МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВОЛИНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ЛЕСІ УКРАЇНКИ
Кафедра комп'ютерних наук та кібербезпеки

На правах рукопису

ОМЕЛЬЧУК АНАСТАСІЯ АНДРІЇВНА

МОНІТОРИНГ ЗЛОВМИСНОСТІ ДОМЕННИХ ІМЕН ТА IP-АДРЕС ЗАСОБАМИ ПЛАТФОРМИ TELEGRAM

Спеціальність: 122 «Комп'ютерні науки»
Освітньо-професійна програма: Комп'ютерні науки та інформаційні технології
Кваліфікаційна робота на здобуття освітнього ступеня «бакалавр»

Науковий керівник:
БУЛАТЕЦЬКА ЛЕСЯ ВІТАЛІЇВНА,
кандидат фізико-математичних наук, доцент
кафедри комп'ютерних наук та кібербезпеки

РЕКОМЕНДОВАНО ДО ЗАХИСТУ
Протокол №
засідання кафедри комп'ютерних наук
та кібербезпеки
від _____ 2024 р.
Завідувач кафедри
(_____) Гришанович Т. О.

Луцьк-2024

ЗМІСТ

ВСТУП

Актуальність даної теми. З огляду на зростання кількості кіберзлочинності, які часто включають в себе атаки на мережеві ресурси, збір та аналіз даних про домени та IP-адреси може надати корисну інформацію для аналітики, такої як тенденції в реєстрації доменів, типи серверів, що використовуються. Такими користувачами, яким потрібна така аналітика найімовірніше можуть бути компанії, які бажають відслідковувати відповідність IP-адрес вебсайтів та їх доступність, щоб переконатися, що їхні онлайн-ресурси працюють належним чином. Деякі компанії можуть використовувати моніторинг домених імен та IP-адрес для аналізу трафіку, визначення популярних регіонів чи географічних точок входу на їхні вебсайти.

Є багато сервісів, які надають інструменти для виявлення потенційних кіберзагроз, аналізу безпеки мережі та ідентифікації шкідливих або небажаних активностей в Інтернеті. Збір даних з різних сервісів та їх аналіз дозволить ефективно відстежувати та реагувати на загрози, пов’язані з IP-адресами та доменними іменами.

Враховуючи те, що на сьогодні месенджери та соціальні мережі стали невід’ємною частиною нашого повсякденного життя і не лише засобом спілкування, а й універсальним інструментом для організації робочих процесів, планування подій, обміну файлами та важливою платформою для взаємодії з клієнтами, то збір даних зручно виконувати за допомогою ботів месенджерів[11]. Одним із найпопулярніших месенджерів є Telegram [12], який надає зручну платформу для розробки та використання ботів, які можуть забезпечити різноманітні послуги та функції. Боти дозволяють користувачам отримувати інформацію та виконувати операції, не покидаючи месенджер, що робить їх використання більш зручним та ефективним. Крім того, бот може бути корисним інструментом для навчання користувачів про те, як працюють доменні імена та IP-адреси, і як правильно використовувати інструменти для їх перевірки.

Мета роботи полягає в реалізації алгоритмів отримання, аналіз та відображення інформації про зловмисність доменних імен та IP-адрес засобами платформи Telegram.

Для досягнення поставленої мети були визначені наступні **завдання**:

- розглянути принципи функціонування ботів у месенджерах;
- розглянути принцип організації доменних імен та IP адрес;
- розглянути основні методи виявлення загроз доменних імен та IP адрес;
- здійснити порівняння існуючих аналогів чат-ботів, які використовуються для моніторингу загроз доменних імен та IP адрес;
- підібрати сервіси, які надають інструменти для виявлення потенційних кіберзагроз та визначити вимоги до бота;
- вибрати оптимальні технології та середовища розробки;
- розробити Telegram бот для моніторингу зловмисності доменних імен та IP адрес;
- провести тестування Telegram бота.

Об’єктом дослідження кваліфікаційної роботи є інформація про зловмисність доменних імен та IP-адрес на платформі Telegram.

Предметом дослідження є процес організації, отримання, аналізу та представлення інформації зловмисність IP-адрес та доменних імен.

РОЗДІЛ 1
ПОНЯТТЯ ТА ПРИНЦИПИ ФУНКЦІОНУВАННЯ БОТІВ У МЕСЕНДЖЕРАХ ТА ТЕХНІКА ЇХ СТВОРЕННЯ

1.1. Принципи функціонування ботів у месенджерах

1.1.1. Популярність сучасних месенджерів та їх функції

У сучасному світі месенджери [10] стали невід'ємною складовою комунікації, дозволяючи людям зберігати зв'язок незалежно від часу та місця розташування. За останні роки їх популярність значно зросла, і тепер майже кожен користується хоча б одним з них. Месенджери відіграють важливу роль у нашому повсякденному житті, допомагаючи не лише спілкуватися з друзями та родиною, а й проводити ділові зустрічі.

Використання месенджерів робить спілкування простішим та зручнішим. Завдяки їм можна надіслати повідомлення в реальному часі, здійснити дзвінок або поділитися файлами, що робить комунікацію більш приємною та ефективною. Месенджери також є економічно ефективною альтернативою традиційним методам спілкування, таким як SMS і телефонні дзвінки. Зазвичай для їх використання потрібне лише підключення до Інтернету, що дозволяє економити гроші на телефонних рахунках, особливо у випадку міжнародних спілкувань.

Крім того, месенджери надають великий вибір налаштувань, які дозволяють користувачам персоналізувати свій досвід. Також вони забезпечують високий рівень безпеки, використовуючи наскрізне шифрування для збереження приватності розмов.

Багато месенджерів також пропонують різноманітні функції, такі як відеодзвінки, обмін файлами та ігри, що робить їх універсальним інструментом для різних потреб.

Отже, месенджери є важливим елементом сучасного життя, забезпечуючи зручність, безпеку та багатий функціонал для користувачів у всьому світі. Очікується, що з розвитком технологій вони будуть продовжувати покращуватися, забезпечуючи ще більш швидке, просте та захопливе спілкування.

1.1.2. Боти у складі месенджерів

Боти [24] стали невід'ємною частиною сучасних месенджерів, розширюючи їх функціональні можливості та забезпечуючи користувачам нові сервіси. Завдяки ботам, месенджери стають не лише засобом спілкування, а й універсальним інструментом для вирішення різноманітних завдань та отримання інформації.

Боти в месенджерах можуть виконувати різноманітні функції, починаючи від отримання новин та прогнозу погоди, і закінчуючи замовленням їжі чи квитків на події. Вони дозволяють користувачам отримувати інформацію та виконувати операції, не покидаючи месенджер, що робить їх використання більш зручним та ефективним.

Крім того, боти можуть бути використані для автоматизації різних процесів, таких як планування зустрічей, нагадування про події або навіть проведення опитувань. Вони спрощують взаємодію з різними сервісами та програмами, роблячи комунікацію з ними більш природною та зручною для користувача.

Загалом, боти є важливим елементом сучасних месенджерів, що розширюють їхні можливості та забезпечують більш широкий спектр сервісів та функцій для користувачів. Їхнє використання дозволяє зробити комунікацію більш продуктивною та зручною, відкриваючи нові можливості для взаємодії та отримання інформації.

Існує кілька технологій для створення ботів, які включають у себе різні підходи та інструменти.

1.1.3. Огляд технологій для створення ботів

Багато месенджерів, таких як Telegram, Facebook Messenger, Viber тощо, надають API для створення ботів [7]. Розробники можуть використовувати ці API для розробки ботів, які можуть спілкуватися з користувачами через ці платформи.

Існують різні фреймворки, призначені спеціально для створення ботів. Наприклад, для Python є такі фреймворки [2], як BotKit, Telebot, aiogram тощо. Ці фреймворки забезпечують зручний інтерфейс для роботи з API месенджерів та надають набір інструментів для розробки функціоналу бота.

Деякі боти використовують технології штучного інтелекту та обробки природної мови для розуміння запитів користувачів та надання відповідей. Тут можна використовувати такі інструменти, як Dialogflow, Wit.ai, Microsoft Bot Framework тощо.

Платформи [9] також дозволяють створювати ботів без необхідності в глибокому розумінні програмування. Вони надають інтерфейси для візуального створення ботів та використання готових компонентів для розробки функціоналу бота. Деякі з них включають в себе ManyChat, Chatfuel, Botpress тощо.

Ряд ботів використовують блокчейн технології для забезпечення безпеки та прозорості взаємодії з користувачами. Такі боти можуть бути використані для реалізації різноманітних фінансових, логістичних та інших додатків.

Це лише декілька прикладів технологій, які використовуються для створення ботів.

1.2. Організація доменних імен та IP адрес

1.2.1. Доменні імена

Доменне ім’я [17] це унікальний ідентифікатор, який використовується для локалізації та доступу до вебресурсів в Інтернеті. Це текстовий маркер, який дозволяє людям легко запам’ятати та використовувати вебадреси. Наприклад, у вебадресі “www.google.com” доменне ім’я – “google.com”.

Домени відіграють ключову роль у Інтернеті і мають декілька важливих функцій:

- використовуються для ідентифікації веб-сайтів, електронної пошти та інших ресурсів в Інтернеті;
- допомагають компаніям та організаціям створювати власну онлайн-ідентичність та бренд, який може бути легко впізнаним та запам’ятовуваним користувачами;
- використовуються для навігації в Інтернеті. Вони дозволяють користувачам зайти на певний вебсайт, вказавши в браузері його доменне ім’я;
- використовуються для створення адрес електронної пошти. Наприклад, користувач може мати електронну адресу в форматі nastya@google.com, де “google.com” – це домен;
- країнові домени верхнього рівня (CTLDs) можуть вказувати на географічне розташування вебсайту або організації. Наприклад, .uk вказує на Велику Британію, .de - на Німеччину, .fr - на Францію і т.д.

Перший рівень доменного імені, також відомий як верхній рівень, є частиною імені, розташованою праворуч від останньої крапки. При реєстрації неможливо створити власний перший рівень домену, оскільки потрібно обрати з наявних варіантів, ось декілька з них: .com, .net, .org, .biz, .info, .ua, .me.

Другий рівень доменного імені, також відомий як основний чи материнський, є частиною імені, розташованою ліворуч від останньої крапки. Наприклад, у домені google.com, "google" є другим рівнем домену.

Третій рівень доменного імені, також відомий як субдомен чи піддомен, є частиною імені, розташованою ліворуч від передостанньої крапки. Субдомени використовуються, щоб надати унікальну адресу різним розділам на сайті.

Доменні імена діляться на дві групи: загальні та національні. Загальні домени означають якусь сферу діяльності. Наприклад:

- .com – для комерційний підприємств, скорочено від company;
- .org – для некомерційних організацій, скорочено від organization;

- .edu – для освітніх закладів, скорочено від education.

З 2011 року до загальних доменів додали нову категорію — «нові загальні домени». Ці домени складаються з цілих слів. Ось кілька з них: .bank, .site, .shop, .website.

Національні домени верхнього рівня виділяють для конкретних держав. У більшості випадків за основу національних доменів верхнього рівня брали дволітерні коди країн. Тому всі національні домени першого рівня складаються з двох літер. Таким чином, .UK означає – Велика Британія, .UA – Україна, .ME – Чорногорія тощо.

Домени верхнього рівня деяких держав популярні у всьому світі, тому що нагадують загальновідомі абревіатури. Наприклад, національні домени островів Тувалу .tv реєструють для сайтів телеканалів, а домени Федеративних Штатів Мікронезії .fm – для сайтів радіостанцій.

Зареєструвати національний домен зазвичай складніше, ніж загальний. Деякі держави дозволяють реєструвати їхні національні домени лише своїм громадянам чи офіційним резидентам. У такому разі під час реєстрації доведеться вказати адресу всередині країни, номер паспорта чи водійського посвідчення.

Наприклад, реєструючи домен Франції (.fr), потрібно вказати своє ім’я, рік і місце народження. Якщо цього не зробити, домен не зареєструється. У різних національних доменів бувають різні вимоги.

Деякі домени можна зареєструвати національною мовою держави. Такі домени називають «інтернаціоналізованими» чи «IDN-доменами». Наприклад, президент.укр повністю складається із символів кирилиці.

Для реєстрації доменного імені потрібно звернутися до доменного реєстратора, що дозволить забронювати унікальне доменне ім’я за певною ціною та на певний період часу. Після реєстрації можна використовувати це доменне ім’я для розміщення вебсайту, налаштування електронної пошти та інших цифрових сервісів.

1.2.2. IP адреси

IP-адреса [1] (Internet Protocol Address) – це числовий ідентифікатор, який призначений для ідентифікації та локалізації пристроїв у мережі Інтернет. Кожен пристрій, що підключений до Інтернету, має свою унікальну IP-адресу, яка використовується для передачі даних між пристроями.

IP-адреси (Internet Protocol addresses) є основою мережевого з'єднання в Інтернеті. Вони виконують кілька важливих функцій:

- IP-адреса дозволяє ідентифікувати кожен пристрій в мережі, щоб відправляти та отримувати дані;
- IP-адреси використовуються для визначення маршруту, яким будуть пересилатися пакети даних від відправника до отримувача через мережу, що дозволяє забезпечити доставку даних відправнику до призначення;
- IP-адреси використовуються для ідентифікації джерела та призначення мережевого трафіку, що дозволяє мережевим пристроям правильно маршрутизувати, фільтрувати та керувати трафіком в мережі;
- коли користувач вводить URL-адресу веб-сайту у веб-браузері, ця адреса перетворюється на відповідну IP-адресу за допомогою DNS-серверів, потім ця IP-адреса використовується для з'єднання з вебсайтом і отримання відповіді;
- IP-адреси необхідні для реалізації різноманітних мережевих послуг, таких як електронна пошта, чат, відеодзвінки, файлообмін та інші.

IP-адреси можуть бути в двох форматах: IPv4 і IPv6. IPv4 складається з чотирьох десяткових чисел, розділених крапками, наприклад, "192.0.2.1", тоді як IPv6 використовує довший формат, що містить шістнадцять груп символів у шістнадцятковій системі числення, розділених двокрапками, наприклад, "2001:0db8:85a3:0000:0000:8a2e:0370:7334".

У сучасному Інтернеті, IP-адреси використовуються для багатьох цілей, включаючи розміщення вебсайтів, обмін даними, підключення до мережевих ресурсів та багато іншого. Однак через обмежену кількість доступних IPv4-

адрес, був розроблений новий стандарт IPv6, щоб забезпечити досить IP-адрес для розширення мережі Інтернет.

1.3. Використання доменних імен та IP-адрес для злочинних дій в Інтернеті

Існує кілька видів загроз в Інтернеті [4, 22], пов’язаних з IP-адресами та доменними іменами. Однією з таких загроз є шахрайство з використанням доменних імен. Ця загроза полягає у створенні фішингових вебсайтів, які мають схожі доменні імена з відомими та довіреними сайтами. Користувачі можуть бути обмануті і ввести особисту інформацію, таку як паролі або номери кредитних карт. Також розповсюдженими загрозами є DDoS-атаки (розподілені атаки з відмовою у обслуговуванні). Атаки цього типу спрямовані на переповнення мережевих ресурсів цільового сервера або мережі, надсилаючи велику кількість запитів на його IP-адресу або доменне ім’я. Це може призвести до відмови у обслуговуванні для легітимних користувачів.

Також, якщо IP-адреси або доменні імена вебсайтів вразливі до атак, зловмисники можуть здійснити атаку на сайт, використовуючи такі вразливості для отримання несанкціонованого доступу або розповсюдження шкідливого коду. Зловмисники можуть використовувати IP-адреси та доменні імена для відправки спаму, фішингових електронних листів або інших шкідливих повідомлень. Вони можуть фальшиво підробляти відправників, щоб видавати себе за легітимних користувачів або компанії. Зловмисники можуть використовувати IP-адреси та доменні імена для розповсюдження вірусів, троянців, червів та іншого шкідливого програмного забезпечення, яке може завдати шкоди користувачам або мережам. Ці загрози підкреслюють важливість захисту IP-адрес та доменних імен в Інтернеті, а також впровадження заходів безпеки для запобігання подібним атакам.

1.4. Основні методи виявлення загроз доменних імен та IP адрес

Основні методи [14] виявлення загроз доменних імен та IP-адрес можуть включати такі аспекти:

- використання спеціальних сервісів для моніторингу реєстрації нових доменних імен, які можуть бути використані для фішингу, шахрайства або інших кібератак;
- перевірка WHOIS-інформації для доменних імен для виявлення підозрілих відомостей, таких як неправдива контактна інформація або короткий термін реєстрації;
- використання спеціалізованих систем, які блокують доступ до вебресурсів з відомих шкідливих IP-адрес або мереж;
- встановлення правил на рівні DNS для блокування доступу до відомих шкідливих доменних імен;
- моніторинг мережевого трафіку для виявлення незвичайних або підозрілих підключень до шкідливих IP-адрес або доменних імен;
- встановлення спеціалізованих систем, які автоматично виявляють та блокують потенційні загрози, що включають шкідливі доменні імена та IP-адреси;
- ретельний аналіз логів та журналів подій для виявлення незвичайної активності, яка може вказувати на проникнення через шкідливі доменні імена або IP-адреси.

Ці методи можуть бути використані окремо або в комбінації для забезпечення виявлення та запобігання загрозам, пов’язаним з доменними іменами та IP-адресами.

1.5. Огляд існуючих засобів для моніторингу безпеки доменів та IP адрес

Існує багато засобів [3] для моніторингу безпеки доменів та IP-адрес, які допомагають виявляти загрози та захищати мережі від кібератак. Наведемо кілька з них.

WHOIS Lookup Services [26]**.** Сервіси, такі як WHOIS Lookup, дозволяють перевіряти реєстраційну інформацію для доменних імен, включаючи терміни реєстрації, власника та інші деталі. Це допомагає виявляти підозрілі домени або шахраїв (рис.1.1).

