DOI 10.31891/2307-5732-2025-347-36
УДК 004.72

ПАВЛЮК ОЛЕКСАНДР-ЮРІЙ
Національний університет «Львівська політехніка»
https://orcid.org/0009-0008-6985-203X
e-mail: oleksandr-iurii.s.pavliuk@lpnu.ua
НЄМКОВА ОЛЕНА
Національний університет «Львівська політехніка»
https://orcid.org/0000-0003-0690-2657
e-mail: olena.a.niemkova@lpnu.ua

АНАЛІЗ ПОТЕНЦІАЛУ DNS-ТУНЕЛЮВАННЯ ДЛЯ СТВОРЕННЯ ПРИХОВАНИХ КАНАЛІВ ЗВ'ЯЗКУ

В роботі проаналізовано технологію DNS-тунелювання, яка використовується для створення прихованих каналів зв'язку через DNS-протокол; розглянуто ключові принципи її роботи, зокрема кодування даних у DNS-запитах і відповідях. Запропоновано архітектуру та принцип роботи месенджера на основі DNS-тунелювання. Розкрито як корисні сценарії застосування для забезпечення анонімності та обходу цензури, так і кібератаки, такі як ексфільтрація даних і керування ботнетами. Надано рекомендації для виявлення та запобігання кібератак, включаючи моніторинг трафіку, використання IDS/IPS і аналіз аномалій.

Ключові слова: DNS-тунелювання, шифрування, обхід цензури, кібератаки, обфускація даних, моніторинг трафіку.

PAVLIUK OLEKSAND-IURII
Lviv Polytechnic National University
NIEMKOVA OLENA
Lviv Polytechnic National University

DNS TUNNELING POTENTIAL ANALYSIS FOR CREATING COVERT COMMUNICATION CHANNELS

The article is devoted to DNS tunnelling, a technique used to establish covert communication channels through the Domain Name System (DNS) protocol. DNS tunnelling encodes data within DNS queries and responses, enabling data transmission through network traffic typically permitted by most firewalls and security systems. The study details key implementation methods, including the use of Base64 encoding for binary-to-text transformation and the suitability of various DNS record types, such as TXT, A/AAAA, MX and CNAME, for tunnelling purposes. Techniques like data obfuscation, encryption using AES, and compression algorithms are analyzed to enhance efficiency and stealth in data transmission. The architecture and operating principle of a messenger based on DNS tunnelling are proposed. The messenger uses DNS queries and responses to transmit encrypted messages, employing AES encryption and Base64 encoding to secure and format data for transmission. This approach enables covert communication by masking message traffic as legitimate DNS requests, offering a unique solution for bypassing network restrictions while maintaining data confidentiality. The article highlights both legitimate and malicious applications of DNS tunnelling. Useful applications include creating backup communication channels, bypassing censorship, and enhancing anonymity, particularly in environments with restricted Internet access. Conversely, the risks of data exfiltration, botnet control, and malware distribution via DNS tunnels are thoroughly discussed, emphasizing their role in circumventing traditional security mechanisms. To address these risks, the article offers detection and prevention strategies. Recommendations include monitoring DNS traffic, utilizing Intrusion Detection and Prevention Systems (IDS/IPS), and implementing anomaly detection models. Specific signatures and machine learning techniques are proposed to identify unusual DNS queries and response patterns. Furthermore, access control policies and DNS whitelisting are suggested to limit unauthorized tunnelling activities. In conclusion, future challenges and the dual-use nature of DNS tunnelling are discussed, advocating ethical considerations in its application. The findings underscore the importance of continuous security system updates and adaptive monitoring strategies to mitigate emerging threats in evolving network environments.

Keywords: DNS tunneling, encryption, censorship circumvention, cyberattacks, data obfuscation, traffic monitoring.

