ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Конструкція прихованих каналів у протоколі IPv6 і їх вплив у інформаційній системі»

на здобуття освітнього ступеня бакалавра
зі спеціальності 123 – Комп'ютерна інженерія
(код, найменування спеціальності)
освітньо-професійної програми Комп'ютерна інженерія
(назва)

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

_____ *Артем ДЕМЧЕНКО*
(підпис) *Ім'я, ПРІЗВИЩЕ здобувача*

Виконав: здобувач вищої освіти гр. КІД-42
Артем ДЕМЧЕНКО
Ім'я, ПРІЗВИЩЕ
Керівник: Ігор БУЧЕНКО
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ
Рецензент: _____
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Комп'ютерної інженерії

Ступінь вищої освіти бакалавр

Спеціальність 123 – Комп'ютерна інженерія

Освітньо-професійна програма Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри
Комп'ютерної інженерії
_____ Наталія ЛАЩЕВСЬКА
_____
«_____» _____ 2025р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Демченко Артем Сергійович
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Конструкція прихованих каналів у протоколі IPv6 і їх вплив у інформаційній системі
керівник кваліфікаційної роботи Ігор БУЧЕНКО,
(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «24» лютого 2025р. № 56
2. Строк подання кваліфікаційної роботи «5» червня 2025р.
3. Вихідні дані до кваліфікаційної роботи: науково-технічна література;
мережеві протоколи: IPv6, ICMPv6, TCP, UDP
розширювальні заголовки протоколу IPv6: Hop-by-Hop, Destination Options, Routing Header, Fragment Header;
віртуальне середовище: Oracle VirtualBox, мережеві топології VMnet2, VMnet0, VMnet3.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Теоретичні основи протоколу IPv6 та ICMPv6
2. Методи створення прихованих каналів у мережевих протоколах
3. Реалізація прихованих каналів у IPv6 з використанням Python/Scapy
4. Аналіз вразливостей та способів обходу систем виявлення атак (IDS)
5. Дослідження впливу прихованих каналів на інформаційну безпеку систем
6. Рекомендації щодо виявлення і запобігання прихованим каналам
5. Перелік ілюстративного матеріалу: *презентація*
6. Дата видачі завдання «24» лютого 2025р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Збір даних	25.02-24.03.2025	Виконано
2	Обробка матеріалу	24.03-02.04.2025	Виконано
3	Розробка теоретичної частини	03.04-07.04.2025	Виконано
4	Розробка моделі мережі	08.04-26.04.2025	Виконано
5	Висновки за результатами аналізу	27.05-06.05.2025	Виконано
6	Оформлення пояснювальної записки	07.05-11.05.2025	Виконано
7	Розробка презентації	12.05-19.05.2025	Виконано
8	Передзахист	27.05-29.05.2025	Виконано
9	Здача в деканат	30.05-05.06.2025	Виконано

Здобувач вищої освіти _____ Артем ДЕМЧЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Ігор БУЧЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 42 стор., 25 рис., 5 табл., 20 джерел.

Мета роботи – дослідження механізмів формування прихованих каналів у протоколі IPv6, аналіз їх впливу на інформаційну безпеку, а також створення та моделювання прикладів прихованого передавання даних із використанням сучасних інструментів мережевого аналізу та генерації трафіку.

Об'єкт дослідження – передача даних у комп'ютерних мережах за допомогою прихованих каналів IPv6**.**

Предмет дослідження – приховані канали зв'язку, що створюються шляхом маніпулювання службовими полями заголовків IPv6, зокрема: Hop Limit, Flow Label, розширювальними заголовками та службовими повідомленнями ICMPv6.

Короткий зміст роботи: розглянуто архітектуру протоколу IPv6, зокрема основні та розширювальні заголовки. Проведено аналіз полів, придатних для створення прихованих каналів. Теоретичні положення підтверджені практичною реалізацією на базі Kali Linux із використанням інструментів Wireshark, Suricata, Scapy та віртуальних маршрутизаторів Cisco CSR1000v. Побудовано приклади прихованих каналів, здійснено захоплення трафіку та оцінено можливості їх виявлення сучасними IDS-рішеннями. Результати показали високий потенціал IPv6 як середовища для прихованих комунікацій, що потребує оновлення засобів контролю та моніторингу.