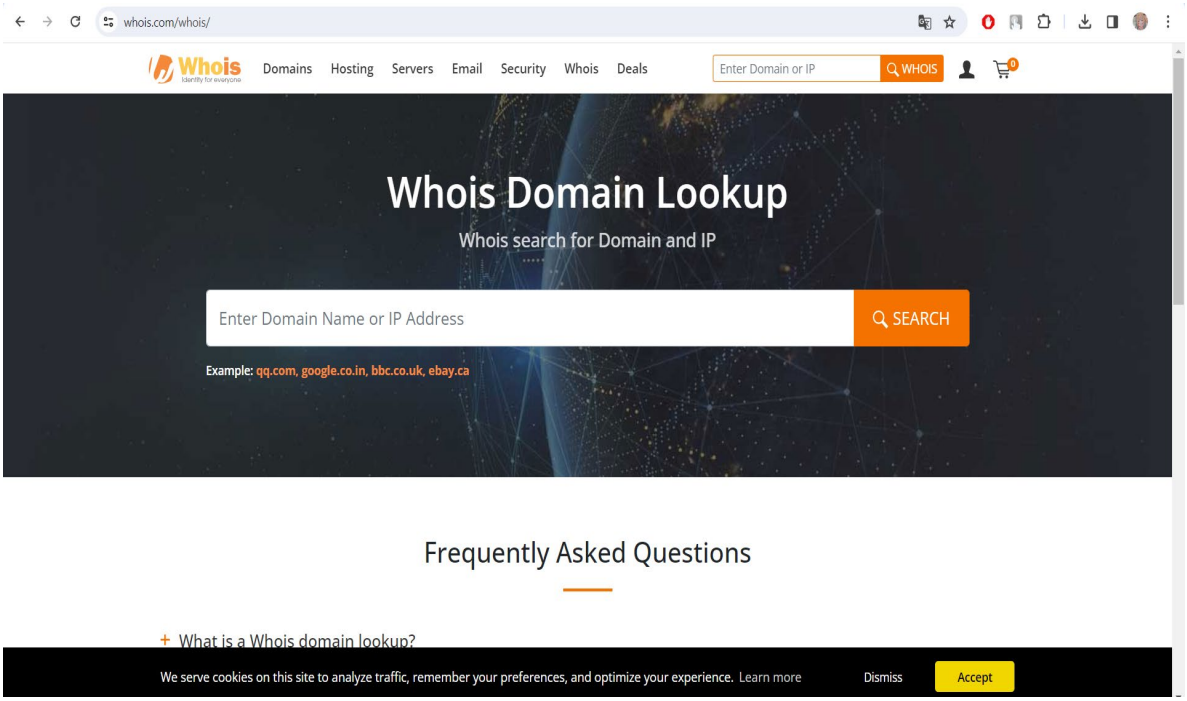


Рисунок 1.1 – Сервіс WHOIS Lookup

DNS Security Extensions (DNSSEC). DNSSEC [6] забезпечує цифровий підпис для DNS-записів, що дозволяє перевірити їх автентичність та цілісність. Це допомагає уникнути DNS-підробок та MITM-атак (рис. 1.2).

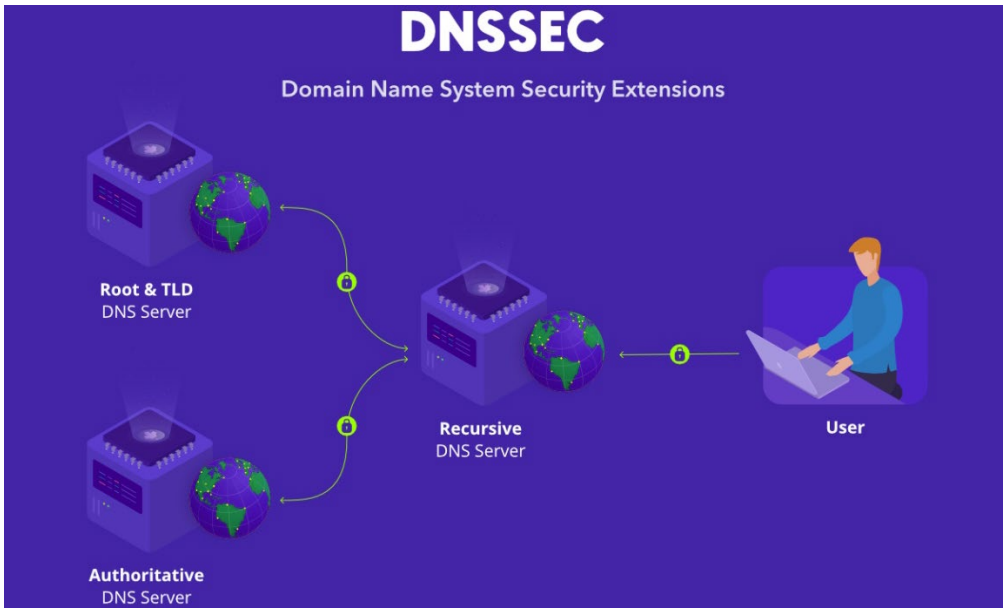


Рисунок 1.2 – Приклад схеми DNS Security Extensions (DNSSEC)

DNS Firewall Services [25]**.** Послуги, які використовуються для блокування доступу до відомих шкідливих доменних імен через встановлення правил на рівні DNS (рис. 1.3).

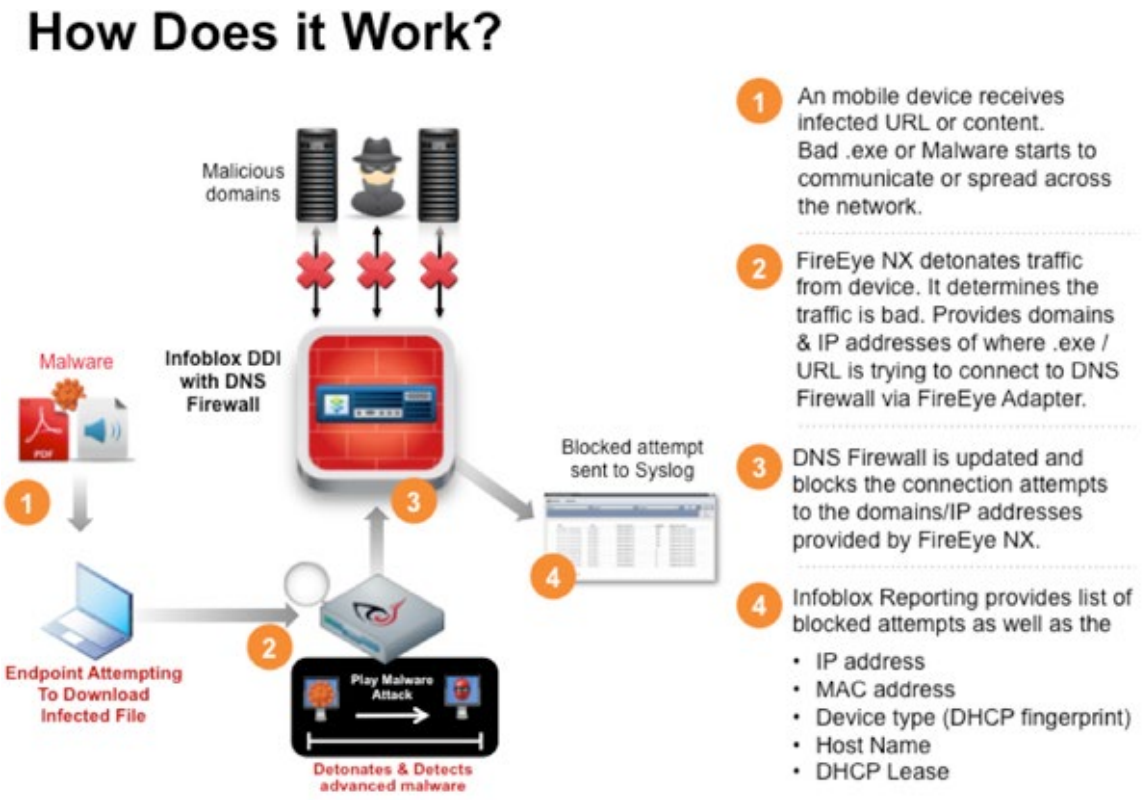


Рисунок 1.3 – Схема роботи DNS Firewall Services

IP Reputation Services. Сервіси, які надають інформацію про репутацію IP-адрес. Вони дозволяють визначати, чи відомий IP-адрес є шкідливим або відомим з проблемами безпеки (рис. 1.4).

Intrusion Detection Systems (IDS) та Intrusion Prevention Systems (IPS) [18]. Системи, які моніторять мережевий трафік для виявлення та блокування незвичайної або підозрілої активності, включаючи атаки через доменні імена та IP-адреси (рис. 1.5).

Malware Scanning Services [8]**.** Сервіси, які проводять сканування вебсайтів для виявлення шкідливих програм, включаючи ті, які можуть бути пов’язані зі шкідливими доменами або IP-адресами (рис. 1.6).

Threat Intelligence Platforms [21]**.** Платформи, які надають інформацію про загрози, що дозволяють моніторити активність та аналізувати потенційні загрози для доменів та IP-адрес (рис. 1.7).

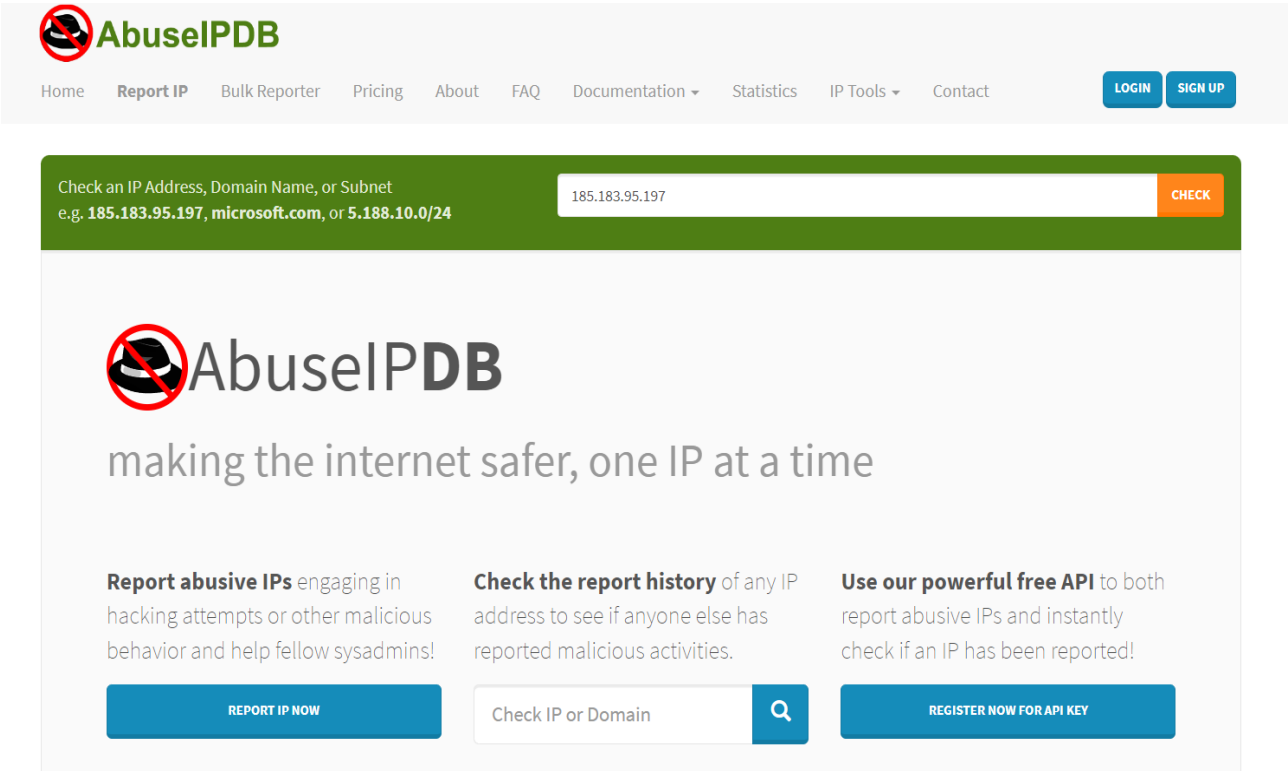


Рисунок 1.4 – Приклад сервісу з IP Reputation Services – AbuseIPDB

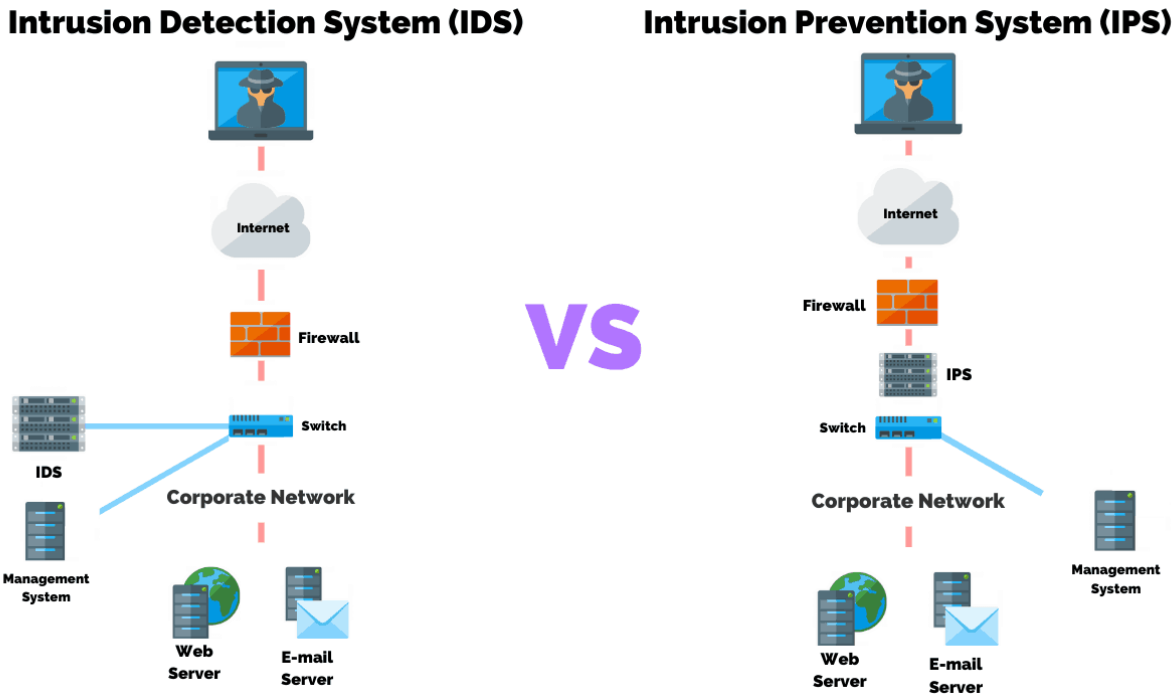


Рисунок 1.5 – Схеми роботи Intrusion Detection Systems (IDS) та Intrusion Prevention Systems (IPS)

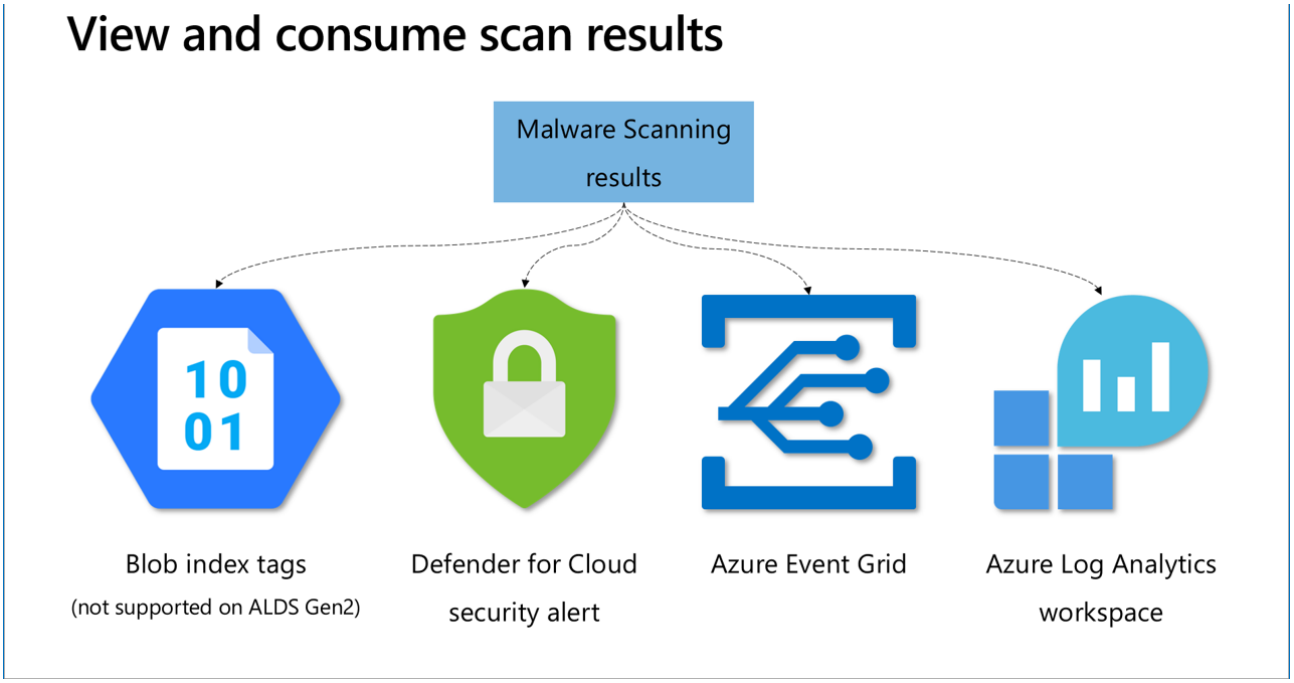


Рисунок 1.6 – Результати Malware Scanning Services

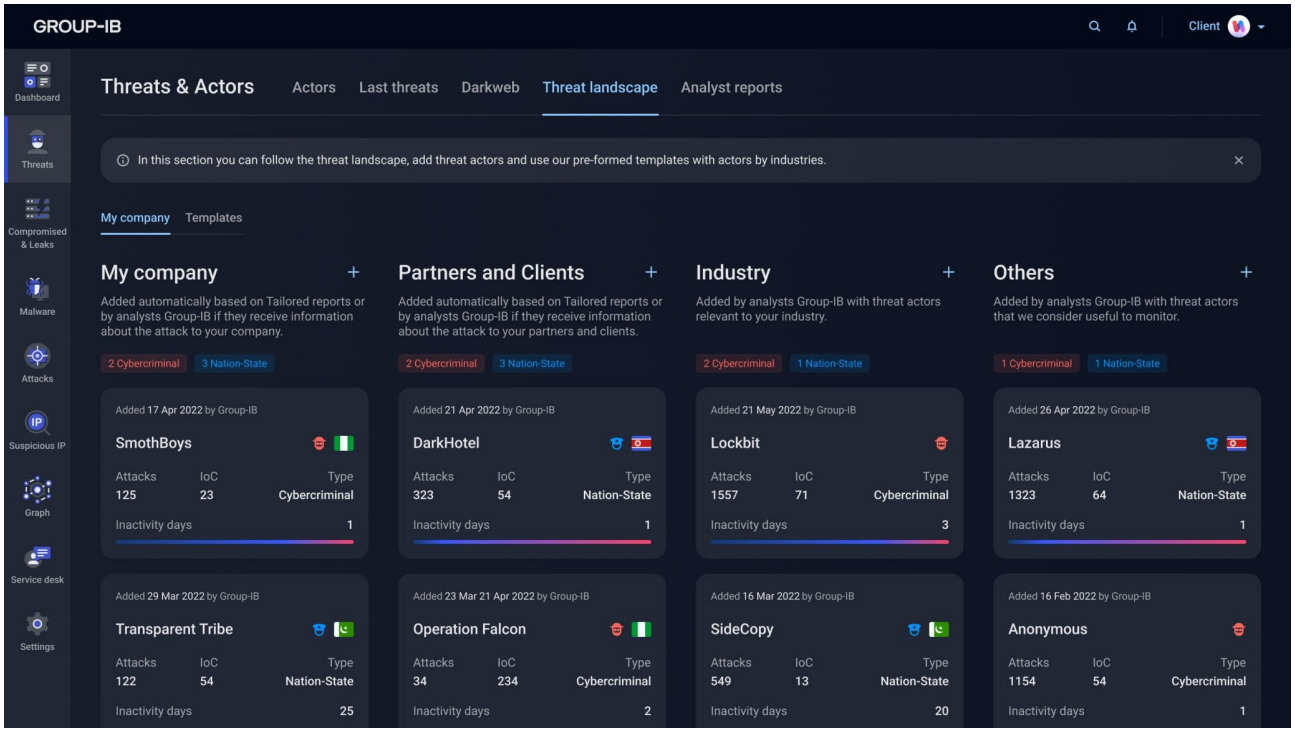


Рисунок 1.7 – Приклад сервісу з Threat Intelligence Platforms — Group-IB

Ці засоби допомагають підтримувати безпеку доменів та IP-адрес, надаючи можливість вчасно виявляти, аналізувати та реагувати на потенційні загрози.

