Міністерство освіти і науки України

Харківський національний університет імені В.Н. Каразіна

Навчально-науковий інститут комп'ютерних наук та штучного інтелекту

Спеціальність 125 «Кібербезпека та захист інформації»

Освітня програма «Безпека інформаційних і комунікаційних систем»

«Допущено до захисту»

В.о. зав. кафедрою БІСТ

Марина ЄСІНА

_____

« » 2024 р.

Пояснювальна записка

до кваліфікаційної роботи магістра

на тему: «Аналіз сучасних методів і технологій DNS фільтрації трафіку, як складової безпеки сучасних інформаційних систем»

оцінка « »

Голова ЕК

Олександр ЛЕМЕШКО _____

Керівник к. т. н. доцент каф.

Малахов С. В.

Рецензент PhD, доцент кафедри інтелектуальних програмних систем і технологій

Родінко М. Ю.

Виконавець : студент групи КБ-61

Чепель Д.О.

Харків – 2024

РЕФЕРАТ

Пояснювальна записка містить 62 сторінки, 34 рисунки, 2 таблиці, 1 додаток, 42 джерела.

Метою дипломної роботи є покращення параметрів захисту сучасних інформаційних систем за рахунок впровадження і вдосконалення підсистеми моніторингу та аналізу DNS-запитів для завчасного виявлення аномалій DNS трафіку, що можуть становити загрозу безпеки для діючих інформаційних систем й мережевого устаткування.

Об’єкт дослідження: - процеси мережевого моніторингу, обробки та відображення інформації, щодо поточного стану роботи DNS сервісів (служб).

Предмет дослідження: - процедури тестування і відображення інтегрованої інформації про поточний стан діючих DNS серверів для визначення їх продуктивності та виявлення ознак відповідних аномалій (атаки і робота алгоритмів генерації доменів (DGA)).

Основними методами дослідження є аналіз профільної інформації, комп’ютерне моделювання, синтез програмної моделі прототипу тестового застосунку та узагальнення отриманих результатів тестувань.

У роботі досліджено питання покращення ефективності DNS-фільтрації. Запропоновано аналіз відомостей, щодо сучасних методів фільтрації DNS та їх застосування в умовах зростаючих кіберзагроз. Систематизовано типові сценарії атак на DNS, із визначенням їх основних механізмів реалізації. Розглянуто методи покращення результативності DNS-фільтрації за допомогою використання зон політик реагування (RPZ), каналів розвідки загроз, шифрування DNS-трафіку та методів виявлення DGA і серверів управління відповідних ботнетів. Розроблено програмний інструмент для моніторингу доступності та тестування часових параметрів хмари DNS-серверів, для різних типів протоколів шифрування DNS.

Тестова програмна модель дозволяє збирати та аналізувати статистику, необхідну для виявлення аномалій у DNS-трафіку. Узагальнено перспективи подальшого розвитку систем DNS-фільтрації з урахуванням вимог до кібербезпеки в умовах сучасних інформаційних систем.

Результати роботи можуть бути застосовані в галузі кібербезпеки для підвищення ефективності систем фільтрації DNS-трафіку та моніторингу мережевих аномалій. Крім того, розроблена програма може знайти застосування у дослідницьких задачах, пов’язаних з аналізом роботи DNS-серверів в умовах підвищених навантажень чи атак.

Ключові слова: DNS, DGA, DoT, DoH, DoQ, RPZ, ІНФОРМАЦІЙНА БЕЗПЕКА, ЗАГРОЗИ БЕЗПЕКИ, ФІЛЬТРАЦІЯ ТРАФІКУ, БОТНЕТ.

ABSTRACT

The explanatory note contains 62 pages, 32 figures, 2 tables, 1 annex and 42 sources.

The aim of the thesis is to enhance the security parameters of modern information systems by improving and implementing a subsystem for monitoring and analyzing DNS queries to detect DNS traffic anomalies in advance that may pose security threats to operational information systems and network infrastructure.

Object of research: - network monitoring processes, processing, and visualization of information related to the current state of DNS services.

Subject of research: - procedures for testing and presenting integrated information about the current state of DNS servers to determine their performance and identify signs of anomalies (such as attacks and domain generation algorithms (DGA)).

The primary research methods include analyzing domain-specific information, computer modeling, synthesizing a prototype software model, and summarizing the test results.

The study explores the issue of improving the efficiency of DNS filtering. It proposes a literature review on modern DNS filtering methods and their application in the context of increasing cybersecurity threats. Typical DNS attack scenarios are systematized, with an emphasis on identifying their key implementation mechanisms. Methods to enhance the effectiveness of DNS filtering are considered, including the use of Response Policy Zones (RPZ), threat intelligence feeds, DNS traffic encryption, and techniques for detecting DGAs (Domain Generation Algorithms) and command-and-control servers of corresponding botnets. A software tool was developed for monitoring the availability and testing the timing parameters of DNS server clouds, supporting various DNS encryption protocols. The test software model enables the collection and analysis of statistics necessary for detecting anomalies in DNS traffic.

The prospects for further development of DNS filtering systems are summarized, considering cybersecurity requirements in modern information systems.

The results of the study can be applied in the field of cybersecurity to improve the efficiency of DNS traffic filtering systems and network anomaly monitoring. Additionally, the developed software can be utilized in research tasks related to analyzing the performance of DNS servers under increased load or attack conditions.

Keywords: DNS, DGA, DoT, DoH, DoQ, RPZ, INFORMATION SECURITY, SECURITY THREATS, TRAFFIC FILTERING, BOTNET.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА СИМВОЛІВ 8
ВСТУП 9
1 ОГЛЯД СИСТЕМИ ДОМЕНИИХ ІМЕН ТА ТИПОВИХ АТАК НА НЕЇ 11
1.1 Система доменних імен 11
1.2 Типові атаки на систему доменних імен 15
1.2.1 DNS Spoofing 15
1.2.2 DNS Amplification 16
1.2.3 DNS Rebinding 18
1.2.4 Typosquatting 20
2 ДОСЛІДЖЕННЯ ІСНУЮЧИХ МЕХАНИЗМІВ ФІЛЬТРАЦІЇ DNS ТРАФІКУ ТА МОЖЛИВОСТІ ПОКРАЩЕННЯ ЇЇ ЕФЕКТИВНОСТІ 22
2.1 Напрями покращення інформативності процедур контролю DNS трафіку 23
2.1.1 Використання каналів розвідки загроз 23
2.1.2 Впровадження інструментів виявлення серверів управління ботнетів 24
2.1.3 Можливості шифрування трафіку DNS 25
2.1.4 Запровадження зон політик реагування 26
2.1.5 Застосування механізмів виявлення роботи алгоритмів генерації доменів ... 27
2.2 Узагальнення досвіду, стосовно практичних реалізацій процедур фільтрації DNS 29
2.2.1 Канали розвідки загроз 29
2.2.2 Виявлення активності ботнетів 30
2.2.3 Шифрування DNS 32
2.2.4 Узагальнення проблематики адміністрування RPZ 33
2.2.5 Специфіка виявлення алгоритмів генерації доменів (DGA) 34

3 РОЗРОБКА ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТЕСТОВОГО ПРОТОТИПУ ДОДАТКУ МОНІТОРИНГУ ПОТОЧНОГО СТАНУ ХМАРИ DNS-СЕРВЕРІВ 39
3.1 Загальна концепція прототипу дослідної програми 39
3.2 Функціонал тестової версії дослідної програми 41
3.3 Графічний (користувацький) інтерфейс програми 42
3.4 Реалізація основних компонентів дослідної програми 45
3.4.1 Локальна складова ПЗ дослідного прототипу 45
3.4.2 Хмарна складова дослідної програми 50
3.5 Експорт результатів тестування 50
4 АНАЛІЗ ТА УЗАГАЛЬНЕННЯ РЕЗУЛЬТАТІВ ВИМІРЮВАНЬ ДЛЯ РІЗНИХ ПАРАМЕТРІВ ДОСЛІДНИЦЬКОЇ ПРОГРАМИ 52
4.1 Результати доступності і часу реакції DNS серверів різних доменних зон 52
4.2 Результати оцінки доступності і часу реакції DNS серверів в залежності від поточного часу доби (в системі «Сервер DNS – Тестер») 57
ВИСНОВКИ 61
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ 63
ДОДАТОК А 68

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА СИМВОЛІВ

DNS	–	Система доменних імен
ІБ	–	Інформаційна безпека
ІКС	–	Інформаційно-комунікаційна система
ПЗ	–	Програмне забезпечення
OSI	–	Модель взаємодії відкритих систем
DDoS	–	Розподілена відмови в обслуговуванні
NTP	–	Мережевий протокол часу
DoT	–	Протокол безпечного передавання DNS через TLS
DoH	–	Протокол безпечного передавання DNS через HTTPS
DoQ	–	Протокол передавання DNS через QUIC
RPZ	–	Зона політики реагування
DGA	–	Алгоритм генерації доменів
AI	–	Штучний інтелект
ML	–	Машинне навчання
CNN	–	Згорткова нейронна мережа
RNN	–	Рекурентна нейронна мережа
LSTM	–	Довга короткочасна пам’ять (архітектура нейронної мережі)
DL	–	Глибинне навчання
VT	–	Технологія віртуалізації
IoT	–	Інтернет речей

ВСТУП

Технологія зіставлення доменних імен - *Domain Name System (DNS)* із їх числовими *IP*-адресами, є найважливішим механізмом сучасного Інтернет, виступаючи в якості умовного «посередника», який перетворює «зручні» для користувачів доменні імена в їх IP-адреси [1]. Система із розгалужених DNS серверів не тільки спрощує процес навігації в Інтернеті але й забезпечує ефективне і точне з'єднання, як між користувачами, так і «місцями призначення» їх пошукових запитів (інформаційними ресурсами) в Інтернеті. Нажаль, сучасні мережеві зловмисники знаходять способи використання *DNS* служб для реалізації різних варіантів атак, причому як окремо, так і в якості складового елементу при реалізації багатоходових - інтегрованих атак [2-3]. Тому, на поточний момент фільтрація DNS трафіку є невід'ємною складовою заходів з інформаційної безпеки (ІБ) для переважної більшості сучасних інформаційно-комунікаційних систем (ІКС) Інструменти управління й фільтрації DNS забезпечують контроль доступу до веб-ресурсів, захист від шкідливого програмного забезпечення (ПЗ) і надає можливість гнучкого впровадження потрібних політик безпеки на рівні мережі (*для моделі OSI*) [4].

Актуальність теми: Наявність інструментів, що здатні контролювати і виявляти аномалії у трафіку DNS-запитів, є важливою складовою у загальній системі мережевої безпеки та покращують поточний стан інформаційної безпеки сучасних інформаційних систем.

Мета роботи: Покращення параметрів захисту сучасних інформаційних систем за рахунок вдосконалення і впровадження підсистеми моніторингу та аналізу DNS-запитів для завчасного виявлення аномалій DNS трафіку, що можуть

становити загрозу безпеки для діючих інформаційних систем й мережевого устаткування.

Результати роботи можуть бути застосовані в галузі кібербезпеки для підвищення ефективності систем фільтрації DNS-трафіку та моніторингу мережевих аномалій. Крім того, програма може знайти застосування у дослідницьких задачах, пов’язаних з аналізом роботи DNS-серверів в умовах підвищених навантажень чи атак.

1 ОГЛЯД СИСТЕМИ ДОМЕНИИХ ІМЕН ТА ТИПОВИХ АТАК НА НЕЇ

1.1 Система доменних імен

Система доменних імен (DNS) відіграє ключову роль у функціонуванні та масштабованості мережі Інтернет, оскільки майже кожен інший протокол залежить від вирішення домену *DNS* для своєї коректної роботи. *DNS* є одним з небагатьох протоколів, що складають умовне «ядро» Інтернету. В загальному випадку, *DNS* використовується переважно для перетворення «зручних» для читання людиною доменних імен у їх цифрові IP-адреси

DNS був розроблений у 1983 році Полом Мокапетрісом для усунення обмежень попередньої файлової системи, що складалася з файлу HOSTS.TXT, яка використовувалася в ARPANET, вимагала ручного оновлення та не була масштабованою зі збільшенням кількості хостів. Оригінальні специфікації DNS були формалізовані в RFC 1034 і RFC 1035, опублікованих у 1987 році. Подальші оновлення, такі як DNSSEC, були спрямовані на усунення проблем із безпекою та забезпечення цілісності даних [5].

Для пошуку потрібного мережевого домену клієнт/користувач надсилає *DNS*-запит до відповідного рекурсивного *DNS* серверу, який, зазвичай, надається поточним інтернет-провайдером та має можливості розпізнавання і кешування (*тимчасового зберігання*) доменних імен. У випадку, коли використовуваний сервер доменних імен не має у своєму кеші необхідних відомостей, то він звертається до кількох інших – «зовнішніх» *DNS* серверів, які зберігають розподілену базу даних доменних імен та їх відповідні IP-адреси. Таким чином, у пошуках потрібного мережевого домену, кожний умовний *DNS* запит транслюється через певні сегменти ієрархічної мережі із довірених серверів імен, доки не знайде потрібну відповідь

(*зіставлення*) та не надішле її клієнту (користувачеві). Отримавши потрібну *IP*-адресу, шукач необхідного інформаційного ресурсу може використовувати її для підключення до хосту призначення [2,4].

DNS організовано ієрархічно, з кореневими серверами на верхньому рівні, за якими йдуть сервери верхнього рівня (наприклад, *.com* або *.org*), а потім авторитетні сервери імен, які зберігають фактичні записи DNS.

Виділяються наступні рівні доменів:

- Домен кореневого рівня:

Найвищий рівень в ієрархії DNS, позначений крапкою ("."), слугує відправною точкою для всіх вирішень доменних імен. Він містить посилання на всі домени верхнього рівня.

- Домени верхнього рівня:

Розташовані безпосередньо під кореневим доменом. До них належать загальні домени верхнього рівня, такі як *.com, .org і .net*, а також домени верхнього рівня з кодом країни, такі як *.uk* і *.ua*. Домени верхнього рівня класифікують на основі їх призначення або географічного походження.

- Домени другого рівня:

Розташовані під доменами верхнього рівня. Зазвичай представляють конкретні організації або проекти. Наприклад, у домені *example.com* компонент *example* є доменом другого рівня.

- Субдомени:

Це підрозділи доменів другого рівня, що забезпечують подальшу категоризацію або вказують на специфічні служби. Наприклад, у www.example.com, компонент www є субдоменом домену другого рівня example.com.

- Хости:

Найнижчий рівень у структурі DNS складається з окремих хостів або служб, які ідентифікуються додатковими мітками. Наприклад, *mail.www.example.com* вказує на певну службу (пошту), пов’язану з субдоменом *www* домену *example.com* [6].

Цю ієрархію можна візуалізувати як перевернуте дерево з різними рівнями, що представляють різні компоненти доменних імен (див. рис. 1.1).

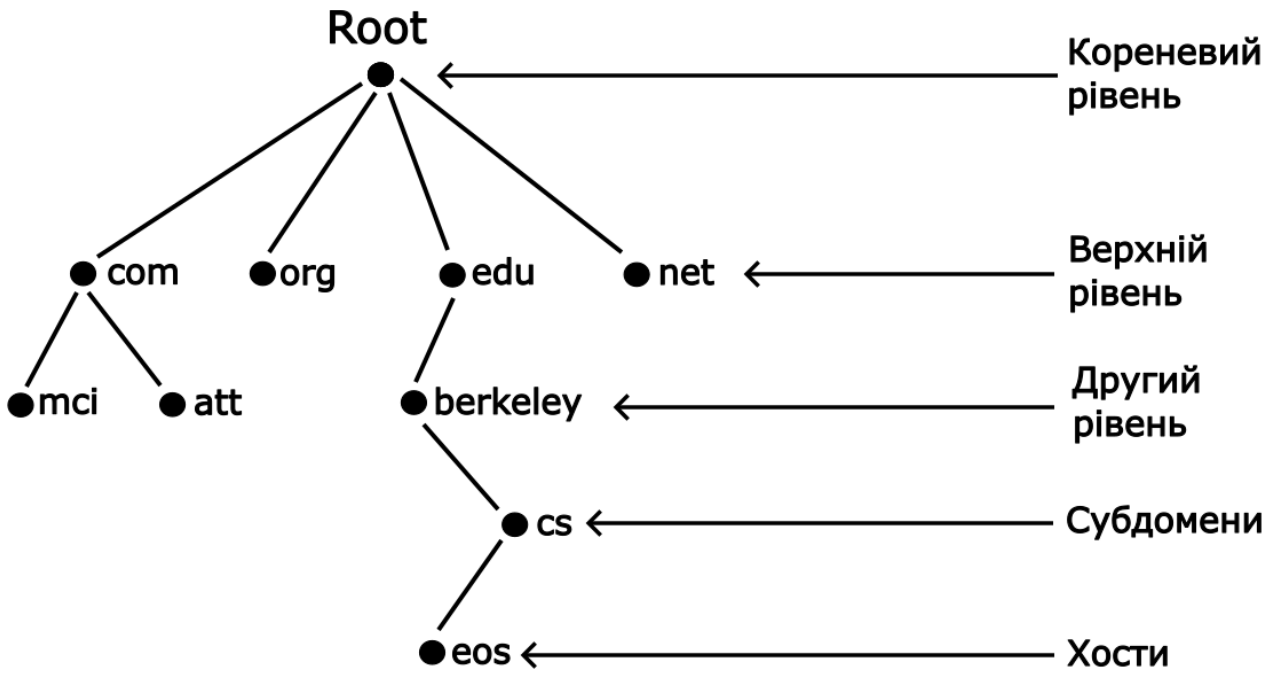


Рисунок 1.1 – Ієрархічна структура DNS (за даними [7])

Система доменних імен зберігає інформацію про домени у вигляді спеціальних записів, які визначають різні аспекти роботи домену. Ці записи забезпечують маршрутизацію трафіку, обслуговування електронної пошти, перевірку безпеки та інші функції. Кожен тип запису виконує специфічне завдання

та є важливим елементом функціонування сучасних інформаційних систем. У таблиці 1.1 наведено основні типи DNS-записів.

Таблиця 1.1 – Основні типи DNS-записів [8].

Тип запису	Опис
A	Зіставляє доменне ім’я на адресу IPv4, дозволяючи користувачам отримувати доступ до веб-сайтів, використовуючи зрозумілі людині імена замість числових IP-адрес. Наприклад, запис типу A для *example.com* може вказувати на *192.0.2.1*
AAAA	Подібно до записів A, але зіставляє ім’я домену з адресою IPv6. Прикладом: запис AAAA для *example.com*, що вказує на *2001:0db8:85a3::8a2e:0370:7334*.
CNAME	Скорочення від Canonical Name; створює псевдонім для доменного імені, дозволяючи кільком доменним іменам вказувати на ту саму IP-адресу. Наприклад, *www.example.com* може бути CNAME для *example.com*
MX	Скорочення від Mail Exchange; вказує поштові сервери, відповідальні за отримання електронних листів для домену. Наприклад, запис MX може спрямовувати електронні листи, надіслані на *example.com*, на *mail.example.com*.
NS	Вказує авторитетні DNS-сервери для домену, які відповідають за керування DNS-записами для цього домену.
SOA	Скорочення від Start of Authority; містить адміністративну інформацію про домен, включаючи основний сервер DNS і контактну інформацію адміністратора.
PTR	Використовується для зворотного DNS-запиту, зіставляє IP-адресу з доменним ім’ям.
TXT	Дозволяє зв’язати довільний текст із доменним ім’ям. Зазвичай використовується для верифікації або безпекових протоколів.
SRV	Вказує місцезнаходження послуг (наприклад, VoIP або миттєвих повідомлень) у домені, зазначаючи ім’я хоста та номер порту.
ALIAS	Подібний до CNAME, але може використовуватися на кореневому рівні домену. Автоматично розв’язується в запис A.

Окрім основних типів DNS-записів, наведених у таблиці, існують й інші, які виконують більш специфічні завдання. Наприклад, деякі з них використовуються для забезпечення динамічного перепису записів у телекомунікаційних системах або для визначення сертифікаційних центрів, яким дозволено видавати SSL-сертифікати для домену. Інші записи можуть містити географічну інформацію про

домен або дані про тип обладнання та операційну систему, що використовується хостом. Також існують записи, які забезпечують додаткову інформацію про відповідальних осіб чи ресурси, пов’язані з доменом [8].

1.2 Типові атаки на систему доменних імен

1.2.1 DNS Spoofing

Спуфінг DNS — це форма зловмисної кібератаки, спрямованої на використання вразливостей у системі доменних імен для перенаправлення інтернет-трафіку на шахрайські або шкідливі сервери. У результаті цього типу атак користувачі, які намагаються отримати доступ до певних веб-сайтів, можуть бути перенаправлені на підроблені ресурси, контрольовані зловмисником. Такі атаки створюють значні ризики, включаючи крадіжку конфіденційної інформації та зараження пристроїв шкідливим програмним забезпеченням.

DNS Spoofing реалізується через кілька основних технік:

- Отруєння кешу DNS (DNS Cache Poisoning):

Цей метод полягає у використанні вразливостей протоколу DNS для додавання підроблених записів у кеш DNS-резолвера. Зловмисник надсилає помилкові відповіді на DNS-запити, переконуючи сервер у їхній достовірності. Унаслідок цього користувачі перенаправляються на IP-адреси, що контролюються атакуючою стороною.

- Викрадення DNS (DNS Hijacking):

У цьому випадку шкідливе програмне забезпечення змінює конфігурації DNS на пристроях користувачів або маршрутизаторах. Це дозволяє зловмисникам перенаправляти трафік на небажані ресурси.