Постановка проблеми

Система доменних імен (DNS) є одним з фундаментальних компонентів Інтернету. Її основна функція полягає в перетворенні зручних для людини доменних імен (наприклад, google.com) на числові IP-адреси, які використовуються комп'ютерами для взаємної ідентифікації у мережі. Цей процес, відомий як резолвінг доменних імен, дозволяє користувачам легко отримувати доступ до веб-сайтів та інших онлайн-ресурсів без необхідності запам'ятовувати складні комбінації цифр [1]. Однак, крім свого основного призначення, DNS може бути використаний і для інших цілей, зокрема для створення прихованих каналів зв'язку. Ця техніка, відома як DNS-тунелювання, дозволяє передавати дані через DNS-запити та відповіді, використовуючи протокол DNS не за призначенням [2]. DNS-тунелювання працює шляхом кодування даних в DNS-запити та відповіді, які потім передаються між клієнтом та сервером. Оскільки DNS-трафік зазвичай дозволений більшістю мережевих фільтрів, це дозволяє обходити обмеження та створювати приховані канали зв'язку, які важко виявити. DNS-тунелювання може бути використане як для легітимних цілей, так і для кібератак. Тому важливо розуміти технічні можливості реалізації даної технології для розроблення рекомендації з ефективного та безпечного її використання.

Аналіз останніх досліджень і публікацій

DNS-тунелювання, як одна з технологій прихованого передавання даних, привертає увагу дослідників та експертів з кібербезпеки, що зумовлює актуальність аналізу останніх публікацій у цій

сфері для розуміння новітніх технік атаки та підходів до їх виявлення. На думку авторів [3], DNS-тунелювання можна розглядати як розділ стеганографії, науки про приховування інформації. В той час як криптографія зосереджена на захисті змісту повідомлення, стеганографія має на меті приховати сам факт передачі даних. DNS-тунелювання саме цим і займається, а саме, приховує обмін даними під виглядом легітимного DNS-трафіку. Для оптимізації передачі даних використовують різні типи запитів [4], самі дані кодують та шифрують [5,6], а також використовують компресію [7]. Застосування конкретного інструменту для створення та використання DNS-тунелів [8-10] залежить від поставлених задач – що є перевагою: швидкість, прихованість, можливість тунелювати TCP-трафік через DNS, або завдана схема кодування [11]. Перший етап – встановлення безпечного з'єднання, потребує протоколу TLS з завданою схемою шифрування [12]. Автори [13] дослідили можливість виявлення кібератаки ексфільтрації даних, у тому числі атаки Advanced Persistent Threat, за допомогою методів машинного навчання. У дослідженні [14] виконано атрибуцію ботів за допомогою фільтра Ходріка–Прескотта на невеликій вибірці і продемонстровано можливі контрзаходи за допомогою аналізу трафіку шляхом створених індикаторів компромісу. Автори [15] розробили методи виявлення DNS-тунелювання за допомогою методів багатошарового персептрону та випадкового лісу. Дослідження [16] підсумовує відомі методи для попередження викрадення даних за допомогою технології DNS-тунелювання. Таким чином, сучасні дослідження зосереджені в основному на попередженні кібератак, що використовують технологію DNS-тунелювання, і практично не розглядають корисні можливості даної технології – аварійний канал зв'язку, анонімність, обхід цензури.

Метою роботи є дослідження технічних аспектів DNS-тунелювання, його потенційних можливостей для створення прихованих каналів зв'язку, а також ризиків, пов'язаних зі зловмисним використанням цієї технології.

Виклад основного матеріалу

Технічні аспекти реалізації та оптимізації DNS-тунелювання

DNS-тунелювання базується на можливості передачі даних, інкапсульованих в DNS-запити та відповіді. Замість запиту IP-адреси, клієнт відправляє спеціально сформовані запити, що містять фрагменти даних. Сервер, налаштований на розпізнавання цих запитів, витягує дані з них та відправляє відповідь, яка також може містити дані. Різні типи DNS-запитів мають різну придатність для тунелювання [4]:

- A/AAAA: Призначені для запиту IPv4/IPv6 адрес, мають обмежену місткість для даних.
- TXT: Дозволяють передавати довільний текст, що робить їх зручними для тунелювання.
- MX: Використовуються для визначення поштових серверів, можуть бути використані для тунелювання, але менш ефективні, ніж TXT.
- CNAME: Створюють аліаси для доменних імен, мають обмежене застосування для тунелювання.