Наведемо декілька інструментів та сервісів, які можна використовувати самостійно для моніторингу безпеки доменів та IP-адрес.

DomainTools [15] надає широкий спектр інструментів для моніторингу безпеки доменів, включаючи можливість перевірки WHOIS, аналізу історії реєстрації доменів, виявлення підроблених доменів і багато іншого (рис. 1.8).

SecurityTrails [19] – це інструмент для моніторингу безпеки, який дозволяє вам відстежувати історію змін IP-адрес, DNS-записів та інших параметрів для доменів. Він також надає інформацію про власників та мережеві ресурси (рис.1.9).

Shodan [20] – це пошукова система для Інтернету, який сканує мережі та збирає інформацію про підключені пристрої. Він може бути використаний для пошуку вразливих пристроїв за їх IP-адресами (рис. 1.10).

GreyNoise Intelligence [16] збирає дані про шум в мережі, що дозволяє виділяти шумовий трафік від потенційно шкідливого. Це допомагає виявляти спам-боти, сканувальні пристрої та інші шкідливі джерела (рис. 1.11).

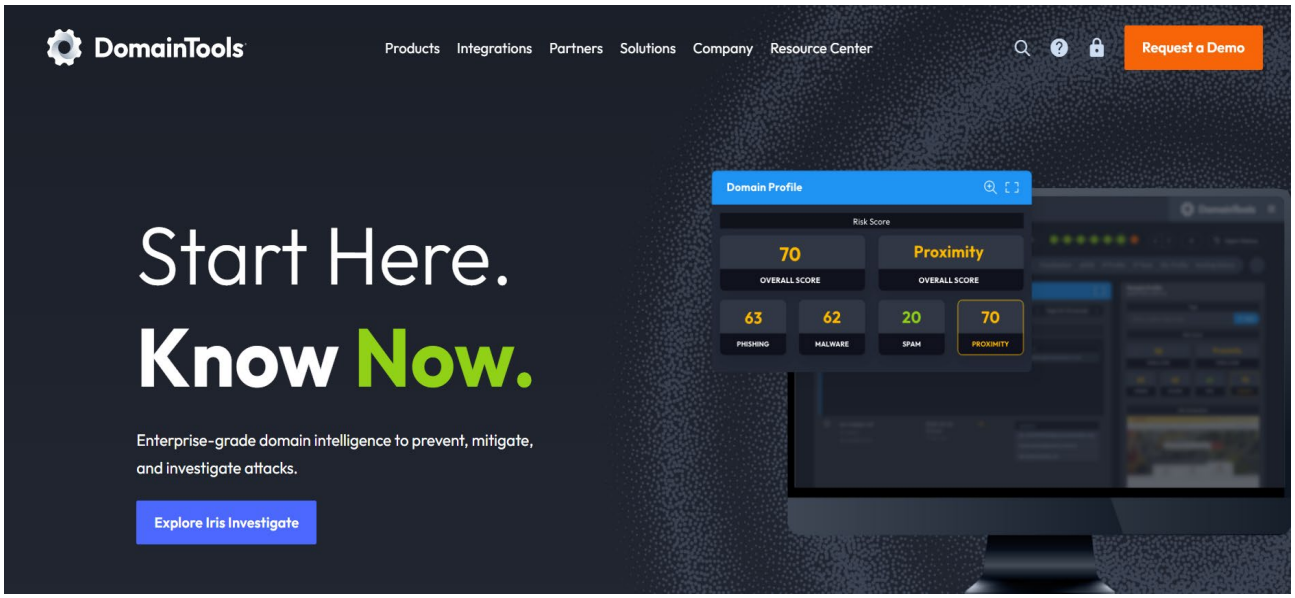


Рисунок 1.8 – Сервіс DomainTools

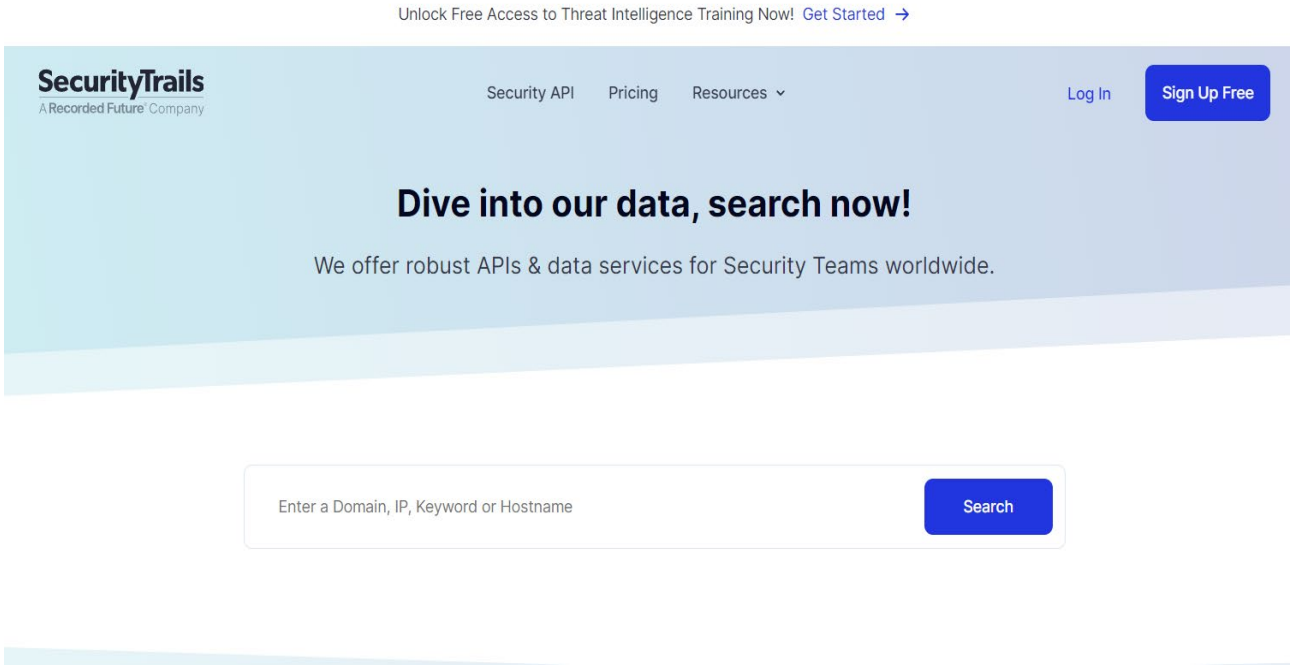


Рисунок 1.9 – Сервіс SecurityTrails

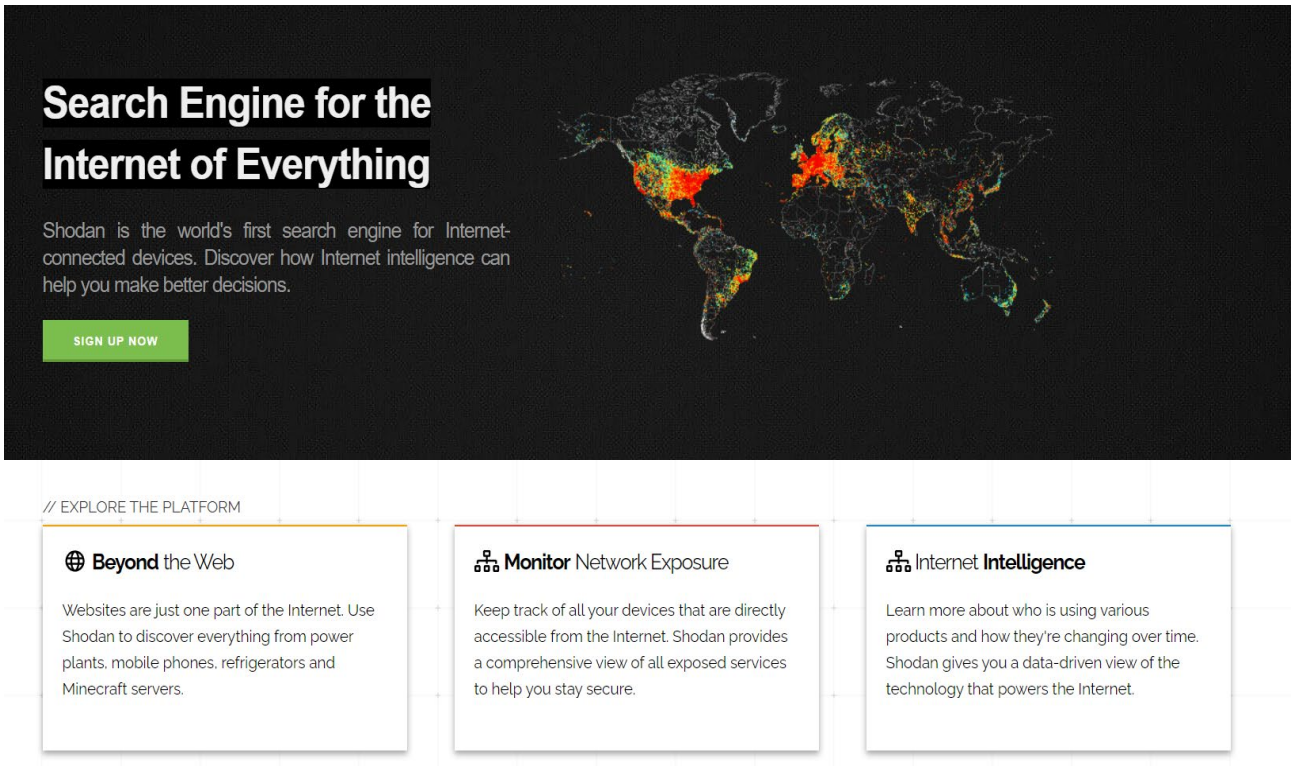


Рисунок 1.10 – Сервіс Shodan

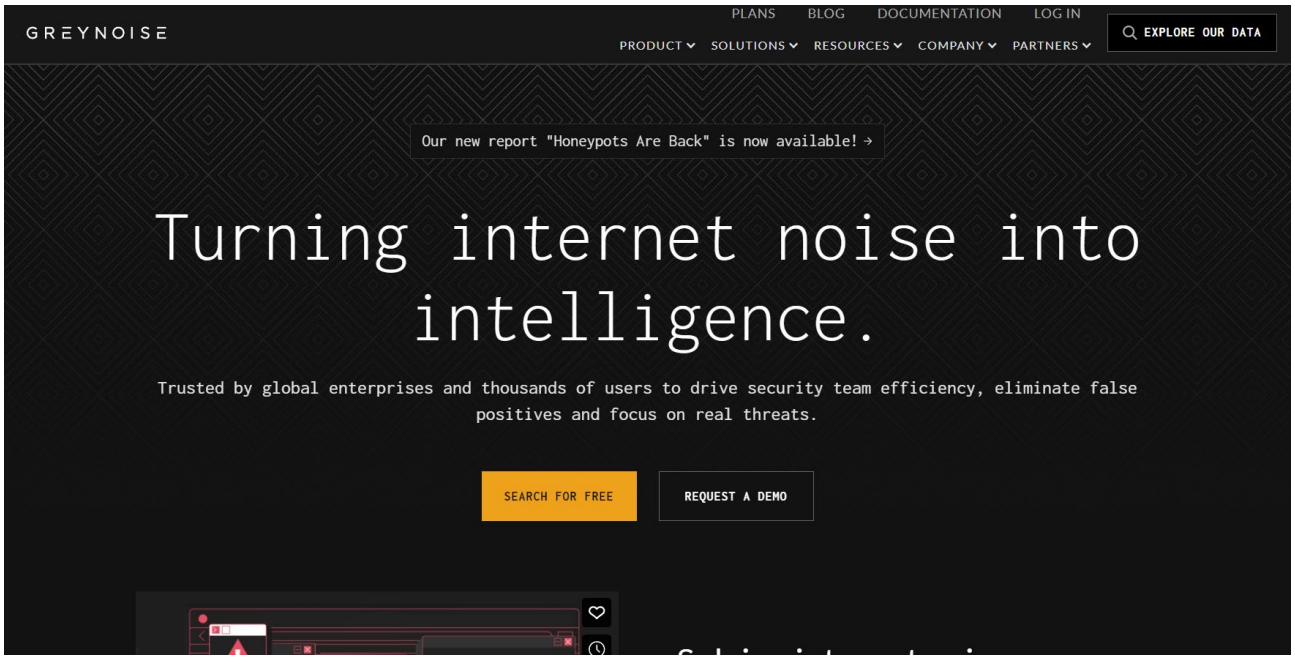


Рисунок 1.11 – Сервіс GreyNoise Intelligence

ThreatConnect [13] – це платформа управління загрозами, яка дозволяє аналізувати, спільно працювати та реагувати на кіберзагрози. Вона надає інформацію про доменні імена, IP-адреси та інші атрибути загроз (рис.1.12).

VirusTotal [23] – це безкоштовний сервіс, який дозволяє перевіряти файли, URL-адреси та IP-адреси на віруси, троянці, черв'яки та інші загрози. Він може бути використаний для аналізу шкідливих доменів та IP-адрес (рис.1.13).

Ці інструменти надають можливості для моніторингу безпеки доменів та IP-адрес, а також допомагають виявляти та вирішувати потенційні загрози в Інтернеті.

Рисунок 1.12 – Сервіс ThreatConnect

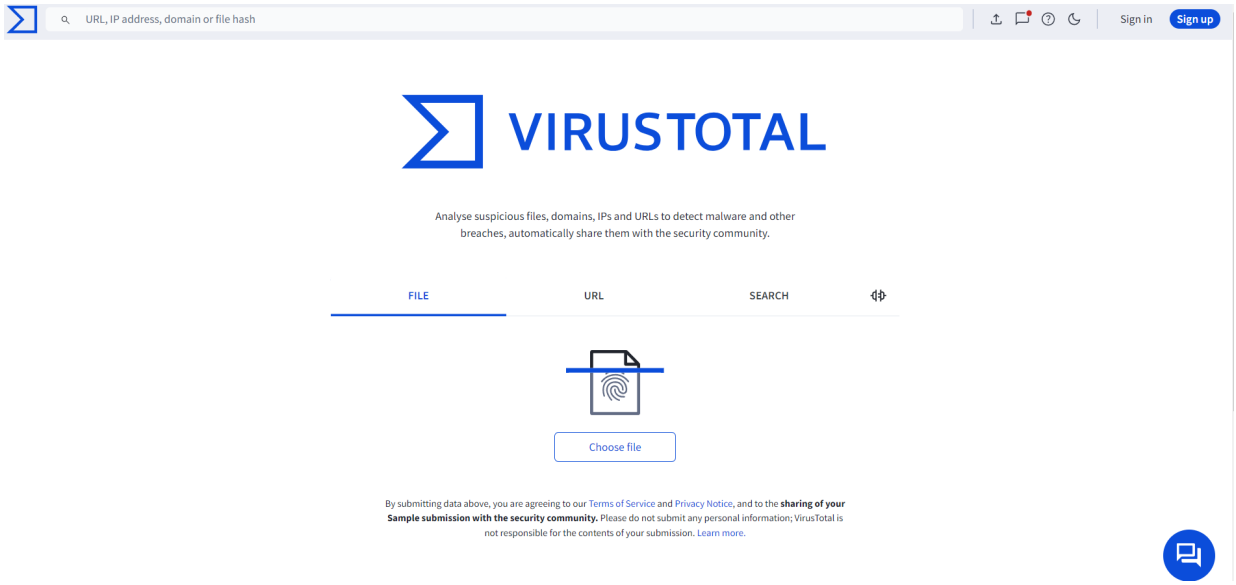


Рисунок 1.13 – Сервіс VirusTotal

1.6. Огляд та аналіз аналогічних програмних розробок

Розглянемо бота, який надає інформацію про домен (рис. 1.14).