- Атака типу "людина посередині" (Man-in-the-Middle):

Ця техніка передбачає, що зловмисник займає проміжну позицію між користувачем і DNS-сервером, що дозволяє перехоплювати DNS-запити та відповіді, змінюючи їх у реальному часі [9].

Схема реалізації атаки спуфінгу DNS приведена на рисунку 1.2.

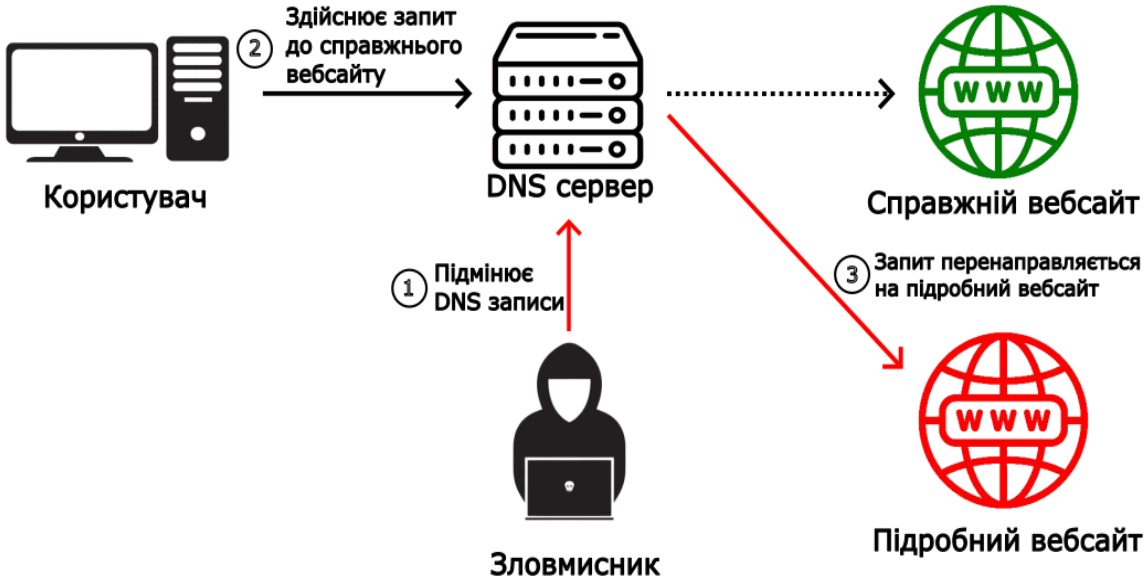


Рисунок 1.2 - Схема реалізації атаки спуфінгу DNS

1.2.2 DNS Amplification

Атака з DNS-посиленням — це тип розподіленої атаки відмови в обслуговуванні, що використовує функціональність системи доменних імен для створення надмірного обсягу трафіку, спрямованого на жертву. Цей тип атаки базується на тому факті, що відповіді DNS можуть бути значно більшими за початкові запити, що дозволяє зловмисникам збільшувати обсяг трафіку й створювати перевантаження мережевих ресурсів цілі.

Атака з DNS-посиленням зазвичай реалізується в кілька етапів:

- Підробка IP-адреси джерела:

Зловмисник надсилає DNS-запит на публічні DNS-сервери, але замість використання власної IP-адреси він підроблює адресу джерела, вказуючи IP-адресу жертви. Унаслідок цього відповіді на запити надсилаються безпосередньо на адресу жертви, а не на адресу зловмисника.

- Формування запитів, що створюють великі відповіді:

Зловмисник конструює DNS-запити таким чином, щоб отримати максимально об’ємні відповіді. Наприклад, він може запитувати інформацію про домени, які мають велику кількість пов’язаних записів (наприклад, TXT-записи або домени з підтримкою DNSSEC). Такі відповіді можуть перевищувати розмір запитів в 10–100 разів.

- Перевантаження цілі:

Після надсилання численних підроблених запитів, кілька DNS-серверів відповідають одночасно, спрямовуючи великі обсяги даних на IP-адресу жертви. Завдяки одночасному використанню кількох серверів, обсяг трафіку значно зростає, що перевантажує пропускну здатність мережі та ресурси жертви. Це призводить до збоїв у роботі або повного припинення обслуговування [10].

Схема реалізації атаки з DNS-посиленням приведена на рисунку 1.3.

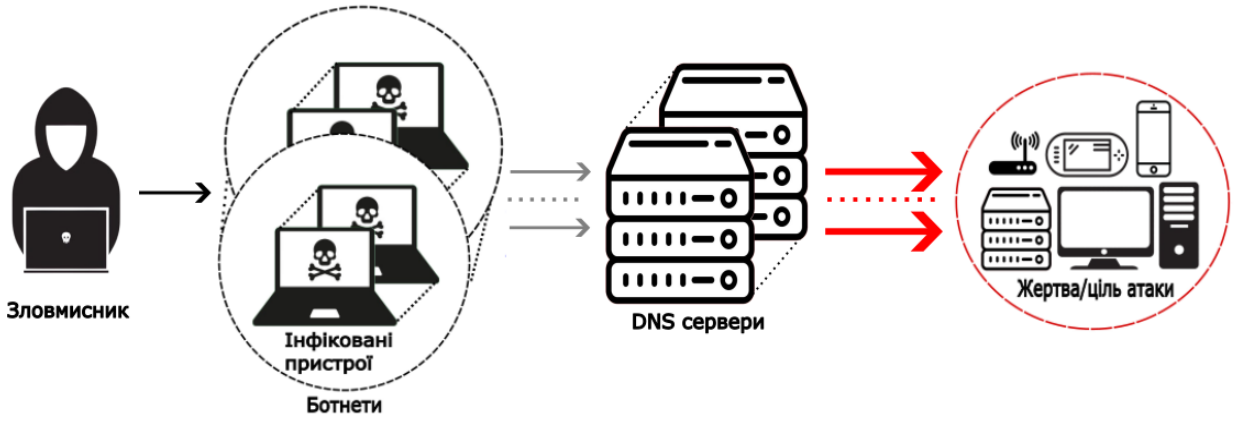


Рисунок 1.3 - Схема реалізації атаки з DNS-посиленням

1.2.3 DNS Rebinding

DNS Rebinding — це тип атаки, який дозволяє зловмиснику обходити політику єдиного походження (same-origin policy), яку застосовують веб-браузери. Ця політика зазвичай обмежує доступ скриптів до ресурсів лише в межах домену, з якого вони були завантажені. Однак DNS Rebinding дає змогу зловмисникам отримувати несанкціонований доступ до конфіденційних даних або послуг у локальній мережі жертви. Атака використовує спосіб, яким DNS перетворює доменні імена в IP-адреси, щоб маніпулювати тим, як браузер визначає довірені домени.

Атака розпочинається, коли користувач відвідує шкідливий веб-сайт, контрольований зловмисником. Спочатку домен цього сайту вирішується у зовнішню IP-адресу, яка належить серверу зловмисника. Після цього зловмисник змінює DNS-записи для цього домену, вказуючи на внутрішню IP-адресу (наприклад, *192.168.1.1*). Оскільки веб-браузер користувача не розрізняє ці дві відповіді, він продовжує надсилати запити до нової IP-адреси. У підсумку браузер вважає, що взаємодіє з оригінальним доменом, але насправді звертається до

внутрішніх ресурсів у локальній мережі жертви. Це дозволяє зловмисникам виконувати JavaScript-код, який може взаємодіяти з локальними пристроями чи сервісами в мережі користувача. Наприклад, якщо у жертви є веб-інтерфейс для маршрутизатора або пристрою розумного дому, зловмисник може отримати доступ до цих систем, змінювати їхні налаштування або викрадати конфіденційні дані [11].

Можливими наслідками атаки DNS Rebinding можуть бути:

- витік конфіденційних даних;
- зміна конфігурацій мережевих пристроїв;
- несанкціонований доступ до внутрішніх ресурсів.

Схема реалізації атаки DNS Rebinding приведена на рисунку 1.4.

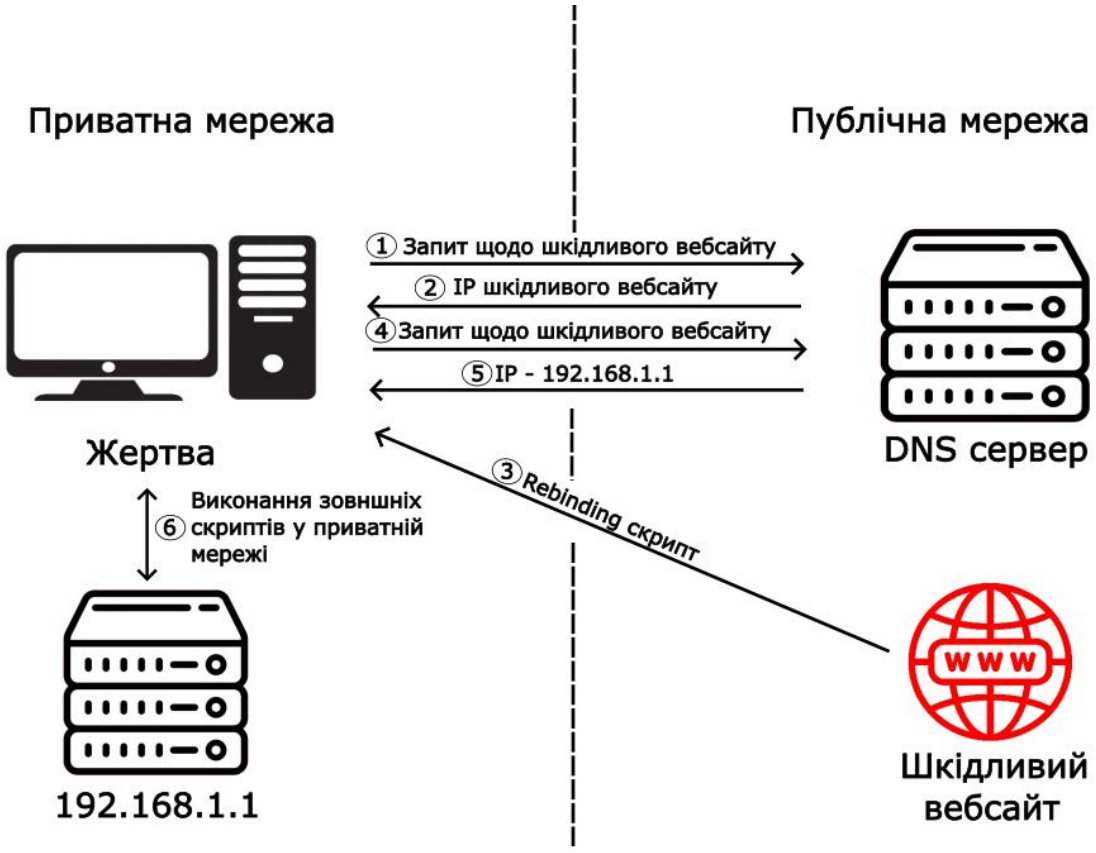


Рисунок 1.4 - Схема реалізації атаки DNS Rebinding (за даними [11])

1.2.4 Typosquatting

Typosquatting — це форма кіберсквотингу, що полягає у реєстрації доменних імен, які є неправильно написаними варіаціями популярних вебсайтів. Головною метою такої техніки є перехоплення трафіку від користувачів, які випадково вводять неправильну адресу або припускаються друкарських помилок під час введення URL. Typosquatting широко використовується для проведення фішингових атак, поширення шкідливого програмного забезпечення або отримання прибутку через рекламні схеми.

Зловмисники аналізують найпоширеніші помилки у написанні популярних доменів і реєструють ці варіації раніше за їхніх законних власників. Наприклад, якщо популярний вебсайт має адресу example.com, зловмисник може зареєструвати домени на кшталт exampel.com або exampl.com.

Користувачі, які помилково вводять такі URL-адреси, перенаправляються на сайт зловмисника, який, зазвичай, виглядає схоже на легітимний ресурс. Основними мотивами здійснення typosquatting є:

- Фішинг:

Користувачів можуть просити ввести конфіденційну інформацію, таку як логіни, паролі або номери кредитних карток. Зібрані дані зловмисники використовують для шахрайства.

- Розповсюдження шкідливого ПЗ:

Зловмисні сайти можуть містити шкідливе програмне забезпечення, яке видається за легітимні додатки. Користувачі, не підозрюючи загрози, завантажують ці файли, що призводить до зараження їхніх пристроїв.

- Отримання рекламного прибутку:

Зловмисники можуть отримувати дохід, розміщуючи на своїх сайтах рекламу або перенаправляючи користувачів на партнерські посилання [12].

Загалом, можна стверджувати, що успіх більшості атак на протокол DNS залежить від того, чи отримує жертва шкідливі або фальшиві відповіді на її запити. Для цього жертва повинна або самостійно відвідати небезпечний домен, або випадково потрапити на такий ресурс через підміну відповіді резоолвера, або ж бути вразливою до флуда зловмисних запитів. Якщо б система жертви не отримала ці відповіді, атака була б неефективною. Це підкреслює важливість використання інструментів DNS-фільтрації для запобігання отриманню шкідливих даних і мінімізації потенційних загроз для користувачів.

2 ДОСЛІДЖЕННЯ ІСНУЮЧИХ МЕХАНИЗМІВ ФІЛЬТРАЦІЇ DNS ТРАФІКУ ТА МОЖЛИВОСТІ ПОКРАЩЕННЯ ЇЇ ЕФЕКТИВНОСТІ

DNS-фільтрація мережевого трафіку є елементом проактивної стратегії кіберзахисту, що діє на рівні системи доменних імен для контролю та управління доступом до Інтернету в мережі. Використовуючи DNS-фільтрацію, організації можуть впроваджувати потрібні політики безпеки, які обмежують доступ до певних веб-сайтів та/чи категорій контенту, які вважаються такими, що не відповідають корпоративним вимогам. Цей механізм фільтрації працює шляхом перехоплення DNS-запитів і їх наступного порівняння із попередньо визначеними реєстрами дозволених чи заблокованих доменних імен та IP-адрес. У разі наявності відповідного ресурсу в стоп листі, DNS-фільтр блокує таке з'єднання.

DNS-фільтрація надає організаціям і окремим мережевим користувачам кілька помітних переваг, зокрема підвищення поточного рівня кібербезпеки шляхом запобігання відвідуванню потенційно зловмисних та/чи небажаних інформаційних ресурсів (наприклад, як функція «батьківській контроль»). Така фільтрація джерел контенту допомагає забезпечити дотримання відомчих нормативних вимог, обмежуючи доступ до сайтів, які не відповідають політиці безпеки компанії/установи. Крім того, DNS-фільтрація сприяє підвищенню продуктивність персоналу, обмежуючи його доступ до інформаційних ресурсів, що не пов'язані з виконанням їх функціональних обов'язків, та зменшує навантаження на корпоративну мережу, шляхом адміністрування небажаного ресурсоємного трафіку [13]. Також слід підкреслити, що делегування функцій DNS-фільтрації на довірені зовнішні ресурси/сервіси, дозволяє в значній мірі позбутися необхідності регулярного перевірки й оновлення стану відповідних реєстрів доступу силами

корпоративних фахівців та забезпечити більшу функціональну стійкість корпоративної інфраструктури в разі відповідних атак [4].

2.1 Напрями покращення інформативності процедур контролю DNS трафіку

2.1.1 Використання каналів розвідки загроз

Канали розвідки загроз є важливим компонентом для реалізації сучасних стратегій ІБ (див. рис. 2.1). Їх використання, в режимі реального часу, забезпечує корпоративних фахівців з ІБ відомостями про нові загрози, вразливості і діяльність кіберзловмисників [2-3]. Вони функціонують шляхом постійного збору та узагальнення даних з різних джерел для виявлення потенційних кіберзагроз і вразливостей [14-15]. Зазвичай ці канали містять індикатори компрометації, наприклад, такі як: - «шкідливі» IP-адреси, доменні імена, хеші файлів та типові шаблони/сценарії підозрілої поведінки [3,16].

Інтеграція каналів розвідки загроз в інструменти та корпоративні системи ІБ дозволяє організаціям автоматично блокувати відомі зловмисні чи скомпрометовані джерела і ранжувати пріоритетність сповіщень системи безпеки на основі відомостей, стосовно актуальності інформації та серйозності наслідків. Це дає змогу приймати обґрунтовані рішення, покращувати час реагування на інциденти та покращити загальну систему ІБ. Використовуючи канали розвідки поточних загроз, сучасні організації можуть покращити показник поінформованості про стан актуальних загроз та завчасно вживати профілактичних заходів для усунення відповідних вразливостей власних ІКС ще до того, як ними спробують скористатися зловмисники [4].

Постійно співвідносячи дані внутрішнього моніторингу безпеки із зовнішніми каналами розвідки загроз, сучасні організації можуть виявляти приховані загрози та превентивно їм протидіяти [16-18]. Такий проактивний підхід до кібербезпеки допомагає випереджати кіберзлочинців, мінімізувати вплив

інцидентів безпеки та ефективно захищати конфіденційні дані та критично важливі активи [4,19].

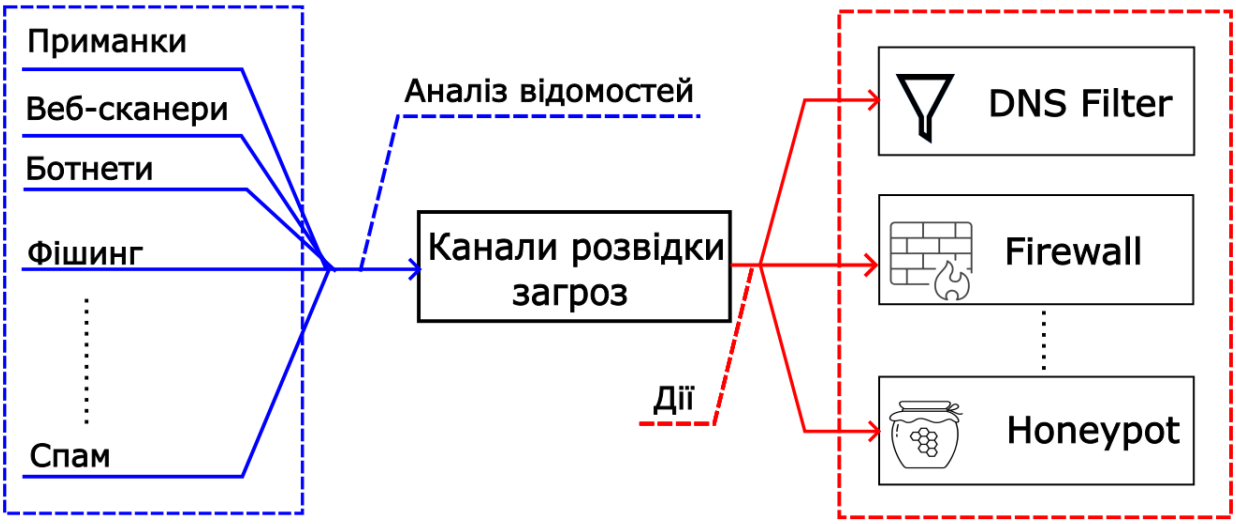


Рисунок 2.1 - Сутність використання каналів розвідки загроз ІБ [4]

2.1.2 Впровадження інструментів виявлення серверів управління ботнетів

Сервер управління ботнет системи являє собою централізований (ведучій) сервер, що використовується зловмисником для дистанційного управління скомпрометованими мережами та/чи пристроями, відомих, як боти або бот-мережі. Командно-контрольні сервери ботнетів (див. рис. 2.2) відіграють ключову роль в організації зловмисних дій, таких як проведення розподілених атак на відмову в обслуговуванні (DDoS-атак, в т.ч. *DNS amplification attack*), DNS та NTP Spoofing, розсилання спаму, здійснення шахрайства, поширення зловмисного ПЗ тощо [3,4,20-21].

Аналізуючи поточні властивості мережевого трафіку, і таким чином ідентифікуючи роботу відповідних серверів, засоби безпеки можуть протидіяти впливу ботнетів, своєчасно парируючи їх зловмисну дію. Переваги виявлення

такого трафіку передбачають зменшення ризику атак з боку ботнетів, захист мережевих пристроїв від їх компрометації з використанням експлойтів та підтримку безпечного корпоративного середовища [4,14-16,22] .