Для передачі через DNS-поля дані кодуються, наприклад, за допомогою Base64, що дозволяє представити бінарні дані у текстовому форматі, сумісному з DNS [5]. Наприклад, клієнт хоче передати повідомлення "Hello". Воно кодується в Base64 ("SGVsbG8=") та розбивається на фрагменти ("SGV", "sbG", "8="). Кожен фрагмент вставляється в DNS-запит. Сервер обробляє запити, витягує фрагменти, об'єднує їх та декодує, отримуючи оригінальне повідомлення.

Для забезпечення конфіденційності даних, що передаються через DNS-тунель, важливо використовувати шифрування. Симетричне шифрування, таке як AES (Advanced Encryption Standard), є оптимальним вибором для DNS-тунелювання. Воно забезпечує високу швидкість шифрування/дешифрування та надійний захист даних. Обидві сторони (клієнт та сервер) використовують один і той самий секретний ключ для шифрування та дешифрування, що спрощує процес [6]. Вибір надійного алгоритму шифрування з достатньою довжиною ключа є критичним для мінімізації ризику перехоплення та читання даних зловмисниками. AES з довжиною ключа 256 біт вважається надійним стандартом на сьогоднішній день.

Для підвищення ефективності та скритності DNS-тунелювання використовуються різні методи оптимізації [5,7]:

- Компресія та мінімізація даних: зменшення обсягу даних, що передаються, дозволяє зменшити кількість DNS-запитів, це знижує навантаження на DNS-сервери та зменшує ймовірність виявлення тунелю. Для цього використовуються алгоритми стиснення, такі як zlib або gzip.
- Обфускація даних: маскування даних під легітимний DNS-трафік ускладнює виявлення тунелю. Це може включати використання випадкових доменних імен, вставку "сміттєвих" даних, або імітацію патернів звичайних DNS-запитів. Наприклад, замість використання очевидних доменних імен (наприклад, tunnel.example.com) можна генерувати випадкові імена (a7f3b9.example.com, e2d1c8.example.com). Це ускладнює ідентифікацію тунельованого трафіку серед легітимних DNS-запитів, втім, балансування рівню обфускації є також важливим, оскільки доменні імена чи вміст DNS-пакетів, що виглядають занадто ентропічно, можуть бути виявленими мережевими фільтрами.

Існує ряд інструментів, які спрощують створення та використання DNS-тунелів. Кожен з цих інструментів має свої переваги та недоліки, вибір залежить від конкретних потреб та умов використання, таблиця 1.

Таблиця 1

Інструменти для створення та використання DNS-тунелів

Інструмент	Коротка характеристика
iodine [8]	Один з найпопулярніших інструментів для DNS-тунелювання. Він використовує протокол UDP та підтримує шифрування, що забезпечує високу швидкість та безпеку. Iodine відносно простий у налаштуванні та використанні, має версії для різних операційних систем.
DNScat2 [9]	Інструмент, орієнтований на використання в penetration testing. Він дозволяє створювати "приховані" канали зв'язку для обходу мережевих фільтрів та брандмауерів. DNScat2 підтримує шифрування та різні методи обфускації, що ускладнює його виявлення.
DNS2TCP [10]	Утиліта, що дозволяє тунелювати TCP-трафік через DNS. Вона підтримує різні типи DNS-запитів (A, AAAA, MX, TXT) та може використовуватись для доступу до TCP-сервісів (наприклад, SSH, HTTP) через DNS-тунель.

Легітимні застосування DNS-тунелювання

Хоча DNS-тунелювання часто асоціюється зі зловмисною діяльністю, воно також може бути використане для легітимних цілей:

- *Аварійний канал зв'язку*. У випадку відмови основного інтернет-з'єднання, DNS-тунелювання може бути використане для встановлення резервного каналу зв'язку. Це особливо актуально для організацій, які потребують постійного доступу до мережі, наприклад, для забезпечення безперервності роботи критичних інфраструктур чи бізнес-процесів.
- *Забезпечення анонімності*. DNS-тунелювання може слугувати додатковим рівнем анонімності при роботі в Інтернеті, особливо в публічних мережах Wi-Fi. Хоча DNS-тунелювання не гарантує повної анонімності, воно може ускладнити відстеження активності користувача та ідентифікацію його реальної IP-адреси.
- *Обхід цензури*. У країнах з жорсткою цензурою Інтернету, DNS-тунелювання може бути використане для доступу до заблокованих веб-сайтів та сервісів. За допомогою DNS-тунелів користувачі можуть обходити обмеження та отримувати доступ до інформації, яка в іншому випадку була б недоступною.

