інтерфейс

Основні зміни:

Інтерфейс побудовано на базі Pygame — користувачі можуть вводити номер телефону, код підтвердження та керувати сесією Telegram у зручному візуальному середовищі.

Забезпечено автоматичне створення сесійного файлу: при першому запуску, якщо сесія Telegram ще не існує, інтерфейс запускає процес авторизації, дозволяючи

пройти його без переходу в консоль.

Візуальні підказки та контроль введення: реалізовано введення даних у вигляді вікна, що знижує ймовірність помилок користувача.

Гнучкість і масштабованість: у майбутньому інтерфейс може бути легко розширено для додаткових функцій — вибору дій, перегляду результатів, керування кількома сесіями тощо.

Очікуваний ефект:

Підвищено зручність використання системи для нетехнічних користувачів

Зменшено кількість помилок при авторизації Telegram

Закладено основу для повноцінної графічної оболонки майбутніх версій

У попередній версії результати обробки зберігались у простому .txt-файлі без структури або форматування, що ускладнювало ознайомлення з великими обсягами інформації та обмежувало подальший аналіз.

У новій реалізації цей недолік було усунено:

пошук інформації та подання результатів перенесено у графічний інтерфейс, де дані структуровано та відображаються у зручній формі.

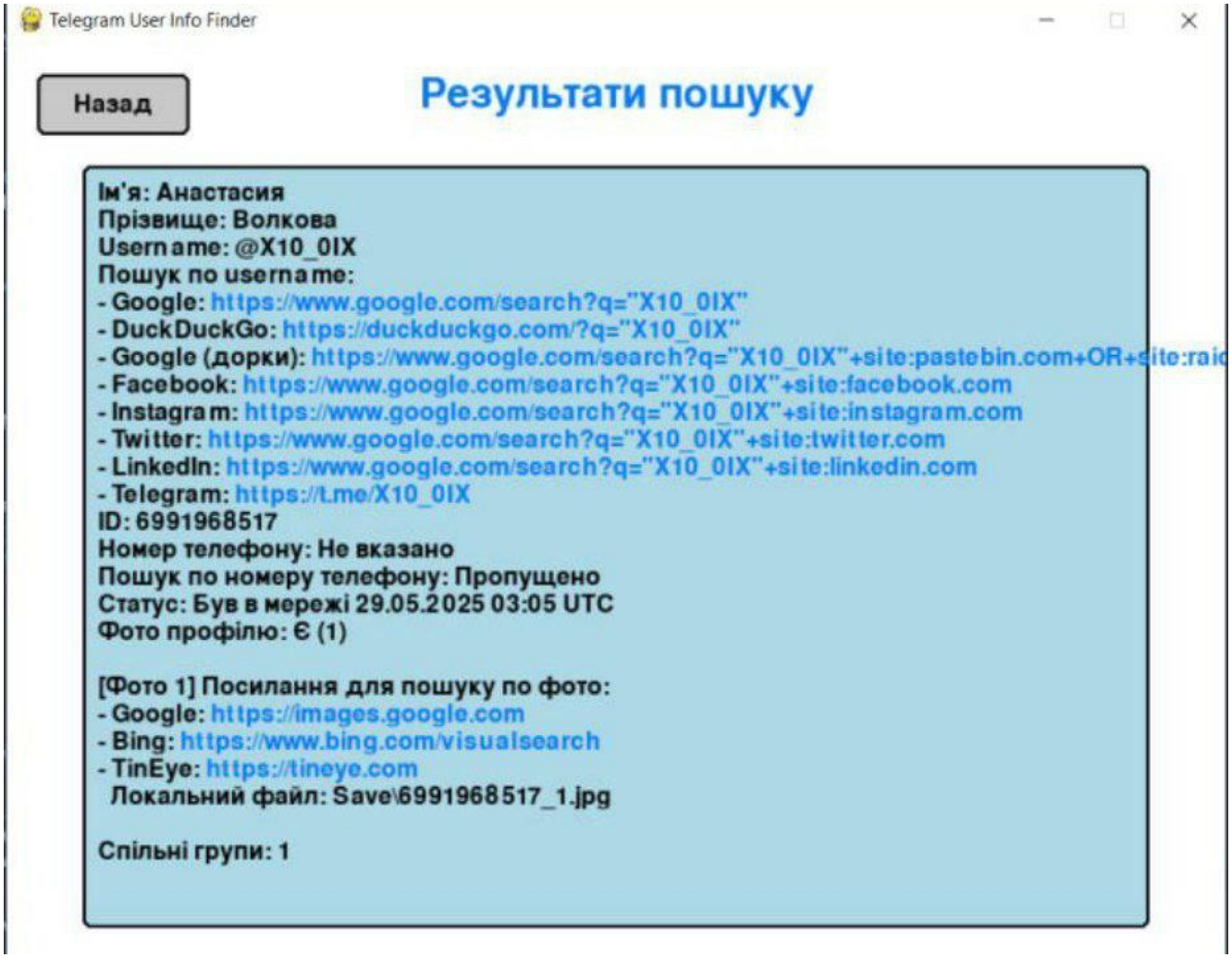


Рисунок 3.11 — Звіт в графічному інтерфейсі

Основні покращення:

Інтегровано систему пошуку всередині інтерфейсу, яка дозволяє швидко знаходити потрібну інформацію без необхідності прокручувати довгі текстові файли.

Всі результати подаються у візуальному вікні з автоматичним форматуванням: кожен блок інформації розділено, виділено ключові елементи (наприклад, імена користувачів, повідомлення, дати).

Посилання стали клікабельними — користувач може натискати на URL без необхідності копіювання вручну, що пришвидшує взаємодію з зовнішніми ресурсами (Telegram, сайти, профілі тощо).