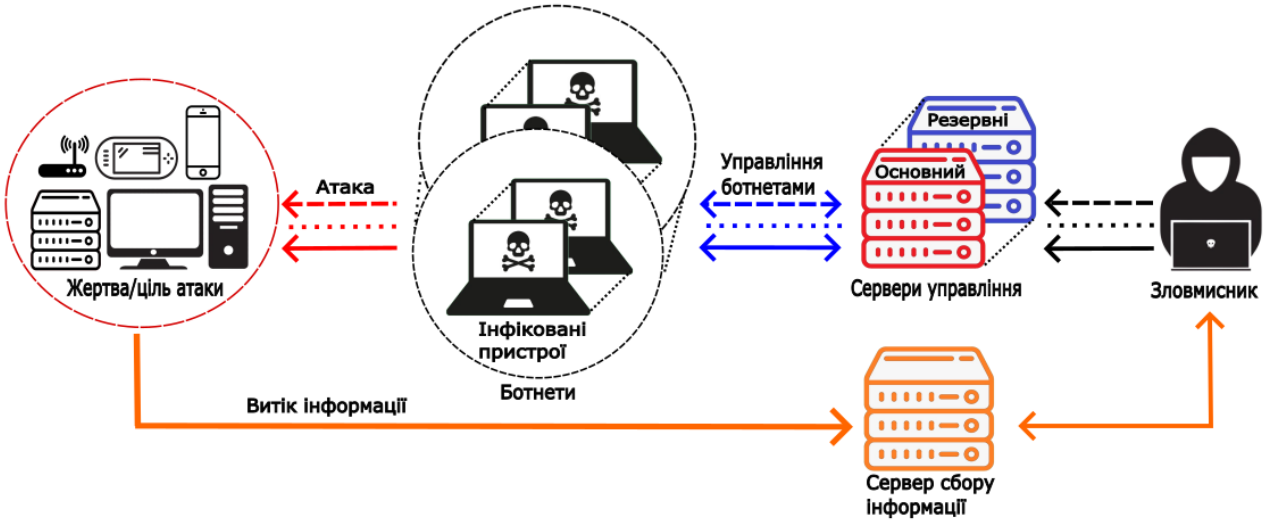


Рисунок 2.2 - Архітектура ботнетуа [4]

2.1.3 Можливості шифрування трафіку DNS

Шифрування DNS трафіку, зокрема DNS-over-TLS (DoT), DNS-over-HTTPS (DoH) та DNS-over-QUIC (DoQ), відіграє важливу роль у підвищенні безпеки та ефективності контролю DNS запитів [23]. Цей процес перетворює інформацію DNS трафіку у зашифрований формат, що гарантує можливість декодування інформації лише довіреними сторонами, такими як DNS-клієнт і сервери DNS провайдера. Шифрування зменшує ризик атак на DNS трафік і маніпуляції з даними цих запитів, знижуючи ймовірність неавторизованих змін у DNS записах. Крім того, існуючи протоколи шифрування DNS підвищують рівень цілісності й конфіденційності відповідного трафіку, протидіючи несанкціонованому моніторингу та відстеженню відомостей DNS-запитів. Процес шифрування DNS трафіку, в цілому, сприяє підвищенню конфіденційності та анонімності онлайн дій користувачів, однак на

цьому шляху є і певні складнощі. Наприклад, складність відстеження вмісту зашифрованого DoH трафіку, з боку адміну безпеки, свідчить про те, що аналіз мережевих DoH з'єднань з метою виявлення потенційного шкідливого трафіку (наприклад з боку зовнішніх командно-контрольних серверів умовних бот-систем [22]), є актуальним напрямом для його подальшого аналізу [4,23].

2.1.4 Запровадження зон політик реагування

Response Policy Zone (RPZ) – це механізм, який використовується при фільтрації DNS для визначення діючих локальних політик у стандартизованому форматі та завантаження/оновлення політик із інших, «зовнішніх» джерел. Використовуючи RPZ, сучасні організації можуть оперативно контролювати, які запити можуть обробляти їхні сервери DNS, а які ні. Це дозволяє оперативно блокувати шкідливі домени й ресурси або виконувати інші дії на основі попередньо встановлених політик безпеки. Політики безпеки формалізуються у вигляді файлів складених для відповідних зон DNS (див. рис. 2.3), котрі формуються за визначеними корпоративними критеріями та/чи діючими нормами мережевої поведінки [24] для основних груп користувачів інформаційних послуг компанії/установи. Цей процес в рівній мірі відноситься, як для власного персоналу, так і користувачів основних послуг компанії. Впровадження концепції RPZ полегшує, масштабування, використання й оновлення актуальних зон безпеки. Парадигма RPZ передбачає їх спільне застосування між декількома DNS-серверами (через передачу зон), що дозволяє оперативно транслювати дані політики за допомогою звичайних протоколів DNS [23]. Політики RPZ складаються з відповідних поведінкових (логічних) тригерів та дій. Тригери визначають, коли потрібно застосувати ту чи іншу політику (тобто декларують певні мережеві умови/обставини), а дії, відповідно, вказують, які саме процедури слід виконати в даному разі.

Впровадження RPZ забезпечує уніфікацію, масштабованість і гнучкість управління політиками DNS. Це підвищує безпеку і цілісність налаштувань фільтрації DNS трафіку, а також дозволяє організаціям визначати й застосовувати власні політики для адміністрування сервісу DNS [4,25].

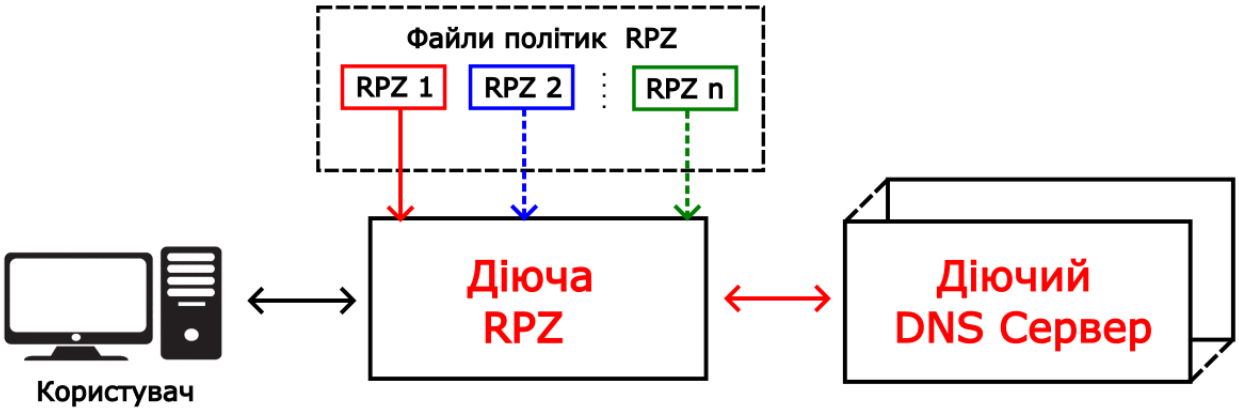


Рисунок 2.3 - Сутність використання RPZ [4]

2.1.5 Застосування механізмів виявлення роботи алгоритмів генерації доменів

Domain Generation Algorithm (DGA) – це алгоритми, які використовуються кіберзловмисниками для генерування великої кількості доменних імен, що слугують умовними точками «зустрічі» в мережі між скомпрометованими пристроями та серверами управління зловмисника. В цьому випадку (див. рис. 2.4 [4]), скомпрометовані/атаковані комп'ютери й інше мережеве устаткування намагаються встановити зв'язок (DNS spoofing) з цими генерованими доменними іменами, щоб отримати від них «оновлення» чи інші команди з боку атакуючої сторони. Ці алгоритми створені для періодичного генерування значної кількості доменних імен, що помітно ускладнює боротьбу з ботнетами. Таким чином DGA використовуються для приховування фактичного розташування командно-контрольних серверів ботнетів серед великої кількості потенційно легітимних адрес, що значно ускладнює пошук та блокування (згодом і видалення) цих доменів.

В широкому сенсі DGA, за своєю парадигмою, є своєрідним аналогом атак типу DNS amplification [21].

Виявлення злочинних доменних імен, що синтезовані DGA, є складної задачею через їх великі обсяги відповідної генерації та випадковий характер цих процесів. Однак, залучення технологій штучного інтелекту і машинного навчання (AI/ML), та впровадження методів глибокого навчання, потенційно демонструють високі показники в ідентифікації злочинних DGA [4,26].

Рисунок 2.4 - Інфографіка порядку дій зловмисника при використанні DGA

2.2 Узагальнення досвіду, стосовно практичних реалізацій процедур фільтрації DNS

Спираючись на аналіз ряду відомих робіт, проведено узагальнення основних відомостей, стосовно особливостей взаємозв'язку технології DNS та питань безпеки сучасних ІКС.

2.2.1 Канали розвідки загроз

В роботі [27] розглянуто проблематику «пасивних» DNS-потоків для отримання актуальної інформації про потенційні загрози безпеки. Автор роботи фокусує увагу на ідентифікації підозрілих і зловмисних доменів, виявленню зловживань DNS та їх аномалій, підкреслюючи важливість перехресної перевірки даних з іншими джерелами інформації про кіберзагрози. Акцентовано необхідність ретельного аналізу DNS-трафіку для виявлення різноманітних кіберзлочинних дій, таких як ботнети, спам і фішинг. Крім того, дослідження автора зосереджене на розробці та впровадженні масштабованої методики «пасивного» аналізу DNS, яка забезпечує своєчасне виявлення актуальних загроз ІБ [4].

Проблематика визначення «якості» каналів розвідки загроз безпеки є вкрай актуальним та доволі складним завданням, оскільки їх важко порівнювати між собою. Так наприклад, у роботі [28] її автором розглядається методологія вимірювання надійності та якості безкоштовних каналів розвідки про кіберзагрози з відкритим кодом. У цьому дослідженні дані аналізуються за допомогою різних метрик, наприклад: - обсягом даних, швидкістю змін, географічним розподілом та збігами між різними каналами. Підкреслено, що різні канали зосереджені на різних питаннях, що ускладнює прямі порівняння. Автором стверджується, що остаточного методу визначення найкращого каналу поки не існує та можна лише зробити певні висновки, стосовно їх якості. Дослідження підкреслює складність

оцінки безкоштовних каналів розвідки загроз та потребу в більш надійних та достовірних інструментах [4,28].

Робота [29] також розглядає проблематику оцінки якості каналів розвідки і пропонує всебічний аналіз каналів розвідки загроз. Автори цієї роботи звертають увагу на необхідність та важливість емпіричних оцінок у цій галузі. Дослідження представляє набір показників для оцінки каналів розвідки загроз, включаючи такі як: - обсяг, диференціальний внесок, унікальний внесок, затримку, точність, охоплення та ін.. Дослідивши 47 каналів із різними IP-адресами та 8 каналів із хешами файлів зловмисного ПЗ, автори роботи виявили значні обмеження в охопленні та точності наявних даних розвідки загроз. Це вказує на високий ступінь унікальності та помилкових узагальнень (прогнозів) для різних каналів. В цілому, ця робота надає обґрунтовану методологію для оцінки каналів розвідки загроз та підкреслює важливість постійного розвитку методів їх оцінки [4,29].

2.2.2 Виявлення активності ботнетів

Дослідницька група авторів роботи [30] пропонує огляд джерел, який включає в себе вичерпний аналіз методів виявлення ботнетів на основі аналізу DNS трафіку. Робота звертає увагу на проблему ботнетів та пов'язаних з ними загроз ІБ. Автори класифікують та порівнюють кілька підходів, що використовують різні властивості DNS трафіку, такі як: – зміна домену, лексичні характеристики доменних імен і шаблони в запитах та DNS відповідях. В роботі аналізуються сильні й слабкі сторони для відповідних підходів. Зазначено, що методи виявлення на основі DNS мають переваги перед методами на основі хосту, оскільки вони можуть надати більш широке уявлення про мережеву активність і уникнути деяких обмежень, пов'язаних з розгортанням систем моніторингу в мережі. У роботі також обговорюються обмеження існуючих підходів, такі як відсутність стандартних метрик і показників

продуктивності та підкреслено потреба в нових методологіях виявлення для протидії поширенню загроз ботнетів [4,30].

В роботі [31] розглядаються різні методології аналізу поведінки ботів та ботнетів, включаючи статистичний аналіз і вимірювання трафіку та наголошується на важливість комплексного підходу до виявлення ботнетів. На думку авторів для ряду попередніх робіт була притаманна деяка невизначеність, насамперед в частині, що підкреслює складність пошуку всіх типів ботнетів через складність моделі їх поведінки. Саме тому дана робота пропонує новий механізм виявлення ботнетів, заснований на моніторингу трафіку DNS для виявлення групової активності розподілених ботів, усуваючи обмеження попередніх підходів і забезпечуючи більш надійний метод для виявлення різних варіацій ботнетів [4,31].

Робота [32] містить огляд існуючих методів виявлення ботнетів та пов’язаних з ними обмежень. Авторами роботи висвітлюються проблеми виявлення нових різновидів ботнетів та звертається увага на необхідності впровадження методів, що не базуються на перевірці корисного навантаження пакетів, оскільки вони могли би працювати із зашифрованими мережевими протоколами [23]. У роботі обговорюється обмеження існуючих методів виявлення при роботі з новими атаками ботнетів. Також підкреслено переваги аналізу поведінки трафіку порівняно з аналізом корисного навантаження пакетів, зокрема можливість працювати із зашифрованим трафіком та менший вплив на продуктивність мережі. У дослідженні запропонована модель виявлення, окреслено параметри її експериментальної оцінки та надано результати для запропонованого детектора, котрі свідчать про його здатність виявляти ботнет-атаки з високою точністю [4,32].

2.2.3 Шифрування DNS

Робота [33] надає ґрунтовний огляд поточного спектру можливостей шифрування DNS [23], зосереджуючись на стандартних методах: - DoT, DoH та DoQ. Автори аналізують статус прийняття цих методів, їх продуктивність, переваги та проблеми безпеки. Основним фокусом статті є зловживання шифруванням DNS для командно-контрольних комунікацій і каналів викрадення/витоку даних, що створює певні труднощі у виявленні та боротьбі з відповідною діяльністю. Також обговорюються методи аналізу зашифрованого DNS-трафіку для профілювання дій користувачів з метою виявлення зловмисної та/чи нештатної мережної активності. Авторами роботи сформульовано напрями майбутніх досліджень, стосовно підвищення продуктивності та безпеки шифрування DNS [4,33].

Автори роботи [34] провели детальний аналіз розгортання і використання протоколів DNS шифрування, зосереджуючись на DoT та DoH. Дослідження охоплює порівняльну оцінку різних протоколів DNS шифрування та проблеми безпеки, що пов’язані з цими протоколами, зокрема оцінку доступності та продуктивності, а також порівняння обсягів трафіку між традиційними й зашифрованими DNS запитами. Автори дослідження підкреслюють, що хоча якість послуг постачальників DNS загалом є задовільна, однак, деякі служби мають неправильні конфігурації, що потребує уваги. Висновки висвітлюють поточний стан галузі телекомунікацій до широкомасштабного впровадження зашифрованих DNS протоколів [4,34].

Робота [35] присвячена дослідженню шифрування DNS запитів для захисту конфіденційності користувачів від атак, що передбачають аналіз трафіку. Автори досліджують питання ефективності аналізу DNS трафіку для виявлення моделей веб-активності користувачів через зашифрований DNS, з особливим акцентом на протокол DoH [23]. Пропонуючи новий набір функцій, розроблений спеціально для аналізу зашифрованого DNS-трафіку, автори демонструють «успішні» атаки з

високою точністю, підкреслюючи існуючи обмеження поточних засобів захисту. Автори досліджень оцінюють рівень захисту, який забезпечують DoH і DoT, акцентуючи, що DoT демонструє кращій рівень безпеки. Підкреслено, що потенціал фільтрації трафіку на основі DNS, навіть у сценаріях із зашифрованим DNS, надає цінну інформацію про вразливості і засоби захисту в рамках зашифрованого DNS [4,35].

2.2.4 Узагальнення проблематики адміністрування RPZ

Робота [36] пропонує аналіз проблематики зон політики реагування DNS та розглядає особливості реалізації RPZ в ПЗ сервера імен BIND, а також приклади їх (політик) розгортання в реальному середовищі. Згідно з дослідженням, впровадження зон політики реагування ефективно блокує спроби DNS запитів з небезпечними доменами, захищаючи клієнтські системи без втрати їхньої продуктивності. Запропоновано набір інструментів для аналізу журнальних даних RPZ, який сприяє завчасної ідентифікації потенційно скомпрометованих систем та небажаної мережевої поведінки користувачів. Також у роботі спрогнозовано подальші напрямки розвитку RPZ, включаючи залучення комерційних постачальників відповідних послуг та вдосконалення механізмів захисту від фішингових атак [3-4, 36-37].

Дослідження роботи [38] присвячено питанням виявлення і блокування аномальних вихідних DNS-запитів, що пов’язані із функціонуванням ботнетів. Автори пропонують систему на основі політик, яка використовує зони політики реагування DNS для вдосконалення свого попереднього підходу, котрий базувався на базі даних MySQL, основним недоліком якого була мережева затримка. Система що розглядається використовує програмно-визначену мережу для контролю мережевого трафіку та аналізу DNS-запитів на відповідність політикам, збереженим у відповідних RPZ, ефективно зменшуючи затримку і поліпшуючи виявлення, та

блокування шкідливого DNS-трафіку. Робота містить попередню оцінку продуктивності і функціональності системи, яка підтверджує її ефективність у локальному мережевому середовищі на основі програмно-визначеної мережі. Автори планують продовжити оцінку цієї системи в умовах роботи реальної мережі та вивчення методів збереження конфіденційності DNS трафіку [4,38].

2.3.5 Специфіка виявлення алгоритмів генерації доменів (DGA)

Робота [39] пропонує всебічний огляд проблематики у напряму виявлення алгоритмів генерації доменів. Автори вказують на важливість тестів, стандартизованих метрик і методів виділення ознак для підвищення надійності та відтворюваності експериментів у дослідженнях DGA. Досліджуючи різні DGA та їх альтернативи проведено розгляд стратегій, котрі використовуються зловмисниками, та потребу в більш складних методах виявлення. Запропонована авторами методологія застосовує ймовірнісні підходи для ефективного виявлення DGA на основі списку слів. Робота пропонує аналіз поточного стану виявлення DGA та окреслює можливі майбутні напрями досліджень для протидії інструментам створення шкідливих доменів [4].

Авторський колектив роботи [40] представляє власний підхід до ідентифікації алгоритмів генерації доменів, шляхом використання методів глибокого навчання. Дослідження зосереджено на вирішенні труднощів виявлення DGA, які генерують домени шляхом псевдовипадкового об’єднання словникових термінів. Використовуючи контекстно-залежне вбудовування слів та «простий» повнозв’язний класифікатор, автори роботи демонструють ефективність свого підходу до класифікації доменів. Використання попередньо підготовлених слів, мінімальні обсяги вихідних даних та коротка тривалість термінів навчання, відрізняють цей підхід від існуючих методів. Дослідники підкреслюють оригінальність своєї методики, яка не потребує розробки функцій вручну та

вивчення списків слів DGA [40], що підтверджує її потенціал для умов реального застосування [4].

У роботі [41] досліджуються особливості виявлення зловмисних доменних імен генерованих DGA, за допомогою реалізації різних архітектур DL. Дослідження охоплює використання згорткової нейронної мережі (CNN), рекурентної нейронної мережі (RNN), довготривалої короткочасної пам'яті (LSTM) та інших моделей. Автори досліджують ефективність різних підходів DL для точного визначення шкідливих доменів, порівнюючи показники продуктивності різних архітектур моделей глибокого навчання. Запропонований фреймворк [41] демонструє високу ефективність у виявленні доменних імен, що синтезовані DGA та може бути використаний для блокування ботів від зовнішніх зв'язків і порушення каналів управління з командно-контрольними серверами ботнетів [4].

Як вже було зазначено вище, канали розвідки загроз є важливими джерелами критично важливих даних про поточні кіберзагрози, які допомагають завчасно виявляти і оперативно реагувати на нові загрози ІБ. Однак при цьому, актуальною проблемою залишається оцінка якості цих каналів. Дослідження за цим напрямом пропонують нові метрики та методики для оцінювання каналів розвідки, в т.ч. на основі аналізу даних DNS-трафіку. Крім того, наголошується на високому ступені унікальності та надмірності помилкових спрацювань у різних каналах розвідки. Узагальнюючи стан напрацювань та поглядів дослідників на подальші можливості більш релевантної екстракції даних з каналів розвідки загроз, можна зробити висновок, про необхідність покращення ефективності механізмів оцінки наявних каналів розвідки та концентрації зусиль на зменшенні частки помилкових спрацювань (*прогнозів*) [4].

Виявлення ознак мережевої активності командних серверів ботнетів може надати важливу інформацію для систем безпеки з фільтрацією DNS-трафіку.

Впровадження такого механізму протидії дозволяє відокремити інфіковані засоби у мережі від відповідних командних центрів або завчасно захистити мережу від атак ботнетів, наприклад за рахунок нав'язування мережевої «гри» з використанням мережевих пасток [15-17]. Поточний стан досліджень тримає у фокусі питання виявлення каналів управління ботнетів, що в свою чергу, зумовлює постійне вдосконалення методик аналізу їх поведінки шляхом комплексного моніторингу DNS-трафіку, включаючи «роботу» з зашифрованим трафіком [23]. Отримані напрацювання декларують досить високу точність виявлення для відомих різновидів ботнетів, однак дослідники звертають увагу на обмеження існуючих підходів [4,39-41].

Стосовно напряму шифрування DNS, слід чітко розділяти дві взаємопов'язані іпостасі подальшого розвитку подій. Так, з одного боку шифрування DNS, може сприяти підвищенню якості фільтрації DNS, теоретично забезпечуючи при цьому більш безпечний канал для DNS-запитів, запобігаючи підробці та/чи втручанню в трафік зловмисників. Однак, з іншого боку, із впровадженням шифрування DNS виникає проблема ускладнення процесу виявлення зловмисного трафіку [23], зокрема трафіку командно-контрольних центрів ботнетів. Більш того, у аналізованих роботах є приклади «успішних» атак на шифрування DNS (наприклад у [35]), що ставить під сумнів ефективність шифрування, як інструменту із забезпечення конфіденційності мережевої активності користувачів. Автори робіт пропонують власні методи аналізу зашифрованого трафіку для профілювання активності користувачів з метою виявлення зловмисних дій та вказують на перевагу DNS-через-TLS у ефективності захисту. В якості основної проблеми, у межах напряму шифрування *DNS*, слід зазначити можливість використання зловмисниками технологій шифрування цього трафіку з метою ускладнення виявлення їх деструктивної діяльності (*фішинг, спам, DDos, DNSA та ін.*) [4].