Проектування примітивного месенджера на основі DNS-тунелювання

Для демонстрації принципу застосування DNS-тунелів для пересічних користувачів на практиці, пропонується механізм роботи примітивного застосунку−месенджера на основі тунелів, опис та функції учасників процесу наведено у таблиці 2.

Таблиця 2

Складові частини месенджера на основі DNS-тунелювання

Учасники процесу	Використане технологічне підґрунтя
Клієнт: Програмне забезпечення, встановлене на пристрої користувача, яке відповідає за відправку та отримання повідомлень через DNS-тунель.	TLS (Transport Layer Security): Забезпечує безпечне з'єднання між клієнтом та сервером для початкового обміну ключами та аутентифікації
DNS-сервер: Стандартний DNS-сервер, який використовується для резолвінгу доменних імен. Він виступає посередником між клієнтом та сервером месенджера, але не вимагає додаткового налаштування, оскільки сам факт його використання для тунелювання приховується.	AES (Advanced Encryption Standard): Симетричний алгоритм шифрування, який використовується для захисту повідомлень від несанкціонованого доступу.
Веб-сервер: Центральний сервер месенджера, який приймає та обробляє повідомлення від клієнтів, а також надсилає повідомлення адресатам.	Base64: Схема кодування, яка дозволяє перетворювати бінарні дані у текстовий формат, придатний для передачі в DNS-запитах. Base64 збільшує розмір даних приблизно на 33% і є досить компактним для використання в DNS-запитах [11]

Принцип роботи:

1. *Встановлення з'єднання*. Клієнт ініціює TLS-з'єднання з сервером. Під час цього процесу відбувається автентифікація та обмін секретним ключем AES, який буде використовуватися для шифрування повідомлень [12].

2. *Підготовка повідомлення*. Клієнт шифрує повідомлення за допомогою AES, кодує його за допомогою Base64 та розбиває на фрагменти, які можна вмістити в DNS-запити.

3. *Відправка повідомлення*. Клієнт формує DNS-запити, вставляючи фрагменти повідомлення в поля запиту (наприклад, ім'я домену або TXT-запис). Ці запити надсилаються на DNS-сервер.

4. *Обробка запитів*. DNS-сервер перенаправляє запити на сервер месенджера.

5. *Отримання та дешифрування*. Сервер витягує фрагменти повідомлення з DNS-запитів, декодує їх за допомогою Base64, розшифровує за допомогою AES та об'єднує в оригінальне повідомлення.

6. *Доставка повідомлення*. Сервер зберігає повідомлення та доставляє його адресату, коли той буде онлайн.

У даній моделі DNS-сервер виступає як посередник і не бере участі в обробці повідомлень. Вся комунікація після TLS-рукостискання відбувається через DNS-запити та відповіді. Стеганографія досягається за рахунок маскування повідомлень під звичайні DNS-запити. На рис.1 наведено спрощену схему роботи месенджера.