ВИСНОВКИ

Отже, з огляду на стрімке зростання обсягів відкритих даних у соціальних мережах, виникає потреба у створенні ефективних інструментів для їх збору, обробки та аналізу. Методи OSINT відкривають широкі можливості для виявлення загроз, моніторингу суспільно важливих подій, проведення розслідувань та прогнозування ризиків. Розробка системи аналізу даних із соцмереж із використанням OSINT має вагоме значення як для державного сектору, так і для приватних структур, сприяючи підвищенню рівня інформаційної безпеки та обґрунтованості управлінських рішень.

У межах реалізації було створено програмне забезпечення, яке дозволяє автоматизовано здійснювати збір, обробку та первинний аналіз даних із месенджера Telegram з використанням бібліотеки Telethon. Розроблений скрипт забезпечує витягнення ключової інформації про користувача, включно з ідентифікатором, username, номером телефону (за наявності), статусом активності, профільними зображеннями та загальними групами.

Також була реалізована можливість формування зовнішніх запитів до популярних пошукових систем для проведення ручного OSINT-пошуку за отриманими даними (username, телефон, фото тощо), що значно підвищує гнучкість та ефективність інструменту.

Важливою особливістю реалізованої системи є наявність графічного інтерфейсу, який спрощує взаємодію з інструментом для кінцевих користувачів. Завдяки цьому навіть користувачі без глибоких технічних знань можуть ефективно працювати з програмою, отримуючи повноцінні OSINT-звіти у зручному форматі.

У цілому, результати дипломної роботи підтверджують доцільність використання технологій відкритої розвідки (OSINT) у сфері інформаційної безпеки, аналізу соціальних мереж та моніторингу загроз. Запропонований підхід може бути адаптований і масштабований під потреби конкретних структур, а також інтегрований у більш комплексні системи кіберзахисту.