Практика використання зон політик реагування (*RPZ*) незмінно продовжує привертати увагу дослідників [36,38] до цього напряму та є ефективним інструментом управління безпекою сучасних ІКС на рівні служби *DNS*, що забезпечує завчасне виявлення й блокування небезпечних доменів [4].

Виявлення ознак роботи алгоритмів генерації доменів (*DGA*), що виступають головним інструментом для створення сукупності умовних «точок зустрічі» кіберзловмисників з їх власними командно-контрольними серверами ботнетів, залишається вкрай важливим завданням. Стрімке й одночасне поширення Інтернету речей (*IoT*), технологій віртуалізації (*VT*), штучного інтелекту і машинного навчання (*AI/ML*), лише додатково підкреслюють вектор на невідкладність розробок, що адекватні рівню та темпу появи нових загроз ІБ. Вочевидь, що проблематика оперативної ідентифікації, блокування генерації і поширення злочинних доменів [39-41] ще до того, як вони зможуть завдати шкоди, є безумовно пріоритетним напрямом для подальших досліджень профільних фахівців з питань ІБ [4].

Останні дослідження [31-33,39-41] в галузі протидії впливу *DGA* розглядають тести, метрики та нові методи виявлення відповідних алгоритмів, котрі базуються на ймовірнісному аналізі і техніках *ML*/*DL*. Узагальнюючи цей досвід можна стверджувати, що пріоритетним напрямом з протидії *DGA*, є інтеграція нових - більш вдосконалених методик у єдину інтегровану систему фільтрації *DNS*, що втілює парадигму проактивного захисту та базується на останніх напрацюваннях в галузі штучного інтелекту і машинного навчання. Впровадження цих технологій розширює можливості поведінкового аналізу мережевої активності [15-16] та вдосконалює евристичні алгоритми, що здатні завчасно й оперативно виявляти і класифікувати підозрілі домені, тим самим усуваючи основні передумови для поширення активності ботнетів. Тож найбільш ймовірним напрямом досліджень в

галузі проблематики фільтрації *DNS*, можна вважати синтез нових структур та логіки роботи інтегрованих підсистем *DNS* безпеки, що впроваджують останні напрацювання на рівні стеку *VT-AI-LM-DL* технологій. Вочевидь, що саме таке поєднання, може дати паритетну відповідь на загрозу масштабного впровадження *AI* в алгоритми зловмисної генерації доменів. Іншими словами, де-факто потрібно вести розмову про необхідність ефективної протидії впливу *AI* як основного елемента протиборчої сторони [4,42].

3 РОЗРОБКА ТА ЕКСПЕРЕМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТЕСТОВОГО ПРОТОТИПУ ПРОГРАМНОГО ДОДАТКУ МОНІТОРИНГУ ПОТОЧНОГО СТАНУ ХМАРИ DNS-СЕРВЕРІВ

В рамках виконання досліджень було розроблено прототип тестової версії програми для виявлення аномалій DNS трафіку, що дозволяє проводити збір, обробку, зберігання і відображення статистичних відомостей, стосовно поточних параметрів часових затримок DNS-запитів в умовах використання різних типів протоколів та різних доменних зон.

3.1 Загальна концепція прототипу дослідної програми

Концепція програми ґрунтується на задачі тестування та аналізу продуктивності DNS-серверів з урахуванням сучасних вимог безпеки. Основна мета програми – виявлення аномалій у роботі DNS-серверів та оцінка їхньої продуктивності для різних методів шифрування із подальшим використанням цієї інформації адаптивною RPZ. Програма поєднує локальні обчислення та розподілені хмарні функції для забезпечення точності вимірювань та широкого географічного охоплення. Перед початком проведення вимірювань DNS-запитів здійснюється синхронізація поточного часу між основною консоллю адміністратора та хмарними ботами-тестерами. Синхронізація виконується одночасно для всіх учасників процесу за допомогою звернення до найближчого доступного NTP-сервера. Для синхронізації використовується NTP сервер у тій самій зоні, що і консоль адміністратора та боти-тестери. Візуалізація концепції програми наведена на рис. 3.1 (зелені стрілки відповідають функціям звірення між серверами одного ресурсу і між серверами різних ресурсів в один і той же час). На рисунку 3.2 зображено структурно-логічну схему програми.

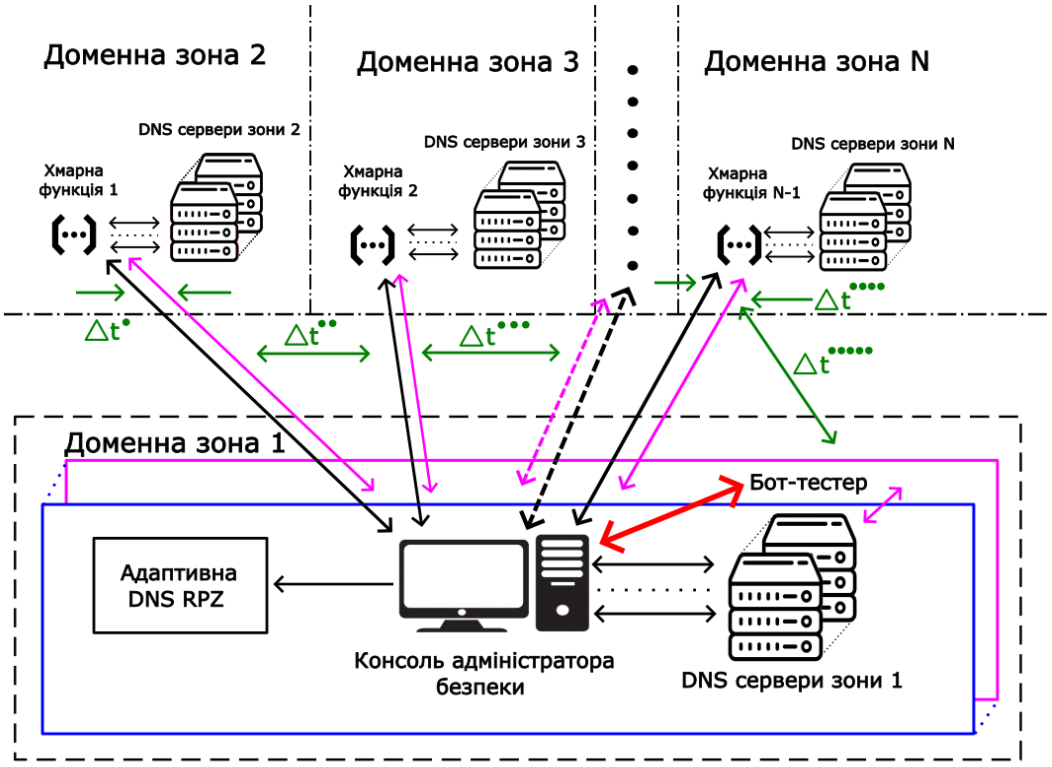


Рисунок 3.1 – Візуалізація концепції програми *(авторська розробка)*

Рисунок 3.2– Структурно-логічна схема програми *(авторська розробка)*

Під час виконання тестів програма застосовує два типи затримок:

- затримка між тестуванням серверів із одної пари (основний та резервний сервер);
- затримка між відправленням аналогічних запитів за різними протоколами.

Перша затримка виконується для зменшення впливу сценарію виходу з ладу основного сервера на результати тестування. Друга затримка призначена для мінімізації впливу кешування серверами попередніх запитів.

3.2 Функціонал тестової версії дослідної програми

Основні завдання та функції:

1) Підтримка трьох протоколів для виконання DNS запитів:
   - Незашифровані DNS запити;
   - DNS over HTTPS (DoH) – для відправки запитів із використанням шифрування через HTTPS ;
   - DNS over TLS (DoT) - для відправки запитів із використанням шифрування через TLS.
2) Одночасне тестування визначеного пулу DNS серверів на їх підтримку протоколів шифрування запитів.
3) Виконання таргетованого тестування для визначених пар DNS серверів *(у зв'язці «основний-резервний»)*.
4) Вимірювання часу «відгуку» контрольованих серверів для кожного з протоколів, що підтримуються.
5) Збереження результатів тестів, включаючи текстові «відповіді» серверів, часові мітки початку та завершення тестів (Сеансу виміру).
6) Відображення результатів тестів в інтерактивній таблиці із полями:
   - назва серверу;
   - домен для тестування;

- IP серверу;
- часи виконання запиту по трьом протоколам;
- відповіді серверів по трьом протоколам;
- час початку та кінця тестування серверу.

7) Побудова гістограм в реальному часі, що відображають часові показники для кожного протоколу на конкретному сервері.
8) Можливість виконувати замірювання із багатьох точок світу завдяки хмарним функціям.
9) Можливість компенсування впливу кешування тестових запитів, шляхом рознесення тестів (*Сеансів вимірів*) для різних протоколів у часі та локації застосованих ботів.
10) Гнучкість налаштування: легкість додавання серверів та доменів у тести, а також зміни поточних параметрів Сеансів вимірювань.

3.3 Графічний (користувацький) інтерфейс програми

Графічний інтерфейс програми складається з наступних вікон:

1) Вікно вибору режиму роботи програми (рис. 3.3):

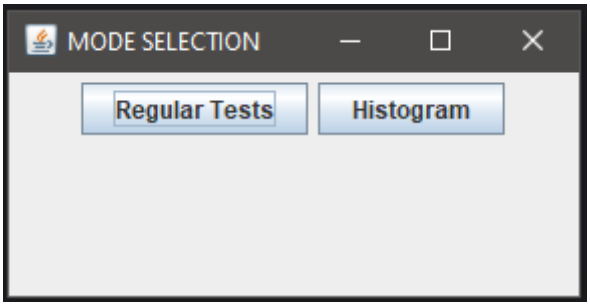


Рисунок 3.3 – Вікно вибору режиму роботи тестової програми

2) Вікно, що вказує усі домени, які будуть використовуватися при тестуванні, з можливістю додати потрібний домен (рис. 3.4):

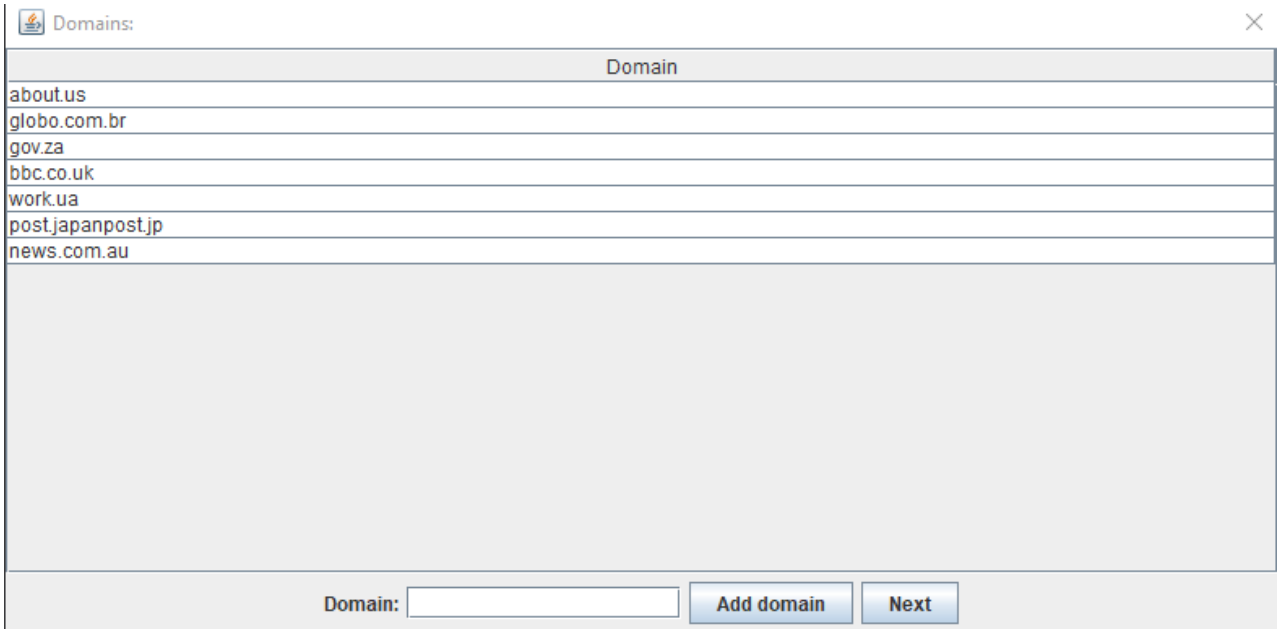


Рисунок 3.4 – Вікно із таблицею контрольованих доменів

3) Вікно, що вказує усі DNS сервери, які обрані для тестувань, з можливістю додати сервер (рис. 3.5):

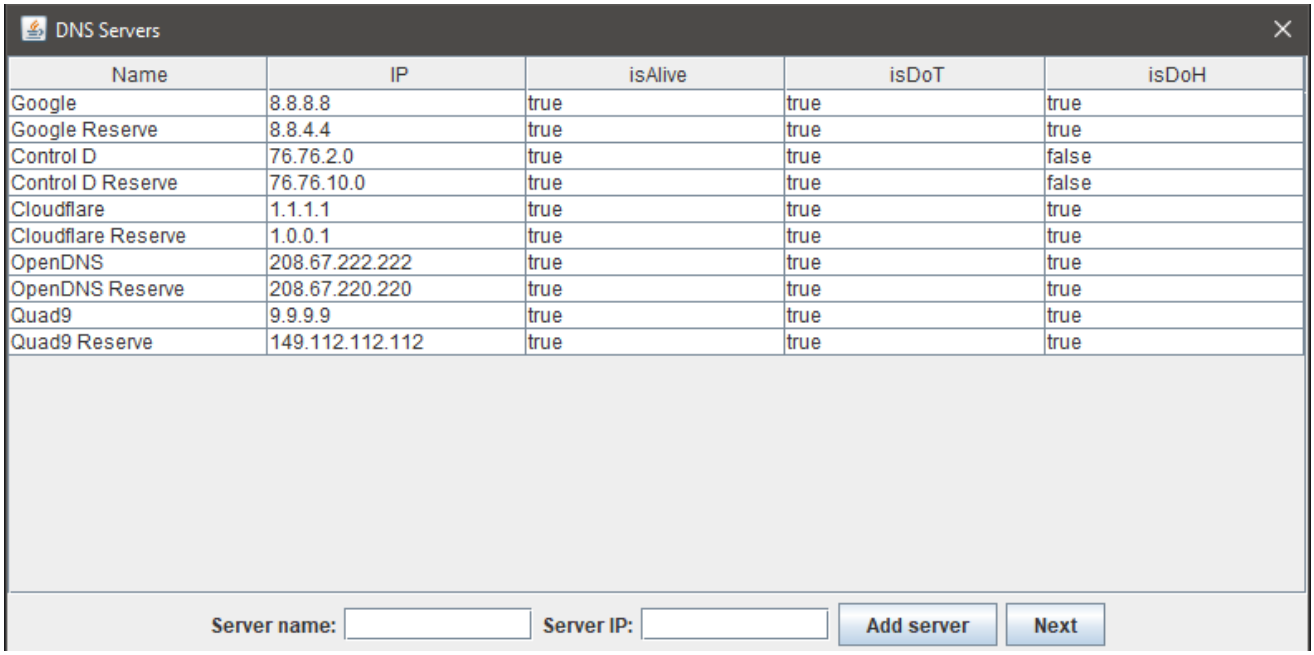


Рисунок 3.5 – Перелік (пул) тестованих DNS серверів

4) Вікно результатів, з можливістю сортування таблиці та пошуку по назві сервера (див. Додаток А, рис. А.1).

5) Вікно вводу даних для режиму гістограми (рис. 3.6):

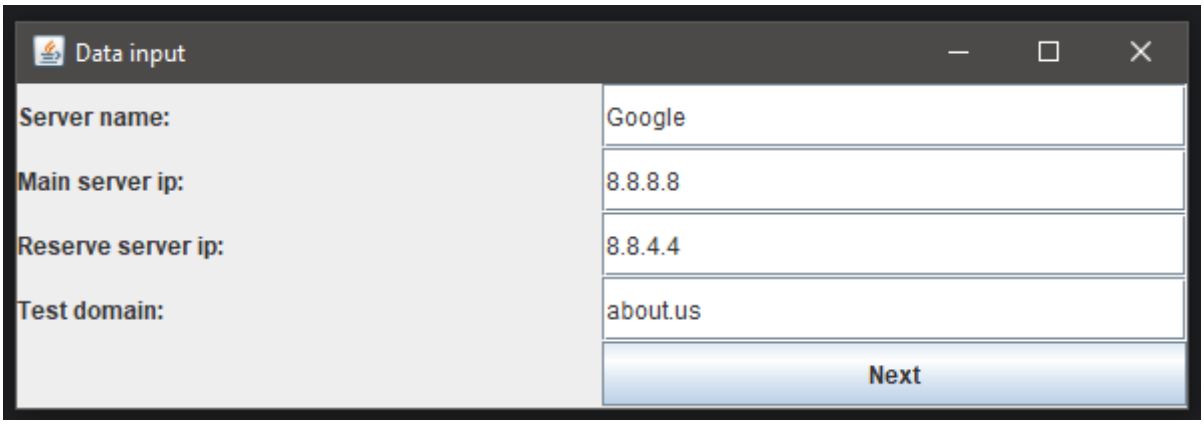


Рисунок 3.6 – Вікно вводу даних

6) Вікно із гістограмою, що оновлюється у реальному часі (рис. 3.7):

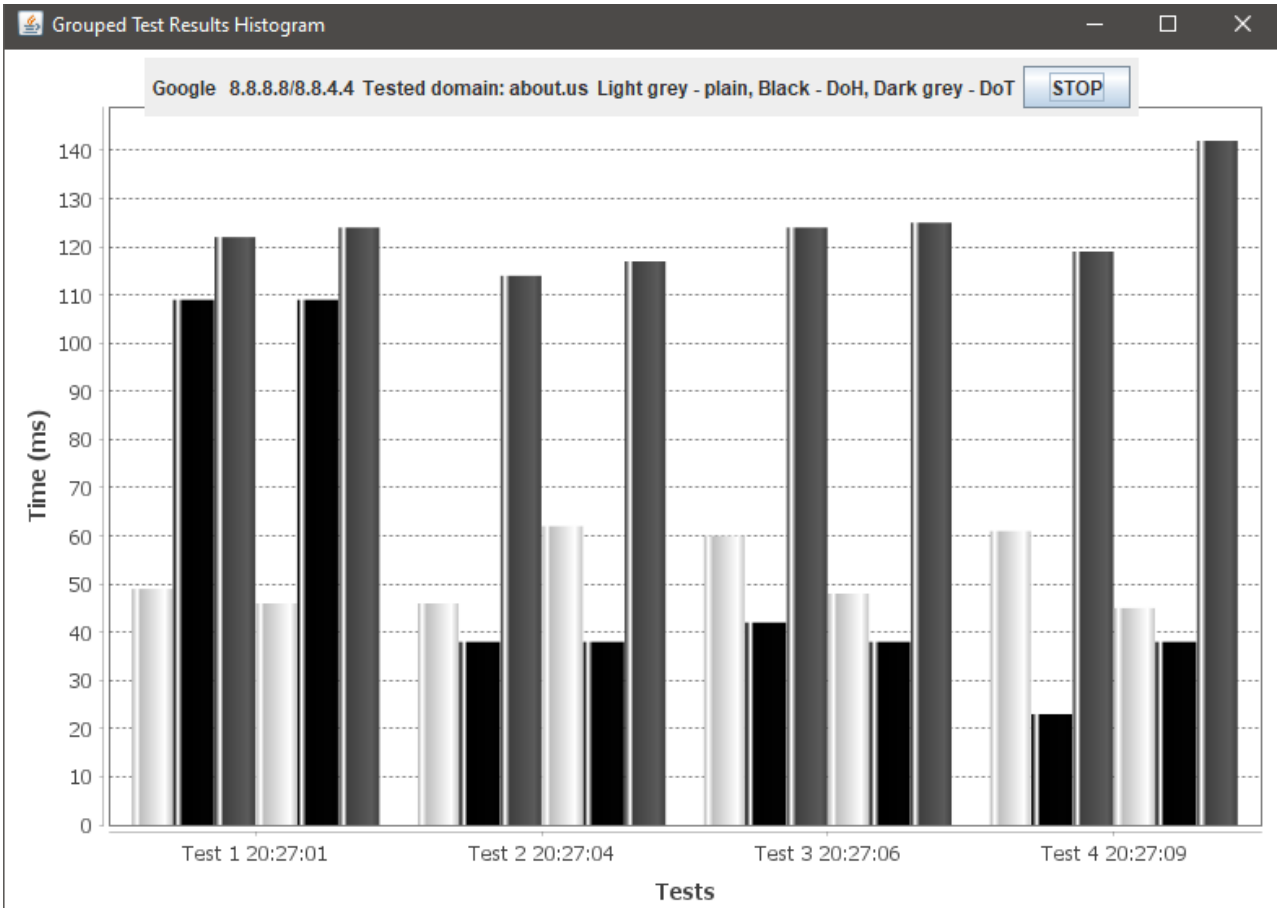


Рисунок 3.7 – Вікно гістограм тестових вимірів

3.4 Реалізація основних компонентів дослідної програми

Програма реалізована на мові програмування Java. Для виконання тестових вимірів використовується бібліотека JavaDNS, для візуалізації – бібліотеки *Swing* та *JfreeChart*.