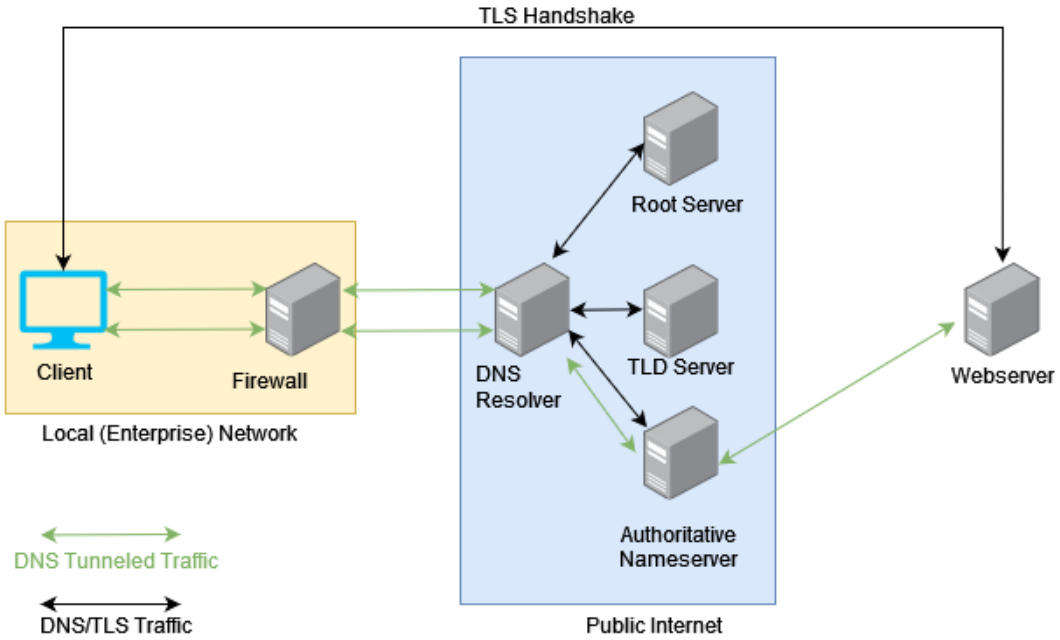

Рис. 1. Спрощена схема роботи месенджера

Очікувано, у такому підході до архітектури месенджера є свої переваги та недоліки. Використання DNS-тунелювання дозволяє приховати факт передачі даних, оскільки трафік маскується під легітимні DNS-запити, проте DNS-протокол не має механізмів контролю цілісності даних, тому виявлення та виправлення помилок може бути складним. Також, хоч протокол DNS є відносно простим (що спрощує реалізацію месенджера), обмежений розмір DNS-запитів накладає обмеження на розмір повідомлень; це призводить до низкої пропускної здатності і відповідно високої латентності (затримки) у передачі даних. Як наслідок, найбільш примітивна реалізація такого принципу передачі даних зможе оперувати лише з інформацією невеликих об’ємів.

Застосування DNS-тунелювання для проведення кібератак

Хоча DNS-тунелювання може бути корисним інструментом для обходу цензури або доступу до обмежених ресурсів, його часто використовують зловмисники для реалізації кібератак, адже DNS-тунелювання надає можливість обходити традиційні засоби безпеки (брандмауери та системи виявлення вторгнень), які часто не контролюють DNS-трафік належним чином. Це створює можливість різноманітних шкідливих дій:

- *Ексфільтрація даних*. Зловмисники можуть використовувати DNS-тунелі для викрадення конфіденційних даних, таких як паролі, фінансова інформація, або інтелектуальна власність. Дані кодуються та передаються у вигляді DNS-запитів, що ускладнює їх виявлення традиційними методами [13].
- *Канали управління ботнетами*. DNS-тунелювання може бути використано для створення прихованих каналів зв'язку між ботнетом та командним сервером. Це дозволяє зловмисникам керувати зараженими комп'ютерами, віддавати їм команди та отримувати інформацію, залишаючись непоміченими [14].
- *Приховане завантаження шкідливого ПЗ*. Зловмисники можуть використовувати DNS-тунелі для доставки шкідливого програмного забезпечення на цільові комп'ютери. Шкідливий код може бути розбитий на фрагменти та переданий у вигляді DNS-запитів, а потім зібраний та запущений на зараженому комп'ютері.

DNS-тунелювання особливо небезпечне в середовищах з обмеженим доступом до Інтернету, таких як корпоративні або державні мережі. У таких мережах часто використовуються брандмауери та

фільтри, які блокують доступ до більшості веб-сайтів та сервісів. Однак DNS-трафік зазвичай дозволений, що дозволяє зловмисникам використовувати DNS-тунелі для обходу цих обмежень та отримання доступу до заборонених ресурсів. Це ставить під загрозу безпеку всієї мережі та може призвести до серйозних наслідків, таких як витік даних, фінансові втрати, або порушення роботи критичної інфраструктури.