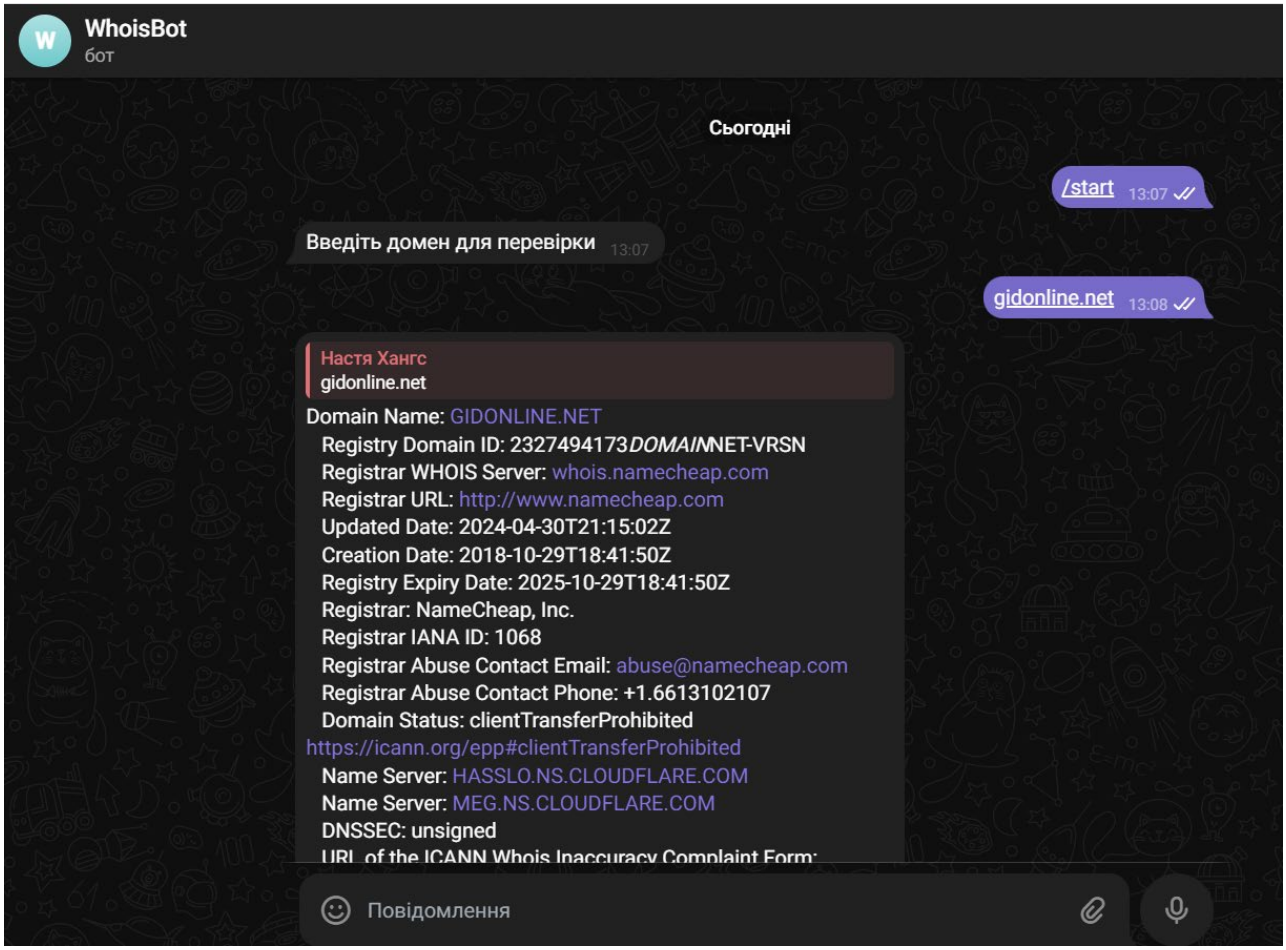


Рисунок 1.14 – Отримання інформації про домен «gidonline.net»

Назва бота: «@BesthostingBot».

Призначення: «@BesthostingBot» [32] створений для полегшення пошуку інформації про власників доменів та IP-адрес. Він виконує функції Whois-бота, що дозволяє користувачам швидко отримувати інформацію про наявність доменного імені та деталі про власника домену чи IP-адреси.

Основні функції бота:

- надання інформації про те, чи вільний домен, який ввів користувач;
- надання детальної інформації про власника доменного імені, включаючи дані реєстрації, термін дії, контактні дані реєстратора тощо;

- надання інформації про власників IP-адрес, що допомагає ідентифікувати та перевіряти ресурси в Інтернеті.

Бот написаний на мові програмування Python, використовуючи фреймворк python-telegram-bot. Для обробки та зберігання даних можуть використовуватися бази даних, такі як PostgreSQL [33] або MySQL [34]. Хостинг та розгортання бота можуть здійснюватися на сервісах, таких як Heroku [35] або AWS [36].

Власник бота: besthosting.ua.

Користувачі можуть почати взаємодію з ботом, відправивши команду «/start». Наприклад, при перевірці домену «gidonline.net» бот надає інформацію подану на рис 1.14.

РОЗДІЛ 2
РОЗРОБКА TELEGRAM-БОТА ДЛЯ МОНІТОРИНГУ БЕЗПЕКИ ДОМЕННИХ ІМЕН ТА IP АДРЕС

2.1 Постановка задачі, призначення та вимоги до програмного засобу

Основним завданням кваліфікаційної роботи було розробка засобу для отримання та аналізу інформації про зловмисність доменних імен та IP-адрес за допомогою інструментарію платформи Telegram. Для цього потрібно спроєктувати та розробити Telegram-бот для моніторингу доменних імен та IP-адрес з метою виявлення потенційно небезпечних об’єктів в мережі Інтернет. Бот повинен відправляти звіти користувачам у разі виявлення аномальних ситуацій при отриманні великої кількості доменних імен чи IP-адрес, що потребують уваги або відправляти повідомлення з коротким аналізом конкретної IP-адреси або доменного імені. Основними завданнями бота є виявлення підозрілих або потенційно небезпечних доменів та IP-адрес, а також інформування користувачів про виявлені загрози.

Наведемо вимоги до програмного засобу.

Функціональні вимоги:

- моніторинг доменних імен та IP-адрес на предмет зловмисних дій та аномальної активності;
- відправлення звітів користувачам у разі виявлення підозрілих об’єктів;
- надсилання повідомлень з аналізом домену або IP-адреси, що потребують уваги користувачів;
- забезпечення безпеки та конфіденційності оброблюваних даних.

Нефункціональні вимоги:

- висока швидкодія та ефективність аналізу доменів та IP-адрес;
- надійність та стабільність роботи програмного засобу;
- простий та зрозумілий інтерфейс для користувачів;
- масштабованість для обробки великого обсягу даних;

- сумісність з платформою Telegram для надсилання повідомлень.

Ці вимоги стануть основою для розробки програмного засобу, який забезпечить ефективний моніторинг доменів та IP-адрес та надійно захистить користувачів від потенційних кіберзагроз.

2.2 Загальний опис проєкту

Основною метою кваліфікаційної роботи є розробка чат-бота, який буде надавати користувачам інформацію про потенційно зловмисні домени та IP-адреси через месенджер Telegram. Перед початком проєктування та розробки бота необхідно визначити основні вимоги та описати їх.

Сучасні методи виявлення та запобігання зловмисним діям в Інтернеті постійно вдосконалюються. Підозрілі домени та IP-адреси можуть бути використані для різних шахрайських атак, включаючи фішинг, розповсюдження шкідливих програм та інші види кіберзлочинності. Тому одним із головних завдань у цій галузі є швидке виявлення потенційно небезпечних доменів та IP-адрес для забезпечення безпеки користувачів Інтернету.

З огляду на цільову аудиторію та важливість забезпечення їхньої безпеки, головною метою даного проєкту є створення чат-бота для моніторингу потенційно зловмисних доменів та IP-адрес через месенджер Telegram. Розроблена система має надавати користувачам зручний сервіс для отримання інформації про підозрілі домени та IP-адреси та забезпечувати швидке реагування на загрози кібербезпеки.

З урахуванням вищезазначених цілей, передбачається розробка Telegram-бота, який виконуватиме наступні функції:

- забезпечення взаємодії з вебсайтами та іншими джерелами для отримання актуальної інформації про потенційно зловмисні домени та IP-адреси;
- здійснення збереження отриманих даних в спеціально створеній базі даних для подальшого аналізу та використання;

- забезпечення належного збереження даних для кожного користувача та надання індивідуалізованої інформації щодо потенційно зловмисних доменів та IP-адрес.

Алгоритм взаємодії користувача та бота виглядає наступним чином:

1. користувач починає взаємодію, відправляє перше повідомлення боту через месенджер Telegram;
2. бот отримує повідомлення від користувача і аналізує його, щоб зрозуміти, які саме дії користувач планує здійснити;
3. обробка запиту користувача, де залежно від отриманого запиту, бот виконує відповідні дії:
   - якщо користувач вимагає моніторингу великої кількості доменів або IP-адрес, бот розпочинає аналіз доступних даних та виявлення потенційних загроз та надсилає звіт;
   - у випадку, якщо користувач вимагає отримати аналіз вказаного домену або IP-адреси, бот проводить необхідні дії для аналізу та надсилає повідомлення з висновком щодо безпеки об’єкта;
4. після обробки запиту бот формує відповідь та відправляє її користувачеві через месенджер Telegram;
5. взаємодія між ботом та користувачем завершується після відправлення відповіді користувачеві.

2.3 Вибір моделі розробки програмного засобу

Модель бота для моніторингу на зловмисність доменів та IP-адрес використовує архітектуру клієнт-сервер [5], де бот, як клієнт, взаємодіє з сервером для обробки запитів користувачів та аналізу даних моніторингу.

Основні компоненти моделі включають клієнтську частину, серверну частину, логіку взаємодії, базу даних та взаємодія з сервером Telegram.

Клієнтська частина:

- користувачі, які взаємодіють з ботом через месенджер Telegram;

- бот, який отримує повідомлення від користувачів та надсилає їм відповіді на їх запити.

Серверна частина:

- сервер, розміщений на комп'ютері з постійним підключенням до Інтернету;
- сервер бази даних, який зберігає інформацію про користувачів та дані моніторингу;

Логіка взаємодії:

- бот приймає повідомлення від користувача через месенджер Telegram;
- після отримання повідомлення бот передає його на сервер для обробки;
- сервер аналізує отримане повідомлення та виконує необхідні дії для моніторингу доменів та IP-адрес;
- якщо знайдено підозрілу активність, сервер генерує відповідь та передає її боту;
- бот надсилає знайдену інформацію користувачеві через месенджер Telegram;

База даних. Сервер бази даних зберігає інформацію про користувачів (ідентифікатори чатів) та дані моніторингу (підозрілі домени та IP-адреси).

Взаємодія з сервером Telegram.

- повідомлення, надіслані користувачем або ботом, спочатку надсилаються на сервери Telegram.
- сервери Telegram відповідають за шифрування повідомлень та нагляд за комунікацією між клієнтами.

2.4 Обґрунтування вибору інструментальних засобів розробки

2.4.1 Середовище розробки PyCharm

Обрано PyCharm [27] як середовище програмування. PyCharm – це інтегроване середовище розробки від JetBrains, яке стало одним з найпопулярніших інструментів для програмістів Python. Його визначає широкий

спектр функцій, які значно полегшують розробку програмного забезпечення. Відмітимо декілька переваг та недоліків. До переваг можна віднести зручний інтерфейс. PyCharm має добре організований інтерфейс, що дозволяє легко переходити між файлами, вкладками, та іншими ресурсами проекту. Також ще однією перевагою є підтримка мови Python. Ця IDE спеціалізується на роботі з Python, тому має розширені можливості для підтримки мови, включаючи автодоповнення, підказки, рефакторинг коду та інше. PyCharm містить вбудовані інструменти для розробки, такі як дебагер, система контролю версій, інструменти для рефакторингу, а також підтримку тестування коду. Завдяки інтеграції з віддаленими інтерпретаторами Python, PyCharm дозволяє розробляти та відебагати код на віддалених серверах або в контейнерах. PyCharm дозволяє встановлювати різноманітні плагіни, що розширюють його можливості.

До недоліків слід віднести великі вимоги до ресурсів системи. PyCharm може вимагати значних ресурсів комп’ютера, що може призводити до повільної роботи на менш потужних пристроях. Повна версія PyCharm є комерційною, хоча є і безкоштовна версія Community Edition з обмеженим функціоналом. Ще одним з недоліків є складний процес освоєння функцій. Як і будь-яка інша складна програма, PyCharm потребує часу для оволодіння всіма його можливостями, що може бути викликаною для початківців.

Наведемо основні інструменти PyCharm:

- відлагодження коду (Debugger) дозволяє крокувати через код, встановлювати точки зупинки та вивчати значення змінних;
- автодоповнення (Autocompletion) пропонує автоматичне завершення коду, що спрощує роботу з бібліотеками та API;
- рефакторинг коду (Code Refactoring) надає можливість автоматизувати процес перетворення та покращення структури коду;
- підтримка віртуальних середовищ (Virtual Environments) дозволяє створювати та керувати ізольованими середовищами Python для проектів;

- інтеграція з системами контролю версій (Version Control Systems Integration), підтримка таких систем, як Git, дозволяє зручно працювати з версіями коду.

2.4.2 Мова програмування Python

Python [28] є відмінним вибором для створення телеграм ботів з кількох причин:

- Python відомий своєю зрозумілим та лаконічним синтаксисом, що спрощує написання та підтримку коду;
- існує багато бібліотек, які значно полегшують створення телеграм ботів, наприклад, «python-telegram-bot», «telepot» та інші;
- Python має велику та активну спільноту розробників, що означає наявність безлічі ресурсів, форумів та готових рішень, які можуть допомогти у вирішенні проблем;
- Python легко інтегрується з іншими системами та сервісами, такими як бази даних, веб-сервіси, інші API тощо, що дозволяє створювати комплексні та функціональні боти;
- існує багато сервісів, які підтримують розгортання Python застосунків, включаючи хмарні платформи як Heroku, AWS, Google Cloud та інші, що спрощує процес деплойменту телеграм бота.

Загалом, Python є потужним інструментом для створення телеграм ботів завдяки своїй зручності, доступним бібліотекам та широким можливостям інтеграції.

2.4.3 Модуль telegram.ext

Модуль telegram.ext [29] в бібліотеці python-telegram-bot надає інструменти для розробки телеграм ботів, спрощуючи обробку повідомлень, команд та інших подій. Цей модуль містить класи та функції, які допомагають будувати більш структуровані та підтримувані боти, зокрема, за допомогою

диспетчера, обробників та конверсійних систем. Наведемо ключові компоненти telegram.ext.

Updater є одним із центральних елементів, що полегшує роботу з API Telegram. Він отримує нові оновлення від Telegram і передає їх диспетчеру для обробки. Updater також може запускати вебхуки [30] для прийому оновлень.

Dispatcher відповідає за маршрутизацію вхідних повідомлень та інших оновлень до відповідних обробників. Можна зареєструвати обробники для різних типів оновлень, таких як команди, текстові повідомлення, зображення тощо.

Обробники використовуються для визначення реакцій на різні типи повідомлень та подій. Є кілька основних типів обробників:

- CommandHandler – обробляє команди, що починаються з /;
- MessageHandler – обробляє текстові повідомлення, які відповідають заданим фільтрам;
- CallbackQueryHandler – обробляє callback-запити з інлайн-кнопок;
- InlineQueryHandler – обробляє інлайн-запити.

Фільтри допомагають обробникам визначати, які повідомлення вони повинні обробляти. Фільтри можуть комбінуватися за допомогою операторів & (AND) та | (OR).

ConversationHandler спрощує створення ботів, які ведуть довгі діалоги з користувачами. Він дозволяє визначати стани та відповідні їм обробники, що допомагає відслідковувати прогрес користувача в діалозі.

JobQueue дозволяє планувати виконання повторюваних завдань, таких як відправка повідомлень або виконання певних дій через задані проміжки часу.

Context об’єкт містить корисні атрибути, такі як bot, job_queue, user_data, chat_data, що полегшує доступ до цих ресурсів.

Модуль telegram.ext надає потужні інструменти для створення телеграм ботів, що дозволяють зручно обробляти різні типи повідомлень, вести діалоги з користувачами та планувати завдання. Використовуючи компоненти цього модуля, можна створити бота будь-якої складності з мінімальними зусиллями.

2.5 Особливості програмної реалізації

Першим кроком у розробці бота є його створення безпосередньо в Telegram за допомогою спеціального бота під назвою «BotFather». Для початку реєстрації бота потрібно скористатися командою «/newbot». Після цього BotFather попросить вас ввести необхідні дані для реєстрації, такі як назву бота (яка буде відображатися у контактній інформації) та його username (коротке ім’я, яке використовується у посиланнях). Після завершення реєстрації ви отримаєте унікальний токен, який дозволить вам взаємодіяти з вашим ботом. Після створення бота та отримання його токену ми можемо зробити додаткові налаштування використовуючи деякі команди (рис 2.1).

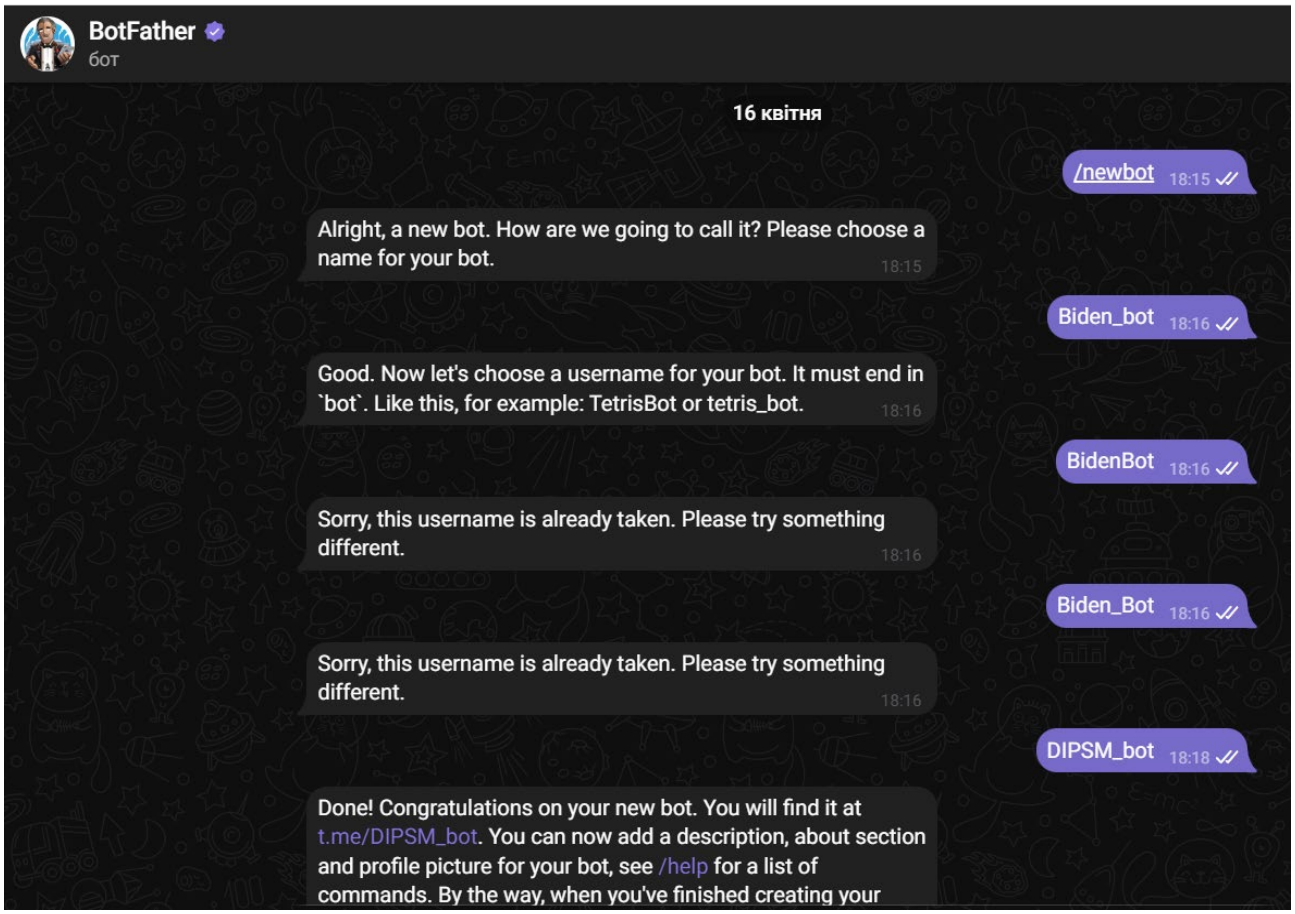


Рисунок 2.1 – Створення бота за допомогою «BotFather»

Таблиця 2.1

Команди для налаштування бота

Команда	**Опис**
/setname	Змінити назву бота
/setdescription	Змінити опис бота (відображається при першому запуску)
/setabouttext	Змінити інформацію про бота (відображається при відкритті профілю)
/setuserpic	Змінити список команд, які підтримує бот
/setcommands	Налаштувати список доступних команд
/token	Згенерувати токен для авторизації
/revoke	Анулювати токен доступу до бота
/setinline	Налаштувати чи доступний бот в inline режимі
/setinlinegeo	Налаштувати можливість передавати дані про місцезнаходження бота з іншого чату
/setinlinefeedback	Налаштувати отримування оновлень щодо результатів, вибраних користувачами
/setjoingroups	Визначити, чи можна додавати бота в групи
/setprivacy	Ввімкнути режим конфіденційності

Щоб створити якісний сервіс, необхідно правильно організувати файлову структуру. Добре структурована система файлів допоможе ефективно розподілити файли за їх функціоналом і полегшить в них навігацію. (рис. 2.1).