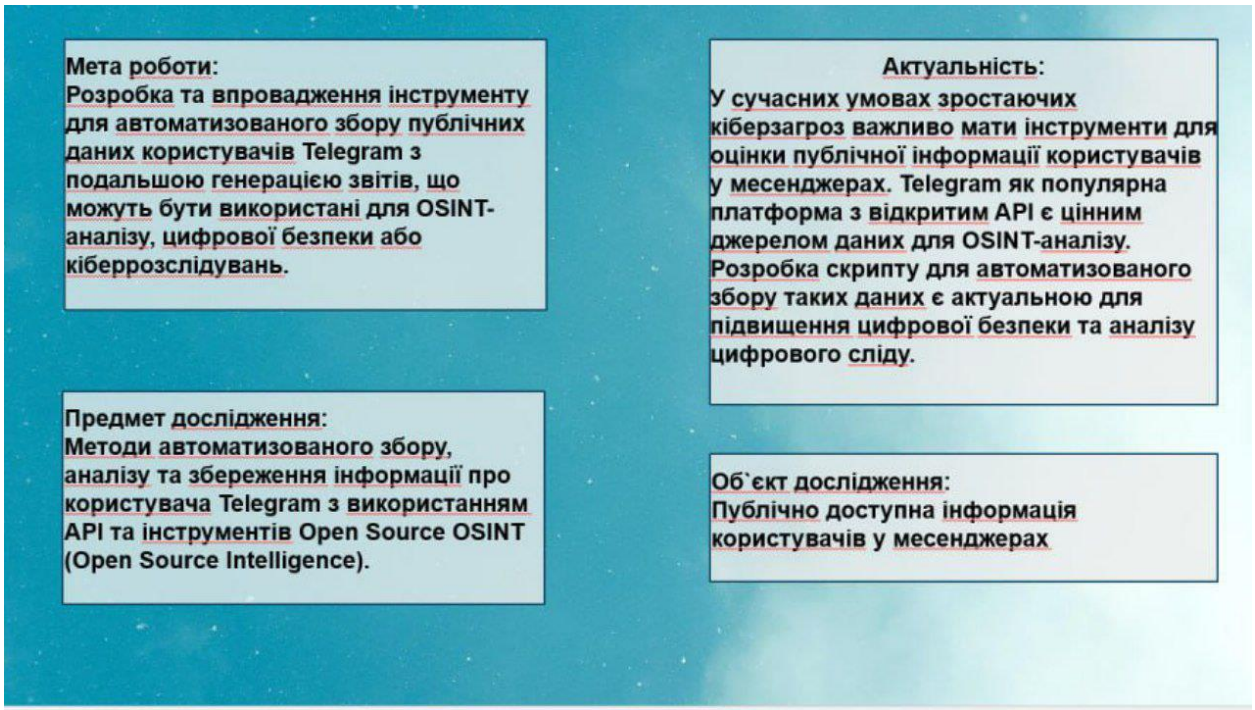
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Електронна енциклопедія Wikipedia. Українськомовна версія // https://uk.wikipedia.org/w/index.php?title=Розвідка_на_основі_відкритих_джерел&stable=0
2. Модель OSINT // https://web.archive.org/web/20090412164950/http://www.computerra.ru/think/kiwi/324966/6
3. Жарков Я.М. Наукові підходи щодо визначення суті розвідки з відкритих джерел / Я.М. Жарков, А.О. Васильєв // Вісник Київського національного університету імені Тараса Шевченка. Військово-спеціальні науки. - 2013. - Вип. 30. - С. 38-41. // http://nbuv.gov.ua/UJRN/VKNU_vsn_2013_30_12
4. Використання технологій OSINT для отримання розвідувальної інформації / О.В. Минько, О. Ю. Іохов, В.Т. Оленченко, К.В. Власов. // http://webcache.googleusercontent.com/search?q=cache:pNCIFjnKvnEJ:irbis-nbuv.gov.ua/cgibin/irbis_nbuv/cgiirbis_64.exe%3FC21COM%3D2%26I21DBN%3DUJRN%26P21DBN%3DUJRN%26IMAGE_FILE_DOWNLOAD%3D1%26Image_file_name%3DPDF/suntz_2016_4_22.pdf+&cd=11&hl=uk&ct=clnk&gl=ua
5. Актуальні питання протидії кіберзлочинності та торгівлі людьми. Харків, 2018 // http://univd.edu.ua/science-issue/issue/3329
6. Распознавание информационных операций / А. Г. Додонов, Д. В. Ландэ, В. В. Цыганок, О. В. Андрейчук, С. В. Каденко, А. Н. Грайворонская. К. : ООО «Инжиниринг», 2017. 282 с
7. NATO Open Source Intelligence Reader (2002). // http://informationretrieval.info/docs/NATO-OSINT.html
8. Розвідка Відкритих Джерел (OPEN SOURCE INTELLIGENCE) Ржевська Н.Ф., Кожушко О.О. // http://ena.lp.edu.ua/bitstream/ntb/19232/1/53-Rzhevska-257-261.pdf
9. Open Source Intelligence (OSINT): Issues for Congress, December 5, 2007. // www.fas.org/sgp/crs/intel/RL34270.pdf
10. Про інформацію: Закон України від 02.10.92 р. № 2657-XII.
11. Про друковані засоби масової інформації (пресу) в Україні: Закон України від 16.11.92 р. № 2782-XII.
12. Про охоронну діяльність: Закон України від 22.03.12 р. № 4616-VI.
13. Про захист персональних даних: Закон України від 01.06.10 р. № 2297-VI.
14. Питання європейської та євроатлантичної інтеграції: Указ Президента України від 20.04.19р. №155/2019.
15. Про Національний координаційний центр кібербезпеки: Указ Президента України від 07.06.16р. №242/2016.
16. Про оперативно-розшукову діяльність: Закон України від 18.02.92 р. № 2135-XII