Виявлення DNS-тунелювання є важливим аспектом захисту мережі. Враховуючи постійний розвиток методів DNS-тунелювання, важливо використовувати комбінований підхід та регулярно оновлювати системи безпеки. Для ефективної ідентифікації цієї загрози адміністратори повинні використовувати комбінацію методів аналізу трафіку та вмісту. Слід звертати увагу на аномально високу кількість DNS-запитів, особливо до підозрілих доменів. DNS-тунелювання може призводити до збільшення розміру DNS-пакетів, тож аналіз запитів та відповідей може виявити нестандартні типи записів або великі обсяги даних. Аналіз вмісту запитів та відповідей може виявити кодовані дані (наприклад, Base64), а використання IDS (Intrusion Detection System) може виявити специфічні сигнатури або патерни у трафіку, що при порівнянні з базами даних відомих сигнатур може визначити інструменти DNS-тунелювання. Можливе також застосування алгоритмів машинного навчання для виявлення аномалій у DNS-трафіку [15].

Рекомендації для ефективного та безпечного використання DNS-тунелювання

При використанні DNS-тунелювання важливо дотримуватися певних рекомендацій для забезпечення ефективності та безпеки. Надійне шифрування є ключовим фактором. Рекомендується використовувати сильні алгоритми шифрування, такі як AES з довжиною ключа 256 біт, щоб захистити дані від перехоплення та несанкціонованого доступу. Дотримання політик безпеки також є критично важливим. Це включає в себе використання складних паролів, регулярне оновлення програмного забезпечення, та обмеження доступу до конфіденційних даних.

Обфускація даних допомагає приховати факт тунелювання та ускладнити його виявлення. Це може включати використання випадкових доменних імен, вставку "сміттєвих" даних або імітацію патернів звичайних DNS-запитів. Балансування навантаження на DNS-сервери також є важливим аспектом. Надмірне навантаження на один сервер може призвести до його перевантаження та відмови в обслуговуванні, що може привернути увагу адміністраторів мережі. Розподіл трафіку між декількома серверами допомагає знизити ризик виявлення.

Для захисту від зловмисного використання DNS-тунелювання організації повинні вживати проактивних заходів. Обмеження доступу до непотрібних DNS-запитів у мережах є важливим кроком. Це може включати блокування запитів до невідомих або підозрілих доменів, а також обмеження кількості DNS-запитів, які може надсилати один пристрій. Запровадження білих списків DNS-серверів дозволяє обмежити використання лише авторизованих серверів, що ускладнює зловмисникам створення тунелів через неконтрольовані сервери. Одночасно, блокування підозрілих запитів, таких як запити з нестандартними типами записів або аномально великими пакетами, допомагає запобігти передачі даних через тунелі.

Регулярний моніторинг трафіку є невід'ємною частиною системи безпеки. Необхідно налаштувати систему моніторингу для швидкого виявлення аномалій у DNS-трафіку, таких як незвичайна кількість запитів, великі пакети, або кодований вміст. Використання спеціалізованих інструментів для моніторингу та аналізу трафіку, таких як Zeek, Splunk, IDS/IPS-системи, дозволяє автоматизувати процес виявлення та реагування на загрози. Ці інструменти можуть аналізувати трафік у режимі реального часу, виявляти підозрілі патерни та генерувати сповіщення для фахівців з безпеки [16].

Проведення тренінгів для співробітників є важливим елементом захисту. Співробітники повинні бути обізнані з ризиками, пов'язаними з DNS-тунелюванням, та знати, як розпізнати підозрілу активність.

Висновки та перспективи подальших досліджень

DNS-тунелювання являє собою потужний інструмент з двояким призначенням. З одного боку, воно надає можливість обходу цензури, доступу до обмежених ресурсів та створення прихованих каналів зв'язку. З іншого боку, DNS-тунелювання може бути використане зловмисниками для реалізації кібератак, таких як ексфільтрація даних, управління ботнетами та поширення шкідливого ПЗ. Ефективне та безпечне використання DNS-тунелювання вимагає дотримання певних рекомендацій, включаючи використання надійного шифрування, обфускацію даних та балансування навантаження на DNS-сервери. Одночасно, організації повинні вживати проактивних заходів для захисту від зловмисного використання DNS-тунелювання, таких як обмеження доступу до непотрібних DNS-запитів, запровадження білих списків DNS-серверів, регулярний моніторинг трафіку та використання спеціалізованих інструментів безпеки.