У директорії «bot/» зосереджена основна логіка бота, що відповідає за його роботу. Файл __init__.py забезпечує ініціалізацію пакету, тоді як main.py (рис. 2.2) містить головний функціонал для запуску бота. У report.py реалізовані функції генерації звітів, user_interface.py відповідає за взаємодію з користувачем, а analytics.py містить методи для перевірки IP-адрес на небезпеку. Файл analytics_domain.py аналізує домен на наявність загроз або шкідливої інформації

про нього. Піддиректорія «api_clients/» містить клієнтів для взаємодії з зовнішніми API сервісами. У shodan_client.py реалізовується клієнт для роботи з Shodan API, abusedb_client.py – з AbuseDB API, a virustotal_client.py – з VirusTotal API. Це дозволяє ботові використовувати зовнішні сервіси для перевірки IP-адрес на шкідливість.

У директорії «config/» зберігаються конфігураційні файли бота. Файл setting.py містить налаштування, необхідні для роботи бота, такі як токени доступу, параметри підключення до бази даних тощо.

```
Baiden_bot/
|-- bot/
|   |-- __init__.py
|   |-- main.py
|   |-- report.py
|   |-- analytics.py
|   |-- analytics_domain.py
|   |-- api_clients/
|       |-- __init__.py
|       |-- shodan_client.py
|       |-- abusedb_client.py
|       |-- virustotal_client.py
|-- config/
|   |-- __init__.py
|   |-- setting.py
|-- utils/
|   |-- __init__.py
|   |-- common.py
|-- Arial-Bold.ttf
|-- requirements.txt
|-- README.md
|-- .env
```

Рисунок 2.2 – Схема файлової структури проєкту.

Директорія «utils/» містить допоміжні функції, які спрощують роботу бота. Файл common.py включає загальні утилітарні функції, які можуть бути використані в різних частинах проекту. Файл requirements.txt містить список залежних бібліотек, необхідних для роботи бота, що дозволяє їх швидко встановити. Файл .env зберігає конфіденційні дані, такі як токени доступу та інші

секрети, які не повинні бути збережені у відкритому вигляді в коді. Таким чином, структура проекту забезпечує чітке розділення логіки, конфігурації та допоміжних функцій, що дозволяє легко підтримувати та розширювати функціональність бота.

Крім того, у проекті використаний файл Arial-Bold.ttf, який містить шрифт Arial Bold. Цей шрифт потрібен для генерації звітів або інших документів, де важлива відповідність певним стилям оформлення тексту.

Функція «start» в проєкті створює інтерфейс для користувача, коли він вперше запускає бота.

Коли користувач починаєте спілкування з ботом, ця функція створює кнопки для вибору: «Створити звіт аналізу IP адрес на шкідливість»; «Відправити повідомлення-звіт про один домен чи IP адресу»; «Отримати детальну інформацію про домен». Також ця функція організовує ці кнопки у зручну інлайнову клавіатуру та відправляє повідомлення з цими кнопками користувачу.

Це дозволяє користувачеві швидко обрати необхідну дію, просто натиснувши на відповідну кнопку.

Функція «button_handler» обробляє вибір користувача, коли він натискає одну з інлайнових кнопок, що відображаються в інтерфейсі бота.

Коли користувач натискає кнопку в боті, ця функція: отримує вибір користувача і підтверджує, що натискання кнопки відбулося успішно; визначає, яку саме кнопку натиснув користувач. Якщо користувач вибрав «Створити звіт аналізу IP адрес на шкідливість», бот запитає користувача чи надіслати список IP-адрес для аналізу. Якщо користувач вибрав «Відправити повідомлення-звіт про один домен чи IP адресу», то бот запитає користувача чи надіслати назву домену для аналізу. Якщо користувач вибрав «Отримати детальну інформацію про домен», то бот запитає користувача чи надіслати назву домену для отримання детальної інформації про нього.

Таким чином, функція реагує на вибір користувача і направляє його до наступного кроку, щоб отримати потрібну інформацію або виконати необхідну дію.

Функція «analyze_and_send_report» (рис. 2.3) обробляє введені користувачем IP-адреси, аналізує їх на шкідливість і надсилає звіт у форматі PDF.

```
def analyze_and_send_report_ip(update, context):
    user_input = update.message.text
    addresses = user_input.split()
    print(addresses)
    update.message.reply_text("Починаємо аналіз IP-адрес...")

    try:
        # Виклик функцій з analytics.py
        ip_info = info_ip_list_total(addresses)
        aggregated_data = aggregate_results_and_mark_maliciousness(ip_info)
        pprint(aggregated_data)

        # Перевірка, чи будь-які IP-адреси виявлені як шкідливі (reputation > 0)
        malicious_ips = [ip for ip in aggregated_data if aggregated_data[ip].get('reputation', 0) > 0]

        if not malicious_ips:
            update.message.reply_text(
                "Не виявлено шкідливих IP-адрес. Перевірте також правильність написання цих адрес.")
            return

        # Викликаємо функцію для створення PDF звіту
        pdf_file_path = 'reportIP.pdf'
        create_pdf_report(aggregated_data, addresses)

        # Перевірка, чи файл існує
        if os.path.exists(pdf_file_path):
            # Відкриття файлу PDF у режимі читання байт ('rb')
            with open(pdf_file_path, 'rb') as pdf_file:
                # Відправка файлу користувачу
                context.bot.send_document(
                    chat_id=update.effective_chat.id,
```

Рисунок 2.3 – функція «analyze_and_send_report» у файлі main.py

Коли користувач надсилає список IP-адрес для аналізу, ця функція: отримує IP-адреси з повідомлення; повідомляє користувача, що аналіз почався; аналізує IP-адреси на шкідливість, використовуючи спеціальні функції «info_ip_list_total», «aggregate_results_and_mark_maliciousness»; перевіряє чи існує IP-адреса або чи є хоча б одна шкідлива IP-адреса; створює звіт у форматі PDF з результатами аналізу за допомогою функції «create_pdf_report»; перевіряє, чи успішно створений файл звіту; надсилає звіт у PDF форматі у чат з

користувачем, якщо файл існує; інформує користувача, якщо сталася помилка при створенні звіту.

Функція «get_domain_details» отримує вказаний користувачем домен та надсилає йому детальну інформацію результатів аналізу домену.

Коли користувач надсилає домен для перевірки, ця функція: отримує домен з повідомлення користувача; інформує користувача, що починається отримання інформації про домен; перевіряє чи існує домен; перевіряє репутацію домену за допомогою спеціальної функції «check_domain_reputation»; формує повідомлення з детальною інформацією; надсилає повідомлення користувачу з цією інформацією; повідомляє користувача, якщо дані про домен не вдалося отримати. При формуванні повідомлення з детальною інформацією, якщо дані знайдено, бот подає наступні дані:

- загальний підсумок аналізів антивірусів;
- DNS-записи;
- HTTPS-сертифікат;
- Alexa-ранг (показує популярність сайту);
- репутаційний рейтинг;
- дата закінчення реєстрації домену.

Функція «get_domain_short_info» надсилає більш стисло результати аналізу вказаного домену користувачем. Функція «main» (рис. 2.4) виконує налаштування та запуск телеграм-бота. Ця функція виконує наступні дії: створює об’єкт оновлювача («Updater») і додає токен бота для авторизації; додає обробника команд для команди «/start», яка викликає функцію «start»; додає обробника для відповіді на натискання кнопок у інтерфейсі бота та викликає функцію «button_handler»; додає обробника для текстових повідомлень, які не є командами та викликає функцію «text_message_handler»; запускає бота для прийому та обробки повідомлень в режимі опитування; утримує бота у режимі роботи, доки він не буде зупинений або програма не буде завершена. Ця функція встановлює всі необхідні обробники та запускає бота, щоб він міг відповідати на команди та повідомлення користувачів.

Файл «settings.py» (рис. 2.5) призначений для зберігання конфіденційних даних та налаштувань, таких як ключі API та токени доступу. Ці дані завантажуються з файлу «.env» за допомогою бібліотеки «dotenv», яка дозволяє зберігати конфіденційні дані у вигляді змінних середовища.

```
def main():
    # Створення оновлювача та додавання токену бота
    updater = Updater(TOKEN, use_context=True)
    dp = updater.dispatcher

    # Додавання командного обробника для команди /start
    updater.dispatcher.add_handler(CommandHandler( command: "start", start))

    # Додавання обробника для кнопок
    updater.dispatcher.add_handler(CallbackQueryHandler(button_handler))

    # Додавання обробника для текстових повідомлень
    dp.add_handler(MessageHandler(Filters.text & ~Filters.command, text_message_handler))


    # Запуск бота
    updater.start_polling()
    updater.idle()


# Виконання функції main
if __name__ == '__main__':
    main()
```

Рисунок 2.4 – функція «main» у файлі main.py

```
import os
from dotenv import load_dotenv

load_dotenv()

SHODAN_API_KEY = os.getenv('SHODAN_API_KEY')
ABUSEDB_API_KEY = os.getenv('ABUSEDB_API_KEY')
VIRUSTOTAL_API_KEY = os.getenv('VIRUSTOTAL_API_KEY')
TELEGRAM_TOKEN = os.getenv('TELEGRAM_TOKEN')
MXTOOLBOX_API_KEY = os.getenv('MXTOOLBOX_API_KEY')
```

Рисунок 2.5 – файл settings.py

У файлі common.py знаходяться допоміжні функції.

Функція «get_index»: приймає список data, якщо список має більше одного елемента, перебирає кожен елемент і перевіряє наявність ключа ssl; повертає індекс першого елемента, що містить ключ ssl, якщо список містить лише один елемент, повертає 0.

Функція «is_valid_domain»: приймає рядок domain; використовує регулярний вираз для перевірки, чи відповідає рядок формату доменного імені; повертає True, якщо домен є валідним, і False в іншому випадку.

Функція «change_str_to_date»: приймає рядок date_number у форматі YYYYMMDD; витягує рік, місяць і день з рядка; перетворює ці значення на дату типу datetime.date; повертає об’єкт date.

Функція «wrap_text» розбиває текст на кілька рядків, щоб він вміщувався у задану ширину. Ця функція: приймає текст, максимальну ширину рядка, шрифт і розмір шрифту; створює PDF документ для вимірювання ширини тексту; розбиває текст на слова; формує рядки, додаючи слова, поки рядок не досягне максимальної ширини; повертає список рядків, які вміщуються у задану ширину. Функція корисна для форматування тексту, щоб він правильно відображався у документі або на екрані.

Функція «create_pdf_report» (рис 2.6) створює PDF звіт про зловмисність IP-адрес. Ця функція виконує такі дії: ініціалізує PDF документ та встановлює шрифти та розмір сторінки; додає заголовок «Звіт про зловмисність IP-адрес» і список IP-адрес; встановлює початкову позицію для тексту на сторінці; виводить інформацію для кожної IP-адреси; завершує і зберігає PDF документ. При виведенні інформації для кожної IP-адреси бот:

- показує IP-адресу, її репутацію та деталі аналізу;
- форматує текст, щоб він вміщувався в ширину сторінки;
- додає інформацію про зловмисність і репутаційні дані;
- якщо текст не вміщується на сторінці, додає нову сторінку.

```python
def create_pdf_report(aggregated_data, ip_list):
    try:
        pdf_file = "reportIP.pdf"
        pdfmetrics.registerFont(TTFont( name: 'Arial-Bold',  filename: 'Arial-BoldMT.ttf'))
        pdfmetrics.registerFont(TTFont( name: 'Arial',  filename: 'arial.ttf'))

        c = canvas.Canvas(pdf_file, pagesize=letter)
        width, height = letter  # Взято з розмірів сторінки letter

        # Заголовок звіту
        c.setFont( psfontname: 'Arial-Bold',  size: 18)
        c.drawString( x: 100, height - 40,  text: "Звіт про зловмисність IP-адрес")

        c.setFont( psfontname: 'Arial-Bold',  size: 12)
        c.drawString( x: 100,  y: 800, "Список IP-адрес : " + ", ".join(ip_list))

        # Підзаголовок
        c.setFont( psfontname: 'Arial-Bold',  size: 12)
        c.drawString( x: 100, height - 60,  text: "Аналіз та оцінка індикаторів компрометації (IoC)")

        # Встановлюємо початкову позицію для тіла звіту
        y_position = height - 100

        # Виводимо інформацію для кожної IP-адреси
        for ip, data in aggregated_data.items():
            reputation = data['reputation']
            if reputation > 0:
                c.setFont( psfontname: 'Arial-Bold',  size: 14)
                c.drawString( x: 100, y_position,  text: f"IP-адреси: {ip}")
                y_position -= 20
```

```python
                c.setFont( psfontname: 'Arial',  size: 10)
                if len(data['reports']) != 0:
                    for report in data['reports']:
                        lines = wrap_text( text: f"Деталі: {report}", width - 220,  font: "Arial",  font_size: 10)
                        for line in lines:
                            c.drawString( x: 120, y_position, line)
                            y_position -= 14

                c.drawString( x: 140, y_position,  text: f"Репутація: {reputation}")
                y_position -= 14
                c.drawString( x: 140, y_position,  text: "(Репутація всіх сервісів, на яких була знайдена інформація про IP-адреси)")
                y_position -= 14

                # Висновок по кожній IP-адресі
                c.setFont( psfontname: 'Arial-Bold',  size: 10)
                maliciousness = "Висока" if data['maliciousness'] >= 5 else "Низька"
                lines = wrap_text( text: f"Зловмисність: {maliciousness} ({data['total_score']:.2f}) \n(Репутація ділиться на суму
                for line in lines:
                    c.drawString( x: 120, y_position, line)
                    y_position -= 20

                # Перевірка, щоб не вийти за межі сторінки
                if y_position < 100:
                    c.showPage()
                    y_position = height - 100

        # Завершуємо PDF
        c.save()

    except Exception as e:
```

Рисунок 2.6 – Функція «create_pdf_report» у файлі report.py

Функція «parse_ip_info» (рис. 2.7) обробляє відповідь від API AbuseDB та витягує з неї інформацію про IP-адресу.

Функція створює словник ip_data, в який зберігає такі дані:

- ip: IP-адреса;
- hostname: імена хостів;
- domain: доменні імена;
- country: код країни;
- isp: ннтернет-провайдер;
- abuse_score: оцінка зловмисності;
- total_reports: загальна кількість скарг.

```python
import requests
from dipr_bot.config.settings import ABUSEDB_API_KEY


1 usage
def parse_ip_info(response):
    if response is not None:
        try:
            ip_data = {
                'ip': response['data']['ipAddress'],
                'hostname': response['data']['hostnames'],
                'domain': response['data']['domain'],
                'country': response['data']['countryCode'],
                'isp': response['data']['isp'],
                'abuse_score': response['data']['abuseConfidenceScore'],
                'total_reports': response['data']['totalReports'],
            }
            return ip_data
        except Exception as e:
            # print(f"AbuseDB response error: {e}")
            return None
```

Рисунок 2.7 — функція «parse_ip_info» для AbuseIPDB

Клас AbuseDBClient (рис. 2.8) створюється для взаємодії з API сервісу AbuseIPDB [30], щоб отримати інформацію про репутацію IP-адреси.

Ініціалізація («__init__»): self.ABUSEDB_API_URL (URL-адреса API сервісу AbuseIPDB); self.ip (збереження переданої IP-адреси).

Метод «get_ip_reputation» надсилає запит до API сервісу AbuseIPDB для отримання репутаційних даних про IP-адресу.

Заголовки запиту включають API-ключ та тип очікуваного контенту (JSON). Параметри запиту це IP-адреса, яку перевіряють та максимальний вік даних у днях (90 днів). Також у цій функції використовується метод «requests.get» для надсилання GET-запиту до API. Функція повертає відповідь у форматі JSON. У випадку помилки запиту, повертає значення None.

```
2 usages
class AbuseDBClient:

    def __init__(self, ip):
        self.ABUSEDB_API_URL = 'https://api.abuseipdb.com/api/v2/check'
        self.ip = ip

    1 usage
    def get_ip_reputation(self):
        headers = {
            'Key': ABUSEDB_API_KEY,
            'Accept': 'application/json'
        }
        params = {
            'ipAddress': self.ip,
            'maxAgeInDays': '90'
        }
        try:
            response = requests.get(self.ABUSEDB_API_URL, headers=headers, params=params)
            return response.json()
        except requests.RequestException as e:
            # print(f"AbuseDB API error: {e}")
            return None
```

Рисунок 2.8 – клас «AbuseDBClient» у файлі abused_client.py

Для кожного модуля (Shodan, AbuseDB, VirusTotal) є своя функція «parse_ip_info», яка витягує дані з JSON файлу, отриманого від відповідного модулю сервісу. Також для кожного модуля створений свій індивідуальний клас для взаємодії з сервісом, наприклад для AbuseDB це клас AbuseDBClient. Функція «info_ip» отримує дані про IP-адресу за допомогою двох функцій: одна для отримання даних , інша – для їх обробки.

Функція «info_ip_list_total» збирає інформацію про список IP-адрес, використовуючи сервіси AbuseDB [31], Shodan [20] та VirusTotal [23]. Для кожної IP-адреси створює об’єкти «AbuseDBClient», «ShodanClient» та «VirusTotalClient». Також використовує функцію «info_ip» для отримання та обробки даних з кожного сервісу для кожної адреси та зберігає отриману інформацію у словник. Ці функції дозволяють автоматизувати процес збору та обробки даних про IP-адреси з різних джерел, що робить їх корисними для аналізу безпеки.

Функція «calculate_shodan_score» (рис. 2.9) перевіряє безпеку IP-адреси за допомогою інформації від Shodan. Вона аналізує інформацію про порти, сертифікат SSL та дати дії сертифікату.