17. Про рішення Ради національної безпеки і оборони України від 27 січня 2016 року “Про Стратегію кібербезпеки України”: Указ Президента України від 15.03.16 р. № 96/2016.
18. Ланде Д.В. Правові питання конкурентної розвідки // http://ippi.org.ua/sites/default/files/7_16.pdf
19. Сучасні інформаційні технології у сфері безпеки та оборони №1 (40)/2021 // http://sit.nuou.org.ua/article/view/232327/232579
20. Закон України “Про основні засади забезпечення кібербезпеки України”
21. ПЕРЕЛІК категорій кіберінцидентів // https://cert.gov.ua/recommendation/16904
22. Журнал “Юридичний бюлетень”. випуск 11. ч. 1. 2019, організаційно-правові засади використання розвідки з відкритих джерел інформації (osint) в діяльності розвідувальних служб європейських країн / Бурба Василь Васильович // http://www.lawbulletin.oduvs.od.ua/archive/2019/11/part_1/3.pdf
23. Розробка системи управління кіберінцидентами в мережах LTE // https://jrnl.nau.edu.ua/index.php/Infosecurity/article/view/13036/18086
24. Розвідка відкритих джерел інформації (osint) у розвідувальній практиці США / Кожушко Ольга Олегівна // http://jrnl.nau.edu.ua/index.php/IMV/article/viewFile/3264/321

Демонстраційні матеріали

(презентація)



Завдання дослідження

1. Проаналізувати можливості Telegram API для доступу до публічної інформації користувачів.
2. Розробити скрипт для автоматизованого збору даних (ім'я, username, ID, номер, фото, статус).
3. Реалізувати генерацію посилань для OSINT-пошуку за username та номером телефону.
4. Забезпечити збереження отриманих даних у зручному текстовому форматі.
5. Перевірити роботу скрипта на прикладах та оцінити його ефективність у практичних сценаріях.