Важливо пам'ятати про етичні аспекти використання DNS-тунелювання. Ця технологія не повинна використовуватися для незаконної діяльності або завдання шкоди іншим. Користувачі повинні усвідомлювати потенційні ризики та використовувати DNS-тунелювання відповідально. Контроль за DNS-трафіком є критично важливим для забезпечення безпеки мережі. Організації повинні впроваджувати системи моніторингу та аналізу трафіку, щоб своєчасно виявляти та запобігати зловмисному використанню DNS-тунелювання.

Враховуючи постійний розвиток технологій, можна очікувати появи нових методів DNS-тунелювання та обходу захисту. Тому важливо слідкувати за новими тенденціями та вдосконалювати системи безпеки. Майбутнє DNS-тунелювання залежить від того, наскільки ефективно зможемо використовувати його потенціал та мінімізувати ризики.

Література

1. Dooley, M., & Rooney, T. (2017). Introduction to the Domain Name System (DNS). In *Cryptography and Network Security* (7th ed., pp. 29–55). Wiley-IEEE Press. https://doi.org/10.1002/9781119328292.ch2
2. Raman, D., Kwon, T., Lee, M. K., & Kwon, D. (2013). DNS tunneling for network penetration. In *T. Kwon, M. K. Lee, & D. Kwon (Eds.), Information Security and Cryptology – ICISC 2012* (Vol. 7839, pp. 65–77). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-37682-5_6
3. Drzymała, M., Szczypiorski, K., & Urbański, M. Ł. (2016). Network steganography in the DNS protocol. *International Journal of Electronics and Telecommunications, 62*(4). https://doi.org/10.1515/eletel-2016-0047
4. Salat, L., Davis, M., & Khan, N. (2023). DNS tunnelling, exfiltration and detection over cloud environments. *Sensors, 23*(5), 2760. https://doi.org/10.3390/s23052760
5. Van Leijenhorst, T., Chin, K.-W., & Lowe, D. (n.d.). On the viability and performance of DNS tunneling. Retrieved November 20, 2024, from https://www.researchgate.net/publication/252673752_On_the_Viability_and_Performance_of_DNS_Tunneling
6. Al-Kasassbeh, M., & Khairallah, T. (2019). Winning tactics with DNS tunneling. *Network Security, 2019*(12), 12–19. https://doi.org/10.1016/S1353-4858(19)30144-8
7. Wang, L., Kim, H., Mittal, P., & Rexford, J. (n.d.). Programmable in-network obfuscation of DNS traffic. Retrieved November 23, 2024, from https://www.ndss-symposium.org/wp-content/uploads/dnspriv21-08-paper.pdf
8. Ekman, E. (n.d.). *iodine*. GitHub. Retrieved November 15, 2024, from https://github.com/yarrick/iodine
9. Dembour, O. (n.d.). *dns2tcp*. GitHub. Retrieved November 29, 2024, from https://github.com/alex-sector/dns2tcp
10. Bowes, R. (n.d.). *dnscat2*. GitHub. Retrieved November 27, 2024, from https://github.com/iagox86/dnscat2
11. Wen, S., & Dang, W. (2018). Research on Base64 encoding algorithm and PHP implementation. In *Proceedings of the 26th International Conference on Geoinformatics* (pp. 1–6). IEEE. https://doi.org/10.1109/GEOINFORMATICS.2018.8557068
12. Bhargavan, K., et al. (2014). Proving the TLS handshake secure (as it is). In *J. A. Garay & R. Gennaro (Eds.), Advances in Cryptology – CRYPTO 2014* (Vol. 8617). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-44381-1_14
13. Das, A., Shen, M.-Y., Shashanka, M., & Wang, J. (2017). Detection of exfiltration and tunneling over DNS. In *Proceedings of the 16th IEEE International Conference on Machine Learning and Applications (ICMLA)* (pp. 737–742). IEEE. https://doi.org/10.1109/ICMLA.2017.00-71
14. Patsakis, C., Casino, F., & Katos, V. (2020). Encrypted and covert DNS queries for botnets: Challenges and countermeasures. *Computers & Security, 88*, 101614. https://doi.org/10.1016/j.cose.2019.101614
15. Berg, A., & Forsberg, D. (2019). Identifying DNS-tunneled traffic with predictive models. *arXiv preprint*. https://doi.org/10.48550/arXiv.1906.11246
16. Dusseault, A. (2021). *Methods for the prevention of data exfiltration by DNS tunneling* (Master's thesis, Utica College). ProQuest. Retrieved November 29, 2024, from https://www.proquest.com/openview/8cffdd3d5c1020bf41f4e9b23c3b96b1/1?pq-origsite=gscholar&cbl=18750&diss=y