3.4.1 Локальна складова ПЗ дослідного прототипу

Програма складається з наступних класів:

1) Клас DnsServer

Цей клас відповідає за тестування DNS серверу, має методи перевірки доступності сервера, підтримки протоколів DoT та DoH, а також замірювання часу виконання запитів.

При створення об’єкту класу DnsServer автоматично виконується перевірка доступності сервера через звичайний DNS запит, а також перевірка на підтримування протоколів шифрування.

Конструктор класу (див. рис. 3.8):

```
public DnsServer(String name,String ip) throws IOException {
  this.name = name;
  this.ip = ip;
  this.isAlive = testPlain();
  this.isDot = testDoT();
  this.isDoh = testDoH();
}
```

Рисунок 3.8 – Конструктор класу DnsServer

Клас включає 3 методи перевірки доступності: *testPlain* (див. рис. 3.9), *testDoT, testDoH*. Тут за вимірювання часу виконання запиту відповідають методи *plainPing*, *dotPing* та *dohPing*. Ці методи створюють та відправляють DNS-запит щодо домену, що вказаний першим аргументом *testDomain*, після цього приймають та зберігають відповідь DNS серверу, повертаючи час виконання операції. Другий аргумент *flag* визначає, чи буде налагоджувальна інформація виводитися у консоль.

```
  private boolean testPlain()
{
    try {

        SimpleResolver resolver = new SimpleResolver(ip);
        Lookup.setDefaultCache(new Cache(), DClass.IN);
        Lookup lookup = new Lookup("google.com", Type.A);
        lookup.setResolver(resolver);
        lookup.run();

        if (lookup.getResult() == Lookup.SUCCESSFUL) {
            return true;
        }
    }
    catch (Exception ignored){}
    return false;
}
```

Рисунок 3.9 – Метод *testPlain*

Метод *dotPing* наведено в Додатку А на рисунку А.2.

За виконання тестів відповідає метод *getTimes* (див. рис. 3.10), змінна *mode* визначає за яким протоколом буде здійснено запит.

```
public void getTimes(long lag, String testDomain, boolean moreData,
int mode) throws Exception {

    if(isAlive && mode==1){
    pq = plainPing(testDomain,moreData)-lag;
    }else pq = 0;

    if(isDoh && mode==2){
        dhq = dohPing(testDomain,moreData)-lag;
    }else dhq = 0;

    if(isDot && mode==3){
        dtq = dotPing(testDomain,moreData)-lag;
    }else dtq = 0;
}
```

Рисунок 3.10 – Метод getTimes

2) Клас TestResult

Цей клас інкапсулює дані, отримані при тестуванні DNS серверів та влючає наступні поля:

- name – ім’я сервера;
- testDomain - домен, що використовується при тестуванні;
- ip – IPv4 адреса DNS сервера;
- pq, dhq, dtq – час виконання незашифрованого запиту, запиту з використанням DoH та DoT відповідно;
- pRes, dhRes, dtRes – текстові відповіді на запити для кожного протоколу;
- timespamp, timespamp2 – час початку та закінчення тестування.

Конструктор класу *TestResult* наведено на рисунку 3.11.

```
public TestResult(String name,String testDomain, String ip, long pq,
long dhq, long dtq, String pRes, String dhRes, String dtRes, long
timestamp, long timestamp2) throws IOException {
    this.name = name;
    this.ip = ip;
    this.testDomain = testDomain;
    this.pq = pq;
    this.dhq = dhq;
    this.dtq = dtq;
    this.pRes = pRes;
    this.dhRes = dhRes;
    this.dtRes = dtRes;
    this.timestamp = timestamp;
    this.timestamp2  =timestamp2;
}
```

Рисунок 3.11 – Конструктор TestResult

3) Клас CloudHandler

Цей клас відповідає за взаємодію із хмарною функцією, що виконує DNS запити та повертає результати тестування. Він надає метод для відправки HTTP-запиту з параметрами тесту та обробки відповіді, перетворюючи її на об’єкт

TestResult. Головним методом цього класу є метод sendRequesToCloudFunction (див. рис. А.3), який виконує серіалізацію запиту та десеріалізацію відповіді у форматі JSON за допомогою бібліотеки Gson.

4) Клас DnsRequest

Цей клас призначений для інкапсуляції даних, необхідних для виконання DNS-запитів у хмарній функції.

5) Клас ModeSelectionFrame

Цей клас призначений для забезпечення графічного інтерфейсу, що дозволяє вибрати між режимами роботи програми. Після вибору режиму вікно закривається, а зміна mode оновлюється відповідно із обраним режимом.

6) Клас RealTimeHistogram

Цей клас призначений для візуалізації результатів тестування DNS-серверів у режимі реального часу. Інформація подається у вигляді гістограми, яка відображає результати тестів пари серверів – основного та резервного, для кожного типу запиту. Також, в поточній версії програми, передбачено можливість призупинення тестування за допомогою відповідної команди (кнопки) та панель із додатковою інформацією, такою як:

- назва серверу, що тестується;
- IP-адреса основного та резервного серверу;
- домен що використовується при проведенні тестів;
- легенда маркування окремих вимірів.

Гістограма оновлюється кожен раз, коли додаються нові результати за допомогою методу addGroupTestResult (див. Додаток, рис. А.4).

7) Клас InputFrame

Цей клас відповідає за графічний інтерфейс, у якому можна вказати домен для тестування, назву та IP-адреси основного та резервного серверів для подальшого використання у режимі гістограми.

8) Клас Main

Клас Main представляє собою основний модуль програми, що відповідає за координацію всіх компонентів. Клас Main можна умовно розбити на такі частини:

- Ініціалізація об’єктів та перемінних (див. рис. 3.12);
- логіка програми, що відповідає за режим гістограми;
- логіка програми, що відповідає за проведення основних тестів;
- вивід результатів тестування на екран та запис результатів у файли;

```
        int roundDelay = 600000;

        int pairDelay = 5000;

        CloudHandler cloudAmerica = new CloudHandler("https://us-
central1-primal-surfer-440715-n8.cloudfunctions.net/dnsUSA");

ArrayList<DnsServer> servers = new ArrayList<>();
ArrayList<TestResult> testResults = new ArrayList<>();

ArrayList<String> domains = new ArrayList<>();
domains.add("about.us");

servers.add(new DnsServer("Google", "8.8.8.8"));
servers.add(new DnsServer("Google Reserve", "8.8.4.4"));
```

Рисунок 3.12 – Приклад ініціалізації об’єктів у класі Main

Крім того, цей клас має наступні методи:

- showServerTable та showDomainTable – відповідають за вікна, що показують таблиці із серверами то доменами, які будуть використовуватися при тестуванні;
- showResultTable та populateTable – відповідають за відображення результатів;
- addPlainResultToList, addDohResultToList та addDotResultToList – відповідають за виклик відповідної хмарної функції та оновлення результатів тестування;
- addLocalPlainResult, addLocalDohResult та addLocalDotResult – відповідають за проведення локального тестування;

- writeResultsToFile та writeResultsToNotReadableFile – відповідають за запис результатів тестування у файли.

3.4.2 Хмарна складова дослідної програми

У якості хмарної складової програми було обрано платформу *Google Cloud*, на якій було розгорнуто 6 функцій з однаковим кодом, але різними географічними розташуваннями. В межах проведених досліджень відповідні хмарні складові програми були розгорнуті в таких регіонах (різних доменних зонах): – Сполучених Штатах Америки, Ізраїлі, Іспанії, Японії, Польщі та Чилі. Коди відповідних хмарних функцій побудовані на основі методів, що присутні у локальній частці програми, із додаванням можливостей прийому запитів та відправлення відповідей на головну консоль адміну.

3.5 Експорт результатів тестування

По закінченню роботи програма створює 14 текстових файлів, по 2 на кожну хмарну функцію та локальне тестування. Результати тестування записуються у 2-х форматах: – перший є зрозумілим для користувача та містить пояснення до даних (див. рис. 3.13), а другий складається виключно з даних (рис. 3.14), розділених комами та розривами рядка, що полегшує їх автоматичну обробку.

Як слід з прикладів лістингу звітів на рис. рис. 3.14 - 3.15, основна увага приділена контролю значень затримки, що згруповані по відповідним текстуальним блокам. Це дозволяє адміністратору з безпеки в потрібний йому момент (в т.ч. в режимі реального часу) здійснювати оцінку поточного стану справ та проводити відповідні коригування діючої RPZ [4], та/або корекцію налаштувань мережевого устаткування *(наприклад, роутерів і proxy серверів)* в частині настроювальних параметрів їх поточних DNS.

Таким чином, за рахунок наявності графічного інтерфейсу, легкості налаштування параметрів тестування та можливості подальшої обробки результатів прототип програми показав себе зручним інструментом для збору інформації щодо стану DNS-серверів.

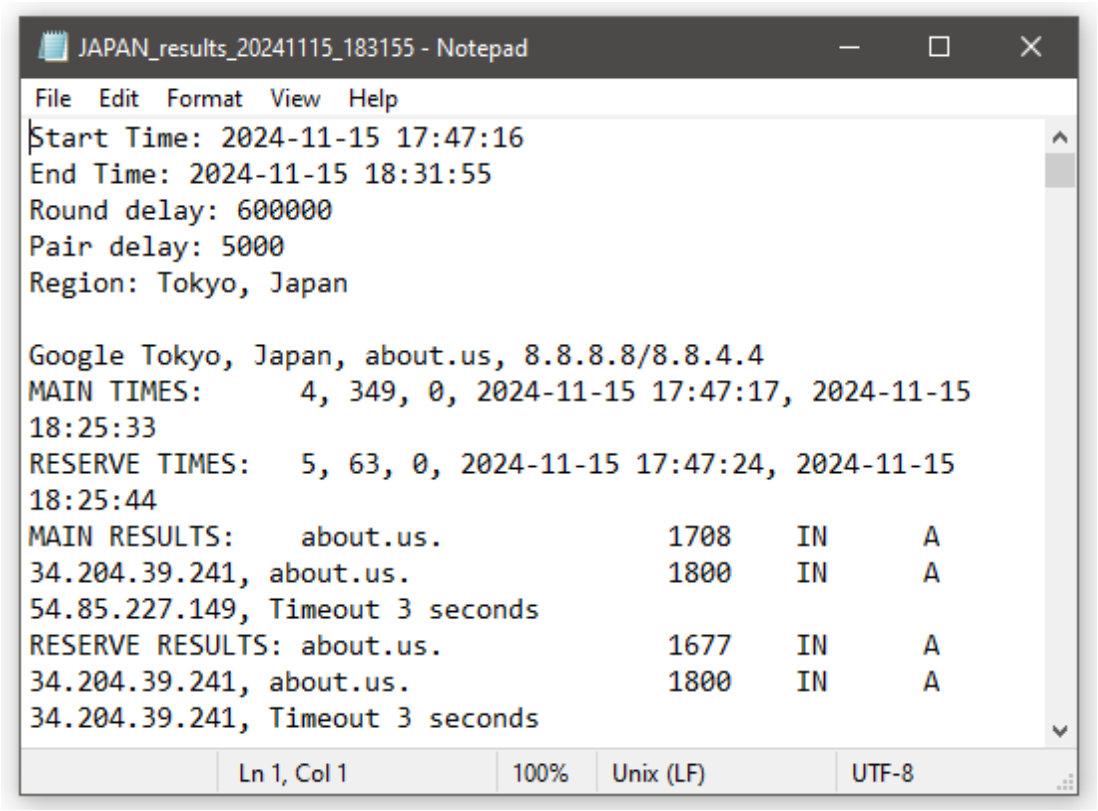


Рисунок 3.13 – Перший формат виводу результатів

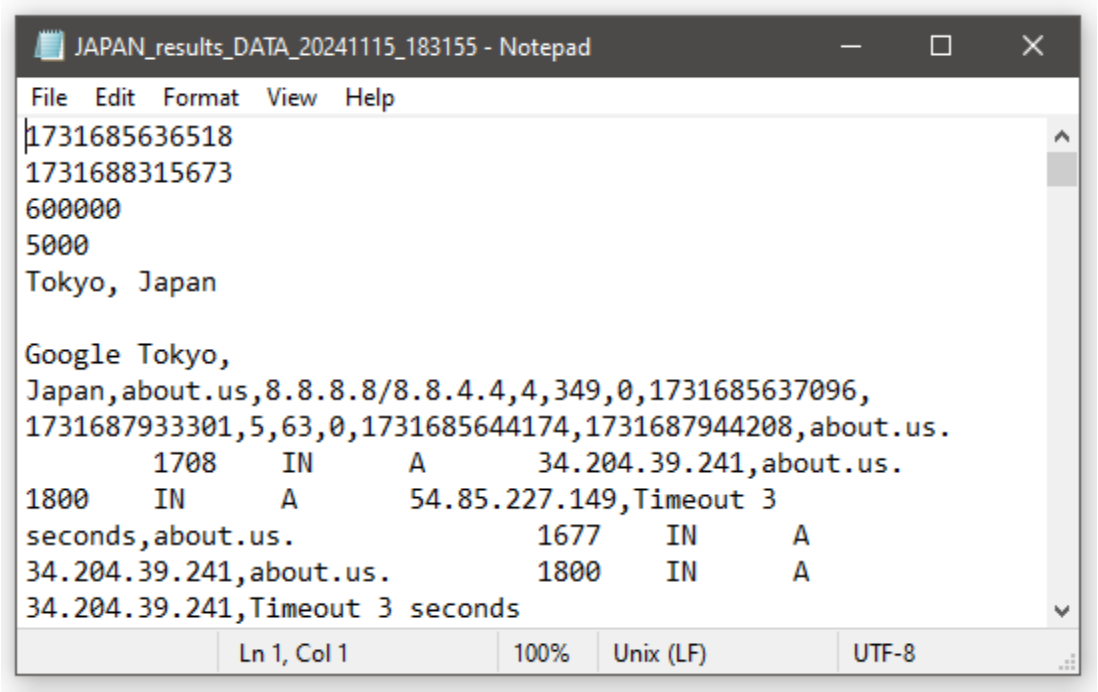


Рисунок 3.14 – Другий формат виводу результатів

4 АНАЛІЗ ТА УЗАГАЛЬНЕННЯ РЕЗУЛЬТАТІВ ВИМІРЮВАНЬ ДЛЯ РІЗНИХ ПАРАМЕТРІВ ДОСЛІДНИЦЬКОЇ ПРОГРАМИ

В рамках проведення досліджень було виконано 10 серій вимірювань протягом п’яти діб. Вимірювання проводилися двічі на день, орієнтовно о 18:00 та 00:00 за київським часом. Для тестування було обрано наступні DNS сервери разом із відповідними резервними серверами: – Google; Control D; Cloudflare; OpenDNS; Quad9.

Формовані моніторингові DNS запити включали домени, розташовані у різних доменних зонах. Для тестових вимірів було використано наступні ресурси:

- about.us - США;
- globo.com.br - Бразилія;
- gov.za - ПАР;
- bbc.com.uk – Велика Британія;
- work.ua - Україна;
- post.japanpost.jp - Японія;
- news.com.au - Австралія.

Крім того, за спрощеною схемою були встановлені наступні часові таймінги затримки вимірів: між серверами однієї пари – 5 секунд; між запитами за різними протоколами – 10 хвилин.

4.1 Результати доступності і часу реакції DNS серверів різних доменних зон

Завдяки зібраним програмою даним можна отримати оцінку продуктивності різних DNS серверів у різних регіонах. Далі буде представлено графіки, що відображають середній час затримки обраних серверів.

Згідно з гістограмами на рис.4.1, найкращі показники для незашифрованих запитів демонструють резервні сервери Control D, Cloudflare та Google.

Серед основних серверів найкращий середній час виконання має основний сервер Google. Для запитів через протокол DoH виділяються сервери Cloudflare (резервний) та Google (основний та резервний). У випадку використання протоколу DoT, найкращі результати спостерігаються на серверах Cloudflare (резервний) та Google (*як основний, так і резервний*).

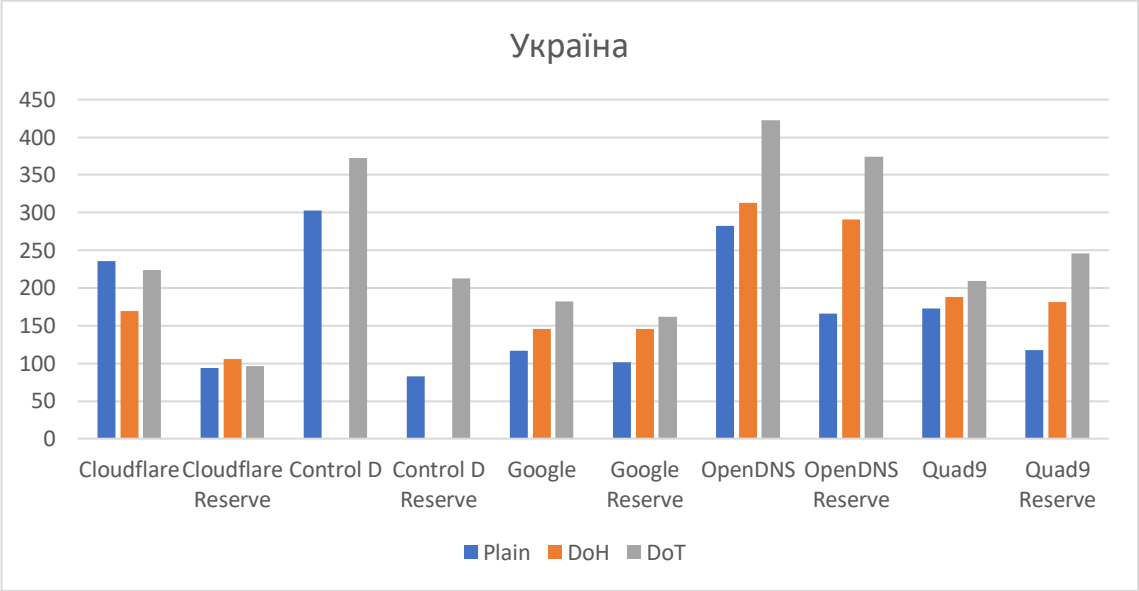


Рисунок 4.1 – Середній час затримки DNS серверів в Україні

Виходячи з рис. 4.2 можна сказати, що в регіоні «США» найкращі показники швидкості для незашифрованих запитів та запитів DoT демонструє резервний сервер Control D, а для DoH – сервери Google.

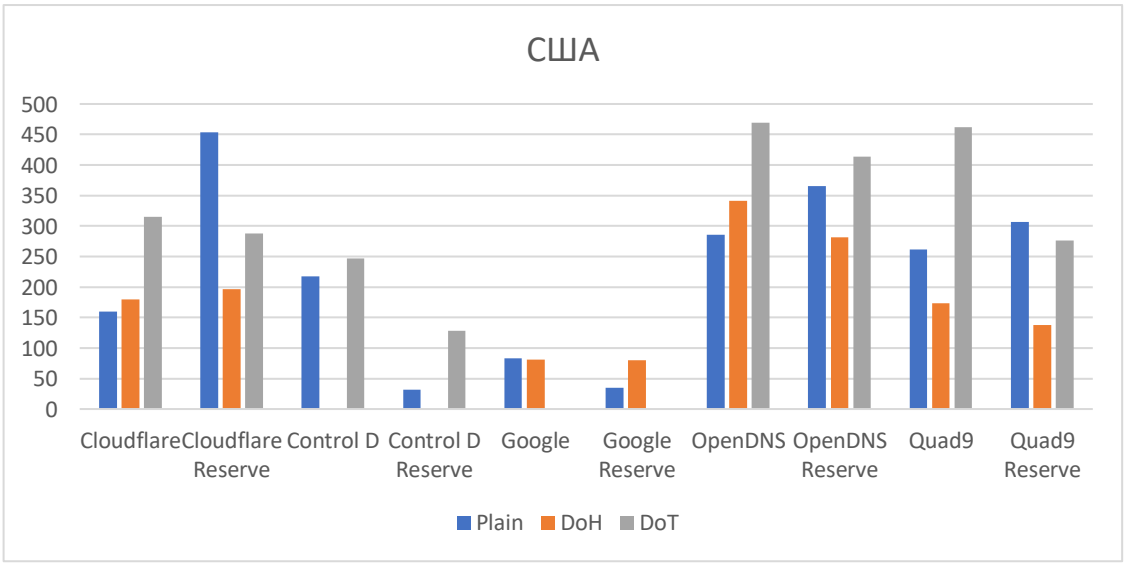


Рисунок 4.2 – Середній час затримки DNS серверів у США

Залежності, представлені на рис. 4.3 свідчать, що в регіоні «Чилі», найшвидшими у виконанні незашифрованих запитів є резервні сервери Control D та Quad9, хоча їх основні сервери демонструють одні з найповільніших результатів.