Якщо знайдено шкідливі порти, або ім’я підписувача та власника сертифікату співпадають, або термін дії сертифікату закінчився, то функція рахує відповідні показники безпеки та надає деталі про виявлені проблеми.

```
def calculate_shodan_score(shodan_info):
    harm_ports = [3389, 21, 23, 3306, 1434, 1433, 1522, 5432, 27017]
    current_date = datetime.datetime.now().date()
    expires_date = change_str_to_date(shodan_info.get('expires_cert_date'))
    subj_cert = shodan_info.get('subject_cert')
    iss_cert = shodan_info.get('issuer_cert')
    ports = shodan_info['ports']
    score = 0
    details = []

    # Приклад оцінки на основі тегів Shodan
    for p in range(0, len(ports)):
        if ports[p] in harm_ports:
            score += 1
            details.append(f'Виявлено шкідливий порт {ports[p]} in {ports}')
            break
    if subj_cert == iss_cert:
        score += 3
        details.append(f'# Ім`я того, хто підписував та того, кому надали сертифікат, співпадають')
        # Ім`я того, хто підписував та того, кому надали сертифікат, співпадають
    if current_date == expires_date:
        score += 3
        details.append(f'Термін дії сертифікату минув (поточна дата: {current_date} and дата закінчення : {expires_date})')

    result = [score, details]
    return result
```

Рисунок 2.9 – функція «calculate_shodan_score»

Функція «calculate_abusedb_score» (рис. 2.10) обчислює показники безпеки IP-адреси за допомогою інформації, отриманої від AbuseDB. Вона використовує дані про рівень зловживання та загальну кількість скарг на IP-адресу. Якщо є дані про рівень зловживання, функція розраховує оцінку шкідливості на основі цього рівня зловживання та додає відповідні деталі до результату. Також, якщо є дані про загальну кількість скарг, вона також враховує їх у розрахунку оцінки шкідливості та додає відповідні деталі до результату.

```
def calculate_abusedb_score(abusedb_info):
    # Припустимо, що abusedb_info містить поле "abuseConfidenceScore"
    score = 0
    details = []
    abuse_score = abusedb_info.get('abuse_score', 0)
    abuse_reports = abusedb_info.get('total_reports', 0)
    if abuse_score > 0:
        score = abuse_score / 100 * 5
        details.append(f'Оцінка зловживання: {abuse_score}% впевненості')
    if abuse_reports > 0:
        score += abuse_reports / 100 * 5
        details.append(f'Загальні скарги: {abuse_reports}')
    result = [score, details]
    return result
```

Рисунок 2.10 – функція «calculate_abusedb_score»

Функція «calculate_virtustotal_score» (рис. 2.11) обчислює оцінку шкідливості за допомогою інформації, отриманої від сервісу VirusTotal. Вона використовує дані про репутацію, кількість нешкідливих та шкідливих голосів. Функція «calculate_country» (рис. 2.12) обчислює оцінку шкідливості на основі переліку популярних країн-хакерів. Якщо одна з країн у списку hackers_countries співпадає з будь-якою країною з вхідного списку country_list, функція присвоює оцінку 4 та повертає деталі про цю країну як частину результату. Якщо ж країни IP-адреси серед таких країн немає, результат буде містити оцінку 0 та пустий список деталей.

```python
def calculate_virustotal_score(virustotal_info):
    # Припустимо, що virustotal_info містить поле "positives" для позитивних знахідок

    reputation = virustotal_info.get('reputation', 0)
    harmless_vt = virustotal_info['total_votes']['harmless']
    malicuous_vt = virustotal_info['total_votes']['malicious']
    score = 0
    details = []
    if 1 < reputation < 100:
        score = (100 - reputation) / 100 * 2
        details.append(f'Репутація низька на VirusTotal: {reputation} ')  # Репутація низька
    if 1 < harmless_vt < 100:
        score += (100 - harmless_vt) / 100 * 2
        details.append(f'Нешкідливість низька на VirusTotal: {harmless_vt} ')  # Нешкідливість низька
    if malicuous_vt > 100:
        score += malicuous_vt / 100 * 2
        details.append(f'Підозрілість висока на VirusTotal: {malicuous_vt} ')  # Підозрілість висока

    result = [score, details]
    return result
```

Рисунок 2.11 – функція «calculate_virtustotal_score»

```python
def calculate_country(country_list):
    hackers_countries = ['China', 'CH', 'India', 'IN', 'Russian Federation', 'RU', 'Brazil', 'BR', 'Iran', 'IR',
                         'North Korea', 'KP', 'Namibia', 'NA', 'Libya', 'LY', 'Ethiopia', 'ET', 'Zimbabwe', 'ZW',
                         'Cameroon', 'CM',
                         'Tanzania', 'TZ', 'Zambia', 'ZM', 'Uganda', 'UG', 'Kenya', 'KE', 'South Africa', 'ZA',
                         'Morocco', 'MA', 'Nigeria', 'NG', 'Tunisia', 'TN']
    score = 0
    details = []
    for c in range(0, len(country_list)):
        if country_list[c] in hackers_countries:
            score = 4
            details = [f'Країна {country_list[c]} в списку найнебезпечніших країн для хакерів']
            break

    result = [score, details]

    return result
```

Рисунок 2.12 – функція «calculate_country»

Функція «aggregate_results_and_mark_maliciousness» обробляє дані про вибрані IP-адрес у словник та обчислює їх загальну репутацію, використовуючи різні джерела інформації, такі як Shodan, AbuseDB, VirusTotal. Крім того, вона обчислює рівень зловмисності для кожної IP-адреси на основі її загального балу репутації (рис. 2.13). Функція «check_domain_reputation» (рис. 2.14) використовує API для перевірки репутації домену за допомогою сервісу VirusTotal. Вона виконує запит до вказаного URL, використовуючи API-ключ, який зберігається в змінній VIRUSTOTAL_API_KEY. Після отримання відповіді в форматі JSON, функція обробляє дані, викликаючи іншу функцію «collect_all_data» для збору всієї необхідної інформації. Якщо відповідь не

містить даних, функція виводить повідомлення про помилку та повертає None, інакше повертає результати обробки.

```python
def aggregate_results_and_mark_maliciousness(ip_data_list):
    # Ініціалізуємо порожній словник для збереження кінцевих даних
    aggregated_data = {}
    # Для порахування балів за країну

    # Визначаємо вагу кожного індикатора
    weights = {
        'shodan': 1.0,
        'abusedb': 1.0,
        'virustotal': 1.0,
    }

    # Сума ваг для нормалізації
    total_weight = sum(weights.values())

    for ip_data in ip_data_list:
        data_country = []
        ip = ip_data['ip']
        # Якщо IP вже існує у словнику, складаємо значення
        if ip not in aggregated_data:
            aggregated_data[ip] = {
                'reputation': 0,
                'reports': [],
                'total_score': 0
            }

        # Обчислюємо репутацію на основі даних від Shodan
        if 'shodan_info' in ip_data:
            data_country.append(ip_data['shodan_info']['country'])
            score = calculate_shodan_score(ip_data['shodan_info'])
            aggregated_data[ip]['reputation'] += score[0] * int(weights['shodan'])
            aggregated_data[ip]['reports'] += score[1]

        # Обчислюємо репутацію на основі даних від AbuseDB
        if 'abusedb_info' in ip_data:
            data_country.append(ip_data['abusedb_info']['country'])
            score = calculate_abusedb_score(ip_data['abusedb_info'])
            aggregated_data[ip]['reputation'] += score[0] * int(weights['abusedb'])
            aggregated_data[ip]['reports'] += score[1]

        # Обчислюємо репутацію на основі даних від VirusTotal
        if 'virustotal_info' in ip_data:
            data_country.append(ip_data['virustotal_info']['country'])
            score = calculate_virustotal_score(ip_data['virustotal_info'])
            aggregated_data[ip]['reputation'] += score[0] * int(weights['virustotal'])
            aggregated_data[ip]['reports'] += score[1]

        if len(data_country) != 0:
            score_country = calculate_country(data_country)
            aggregated_data[ip]['reputation'] += score_country[0]
            aggregated_data[ip]['reports'] += score_country[1]

        # Нормалізуємо репутацію
        aggregated_data[ip]['total_score'] = int(aggregated_data[ip]['reputation'] / total_weight)

    # Маркування зловмисності
    for ip, data in aggregated_data.items():
        data['maliciousness'] = classify_maliciousness(aggregated_data[ip]['total_score'])

    return aggregated_data
```

Рисунок 2.13 – функція «aggregate_results_and_mark_maliciousness»

```python
import requests
from datetime import datetime
from dipr_bot.config.settings import VIRUSTOTAL_API_KEY
from pprint import pprint


3 usages
def check_domain_reputation(domain):
    # Використання API для перевірки репутації домену
    virus_total_api_key = VIRUSTOTAL_API_KEY
    url = f"https://www.virustotal.com/api/v3/domains/{domain}"
    headers = {
        "x-apikey": virus_total_api_key
    }
    response = requests.get(url, headers=headers)
    data_responce = response.json()
    if data_responce is not None:
        result = collect_all_data(data_responce)
        pprint(data_responce)
        return result
    else:
        print("Не вдалося перевірити репутацію домену")
        return None
```

Рисунок 2.14 – функція «check_domain_reputation»

Функція приймає дані про домен у форматі JSON і викликає різні допоміжні функції для збору різноманітної інформації про домен та заповнює словник даними з цих функцій.

Функція «antivirus_results» отримує дані про аналіз різних антивірусів домену та повертає підсумкове повідомлення, що містить кількість виявлених шкідливих та підозрілих елементів. Якщо обидва показники рівні 0, функція повертає значення None. Кожна функція (крім «https_certificate»), яка аналізує дані отриманих з сервісу, має також розширену версію, яка надає більш детальну інформацію про домен.

Функція «dns_records» (рис. 2.15) витягує останні DNS-записи з вхідних даних і перевіряє, чи є серед них записи, які можуть бути підозрілими або

шкідливими. Якщо такі записи знайдені, вони виділяються у підкресленому форматі у вихідному повідомленні.

Якщо ж жодних підозрілих записів не виявлено, функція повертає значення None.

```
def dns_records(data):
    # Отримання списку DNS-записів
    last_dns_records = data.get('data', {}).get('attributes', {}).get('last_dns_records', [])

    # Створення повідомлення з підсумком DNS-записів
    summary_message = ""
    suspicious_records = False
    observed_ns = set()  # Зберігати вже виявлені сервери імен

    for record in last_dns_records:
        record_type = record.get('type', 'Невідомий тип запису')
        record_value = record.get('value', 'Невідоме значення')

        # Перевірка на наявність шкідливих записів
        if record_type in ['NS', 'MX', 'TXT', 'SPF', 'DKIM', 'CAA']:
            suspicious_records = True
            # Перевірка, чи запис NS не вже був виявлений
            if record_value in observed_ns:
                continue  # Пропустити дублікати записів NS
            else:
                observed_ns.add(record_value)  # Додати цей запис NS у список виявлених
            # Підкреслення DNS-запису
            summary_message += f"<u>- Тип: {record_type}, Значення: {record_value}</u>\n"

    if suspicious_records:
        return summary_message
    else:
        return None
```

Рисунок 2.15 — функція «dns_records»

Функція «https_certificate» (рис. 2.16) аналізує дані про HTTPS-сертифікат та створює детальний опис цього сертифіката. Вона перевіряє такі аспекти сертифіката, як основне ім’я (Common Name), термін дії, серійний номер, алгоритм підпису, і виводить відповідні повідомлення щодо кожного з них.

Крім того, функція враховує різні можливі проблеми, такі як співпадіння основного імені з ім’ям видачі, недійсність терміну дії сертифіката або використання застарілих алгоритмів підпису.

```python
def https_certificate(data):
    # Отримуємо дані про HTTPS-сертифікат
    last_https_certificate = data.get('data', {}).get('attributes', {}).get('last_https_certificate', {})
    if not last_https_certificate:
        return None

    print(last_https_certificate.get('subject', {}).get('CN', 'Невідоме CN'))
    print(last_https_certificate.get('issuer', {}).get('CN', ''))

    # Перевірка на співпадіння основного імені
    common_name = last_https_certificate.get('subject', {}).get('CN', 'Невідоме CN')
    if common_name == last_https_certificate.get('issuer', {}).get('CN', ''):
        common_name_message = f"<u>Попередження: Основне ім'я (CN) сертифіката  співпадає з ім'ям видачі.</u>\n"
    else:
        common_name_message = ""

    # Перевірка на термін дії сертифіката
    validity_period_message = ""
    not_before = last_https_certificate.get('validity', {}).get('not_before', '')
    not_after = last_https_certificate.get('validity', {}).get('not_after', '')
    if not_before and not_after:
        from datetime import datetime
        current_date = datetime.utcnow()
        not_before_date = datetime.strptime(not_before, _format: '%Y-%m-%d %H:%M:%S')
        not_after_date = datetime.strptime(not_after, _format: '%Y-%m-%d %H:%M:%S')
        if current_date < not_before_date:
            validity_period_message = "Попередження: Термін дії сертифіката ще не настав.\n"
        elif current_date > not_after_date:
            validity_period_message = "<u>Попередження: Термін дії сертифіката вже минув.</u>\n"
```

```python
        validity_period_message = "Попередження: Термін дії сертифіката ще не настав.\n"
    elif current_date > not_after_date:
        validity_period_message = "<u>Попередження: Термін дії сертифіката вже минув.</u>\n"

# Перевірка серійного номеру сертифіката
serial_number_message = ""
serial_number = last_https_certificate.get('serial_number', '')

# Перевірка алгоритму підпису сертифіката
signature_algorithm_message = ""
signature_algorithm = last_https_certificate.get('cert_signature', {}).get('signature_algorithm', '')
if signature_algorithm in ['MD2', 'MD4', 'MD5', 'SHA1']:
    signature_algorithm_message = "<u>Попередження: Використовується застарілий або слабкий алгоритм підпису.</u>\n"

# Створюємо повідомлення з підсумком даних сертифіката
summary_message = ""
summary_message += f"CN (Common Name/Основне ім'я): {common_name}\n{common_name_message}\n"
summary_message += f"Термін дії: {not_before} - {not_after}\n{validity_period_message}\n"
summary_message += f"Серійний номер: {serial_number}\n{serial_number_message}\n"
summary_message += f"Алгоритм підпису: {signature_algorithm}\n{signature_algorithm_message}\n"

print(summary_message)

return summary_message
```

Рисунок 2.16 – функція «https_certificate»

Функція «alexa_rank» отримує значення рангу Alexa з вхідних даних. Вона перевіряє, чи існує значення рангу, і якщо так, повертає повідомлення з рангом Alexa. Якщо значення рангу відсутнє, функція повертає None.

Функція «reputation» отримує значення репутації з вхідних даних і повертає відповідне повідомлення про репутацію у підкресленому форматі, якщо репутація менша за 0, інакше повертає значення None.

Функція «domain_expiry_date» отримує WHOIS інформацію про домен та шукає дату закінчення реєстрації домену: якщо вона знаходить таку дату, порівнює її з поточною датою; якщо термін дії домену вже минув, функція повертає повідомлення про те, що реєстрація закінчилася, і рекомендує звернутися до реєстратора для подовження; якщо домен ще активний, функція повертає None; якщо дата закінчення реєстрації не знайдена, функція також повертає None.

Для того щоб розпочати розмову з ботом, користувачу необхідно ввести команду /start (рис. 2.17). Після цього бот для користувача дає можливість вибору, який реалізовано у вигляді кнопок у повідомленні. Користувач може обрати:

- створити звіт аналізу на зловмисність списку IP-адрес;
- відправити повідомлення-звіт про вказаний домен;
- отримати детальну інформацію про вказаний домен.

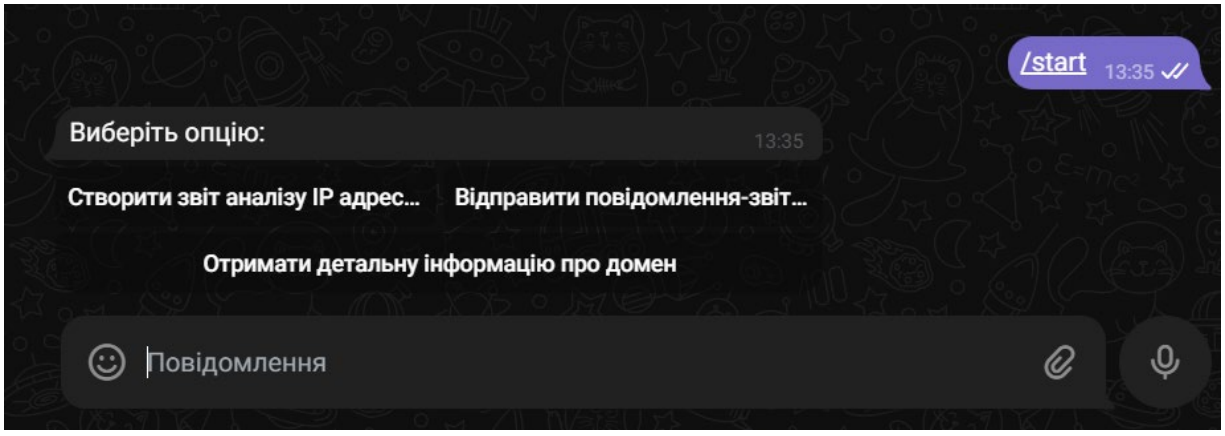


Рисунок 2.17 – Результат виконання команди «/start»

При виборі звіту по IP-адресам, користувачу необхідно ввести список адрес для перевірки на шкідливість. Адреси потрібно вводити через пробіл, також адреси можна вводити як і в протоколі IPv6, так і протоколі IPv4. Для того,

щоб бот перевірив кожну адресу зі списку, потрібно зачекати певний проміжок часу, який залежить від кількості IP-адрес. Після закінчення перевірки, бот надсилає готовий звіт користувачу (рис 2.18). Якщо звіт не вдалося створити, наприклад всі адреси нешкідливі, то також виводиться повідомлення користувачу (рис. 2.20). Звіт формується у PDF форматі та надсилається користувачу (рис. 2.19)