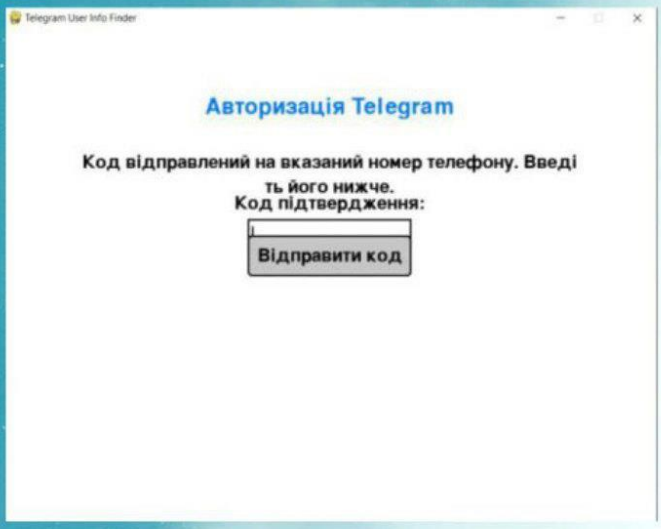
Авторизація

При першому запуску програми користувач проходить авторизацію через Telegram API.

Після успішного входу автоматично створюється сесійний файл, який зберігає дані авторизації.

Завдяки цьому, повторна авторизація не потрібна — додаток одразу підключається до акаунта Telegram при наступних запусках.

Це забезпечує зручність для користувача та безперервну роботу з Telegram API без необхідності вводити код підтвердження щоразу.

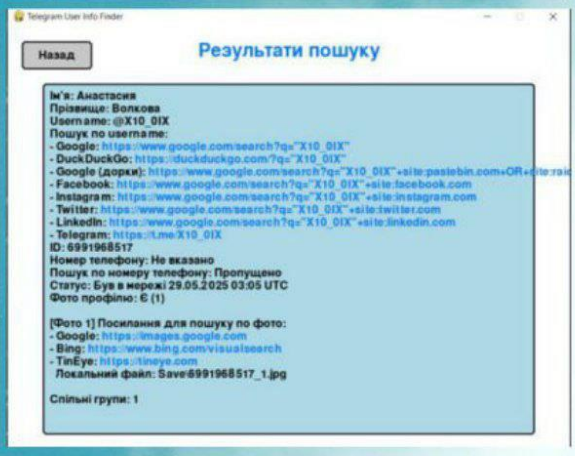


Інтерфейс

Для підвищення зручності використання скрипта розроблено графічний інтерфейс користувача (GUI) на базі бібліотеки Pygame. Хоча Pygame зазвичай використовується для створення ігор, вона також дозволяє створювати прості інтерфейси з полями введення, кнопками та візуальним виводом результатів.

Пошук

Цей інструмент є легким рішенням для поверхневого аналізу. Він ідеально підходить для швидкої перевірки акаунтів, збору базової інформації та генерації дорків. Він автоматизує створення дорків для популярних соціальних мереж (Facebook, Instagram, Twitter тощо) та пошукових систем (Google, DuckDuckGo), що дозволяє швидко знаходити публічні дані.



Аналіз існуючих рішень

На сьогодні існує низка рішень для збору відкритої інформації з Telegram, серед яких:

Телеграм-боти — надають базову інформацію про користувача (ім'я, ID, username, іноді дату створення акаунта). Проте вони обмежені за функціоналом і не завжди відкриті для кастомізації.

OSINT-платформи (Maltego, SpiderFoot) — підтримують інтеграцію з Telegram через спеціальні модулі, але часто вимагають платної підписки або складної конфігурації.

Готові скрипти на GitHub — є багато рішень на основі Telethon або Pyrogram, але більшість з них мають обмежений функціонал (наприклад, тільки збір username або фото) або застарілий код.

Ручний пошук — включає використання Google Dorks та соцмереж для перевірки username або номеру телефону. Це потребує багато часу і не підходить для автоматизації.