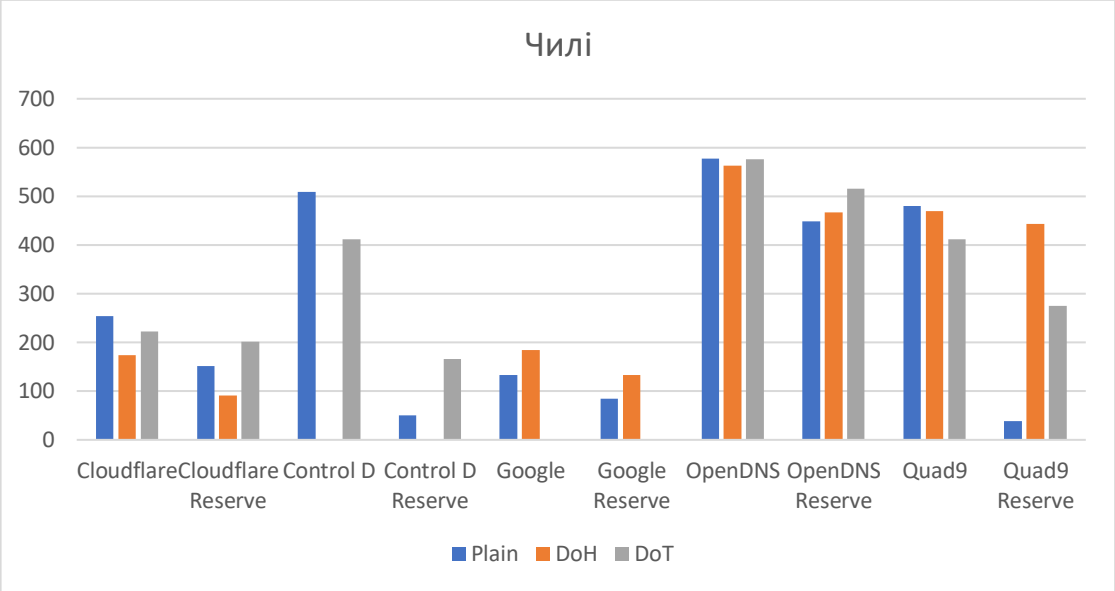


Рисунок 4.3 – Середній час затримки DNS серверів у Чилі

Аналіз рис. 4.4 показує, що в регіоні «Ізраїль» найкращі показники швидкості демонструють сервери Google та резервний сервер Control D.

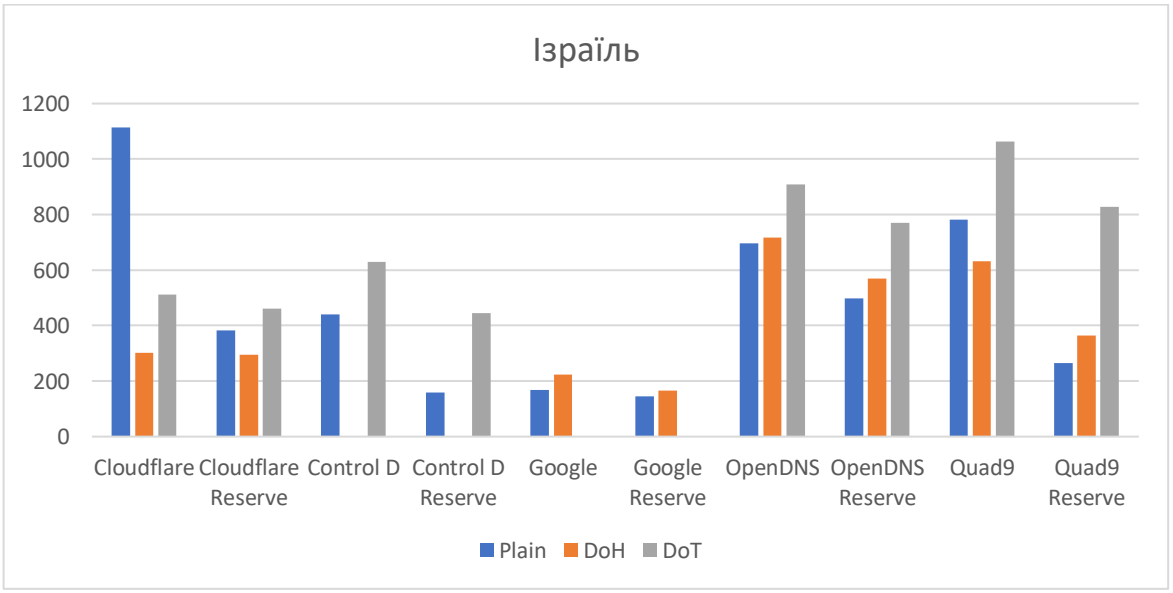


Рисунок 4.4 – Середній час затримки DNS серверів в Ізраїлі

Для регіонів Європи («Польща» (рис. 4.5) та «Іспанія» (рис. 4.6)), резервний сервер Control D має найкращі показники середнього часу виконання *незашифрованих* запитів. Однак для *зашифрованих* запитів (*DoT* та *DoH*) найшвидшими є сервери Google, Cloudflare та Quad9.

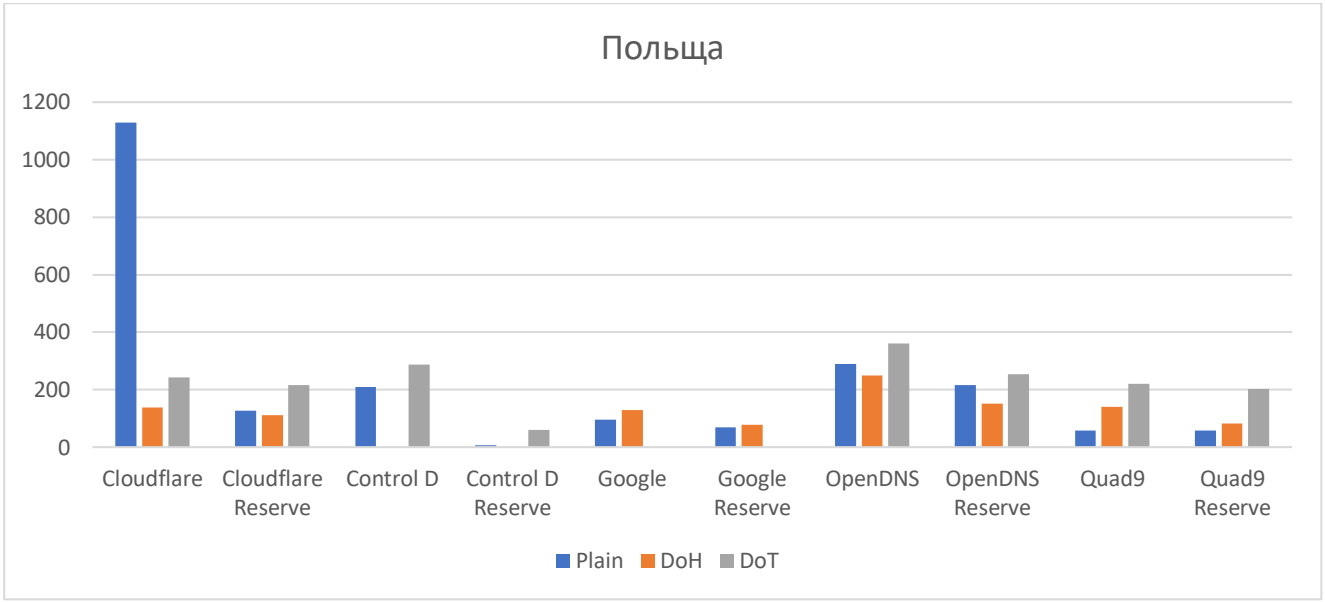


Рисунок 4.5 – Середній час затримки DNS серверів у Польщі

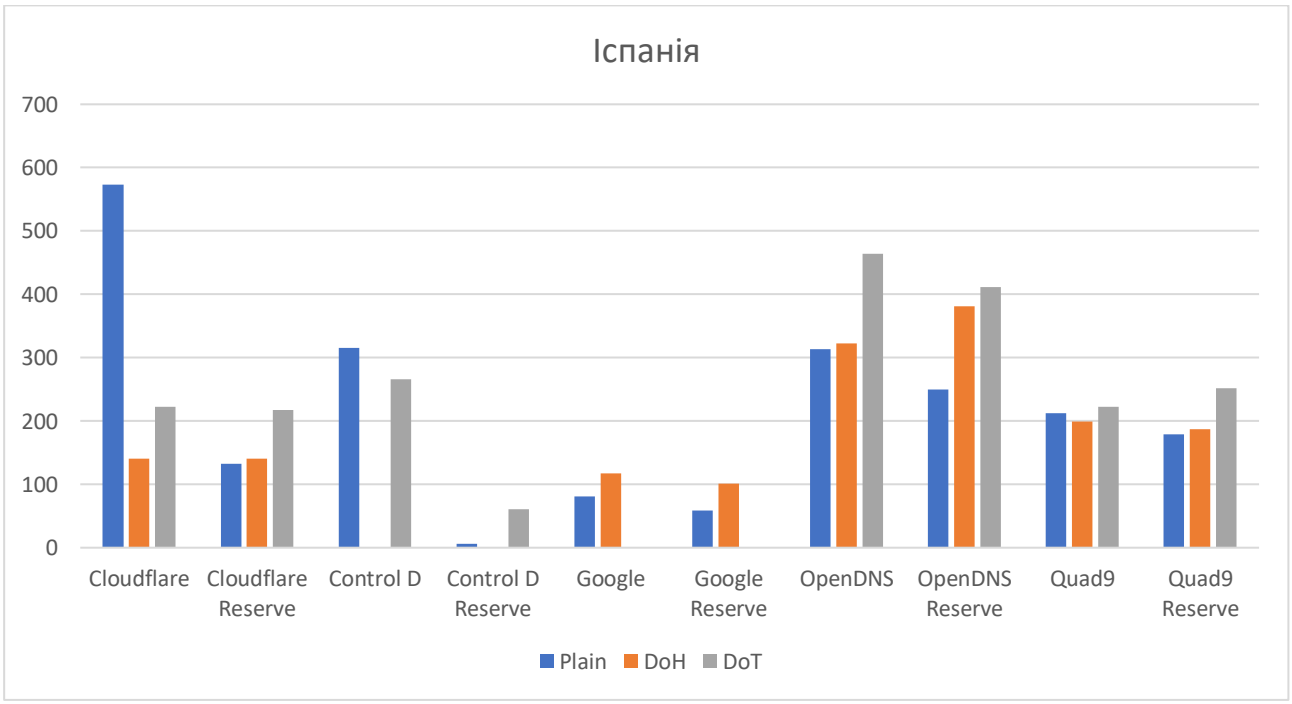


Рисунок 4.6 – Середній час затримки DNS серверів в Іспанії

У Тихоокеанському регіоні *(«Японія» (рис. 4.7))* найшвидшими у виконанні незашифрованих запитів є резервні сервери Control D та Quad9, для зашифрованих запитів найкращі результати продемонстрували сервери Cloudflare.

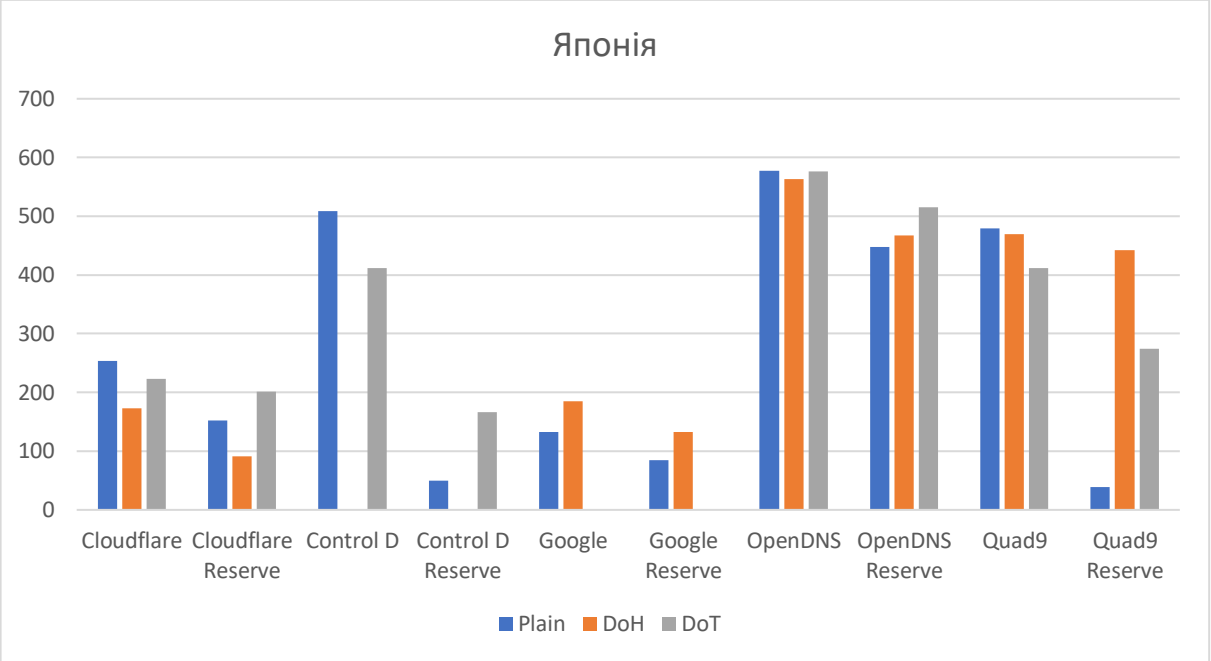


Рисунок 4.7 – Середній час затримки контрольованих DNS серверів в Японії

Окремо слід зазначити, що сервер *Cloudflare* демонструє схильність до різких збільшень затримок у випадках, коли запит надсилається до домену, який географічно віддалений від місця тестування. Водночас, цей сервер є одним із найшвидших при роботі із запитами до місцевих або популярних доменів, як ілюструється у таблиці 4.1.

Таблиця 4.1 – фрагмент даних локального тестування серверу *Cloudflare*

Назва серверу	**Домен**	**Час без шифр.**	**Час DoH**	**Час DoT**
Cloudflare	bbc.co.uk	*20*	*33*	*82*
Cloudflare Reserve	bbc.co.uk	*20*	*40*	*82*
Cloudflare	work.ua	*11*	*44*	*80*
Cloudflare Reserve	work.ua	*49*	*43*	*83*
Cloudflare	post.japanpost.jp	*945*	*648*	*122*
Cloudflare Reserve	post.japanpost.jp	*137*	*87*	*83*

Прим: авторські виміри.

4.2 Результати оцінки доступності і часу реакції DNS серверів в залежності від поточного часу доби (*в системі «Сервер DNS – Тестер»*)

Розглянемо залежності, що відображають середній час затримки пар серверів (*основний-резервний*) у різні часи виконання вимірювань у геолокації США (час, представлений на графіках, відповідає місцевому часу в США).

На рис. 4.8 сервери *Cloudflare* демонструють стабільно приємний час виконання, за винятком двох аномалій, що спостерігалися вранці 16 листопада та ввечері 19 листопада. Ці відхилення пов'язані із запитами до доменів, які знаходяться на значній відстані від зони тестування. Наприклад, запит до резервного серверу Cloudflare щодо домену *globo.com.br* ввечері 19 листопада зайняв 10425 мс.

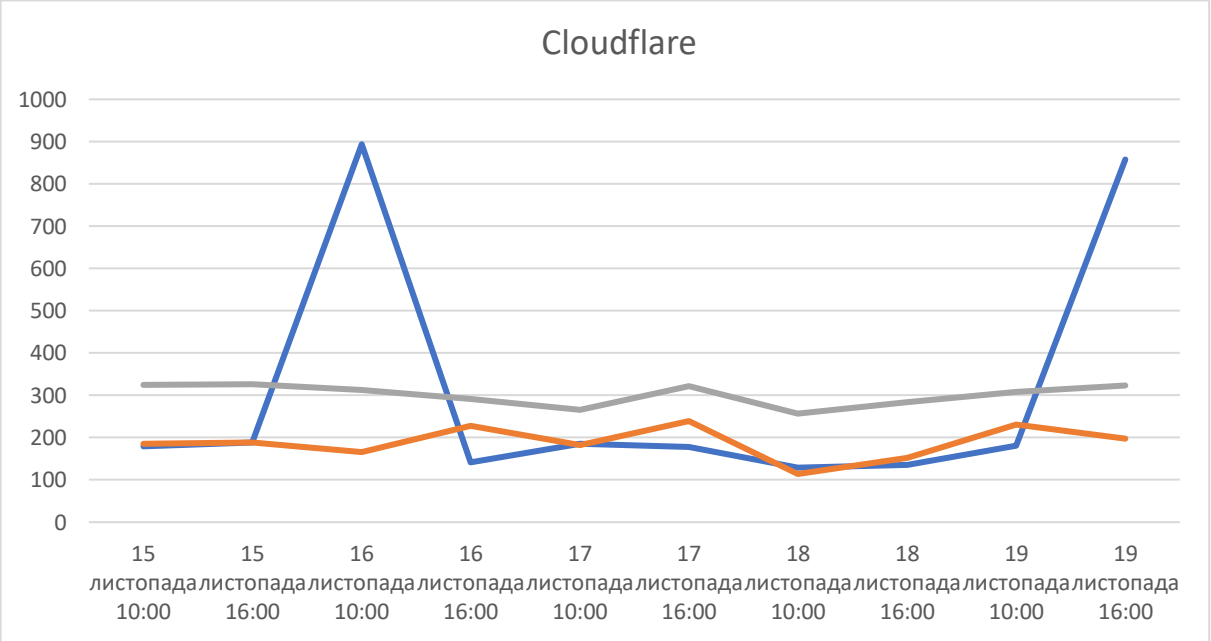


Рисунок 4.8 – Середній час затримки серверів *Cloudflare*

На рис. 4.9 сервери Google демонструють стабільно малий час виконання запитів, що свідчить про їх високу продуктивність і надійність (*принаймні на час виконання тестових вимірів*).

На рис. 4.10 сервери Control D демонструють стабільний час, за винятком аномалії 18.11.24, коли запит щодо домену *globo.com.br*, зайняв 4690мс.

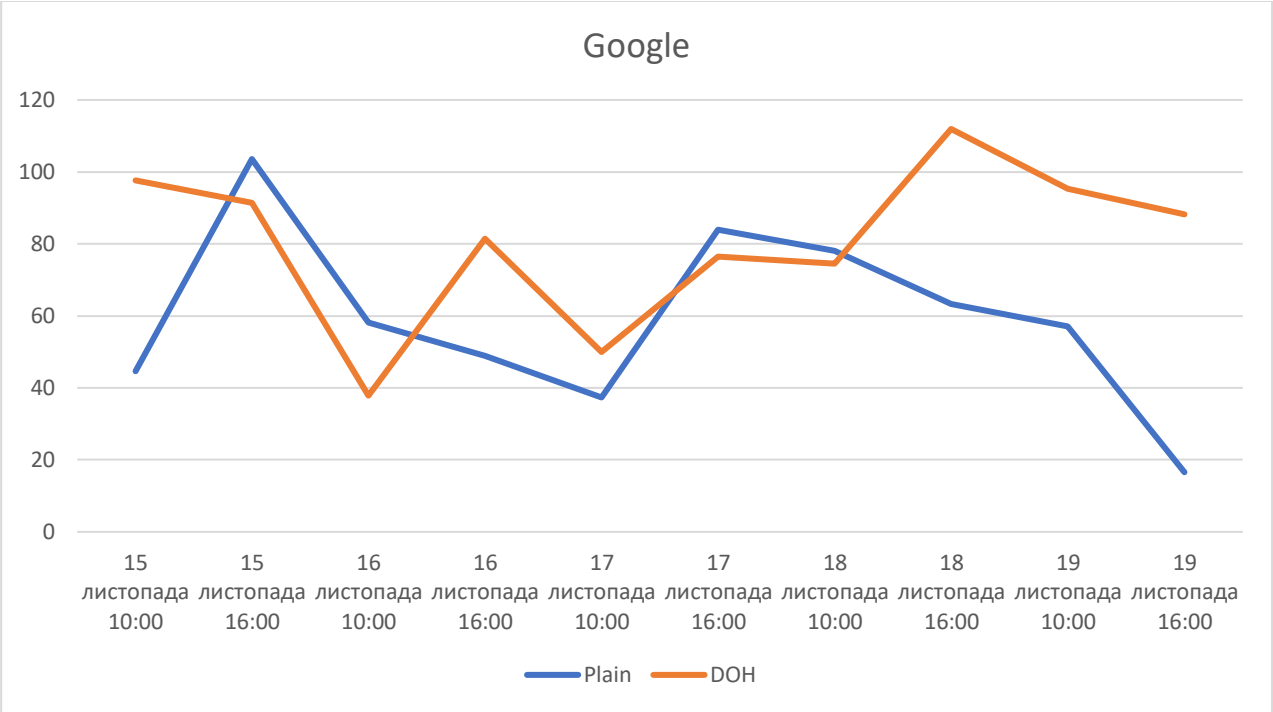


Рисунок 4.9 – Середній час затримки серверів Google

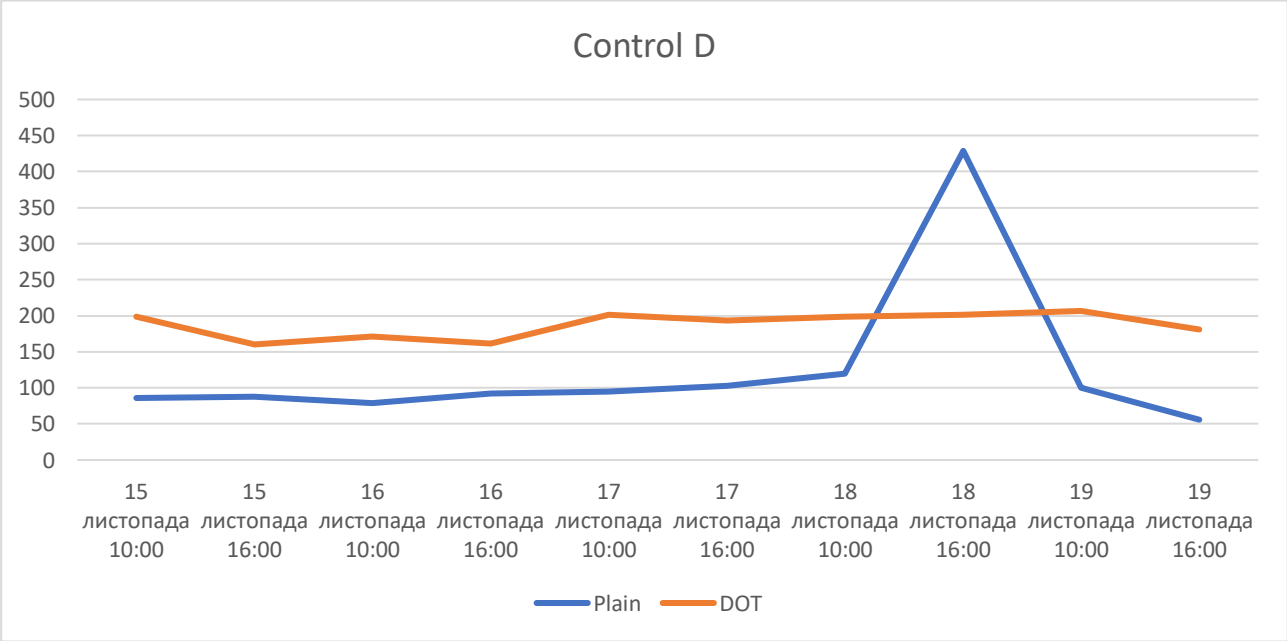


Рисунок 4.10 – Середній час затримки серверів Control D

На графіках рис.4.11, *Open DNS* демонструє збільшення затримки на вихідних 16-17 листопада, що може свідчити про нездатність постачальника послуги DNS компенсувати збільшену активність користувачів у ці дні.