References

1. Dooley, M., & Rooney, T. (2017). Introduction to the Domain Name System (DNS). In Cryptography and Network Security (7th ed., pp. 29–55). Wiley-IEEE Press. https://doi.org/10.1002/9781119328292.ch2
2. Raman, D., Kwon, T., Lee, M. K., & Kwon, D. (2013). DNS tunneling for network penetration. In T. Kwon, M. K. Lee, & D. Kwon (Eds.), Information Security and Cryptology – ICISC 2012 (Vol. 7839, pp. 65–77). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-37682-5_6
3. Drzymała, M., Szczypiorski, K., & Urbański, M. Ł. (2016). Network steganography in the DNS protocol. International Journal of Electronics and Telecommunications, 62(4). https://doi.org/10.1515/eletel-2016-0047
4. Salat, L., Davis, M., & Khan, N. (2023). DNS tunnelling, exfiltration and detection over cloud environments. Sensors, 23(5), 2760. https://doi.org/10.3390/s23052760
5. Van Leijenhorst, T., Chin, K.-W., & Lowe, D. (n.d.). On the viability and performance of DNS tunneling. Retrieved November 20, 2024, from https://www.researchgate.net/publication/252673752_On_the_Viability_and_Performance_of_DNS_Tunneling
6. Al-Kasassbeh, M., & Khairallah, T. (2019). Winning tactics with DNS tunneling. Network Security, 2019(12), 12–19. https://doi.org/10.1016/S1353-4858(19)30144-8

7. Wang, L., Kim, H., Mittal, P., & Rexford, J. (n.d.). Programmable in-network obfuscation of DNS traffic. Retrieved November 23, 2024, from https://www.ndss-symposium.org/wp-content/uploads/dnspriv21-08-paper.pdf

8. Ekman, E. (n.d.). iodine. GitHub. Retrieved November 15, 2024, from https://github.com/yarrick/iodine

9. Dembour, O. (n.d.). dns2tcp. GitHub. Retrieved November 29, 2024, from https://github.com/alex-sector/dns2tcp

10. Bowes, R. (n.d.). dnscat2. GitHub. Retrieved November 27, 2024, from https://github.com/iagox86/dnscat2

11. Wen, S., & Dang, W. (2018). Research on Base64 encoding algorithm and PHP implementation. In Proceedings of the 26th International Conference on Geoinformatics (pp. 1–6). IEEE. https://doi.org/10.1109/GEOINFORMATICS.2018.8557068

12. Bhargavan, K., et al. (2014). Proving the TLS handshake secure (as it is). In J. A. Garay & R. Gennaro (Eds.), Advances in Cryptology – CRYPTO 2014 (Vol. 8617). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-44381-1_14

13. Das, A., Shen, M.-Y., Shashanka, M., & Wang, J. (2017). Detection of exfiltration and tunneling over DNS. In Proceedings of the 16th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 737–742). IEEE. https://doi.org/10.1109/ICMLA.2017.00-71

14. Patsakis, C., Casino, F., & Katos, V. (2020). Encrypted and covert DNS queries for botnets: Challenges and countermeasures. Computers & Security, 88, 101614. https://doi.org/10.1016/j.cose.2019.101614

15. Berg, A., & Forsberg, D. (2019). Identifying DNS-tunneled traffic with predictive models. arXiv preprint. https://doi.org/10.48550/arXiv.1906.11246

16. Dusseault, A. (2021). Methods for the prevention of data exfiltration by DNS tunneling (Master's thesis, Utica College). ProQuest. Retrieved November 29, 2024, from https://www.proquest.com/openview/8cffdd3d5c1020bf41f4e9b23c3b96b1/1?pq-origsite=gscholar&cbl=18750&diss=y