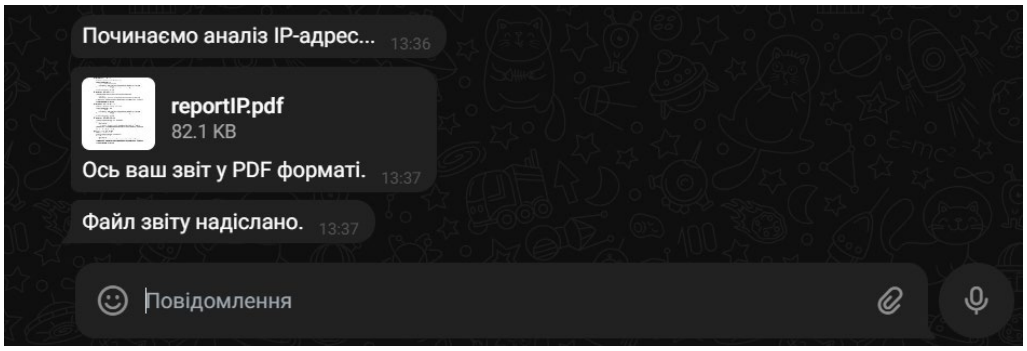
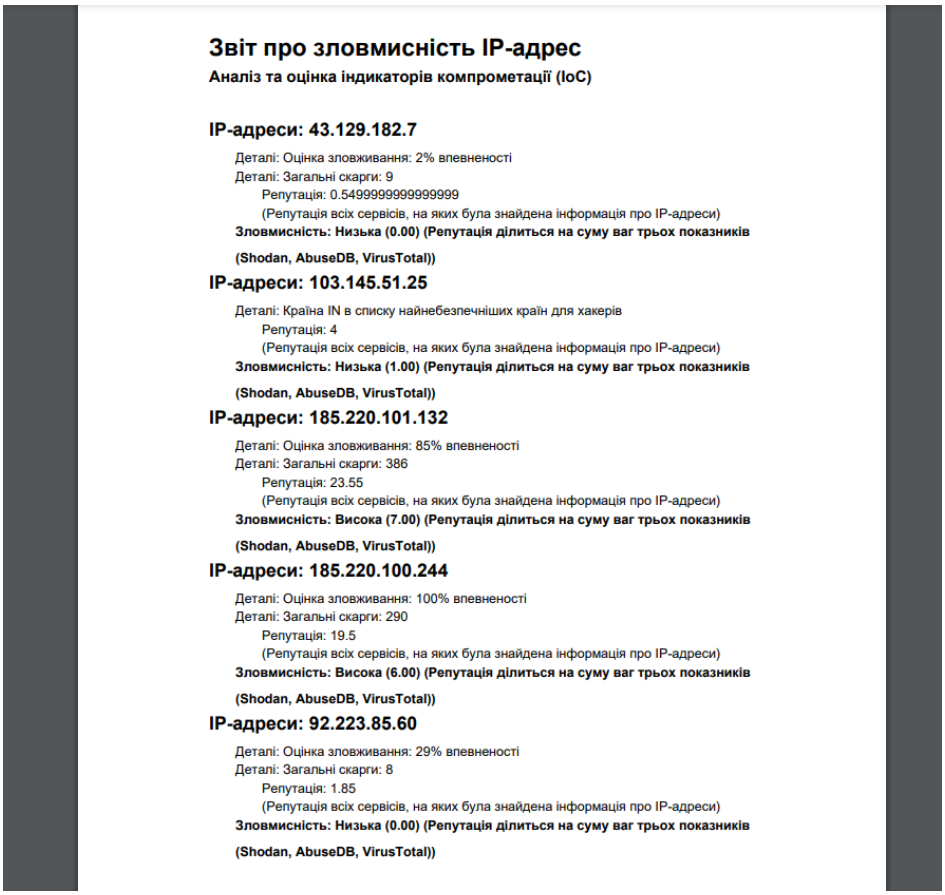


Рисунок 2.18 — Результат вибору «Створити звіт по IP-адресам»



Звіт про зловмисність IP-адрес

Аналіз та оцінка індикаторів компрометації (IoC)

IP-адреси: 43.129.182.7

Деталі: Оцінка зловживання: 2% впевненості
Деталі: Загальні скарги: 9
Репутація: 0.5499999999999999
(Репутація всіх сервісів, на яких була знайдена інформація про IP-адреси)
Зловмисність: Низька (0.00) (Репутація ділиться на суму ваг трьох показників
(Shodan, AbuseDB, VirusTotal))

IP-адреси: 103.145.51.25

Деталі: Країна IN в списку найнебезпечніших країн для хакерів
Репутація: 4
(Репутація всіх сервісів, на яких була знайдена інформація про IP-адреси)
Зловмисність: Низька (1.00) (Репутація ділиться на суму ваг трьох показників
(Shodan, AbuseDB, VirusTotal))

IP-адреси: 185.220.101.132

Деталі: Оцінка зловживання: 85% впевненості
Деталі: Загальні скарги: 386
Репутація: 23.55
(Репутація всіх сервісів, на яких була знайдена інформація про IP-адреси)
Зловмисність: Висока (7.00) (Репутація ділиться на суму ваг трьох показників
(Shodan, AbuseDB, VirusTotal))

IP-адреси: 185.220.100.244

Деталі: Оцінка зловживання: 100% впевненості
Деталі: Загальні скарги: 290
Репутація: 19.5
(Репутація всіх сервісів, на яких була знайдена інформація про IP-адреси)
Зловмисність: Висока (6.00) (Репутація ділиться на суму ваг трьох показників
(Shodan, AbuseDB, VirusTotal))

IP-адреси: 92.223.85.60

Деталі: Оцінка зловживання: 29% впевненості
Деталі: Загальні скарги: 8
Репутація: 1.85
(Репутація всіх сервісів, на яких була знайдена інформація про IP-адреси)
Зловмисність: Низька (0.00) (Репутація ділиться на суму ваг трьох показників
(Shodan, AbuseDB, VirusTotal))

Рисунок 2.19 – Приклад готового звіту

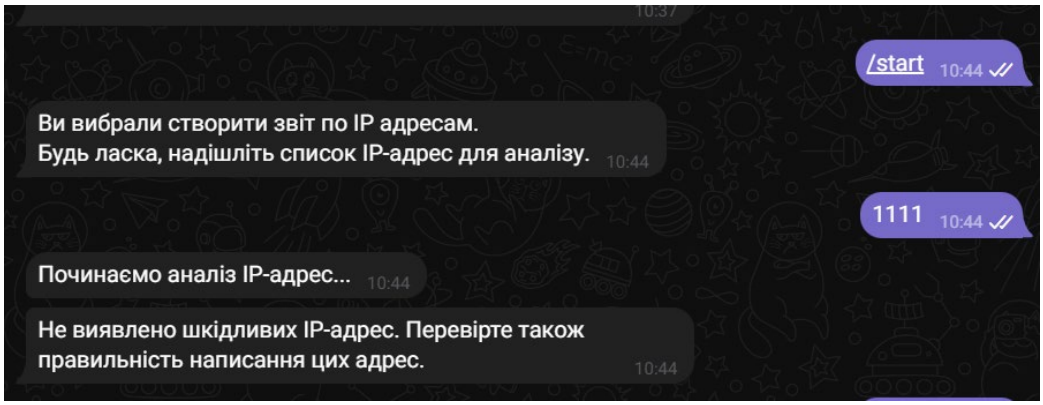


Рисунок 2.20 — Виведення повідомлення у разі нествореного звіту або всі адреси нешкідливі

Для вибору іншої опції, потрібно розпочати з команди /start. Наприклад, користувач вибрав опцію «Відправити повідомлення-звіт про домен». (рис 2.20) Після цього вибору, користувачу потрібно ввести домен для аналізу. Бот розпочне аналіз домену та надішле коротке повідомлення з виділеною інформацію, яка може вказувати на шкідливість домен. Якщо бот не знайшов інформації про домен або виявив домен повністю безпечним, то надсилає відповідне повідомлення.

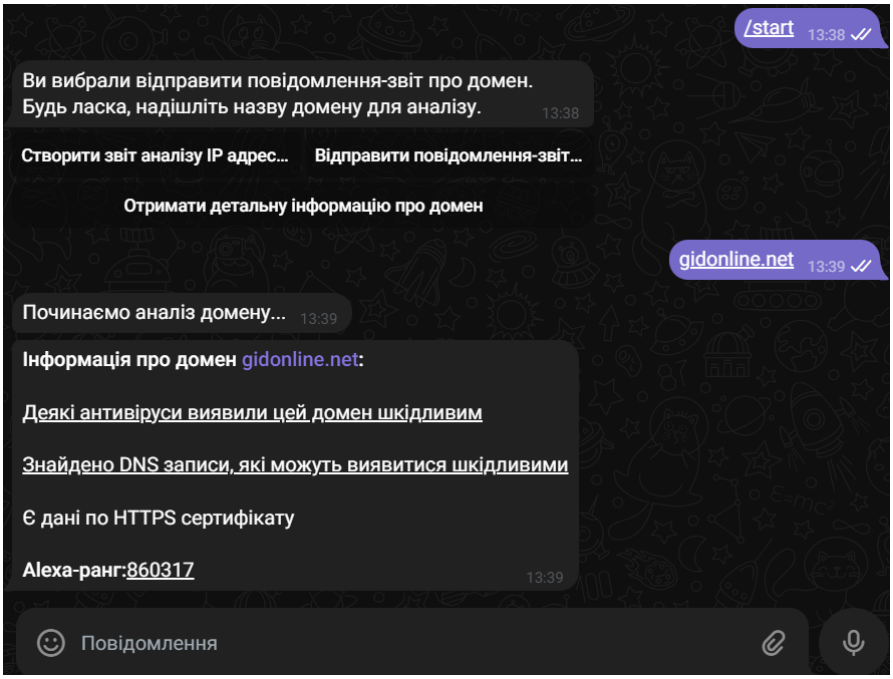


Рисунок 2.21 – Результат вибору «Відправити повідомлення-звіт про домен»

Для отримання більш детальної інформації про домен користувачу потрібно ропочати розмову командою /start та обрати опцію «Отримати детальну інформацію про домен». Результат перевірки буде надіслано повідомлення з детальною інформацією про домен.

Наприклад, в короткому форматі, бот може повідомити про наявність даних по HTTPS-сертифікату, а в більш розширеному форматі бот розписує інформацію про сертифікат (рис. 2.21) (основне ім’я, термін дії, серійний номер, алгоритм підпису)

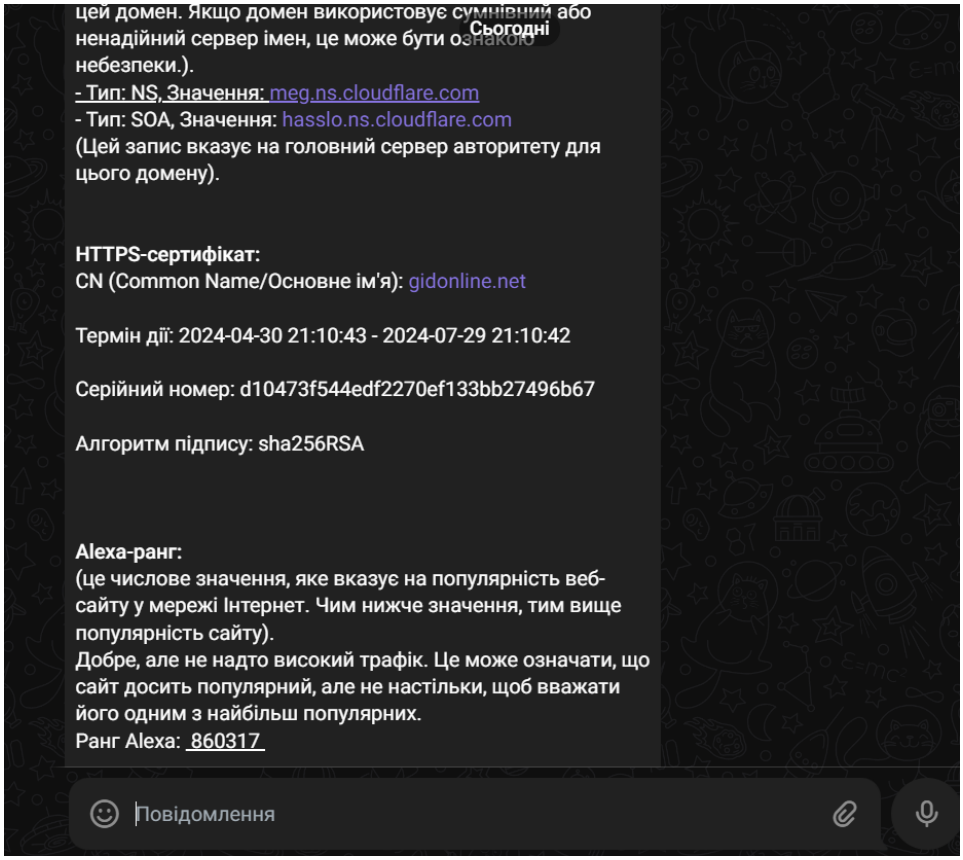


Рисунок 2.22 – Деталізовані дані по HTTPS-сертифікату

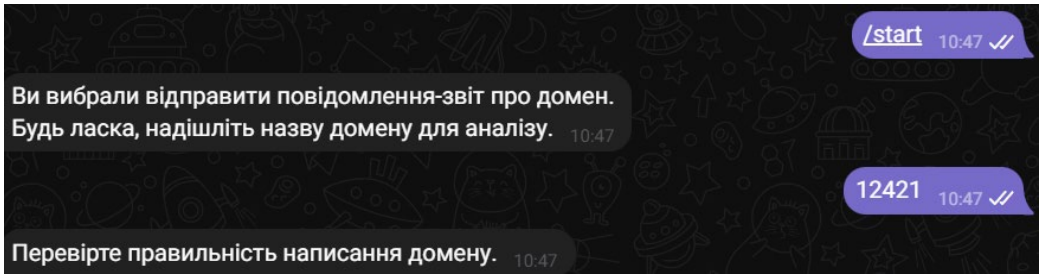


Рисунок 2.23 – Виведення повідомлення у разі неіснуючого домену

2.6 Організація тестування та налагодження програмного засобу

Фінальний етап проєкту включає тестування готового продукту. Тестування означає перевірку та оцінку функціональності програмного забезпечення згідно з його призначенням. Це допомагає виявити та вирішити можливі проблеми в роботі продукту. У таблиці 2.2 показано, як проводиться тестування завершеного продукту за допомогою набору тестових випадків.

Таблиця 2.2

Тестування чат-боту

Опис тесту	**Послідовність кроків відтворення**	**Очікуваний результат**	**Результат тестування**
Перевірка отримання файлу звіту по шкідливим IP-адресам	1. Запустити бота 2. Виконати команду /start 3. Обрати опцію «Створити звіт по IP-адресам» 4. Надіслати обрані IP-адреси Зачекати аналізу адрес 5. Отримати PDF файл у повідомленні від бота	Створення PDF файлу звіту та надсилання його користувачу	Пройдено
Перевірка на отримання даних про домен	1. Запустити бота 2. Виконати команду /start 3. Обрати опцію «Відправити повідомлення-звіт про домен» 4. Надіслати назву домену 5. Отримати повідомлення про аналіз домену	Надсилання повідомлення для користувача про аналіз домену	Пройдено

Перевірка відображення детальної інформації про домен	1. Виконати команду /start 2. Обрати опцію «Отримати детальну інформацію про домен» 3. Надіслати назву домену 4. Отримати повідомлення з детальною інформацією про домен	Надсилання повідомлення для користувача про аналіз домену у розширеному форматі	Пройдено

Всі тести пройдені успішно. Помилок під час тестування виявлено не було.

2.7 Рекомендації з використання та впровадженню програмного засобу

Цей бот розроблено для моніторингу безпеки доменів та IP-адрес. Він надає можливість користувачам отримувати інформацію про підозрілу активність за введеними даними та забезпечує функціонал пошуку та аналізу доменів чи IP-адрес.

Можливості та переваги розробленого Telegram-бота:

- зручний доступ до інформації про безпеку доменів та IP-адрес через месенджер Telegram, що дозволяє швидко та зручно отримувати актуальні дані;
- можливість отримати детальний звіт про підозрілу активність, що допомагає уникнути можливих загроз;
- функціонал пошуку та аналізу доменів чи IP-адрес, що допомагає з’ясувати їхню безпеку та стабільність;
- інтуїтивно зрозумілий і простий у використанні список команд для швидкого доступу до потрібної інформації.

Для забезпечення коректної роботи бота та задоволення потреб користувачів, необхідно мати стабільне Інтернет-підключення та використовувати операційну систему, яка підтримує месенджер Telegram, такі як Windows, macOS, Linux, Android або iOS.

ВИСНОВКИ

Ця кваліфікаційна робота була спрямована на розробку чат-бота для месенджера Telegram, який має за мету забезпечити користувачам зручний доступ до моніторингу інформації про зловмисність доменних імен та IP-адрес.

В процесі виконання кваліфікаційної роботи було виконано ряд завдань:

- розглянуто принцип організації доменних імен та IP адрес;
- проаналізовано основні методи виявлення загроз доменних імен та IP адрес;
- розглянуто основні сервіси, які використовуються для аналізу на зловмисність IP-адрес та доменних імен;
- розглянуто принципи функціонування ботів у месенджерах та існуючі чат-боти, які використовуються для моніторингу загроз доменних імен та IP адрес, проаналізовано їхні переваги й недоліки;
- обрано технології та середовища розробки власного бота;
- розроблено Telegram бот для отримання інформації про зловмисність доменних імен та IP адрес;
- проведено тестування Telegram бота.

Завдяки розробленому програмному продукту користувачі мають можливість зручно моніторити безпеку доменів та IP-адрес. Цей продукт робиться доступним для кожного користувача платформи Telegram.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Все, що потрібно знати про IP-адресу. *Макснет – Гігабітний інтернет-провайдер та оператор зв'язку для дому та бізнесу*. URL: https://maxnet.ua/blog/vse-chto-nuzhno-znat-ob-ip-adrese/ (дата звернення: 09.04.2024).
2. Дмитро. Telegram-боти на Python: огляд п'яти найкращих фреймворків/бібліотек. *Друкарня*. URL: https://drukarnia.com.ua/articles/telegram-boti-na-python-oglyad-p-yati-naikrashikh-freimvorkiv-bibliotek-L7UA7 (дата звернення: 09.04.2024).
3. ТОП сервісів для моніторингу доступності веб-сайтів і серверів | KR. Laboratories. *KR. Laboratories*. URL: https://kr-labs.com.ua/blog/top-uptime-monitoring-tools/ (дата звернення: 09.04.2024).
4. Учасники проектів Вікімедіа. Інтернет-безпека – Вікіпедія. *Вікіпедія*. URL: https://uk.wikipedia.org/wiki/Інтернет-безпека (дата звернення: 09.04.2024).
5. Учасники проектів Вікімедіа. Клієнт-серверна архітектура – Вікіпедія. *Вікіпедія*. URL: https://uk.wikipedia.org/wiki/Клієнт-серверна_архітектура (дата звернення: 09.04.2024).
6. Учасники проектів Вікімедіа. DNSSEC – Вікіпедія. *Вікіпедія*. URL: https://uk.wikipedia.org/wiki/DNSSEC (дата звернення: 09.04.2024).
7. Як створити чат-бота для сайту: 7 корисних сервісів. *Блог HelpCrunch*. URL: https://helpcrunch.com/blog/uk/yak-stvoryty-chat-bota-dlia-saitu/ (дата звернення: 09.04.2024).
8. Top Website Malware Scanners for Better Online Security. *Geekflare*. URL: https://geekflare.com/best-website-malware-scanners/ (date of access: 09.04.2024).
9. Best AI Chatbot Platforms: A Comprehensive Guide (2024) | Botpress Blog. *Botpress | the Generative AI platform for ChatGPT Chatbots*. URL: https://botpress.com/blog/9-best-ai-chatbot-platforms (date of access: 09.04.2024).
10. amoCRM. Why Messengers are Becoming More Popular Than Social Media. *Medium*. URL: https://medium.com/@amocrmglobalmarketing/why-

messengers-are-becoming-more-popular-than-social-media-3bf2ec80e146 (date of access: 09.04.2024).