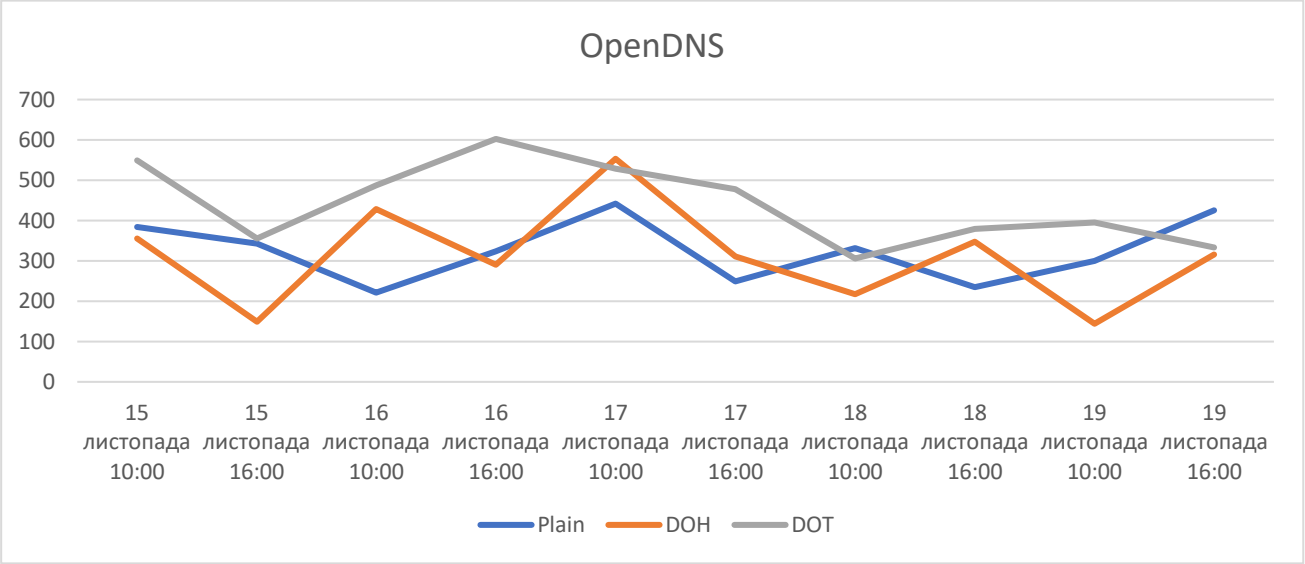


Рисунок 4.11 – Середній час затримки серверів OpenDNS

На рис. 4.12 сервери Quad9 демонструють стабільне значення затримки в усі дні, крім 16 листопада.

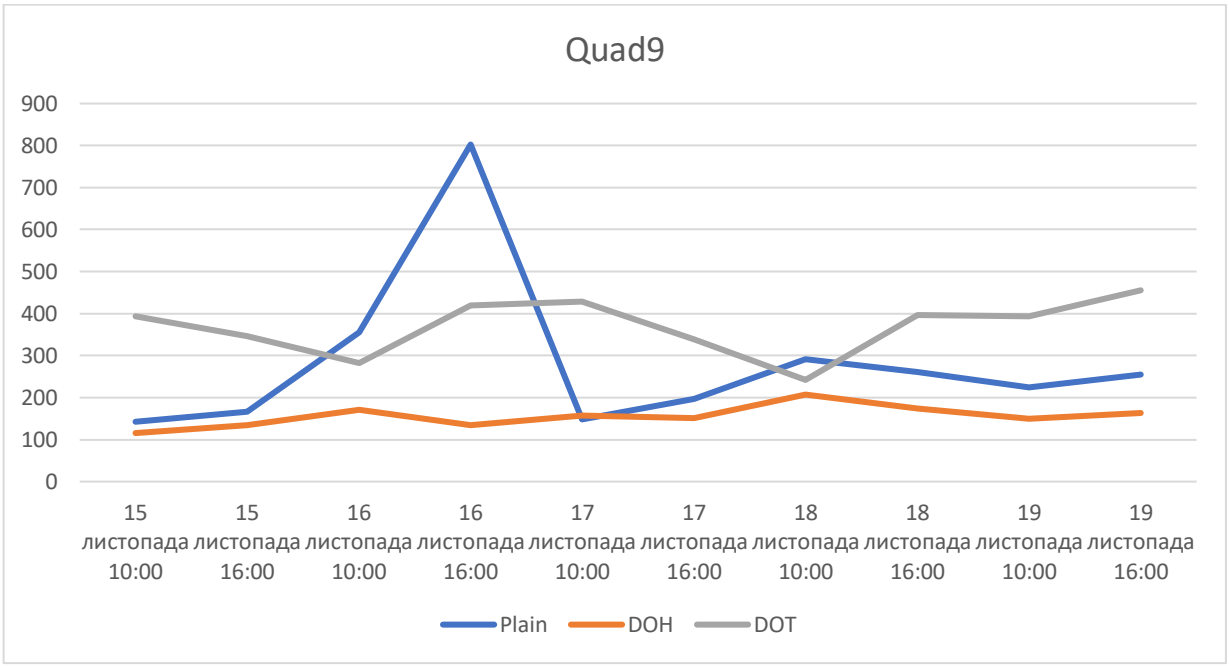


Рисунок 4.12 – Середній час затримки серверів Quad9

Стосовно отриманих залежностей (рис. 4.1-4.12) можна зробити наступні висновки:

1) Сервери Google демонструють найвищу стабільність серед усіх досліджених серверів DNS, незалежно від географічного розташування;
2) Хоча сервери Cloudflare можуть демонструвати найнижчі часи виконання запитів, вони є схильними до значних сплесків затримок, особливо при роботі з віддаленими доменами;
3) У багатьох випадках резервні сервери, такі як Control D або Quad9, демонструють кращі результати, ніж основні, що свідчить про можливість оптимізації вибору серверів залежно від потреб. У ситуаціях, коли критичною є швидкість відповіді, використання резервних серверів як основних може стати ефективним рішенням, особливо якщо попередній аналіз показує стабільно нижчі затримки порівняно з основними серверами.

Таким чином, в цілому, можна зробити висновок, що розроблена тестова версія дослідного програмного додатку є ефективним та зручним інструментом для збору інформації, щодо доступності й продуктивності обраної хмари DNS серверів. Діюча версія тестової програми дозволяє автоматизувати процес тестування, забезпечує підтримку різних схем/профілів моніторингу та різних протоколів шифрування DNS-запитів, що надає широкі можливості для порівняльного аналізу отриманих результатів. Зібрані дані можуть використовуватися для виявлення потенційних вразливостей та аномалій трафіку DNS, оцінки надійності DNS серверів у різних мережевих умовах (місцевий поточний час та геолокація) та визначення найбільш оптимальних станів RPZ для забезпечення швидкодії зворотних реакцій корпоративних засобів мережевого захисту та покращення поточного рівня безпеки при виконанні DNS-запитів.

ВИСНОВКИ

1) В роботі досліджено ключові аспекти функціонування системи доменних імен та розглянуті поширені атаки на неї, такі як DNS спуфінг, атака з DNS-посиленням, DNS rebinding і typosquatting. Встановлено, що успішність цих атак спирається на отримання жертвою зловмисних DNS-відповідей, що вказує на важливість DNS-фільрації, як складової безпеки сучасних інформаційних систем.

2) Виконано аналіз сучасні методики підвищення ефективності фільтрації DNS, серед яких – використання каналів розвідки загроз, шифрування DNS-запитів, зон політик реагування (RPZ) та методи виявлення алгоритмів генерації доменів (DGA) та серверів управління ботнетів. Однак, як показує аналіз існуючих досліджень, ці підходи все ще мають обмеження у виявленні нових і раніше невідомих загроз. Перспективним напрямом є інтеграція технологій штучного інтелекту та машинного навчання, які мають потенціал для подолання цих недоліків і побудови більш адаптивної системи кіберзахисту.

3) Розроблено прототип дослідної програми, що дозволяє аналізувати дані затримки DNS-серверів для трьох типів DNS-запитів (Незашифровані, DoH, DoT) у різних регіонах. Програма також забезпечує збереження та візуалізацію результатів, що полегшує здійснення аналізу даних.

4) Отримані результати моделювання дозволили оцінити ефективність DNS-серверів, які були протестовані, та виявити аномалії у затримках, зокрема значне зростання середнього часу відповіді деяких серверів при виконанні запитів до віддалених доменних зон.

5) Результати проведених досліджень можуть бути використанні для оптимізації налаштувань корпоративної DNS інфраструктури (адміністрування політик RPZ), вибору оптимальних, за часом та локацією, серверів та сприяти виконанню завдань, щодо виявлення аномалій та збоїв у роботі DNS-серверів.

6) Перспективами подальшої розробки програми є удосконалення користувацького інтерфейсу, впровадження режиму мультибазових хмарних вимірювань та створення інструментів для автоматичного аналізу вихідних даних, за рахунок залучення можливостей технології штучного інтелекту.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. What is DNS? Режим доступу: https://www.cloudflare.com/learning/dns/what-is-dns/
2. Погоріла, К., Лєсная, Ю., Богданова, Є., & Малахов, С. (2022). Соціальний інжиніринг, як фактор реалізації інсайдерських загроз. Scientific Collection «InterConf», (111): with the Proceedings of the 1st International Scientific and Practical Conference «Scientific Community: Interdisciplinary Research» (June 6-8, 2022). Boston, USA; С. 494-501. Режим доступу: https://archive.interconf.center/index.php/conference-proceeding/article/view/645/666
3. Лєсная, Ю., Малахов, С. Узагальнення основних передумов реалізації фішингових атак. Proceedings of the XVII International Scientific and Practical Conference. Ankara, Turkey. 2023. С. 453-457. Режим доступу: https://isg-konf.com/wp-content/uploads/2023/05/SYSTEM-ANALYSIS-AND-INTELLIGENT-SYSTEMS-FOR-MANAGEMENT.pdf
4. Danylo Chepel, & Serhii Malakhov. (2024). Summary of DNS traffic filtering trends as a component of modern information systems security. *Computer Science and Cybersecurity*, (1), 6-21. Режим доступу: https://doi.org/10.26565/2519-2310-2024-1-01
5. DNS history. When and why was DNS created? Режим доступу: https://www.cloudns.net/blog/dns-history-creation-first/
6. What is DNS Hierarchy Architecture with Examples (Explained). Режим доступу: https://cloudinfrastructureservices.co.uk/what-is-dns-hierarchy/
7. DNS Hierarchy. Режим доступу: https://www.inetdaemon.com/tutorials/internet/dns/operation/hierarchy.shtml
8. DNS Record types. Режим доступу: https://simpledns.plus/help/dns-record-types
9. What is DNS Spoofing? Режим доступу: https://www.myrasecurity.com/en/dns-spoofing/

10. What Is a DNS Amplification Attack? Режим доступу: https://www.checkpoint.com/cyber-hub/network-security/what-is-dns-security/what-is-a-dns-amplification-attack/
11. What is DNS Rebinding? Режим доступу: https://www.geeksforgeeks.org/what-is-dns-rebinding/
12. What is typosquatting? Режим доступу: https://support.microsoft.com/en-us/topic/what-is-typosquatting-54a18872-8459-4d47-b3e3-d84d9a362eb0
13. What is DNS filtering? Режим доступу: https://www.cloudflare.com/learning/access-management/what-is-dns-filtering/
14. Яремчук, К., Воскобойников, Д., & Мелкозьорова, О. (2022). Сучасні загрози та способи забезпечення безпеки веб-застосунків. *Комп'ютерні науки та кібербезпека*, (2), 28-34. Режим доступу: https://doi.org/10.26565/2519-2310-2022-2-03
15. Богданова, Є., Чорна, Т., & Малахов, С. (2022). Огляд поточного стану загроз, що обумовлені впливом експлойтів. *Комп'ютерні науки та кібербезпека*, (2), 35-40. Режим доступу: https://doi.org/10.26565/2519-2310-2022-2-04
16. Кохановська, Т., Нарежний, О., & Дьяченко, О. (2020). Дослідження можливостей технології Honeypot. Комп'ютерні науки та кібербезпека, 1(1), 33-42. Режим доступу: https://doi.org/10.26565/2519-2310-2020-1-03
17. Михайленко Д., Нємцев М. Особливості технології мережевих пасток як інструменту активного захисту та аналізу дій атакуючої сторони. Proceedings of the XXI International Scientific and Practical Conference. Melbourne, Australia. 2023. С. 483-487. Режим доступу: https://isg-konf.com/wp-content/uploads/2023/05/Scientists-and-methods-of-using-modern-technologies.pdf
18. Січкар, М., & Малахов, С. (2024). Узагальнення особливостей відомих засобів міжмережевого екранування. Proceedings of the XXI International

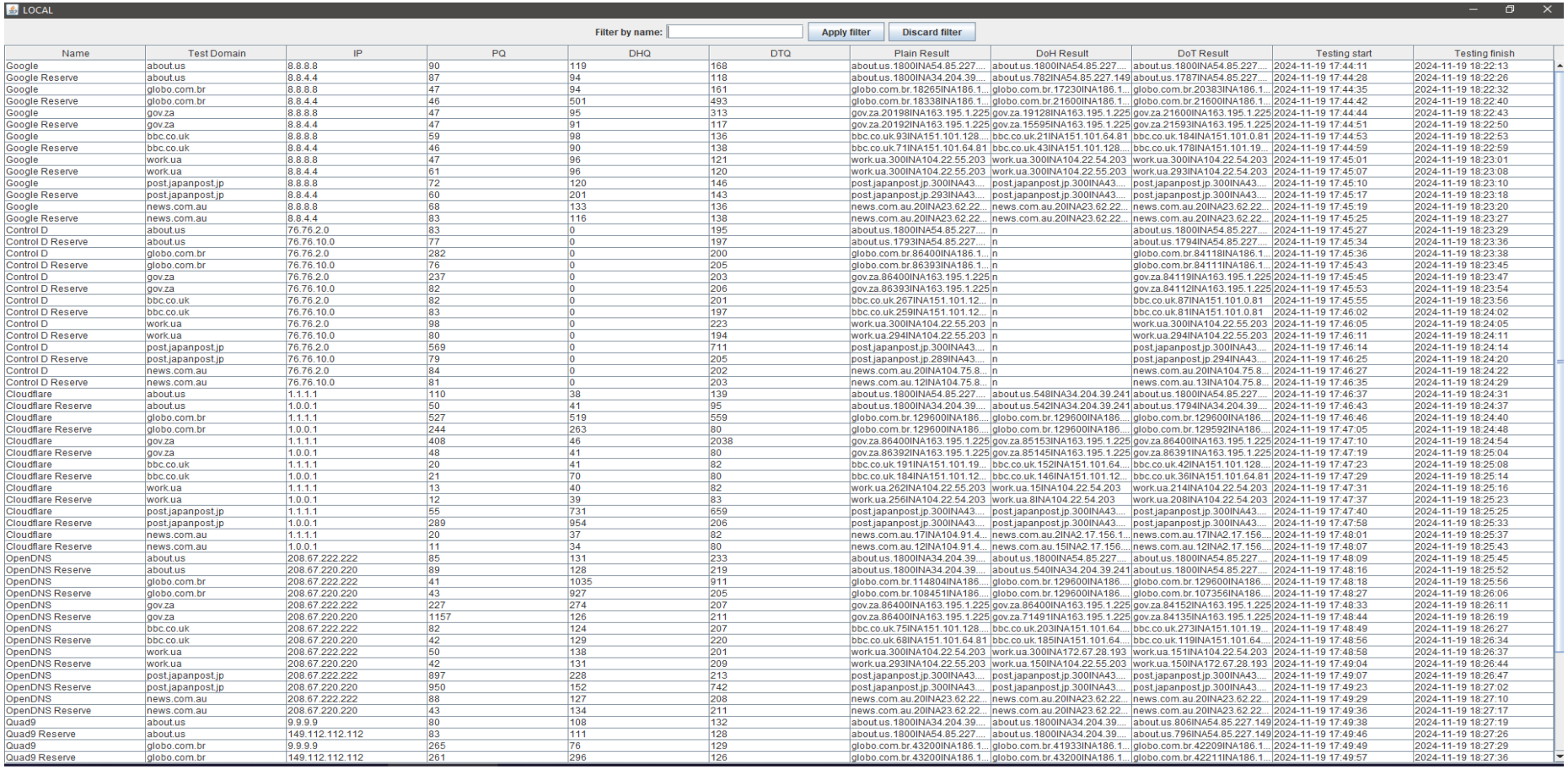
Scientific and Practical Conference. Sofia, Bulgaria. 2024. С. 370-376. Режим доступу: https://isg-konf.com/uk/innovative-solutions-in-public-communications-and-international-relations/

19. What is a Threat Intelligence Feed? Режим доступу: https://www.crowdstrike.com/cybersecurity-101/threat-intelligence/threat-intelligence-feeds/
20. Guofei Gu, Junjie Zhang & Wenke Lee. BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic. Режим доступу: https://people.engr.tamu.edu/guofei/paper/Gu_NDSS08_botSniffer.pdf
21. How Does a DNS Amplification Attack Work? (2024). Check Point. Режим доступу: https://www.checkpoint.com/cyber-hub/network-security/what-is-dns-security/what-is-a-dns-amplification-attack/
22. Albulayhi, K., Smadi, A., Sheldon, F., & Abercrombie, R. (2021). IoT Intrusion Detection Taxonomy, Reference Architecture, and Analyses. Sensors, (6432), 21. Режим доступу: https://doi.org/10.3390/s21196432
23. Коробейнікова, Т., & Федчук, Т. (2024). Огляд протоколів DNS, DOH та DOT. Collection of Scientific Papers «ΛΌГОΣ», (March 1, 2024; Paris, France), 253–256. Режим доступу: https://doi.org/10.36074/logos-01.03.2024.056
24. Гайкова, В., & Малахов, С. (2021). Аналіз факторів і умов реалізації кібербулінгу з урахуванням можливостей сучасних інформаційних систем. *Комп'ютерні науки та кібербезпека*, (1), 50-59. Режим доступу: https://doi.org/10.26565/2519-2310-2021-1-04
25. Hugo M. Connery. DNS Response Policy Zones History, Overview, Usage and Research. Режим доступу: https://www.dnsrpz.info/RPZ-History-Usage-Research.pdf
26. What Are Domain Generation Algorithms? Режим доступу: https://www.akamai.com/glossary/what-are-dgas

27. Anhar Haneef. On the Scalable Generation of Cyber Threat Intelligence from Passive DNS Streams. Режим доступу: http://surl.li/phbham
28. Keijo Korte. Measuring the quality of Open Source Cyber Threat Intelligence Feeds. Режим доступу: http://surl.li/yhiqoe
29. Vector Guo Li, Matthew Dunn, Paul Pearce, Damon McCoy, Geoffrey M. Voelker, Stefan Savage, Kirill Levchenko. Reading the Tea Leaves: A Comparative Analysis of Threat Intelligence. Режим доступу: https://www.usenix.org/system/files/sec19-li-vector_guo.pdf
30. Constantinos Patsakis, Fran Casino. Exploiting Statistical and Structural Features for the Detection of Domain Generation Algorithms. Режим доступу: https://arxiv.org/pdf/1912.05849
31. Joewie J. Koh, Barton Rhodes. Inline Detection of Domain Generation Algorithms with Context-Sensitive Word Embeddings. Режим доступу: https://arxiv.org/pdf/1811.08705
32. Amara Dinesh Kumar, Harish Thodupunoori, R. Vinayakumar, K. P. Soman, Prabaharan Poornachandran, Mamoun Alazab, and Sitalakshmi Venkatraman. Enhanced Domain Generating Algorithm Detection Based on Deep Neural Networks. Режим доступу: http://surl.li/sgufmu
33. Minzhao Lyu, Hassan Habibi Gharakheili, Vijay Sivaraman. A Survey on DNS Encryption: Current Development, Malware Misuse, and Inference Techniques. Режим доступу: https://arxiv.org/pdf/2201.00900
34. Chaoyi Lu, Baojun Liu, Zhou Li, Shuang Hao, Haixin Duan, Mingming Zhang, Chunying Leng, Ying Liu, Zaifeng Zhang & Jianping Wu. An End-to-End, Large-Scale Measurement of DNS-over-Encryption: How Far Have We Come? Режим доступу: http://surl.li/ebouep
35. Sandra Siby, Marc Juarez, Claudia Diaz, Narseo Vallina-Rodriguez, Carmela Troncoso. Encrypted DNS - Privacy? A Traffic Analysis Perspective. Режим доступу: https://arxiv.org/abs/1906.09682