Вибір технологій

Python — проста й гнучка мова, швидкий розвиток функціоналу, велика кількість бібліотек.

Було обрано через швидкість розробки, легке тестування та потужну спільноту підтримки.

datetime — стандартна бібліотека для фіксації часу активності та ведення історії.

Було обрана через вбудованість у Python і простоту роботи з датами/часом без зайвих залежностей.

Telethon — асинхронна бібліотека для Telegram API, зручно працювати з акаунтами, фото, групами.

Було обрано за гнучкість, стабільність роботи з Telegram, підтримку всіх офіційних методів API.

Telegram API — офіційне API Telegram, дає доступ до даних акаунтів, чатів, медіа через авторизований клієнт.

Було обрано за достовірність і можливість отримання повної інформації напряму з Telegram без посередників.

Pygame — простий графічний інтерфейс, підходить для демонстрації результатів без командного рядка.

Було обрано як легке рішення для візуального запуску скриптів, без потреби складного GUI-фреймворку.

os — робота з файлами: створення папок, збереження фото, текстових звітів.

Була обрана для зручного управління файлами та кросплатформеності без сторонніх бібліотек.

Функціональні можливості

— Вхід через session-файл — зручно, без повторної авторизації при кожному запуску.

— Підтримка різних типів ідентифікаторів: username, ID або номер телефону.

— Отримання основної інформації про користувача: ім'я, прізвище, username, ID, номер телефону.

— Завантаження профільних фотографій користувача на локальний диск.

— Генерація посилань для ручного пошуку по зображенню (Google Images, TinEye, Bing).

— Отримання статусу активності користувача (останній онлайн, активний зараз тощо).

— Авторизація через Telegram API за допомогою Telethon.

— Формування дорків — автоматичне створення посилань для OSINT-пошуку по username та номеру телефону (Google, DuckDuckGo, соцмережі).

— Збереження звітів у текстовий файл (user_info.txt) з усіма даними.

— Ведення історії запитів із фіксацією дати та часу (history.txt).

— Підрахунок спільних груп з користувачем (якщо можливо).

— Графічний інтерфейс на Pygame — зручне візуальне представлення та запуск додатку без командного рядка.

Перспективи та Недоліки системи

— Авторизація через Telegram API обмежує доступ до випадкових користувачів — для перегляду інформації потрібно, щоб людина була в контактах або не мала приватних обмежень.

— Відсутність бази даних — історія зберігається у текстовому файлі.

— Було свідомо вирішено не використовувати бази даних через можливі правові ризики при обробці персональних даних.

— Залежність від стабільності Telegram API — при блокуваннях або змінах у політиці Telegram функціональність може бути обмежена.

Проте, це більше легкий інструмент поверхневого аналізу,
який підходить для швидкої перевірки акаунтів, збору базової інформації та генерації дорків —
але не замінює повноцінні системи глибокої аналітики чи OSINT-платформи.

Висновки

— Реалізовано ефективний інструмент для збору OSINT-даних через Telegram.

— Система автоматизує рутинну перевірку акаунтів за username, ID або номером телефону.

— Отримані результати зберігаються у вигляді звітів із посиланнями для подальшого аналізу.

— Графічний інтерфейс робить використання зручним навіть для користувачів без технічної підготовки.

— Авторизація через session-файл забезпечує безпечну та постійну роботу без повторного входу.

Розроблений застосунок поєднує в собі зручність, функціональність та актуальність для задач OSINT-аналізу.Він демонструє можливості автоматизованого збору даних із Telegram у поєднанні з простим інтерфейсом та безпечною авторизацією.

Апробація

Теза на тему «Система аналізу данних з соціальних мереж за допомогою методів OSINT»

VI Міжнародна науково-практична конференція для студентів, аспірантів, докторантів, молодих учених, присвяченій Дню науки, «Наука та освіта в дослідженнях молодих учених».

Дякую за увагу!