КЛЮЧОВІ СЛОВА: IPV6, ПРИХОВАНІ КАНАЛИ, РОЗШИРЮВАЛЬНІ ЗАГОЛОВКИ, SCAPY, ІНФОРМАЦІЙНА БЕЗПЕКА, HOP LIMIT, FLOW LABEL, ICMPV6, DOH, WIRESHARK, SURICATA.

ЗМІСТ

ВСТУП……..……..... 8
1 ПРОТОКОЛ IPV6 ТА ЙОГО ОСОБЛИВОСТІ…… 9
1.1 Походження IPv6……..... 9
1.2 Властивості протоколу……..... 10
1.3 Структура заголовка IPv6……..... 12
1.4 Розширювальні заголовки IPv6……..... 16
1.5 Протокол управління повідомленнями Інтернету для IPv6……..... 20
2 ПРИХОВАНІ КАНАЛИ В IPV6……..... 22
2.1 Основи побудови та класифікація прихованих каналів……... 22
2.2 Вплив прихованих каналів на безпеку інформаційних систем ……..... 28
3 АРХІТЕКТУРА ТА РЕАЛІЗАЦІЯ СЕРЕДОВИЩА ……..….... 30
3.1 Структури відправника та приймача……..... 30
3.2 Використані інструменти, платформи та застосунки……..…… 33
3.3 Реалізація прихованих каналів у протоколі IPv6……..…… 35
3.4 Аналіз із використанням Suricata IDS…… 39
3.5 Виявлення трафіку через Wireshark…… 41
3.6 Оцінка надійності каналів та порівняння……... 44
ВИСНОВКИ…… 48
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ…….…. 50
Додаток А. Демонстраційні матеріали…… 53

ВСТУП

Протокол IPv6 залишається відносно новим і недостатньо вивченим стандартом інтернет-протоколу. У порівнянні зі своїм попередником – IPv4 – він надає значно більше адресного простору та низку нових функцій. Проте з цим зростає і ризик появи нових векторів атак, які можуть бути використані зловмисниками. Однією з таких загроз є використання прихованих каналів зв'язку в межах протоколів IPv6 та ICMPv6. Приховані канали – це методи передачі інформації, що маскуються під легітимний мережевий трафік. Вони дозволяють непомітно передавати дані, обходячи системи контролю, що створює серйозні ризики для інформаційної безпеки. Завдяки широким можливостям протоколу IPv6, зокрема наявності розширювальних заголовків і різноманітних службових полів, відкриваються нові можливості для вбудовування прихованих повідомлень у мережевий трафік. Такі канали можуть залишатися невиявленими навіть при активному моніторингу системами виявлення вторгнень (IDS), такими як Suricata. Метою цієї роботи є вивчення методів створення прихованих каналів у протоколі IPv6, реалізація прототипів таких каналів із використанням мови програмування Python та бібліотеки Scapy, а також дослідження їх впливу на інформаційну систему та ефективність детектування з боку IDS. У межах роботи будуть розглянуті як теоретичні аспекти роботи протоколів IPv6/ICMPv6, так і практична реалізація кількох видів прихованих каналів (через поля Hop Limit, Flow Label, Destination Options Header тощо). Окрему увагу буде приділено аналізу здатності IDS Suricata виявляти такі канали та визначенню їхньої ефективності, прихованості та потенційної загрози для корпоративних інформаційних систем.