11. Contributors to Wikimedia projects. Messenger (software) - Wikipedia. *Wikipedia, the free encyclopedia*. URL: https://en.wikipedia.org/wiki/Messenger_(software) (date of access: 09.04.2024).
12. Contributors to Wikimedia projects. Telegram (software) - Wikipedia. *Wikipedia, the free encyclopedia*. URL: https://en.wikipedia.org/wiki/Telegram_(software) (date of access: 09.04.2024).
13. Cyber Threat Intelligence and Cyber Risk Quantification Company | ThreatConnect. *ThreatConnect*. URL: https://threatconnect.com/ (date of access: 09.04.2024).
14. Detection and prevention of DNS anomalies | Infosec. *Cybersecurity Training & Certifications | Infosec*. URL: https://www.infosecinstitute.com/resources/malware-analysis/detection-prevention-dns-anomalies/ (date of access: 09.04.2024).
15. DomainTools - The first place to go when you need to know. *DomainTools | Start Here. Know Now.* URL: https://www.domaintools.com/ (date of access: 09.04.2024).
16. GreyNoise is the source for understanding internet noise. *GreyNoise is the source for understanding internet noise*. URL: https://www.greynoise.io/ (date of access: 09.04.2024).
17. HOSTiQ.ua. *Хостинг в Украине от HOSTiQ – лучший украинский хостинг*. URL: https://hostiq.ua/ukr/info/what-is-domain/ (date of access: 09.04.2024).
18. Intrusion Detection Systems (IDS) та Intrusion Prevention Systems (IPS) – UA5.org. *UA5.org – Матеріали з інформаційних технологій*. URL: https://ua5.org/protect/3070-intrusion-detection-systems-ids-ta-intrusion-prevention-systems-ips.html (дата звернення: 09.04.2024).
19. SecurityTrails. *securitytrails.com*. URL: https://securitytrails.com/ (date of access: 09.04.2024).

20. Shodan. *shodan.io*. URL: https://www.shodan.io/ (date of access: 09.04.2024).
21. Top 7 Threat Intelligence Platforms & Tools | eSP. *eSecurity Planet*. URL: https://www.esecurityplanet.com/products/threat-intelligence-platforms/ (date of access: 09.04.2024).
22. Types of Domain Name Attacks - DNS. -. URL: https://news.registro.gt/en/2023/09/28/types-of-domain-name-attacks/ (date of access: 09.04.2024).
23. VirusTotal. *VirusTotal*. URL: https://www.virustotal.com/gui/home/upload (date of access: 09.04.2024).
24. What Is a Chatbot?. *Oracle | Cloud Applications and Cloud Platform*. URL: https://www.oracle.com/chatbots/what-is-a-chatbot/ (date of access: 09.04.2024).
25. What is a DNS Firewall? Benefits of DNS Firewall | Fortinet. *Fortinet*. URL: https://www.fortinet.com/resources/cyberglossary/dns-firewall (date of access: 09.04.2024).
26. What is Whois Information and Why is it Valuable? - DomainTools | Start Here. Know Now. *DomainTools | Start Here. Know Now.* URL: https://www.domaintools.com/support/what-is-whois-information-and-why-is-it-valuable/ (date of access: 09.04.2024).
27. Опис продукту JetBrains PyCharm - ITPRO.UA. *ITPRO*. URL: https://itpro.ua/product/jetbrains-pycharm/?tab=description (дата звернення: 26.04.2024).
28. Учасники проектів Вікімедіа. Python – Вікіпедія. *Вікіпедія*. URL: https://uk.wikipedia.org/wiki/Python (дата звернення: 26.04.2024).
29. python-telegram-bot. *PyPI*. URL: https://pypi.org/project/python-telegram-bot/ (date of access: 26.05.2024).
30. Учасники проектів Вікімедіа. Webhook – Вікіпедія. *Вікіпедія*. URL: https://uk.wikipedia.org/wiki/Webhook (дата звернення: 26.04.2024).
31. AbuseIPDB. *https://www.abuseipdb.com/*. URL: https://www.abuseipdb.com/ (date of access: 26.05.2024).

32. BestHostingBot. *https://besthosting.ua/ua/news/147*. URL: https://besthosting.ua/ua/news/147 (date of access: 27.05.2024).
33. Учасники проектів Вікімедіа. PostgreSQL – Вікіпедія. *Вікіпедія*. URL: https://uk.wikipedia.org/wiki/PostgreSQL (дата звернення: 27.05.2024).
34. Учасники проектів Вікімедіа. MySQL – Вікіпедія. *Вікіпедія*. URL: https://uk.wikipedia.org/wiki/MySQL (дата звернення: 27.05.2024).
35. Учасники проектів Вікімедіа. Heroku – Вікіпедія. *Вікіпедія*. URL: https://uk.wikipedia.org/wiki/Heroku (дата звернення: 27.05.2024).
36. Учасники проектів Вікімедіа. Amazon Web Services – Вікіпедія. *Вікіпедія*. URL: https://uk.wikipedia.org/wiki/Amazon_Web_Services (дата звернення: 27.05.2024).

ДОДАТОК А

Технічне завдання

Тематика

Чат-бот повинен містити функціонал для моніторингу безпеки доменних імен та IP-адрес у реальному часі.

Цільова аудиторія

Адміністратори мереж, спеціалісти з кібербезпеки, користувачі Telegram.

Платформа

Чат-бот повинен працювати на всіх доступних платформах, на яких працює Telegram.

Основні функції

- перевірка на відкриття шкідливих портів, які можуть бути використані для несанкціонованого доступу або атак;
- перевірка чи не минув термін дії сертифікату-SSL, що може свідчити про можливу вразливість системи;
- оцінка рівня впевненості у визначенні ідентичності IP-адреси та її можливої шкідливості;
- аналіз кількості скарг на дану IP-адресу, що може вказувати на її зловживання;
- оцінка репутації IP-адреси на інших сайтах щодо її шкідливості або небезпечності;
- перевірка на низький рівень небезпечності IP-адреси;
- аналіз рівня підозрілості IP-адреси на можливу шкідливість або зловживання;
- оцінка ризику згідно з тим, чи IP-адреса належить країнам, відомим своєю активністю у сфері кіберзлочинності;

- надання актуальної інформації про безпеку доменних імен та IP-адрес на основі введених даних;
- надання детальної інформації про конкретний домен;
- можливість ручного введення даних для перевірки;
- надання користувачу можливості отримувати зведені звіти у форматі PDF про стан безпеки IP-адрес на регулярній основі.

Дизайн

Бот повинен мати зрозумілий та логічний функціонал, доступний для кожного користувача. Вивід інформації повинен бути у дружньому тоні та містити візуальні підказки.

ДОДАТОК Б

Керівництво користувачу

Для запуску бота, користувачу необхідно ввести команду «/start» (рис. Б.1). Після цього користувачу надається вибір опцій, які реалізовано у вигляді кнопок у повідомленні. Користувач може обрати:

- створити звіт аналізу на зловмисність списку IP-адрес;
- відправити повідомлення-звіт про вказаний домен;
- отримати детальну інформацію про вказаний домен.

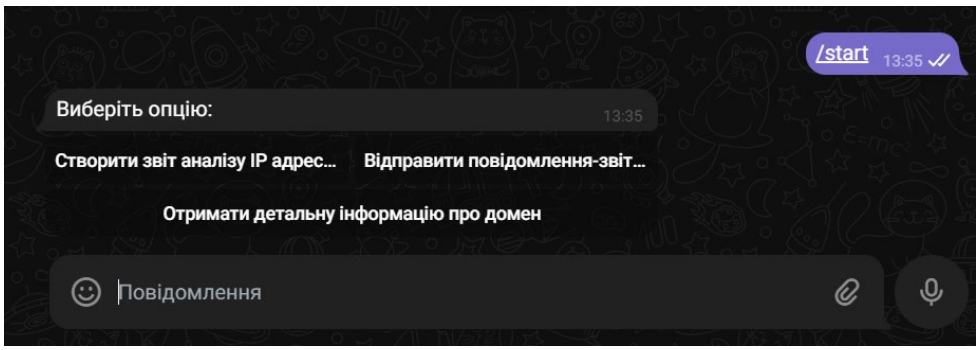


Рисунок Б.1 – Виконання команди «/start»

При виборі звіту по IP-адресам, користувачу потрібно ввести список адрес для перевірки на зловмисність. Адреси слід вводити через пробіл, і можна ввести адреси як в протоколі IPv6, так і в протоколі IPv4. Для того щоб бот перевірив кожну адресу зі списку, потрібно зачекати певний проміжок часу, який залежить від кількості IP-адрес. Після завершення перевірки бот надсилає готовий звіт користувачу. Звіт формується у форматі PDF та надсилається користувачу (рис. Б.2).

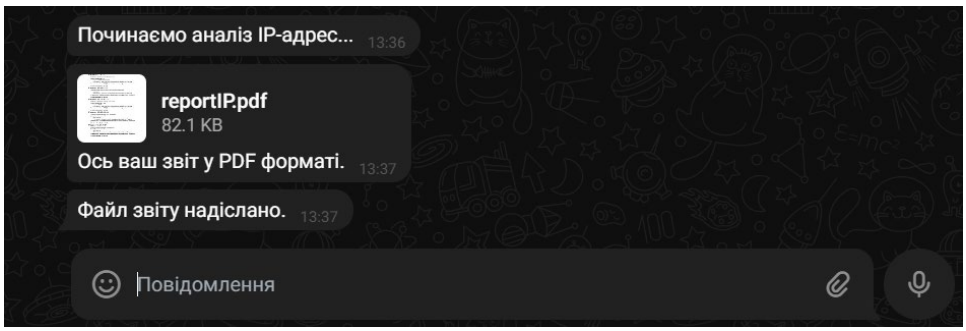


Рисунок Б.2 – Отримання файлу звіту у форматі PDF

Щоб вибрати іншу опцію, необхідно почати з команди «/start». При виборі опції «Відправити повідомлення-звіт про домен» (рис. Б.4). Користувачу потрібно ввести домен для аналізу. Бот розпочне аналіз домену та надішле коротке повідомлення з інформацією, яка може вказувати на шкідливість домену.

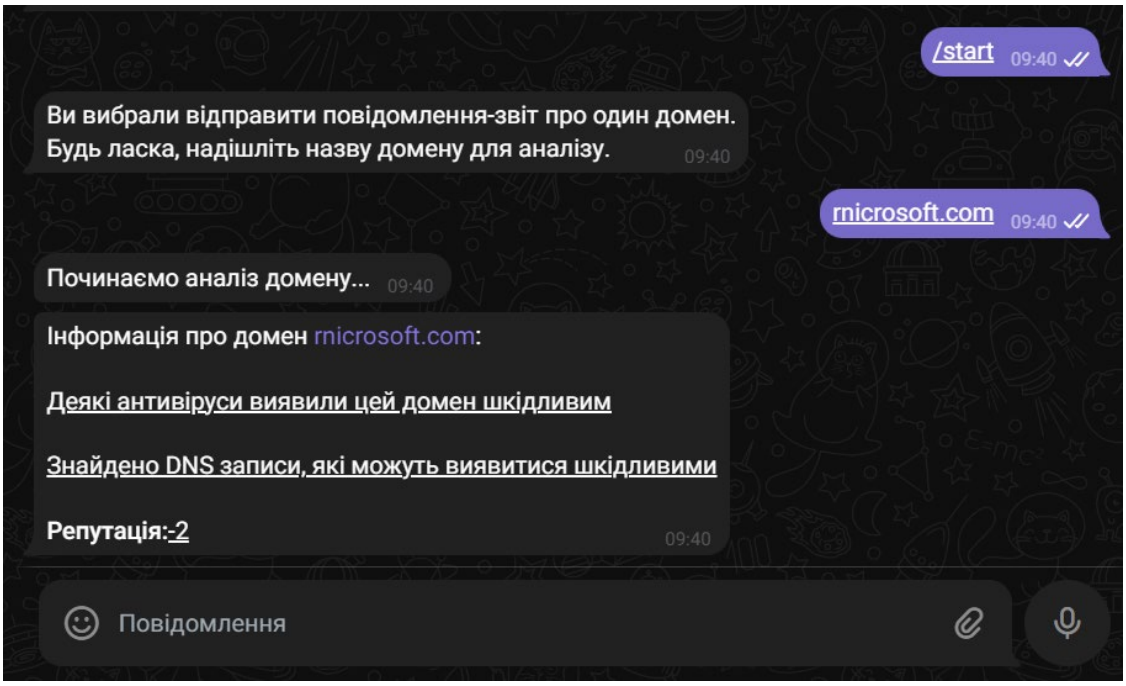


Рисунок Б.3 – Отримання основної інформації у вигляді короткого повідомлення про домен

Для отримання більш детальної інформації про домен користувачу потрібно ропочати розмову командою «/start» та обрати опцію «Отримати детальну інформацію про домен». Результат перевірки буде надіслано повідомлення з детальною інформацією про домен (рис. Б.5).

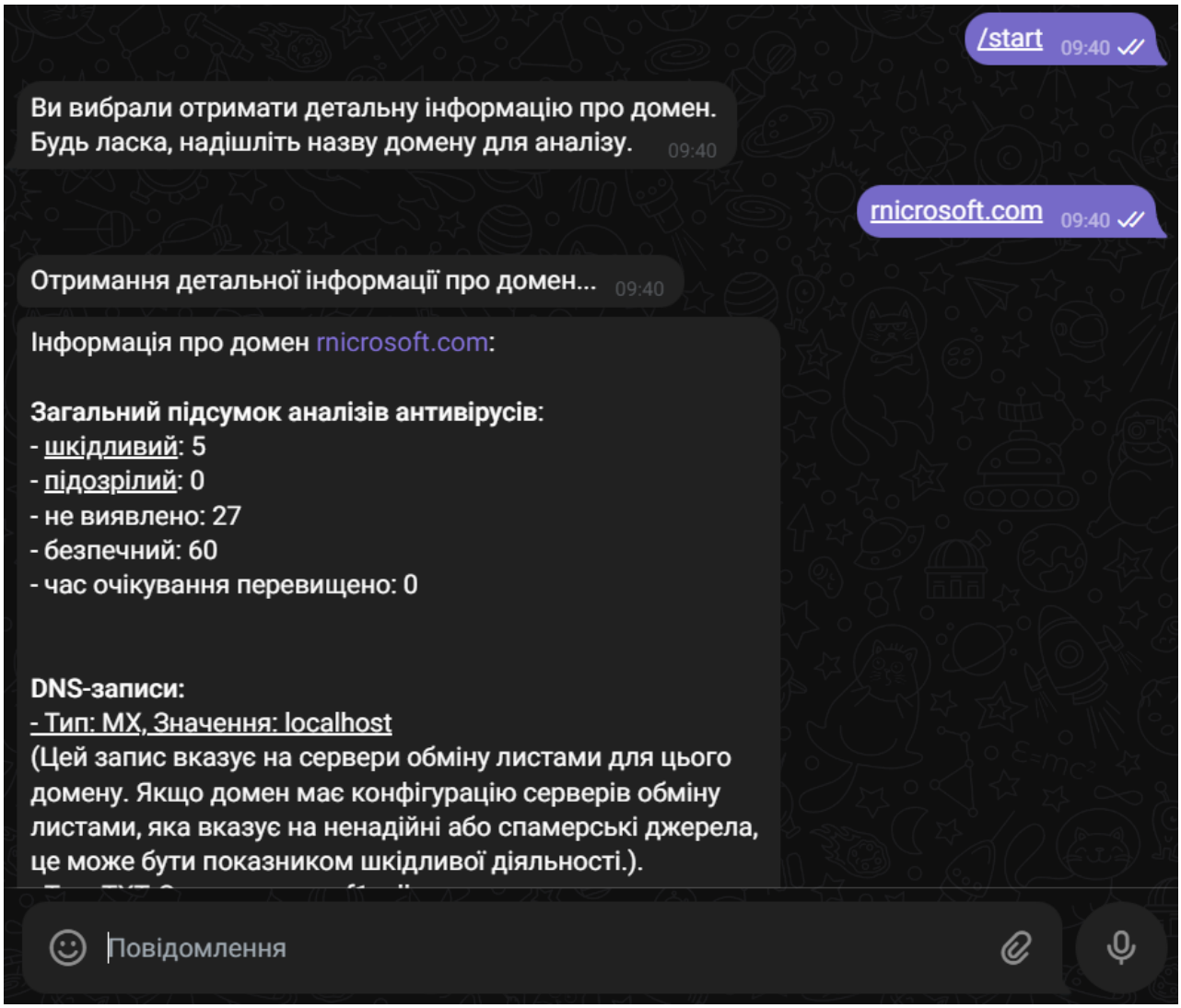


Рисунок Б.4 – Отримання детальної інформації про домен

АНОТАЦІЯ

Омельчук А.А. Моніторинг зловмисності доменних імен та IP-адрес засобами платформи Telegram. Рукопис

Кваліфікаційна робота на здобуття освітнього ступеня «бакалавр» за спеціальністю 122 Комп'ютерні науки. Волинський національний університет імені Лесі Українки, Луцьк, 2024 р.

Робота містить інформацію про дослідження технологій для створення сервісу моніторингу безпеки доменних імен та IP-адрес. Проведено аналіз існуючих рішень та підходів до розробки такого сервісу. Ці методи були використані під час створення програмного продукту. Програма є комплексною розробкою та складається з трьох основних компонентів: бази даних, клієнтської частини та серверної частини.

Компонент бази даних забезпечує зберігання та організацію великої кількості даних, необхідних для моніторингу. Це дозволяє ефективно здійснювати пошук та управління інформацією, забезпечуючи безперебійну роботу користувачів. Клієнтська частина програмного продукту була ретельно розроблена для забезпечення інтуїтивно зрозумілого та візуально привабливого інтерфейсу для користувачів. Тим часом серверний компонент відігравав ключову роль в обробці запитів користувачів, обробці даних та забезпеченні зв'язку між клієнтом і базою даних.

Протягом усього процесу дослідження та розробки було впроваджено процедури тестування та забезпечення якості для гарантування надійності та стабільності програмного продукту. Було створено комплексний набір тестових сценаріїв для оцінки функціональності, продуктивності та швидкості реагування сервісу моніторингу.

Таким чином, у роботі досліджено різні технології та методології для створення сервісу, що полегшує моніторинг безпеки доменних імен та IP-адрес. Завдяки ретельному аналізу існуючих рішень та використанню відповідних

інструментів, таких як Python, telegram.ext, Shodan, VirusTotal та PyCharm, було розроблено комплексний та зручний для користувача програмний продукт.

Ключові слова: месенджер, чат-бот, IP-адреса, домен, python, telegram.ext, shodan, abusedb, virustotal.