36. Hugo M. Connery. DNS Response Policy Zones History, Overview, Usage and Research. Режим доступу: https://www.dnsrpz.info/RPZ-History-Usage-Research.pdf
37. Скибун, О. (2023). Фішинг та фішери в сучасному світі. Grail of Science, (23), 259–264. Режим доступу: https://doi.org/10.36074/grail-of-science.23.12.2022.38
38. Hikaru Ichise, Yong Jin & Katsuyoshi Iida. Policy-based Detection and Blocking System for Abnormal Direct Outbound DNS Queries using RPZ. Режим доступу: https://eprints.lib.hokudai.ac.jp/dspace/handle/2115/86951
39. Kamal Alieyan, Ammar Almomani, Ahmad Manasrah, Mohammed M. Kadhum. A survey of botnet detection based on DNS. Режим доступу: http://surl.li/vqinqn
40. Hyunsang Choi, Hanwoo Lee, Heejo Lee, Hyogon Kim. Botnet Detection by Monitoring Group Activities in DNS Traffic. Режим доступу: http://surl.li/nbbypc
41. David Zhao, Issa Traore, Bassam Sayed, Wei Lu, Sherif Saad, Ali Ghorbani, Dan Garant. Botnet detection based on traffic behavior analysis and flow intervals. Режим доступу: http://surl.li/ydwehw
42. Chepel D., Malakhov S., Kolovanova Ie. An overview of the capabilities of dns filtering as a security tool for modern information systems. Режим доступу: https://archive.journal-grail.science/index.php/2710-3056/issue/view/02.08.2024/30

ДОДАТОК А



Name	Test Domain	IP	PQ	DHQ	DTQ	Plain Result	DoH Result	DoT Result	Testing start	Testing finish
Google	about.us	8.8.8.8	90	119	168	about.us.1800INA54.85.227....	about.us.1800INA54.85.227....	about.us.1800INA54.85.227....	2024-11-19 17:44:11	2024-11-19 18:22:13
Google Reserve	about.us	8.8.4.4	87	94	118	about.us.1800INA34.204.39....	about.us.782INA54.85.227.149	about.us.1787INA54.85.227....	2024-11-19 17:44:28	2024-11-19 18:22:26
Google	globo.com.br	8.8.8.8	47	94	161	globo.com.br.18265INA186.1...	globo.com.br.17230INA186.1...	globo.com.br.20383INA186.1...	2024-11-19 17:44:35	2024-11-19 18:22:32
Google Reserve	globo.com.br	8.8.4.4	46	501	493	globo.com.br.18338INA186.1...	globo.com.br.21600INA186.1...	globo.com.br.21600INA186.1...	2024-11-19 17:44:42	2024-11-19 18:22:40
Google	gov.za	8.8.8.8	47	95	313	gov.za.20198INA163.195.1.225	gov.za.19128INA163.195.1.225	gov.za.21600INA163.195.1.225	2024-11-19 17:44:44	2024-11-19 18:22:43
Google Reserve	gov.za	8.8.4.4	47	91	117	gov.za.20192INA163.195.1.225	gov.za.15595INA163.195.1.225	gov.za.21593INA163.195.1.225	2024-11-19 17:44:51	2024-11-19 18:22:50
Google	bbc.co.uk	8.8.8.8	59	98	136	bbc.co.uk.93INA151.101.128....	bbc.co.uk.21INA151.101.64.81	bbc.co.uk.184INA151.101.0.81	2024-11-19 17:44:53	2024-11-19 18:22:53
Google Reserve	bbc.co.uk	8.8.4.4	46	90	138	bbc.co.uk.71INA151.101.64.81	bbc.co.uk.43INA151.101.128....	bbc.co.uk.178INA151.101.19...	2024-11-19 17:44:59	2024-11-19 18:22:59
Google	work.ua	8.8.8.8	47	96	121	work.ua.300INA104.22.55.203	work.ua.300INA104.22.54.203	work.ua.300INA104.22.54.203	2024-11-19 17:45:01	2024-11-19 18:23:01
Google Reserve	work.ua	8.8.4.4	61	96	120	work.ua.300INA104.22.55.203	work.ua.300INA104.22.55.203	work.ua.293INA104.22.54.203	2024-11-19 17:45:07	2024-11-19 18:23:08
Google	post.japanpost.jp	8.8.8.8	72	120	146	post.japanpost.jp.300INA43....	post.japanpost.jp.300INA43....	post.japanpost.jp.300INA43....	2024-11-19 17:45:10	2024-11-19 18:23:10
Google Reserve	post.japanpost.jp	8.8.4.4	60	201	143	post.japanpost.jp.293INA43....	post.japanpost.jp.300INA43....	post.japanpost.jp.300INA43....	2024-11-19 17:45:17	2024-11-19 18:23:18
Google	news.com.au	8.8.8.8	68	133	136	news.com.au.20INA23.62.22...	news.com.au.20INA23.62.22...	news.com.au.20INA23.62.22...	2024-11-19 17:45:19	2024-11-19 18:23:20
Google Reserve	news.com.au	8.8.4.4	83	116	138	news.com.au.20INA23.62.22...	news.com.au.20INA23.62.22...	news.com.au.20INA23.62.22...	2024-11-19 17:45:25	2024-11-19 18:23:27
Control D	about.us	76.76.2.0	83	0	195	about.us.1800INA54.85.227....	n	about.us.1800INA54.85.227....	2024-11-19 17:45:27	2024-11-19 18:23:29
Control D Reserve	about.us	76.76.10.0	77	0	197	about.us.1793INA54.85.227....	n	about.us.1794INA54.85.227....	2024-11-19 17:45:34	2024-11-19 18:23:36
Control D	globo.com.br	76.76.2.0	282	0	200	globo.com.br.86400INA186.1...	n	globo.com.br.84118INA186.1...	2024-11-19 17:45:36	2024-11-19 18:23:38
Control D Reserve	globo.com.br	76.76.10.0	76	0	205	globo.com.br.86393INA186.1...	n	globo.com.br.84111INA186.1...	2024-11-19 17:45:43	2024-11-19 18:23:45
Control D	gov.za	76.76.2.0	237	0	203	gov.za.86400INA163.195.1.225	n	gov.za.84119INA163.195.1.225	2024-11-19 17:45:45	2024-11-19 18:23:47
Control D Reserve	gov.za	76.76.10.0	82	0	206	gov.za.86393INA163.195.1.225	n	gov.za.84112INA163.195.1.225	2024-11-19 17:45:53	2024-11-19 18:23:54
Control D	bbc.co.uk	76.76.2.0	82	0	201	bbc.co.uk.267INA151.101.12...	n	bbc.co.uk.87INA151.101.0.81	2024-11-19 17:45:55	2024-11-19 18:23:56
Control D Reserve	bbc.co.uk	76.76.10.0	83	0	197	bbc.co.uk.259INA151.101.12...	n	bbc.co.uk.81INA151.101.0.81	2024-11-19 17:46:02	2024-11-19 18:24:02
Control D	work.ua	76.76.2.0	98	0	223	work.ua.300INA104.22.55.203	n	work.ua.300INA104.22.55.203	2024-11-19 17:46:05	2024-11-19 18:24:05
Control D Reserve	work.ua	76.76.10.0	80	0	194	work.ua.294INA104.22.55.203	n	work.ua.294INA104.22.55.203	2024-11-19 17:46:11	2024-11-19 18:24:11
Control D	post.japanpost.jp	76.76.2.0	569	0	711	post.japanpost.jp.300INA43....	n	post.japanpost.jp.300INA43....	2024-11-19 17:46:14	2024-11-19 18:24:14
Control D Reserve	post.japanpost.jp	76.76.10.0	79	0	205	post.japanpost.jp.289INA43....	n	post.japanpost.jp.294INA43....	2024-11-19 17:46:25	2024-11-19 18:24:20
Control D	news.com.au	76.76.2.0	84	0	202	news.com.au.20INA104.75.8...	n	news.com.au.20INA104.75.8...	2024-11-19 17:46:27	2024-11-19 18:24:22
Control D Reserve	news.com.au	76.76.10.0	81	0	203	news.com.au.12INA104.75.8...	n	news.com.au.13INA104.75.8...	2024-11-19 17:46:35	2024-11-19 18:24:29
Cloudflare	about.us	1.1.1.1	110	38	139	about.us.1800INA54.85.227....	about.us.548INA34.204.39.241	about.us.1800INA54.85.227....	2024-11-19 17:46:37	2024-11-19 18:24:31
Cloudflare Reserve	about.us	1.0.0.1	50	41	95	about.us.1800INA34.204.39....	about.us.542INA34.204.39.241	about.us.1794INA34.204.39....	2024-11-19 17:46:43	2024-11-19 18:24:37
Cloudflare	globo.com.br	1.1.1.1	527	519	559	globo.com.br.129600INA186....	globo.com.br.129600INA186....	globo.com.br.129600INA186....	2024-11-19 17:46:46	2024-11-19 18:24:40
Cloudflare Reserve	globo.com.br	1.0.0.1	244	263	80	globo.com.br.129600INA186....	globo.com.br.129600INA186....	globo.com.br.129592INA186....	2024-11-19 17:47:05	2024-11-19 18:24:48
Cloudflare	gov.za	1.1.1.1	408	46	2038	gov.za.86400INA163.195.1.225	gov.za.85153INA163.195.1.225	gov.za.86400INA163.195.1.225	2024-11-19 17:47:10	2024-11-19 18:24:54
Cloudflare Reserve	gov.za	1.0.0.1	48	41	80	gov.za.86392INA163.195.1.225	gov.za.85145INA163.195.1.225	gov.za.86391INA163.195.1.225	2024-11-19 17:47:19	2024-11-19 18:25:04
Cloudflare	bbc.co.uk	1.1.1.1	20	41	82	bbc.co.uk.191INA151.101.19...	bbc.co.uk.152INA151.101.64....	bbc.co.uk.42INA151.101.128....	2024-11-19 17:47:23	2024-11-19 18:25:08
Cloudflare Reserve	bbc.co.uk	1.0.0.1	21	70	80	bbc.co.uk.184INA151.101.12...	bbc.co.uk.146INA151.101.12...	bbc.co.uk.36INA151.101.64.81	2024-11-19 17:47:29	2024-11-19 18:25:14
Cloudflare	work.ua	1.1.1.1	13	40	82	work.ua.262INA104.22.55.203	work.ua.15INA104.22.54.203	work.ua.214INA104.22.54.203	2024-11-19 17:47:31	2024-11-19 18:25:16
Cloudflare Reserve	work.ua	1.0.0.1	12	39	83	work.ua.256INA104.22.54.203	work.ua.8INA104.22.54.203	work.ua.208INA104.22.54.203	2024-11-19 17:47:37	2024-11-19 18:25:23
Cloudflare	post.japanpost.jp	1.1.1.1	55	731	659	post.japanpost.jp.300INA43....	post.japanpost.jp.300INA43....	post.japanpost.jp.300INA43....	2024-11-19 17:47:40	2024-11-19 18:25:25
Cloudflare Reserve	post.japanpost.jp	1.0.0.1	289	954	206	post.japanpost.jp.300INA43....	post.japanpost.jp.300INA43....	post.japanpost.jp.300INA43....	2024-11-19 17:47:58	2024-11-19 18:25:33
Cloudflare	news.com.au	1.1.1.1	20	37	82	news.com.au.17INA104.91.4....	news.com.au.2INA2.17.156.1...	news.com.au.17INA2.17.156....	2024-11-19 17:48:01	2024-11-19 18:25:37
Cloudflare Reserve	news.com.au	1.0.0.1	11	34	80	news.com.au.12INA104.91.4....	news.com.au.15INA2.17.156....	news.com.au.12INA2.17.156....	2024-11-19 17:48:07	2024-11-19 18:25:43
OpenDNS	about.us	208.67.222.222	85	131	233	about.us.1800INA34.204.39....	about.us.1800INA54.85.227....	about.us.1800INA54.85.227....	2024-11-19 17:48:09	2024-11-19 18:25:45
OpenDNS Reserve	about.us	208.67.220.220	89	128	219	about.us.1800INA34.204.39....	about.us.540INA34.204.39.241	about.us.1800INA54.85.227....	2024-11-19 17:48:16	2024-11-19 18:25:52
OpenDNS	globo.com.br	208.67.222.222	41	1035	911	globo.com.br.114804INA186....	globo.com.br.129600INA186....	globo.com.br.129600INA186....	2024-11-19 17:48:18	2024-11-19 18:25:56
OpenDNS Reserve	globo.com.br	208.67.220.220	43	927	205	globo.com.br.108451INA186....	globo.com.br.129600INA186....	globo.com.br.107356INA186....	2024-11-19 17:48:27	2024-11-19 18:26:06
OpenDNS	gov.za	208.67.222.222	227	274	207	gov.za.86400INA163.195.1.225	gov.za.86400INA163.195.1.225	gov.za.84152INA163.195.1.225	2024-11-19 17:48:33	2024-11-19 18:26:11
OpenDNS Reserve	gov.za	208.67.220.220	1157	126	211	gov.za.86400INA163.195.1.225	gov.za.71491INA163.195.1.225	gov.za.84135INA163.195.1.225	2024-11-19 17:48:44	2024-11-19 18:26:19
OpenDNS	bbc.co.uk	208.67.222.222	82	124	207	bbc.co.uk.75INA151.101.128....	bbc.co.uk.203INA151.101.64....	bbc.co.uk.273INA151.101.19...	2024-11-19 17:48:49	2024-11-19 18:26:27
OpenDNS Reserve	bbc.co.uk	208.67.220.220	42	129	220	bbc.co.uk.68INA151.101.64.81	bbc.co.uk.185INA151.101.64....	bbc.co.uk.119INA151.101.64....	2024-11-19 17:48:56	2024-11-19 18:26:34
OpenDNS	work.ua	208.67.222.222	50	138	201	work.ua.300INA104.22.54.203	work.ua.300INA172.67.28.193	work.ua.151INA104.22.54.203	2024-11-19 17:48:58	2024-11-19 18:26:37
OpenDNS Reserve	work.ua	208.67.220.220	42	131	209	work.ua.293INA104.22.55.203	work.ua.150INA104.22.55.203	work.ua.150INA172.67.28.193	2024-11-19 17:49:04	2024-11-19 18:26:44
OpenDNS	post.japanpost.jp	208.67.222.222	897	228	213	post.japanpost.jp.300INA43....	post.japanpost.jp.300INA43....	post.japanpost.jp.300INA43....	2024-11-19 17:49:07	2024-11-19 18:26:47
OpenDNS Reserve	post.japanpost.jp	208.67.220.220	950	152	742	post.japanpost.jp.300INA43....	post.japanpost.jp.300INA43....	post.japanpost.jp.300INA43....	2024-11-19 17:49:23	2024-11-19 18:27:02
OpenDNS	news.com.au	208.67.222.222	88	127	208	news.com.au.20INA23.62.22...	news.com.au.20INA23.62.22...	news.com.au.20INA23.62.22...	2024-11-19 17:49:29	2024-11-19 18:27:10
OpenDNS Reserve	news.com.au	208.67.220.220	43	134	211	news.com.au.20INA23.62.22...	news.com.au.20INA23.62.22...	news.com.au.20INA23.62.22...	2024-11-19 17:49:36	2024-11-19 18:27:17
Quad9	about.us	9.9.9.9	80	108	132	about.us.1800INA34.204.39....	about.us.806INA54.85.227.149	about.us.806INA54.85.227.149	2024-11-19 17:49:38	2024-11-19 18:27:19
Quad9 Reserve	about.us	149.112.112.112	83	111	128	about.us.1800INA54.85.227....	about.us.1800INA34.204.39....	about.us.796INA54.85.227.149	2024-11-19 17:49:46	2024-11-19 18:27:26
Quad9	globo.com.br	9.9.9.9	265	76	129	globo.com.br.43200INA186.1...	globo.com.br.41933INA186.1...	globo.com.br.42209INA186.1...	2024-11-19 17:49:49	2024-11-19 18:27:29
Quad9 Reserve	globo.com.br	149.112.112.112	261	296	126	globo.com.br.43200INA186.1...	globo.com.br.43200INA186.1...	globo.com.br.42211INA186.1...	2024-11-19 17:49:57	2024-11-19 18:27:36

Рисунок А.1 – Вікно відображення результатів моніторингу *(в діючому релізі)*

```
private long dotPing(String testDomain, boolean flag)
{
    int port = 853;
    long duration = 0;

    try {
        Record record = Record.newRecord(Name.fromString(testDomain + "."), Type.A, DClass.IN);
        Message query = Message.newQuery(record);

        byte[] queryData = query.toWire();
        long startTime = System.nanoTime();
        long endTime;

        SSLSocketFactory factory = (SSLSocketFactory) SSLSocketFactory.getDefault();
        try (SSLSocket socket = (SSLSocket) factory.createSocket(ip, port)) {
            socket.startHandshake();
            OutputStream outputStream = socket.getOutputStream();
            outputStream.write((queryData.length >> 8) & 0xff);
            outputStream.write(queryData.length & 0xff);
            outputStream.write(queryData);

            InputStream inputStream = socket.getInputStream();
            int responseLength = (inputStream.read() << 8) | inputStream.read();
            byte[] responseData = new byte[responseLength];
            inputStream.read(responseData);
            endTime = System.nanoTime();

            Message response = new Message(responseData);
            Record[] answers = response.getSectionArray(Section.ANSWER);
            if (answers.length > 0) {
                for (Record answer : answers) {
                    dtResult = answer.toString();
                    if(flag)    System.out.println("Answer: " + answer);
                }
            } else {
                if(flag) System.out.println("No answers found.");
            }
        }

        duration = (endTime - startTime) / 1_000_000;
        if(flag)    System.out.println("Query time: " + duration + " ms");
        return duration;

    } catch (Exception e) {
        if(flag)  System.out.println("DOT Failed");;
    }
    return duration;
}
```

Рисунок А.2 – Метод dotPing

```
public TestResult sendRequestToCloudFunction(DnsRequest dnsRequest) throws Exception {
    String urlString = URL;
    URL url = new URL(urlString);
    HttpURLConnection conn = (HttpURLConnection) url.openConnection();
    conn.setRequestMethod("POST");
    conn.setRequestProperty("Content-Type", "application/json");
    conn.setDoOutput(true);

    Gson gson = new Gson();
    String jsonInputString = gson.toJson(dnsRequest);

    try (OutputStream os = conn.getOutputStream()) {
        byte[] input = jsonInputString.getBytes("utf-8");
        os.write(input, 0, input.length);
    }

    BufferedReader br = new BufferedReader(new InputStreamReader(conn.getInputStream(),
"utf-8"));
    StringBuilder response = new StringBuilder();
    String responseLine;
    while ((responseLine = br.readLine()) != null) {
        response.append(responseLine.trim());
    }
    System.out.println("Response from Cloud Function: " + response.toString());

    JsonObject jsonObject = gson.fromJson(String.valueOf(response), JsonObject.class);

    String name = " ";
    String testDomain = " ";
    String ip = " ";
    String pRes = jsonObject.get("plainResult").getAsString();
    String dhRes = jsonObject.get("dohResult").getAsString();
    String dtRes = jsonObject.get("dotResult").getAsString();
    long pq = jsonObject.get("plainQueryTime").getAsLong();
    long dhq = jsonObject.get("dohQueryTime").getAsLong();
    long dtq = jsonObject.get("dotQueryTime").getAsLong();
    long timestamp = jsonObject.get("timestamp").getAsLong();

return new TestResult(name, testDomain, ip, pq, dhq, dtq, pRes, dhRes, dtRes, timestamp, 0);
}
```

Рисунок А.3 – Метод sendRequestToCloudFunction

```
   public void addGroupedTestResult(ArrayList<TestResult> testResults1,
ArrayList<TestResult> testResults2) throws InterruptedException {
    while (stop)
    {
        wait();
    }
    dataset.clear();
    DateTimeFormatter formatter =
DateTimeFormatter.ofPattern("HH:mm:ss").withZone(ZoneId.systemDefault());

    for (int i = 0; i < Math.min(testResults1.size(), testResults2.size()); i++)
{
        TestResult result1 = testResults1.get(i);
        TestResult result2 = testResults2.get(i);
        String groupLabel = "Test " + (i + 1)+ " " +
formatter.format(Instant.ofEpochMilli(result2.getTimestamp2()));
        dataset.addValue(result1.getPq(), "Test 1 - Plain", groupLabel);
        dataset.addValue(result1.getDhq(), "Test 1 - DoH", groupLabel);
        dataset.addValue(result1.getDtq(), "Test 1 - DoT", groupLabel);

        dataset.addValue(result2.getPq(), "Test 2 - Plain", groupLabel);
        dataset.addValue(result2.getDhq(), "Test 2 - DoH", groupLabel);
        dataset.addValue(result2.getDtq(), "Test 2 - DoT", groupLabel);
    }
    repaint();
}
```

Рисунок А.4 – Метод AddGroupedTestResult