1 ПРОТОКОЛ IPV6 ТА ЙОГО ОСОБЛИВОСТІ

1.1 Походження IPv6

На межі XX–XXI століть протокол IPv4 зіткнувся з масштабною проблемою – стрімке зростання кількості пристроїв, підключених до Інтернету, спричинило дефіцит IP-адрес, адже 32-бітний простір забезпечував лише приблизно 4,3 мільярда унікальних комбінацій. У 1990-х роках Інженерна рада Інтернету (IETF) розпочала розробку нового протоколу, що мав усунути недоліки IPv4 і забезпечити достатній запас адрес на майбутнє. Після розгляду кількох варіантів, у 1998 році був офіційно затверджений стандарт IPv6, який отримав номер RFC 2460 (Internet Protocol, Version 6 (IPv6) Specification). Основною інновацією стало впровадження 128-бітного адресного простору, який теоретично дозволяє адресувати до $3{,}4\times10^{38}$ пристроїв, що практично повністю усуває проблему вичерпання адрес. Номер версії був встановлений як 6, оскільки п’ята версія (IPv5) ніколи не використовувалася у глобальних мережах – вона залишилася експериментальною, розробленою для мультимедійної передачі в режимі потокового аудіо та відео. Активне впровадження IPv6 почалося з початку 2010-х років. Великі IT-компанії, такі як Google, Microsoft, Facebook, а також провайдери хостингу й мобільного зв’язку почали підтримувати цей протокол на своїх платформах. Проте, IPv4 все ще залишається найбільш використовуваним, і повний перехід на нову версію вимагає часу, ресурсів і узгодженості на міжнародному рівні. На сьогодні IPv6 не тільки забезпечує більший адресний простір, але й закладає фундамент для нових архітектур, таких як IoT, Smart City, масова віртуалізація, безпека та гнучка маршрутизація[1].

1.2 Властивості протоколу IPv6

IPv6 було розроблено для подолання обмежень, з якими зіткнувся протокол IPv4. Насамперед, це дефіцит адресного простору. Новий протокол має більший розмір адреси (128 біт проти 32 біт у IPv4), що дозволяє значно збільшити кількість доступних IP-адрес. Однією з ключових властивостей IPv6 є спрощення структури заголовка. Це дозволяє пришвидшити обробку пакетів маршрутизаторами та зменшити час затримки. Крім того, IPv6 підтримує автоконфігурацію – вузли можуть автоматично отримувати адресу без необхідності використання DHCP-сервера. Іншою важливою особливістю є вбудована підтримка безпеки. Протокол включає обов’язкову підтримку IPsec, що забезпечує шифрування та автентифікацію на рівні мережі. Це значно підвищує безпечність взаємодії в публічних і приватних мережах. У протоколі IPv6 також було враховано підтримку мобільності, що дозволяє пристроям переміщуватись між мережами без втрати активних з’єднань. Така функціональність є критично важливою в умовах сучасного мобільного середовища. Завдяки ієрархічній структурі адресації та спрощенню маршрутизації, IPv6 покращує ефективність роботи мереж і знижує навантаження на таблиці маршрутизації. Ці властивості є суттєвими при впровадженні протоколу в масштабних корпоративних і глобальних мережах.

Таблиця 1.1 – Порівняльна характеристика протоколів IPv4 та IPv6

Характеристика	**IPv4**	**IPv6**
Розмір адреси	32 біти (~4,3 млрд адрес)	128 бітів (~$3{,}4\times10^{38}$ адрес)
Тип адресації	Unicast, Broadcast, Multicast	Unicast, Multicast, Anycast
Конфігурація адрес	Переважно через DHCP	Підтримка SLAAC (автоконфігурація)
Безпека	IPsec – опціональний	IPsec – обов'язкова підтримка
Підтримка розширень	Обмежена	Через розширювальні заголовки
Сумісність	Масово підтримується, але обмежений	Не сумісний напряму з IPv4

Продовження таблиці 1.1 - Порівняльна характеристика протоколів IPv4 та IPv6

Характеристика	IPv4	IPv6
Фрагментація пакетів	Може виконуватись маршрутизатором	Виконується лише відправником
Підтримка QoS	Через поле Type of Service (ToS)	Через поле Flow Label
Потенціал для прихованих каналів	Обмежений	Високий – завдяки службовим полям і заголовкам

Протокол IPv6 обрано як основний об'єкт дослідження в цій кваліфікаційній роботі з причин:

IPv6 має гнучку систему розширювальних заголовків, які можуть містити довільну службову інформацію. Це створює сприятливе середовище для моделювання прихованих каналів зв’язку. Недостатній рівень моніторингу – на відміну від IPv4, більшість систем виявлення атак (IDS/IPS), зокрема з відкритим кодом (Suricata, Snort), мають обмежену кількість сигнатур для трафіку IPv6. Це дозволяє приховати зловмисну активність.

Через менший рівень розповсюдження IPv6 у порівнянні з IPv4, багато адміністраторів недостатньо обізнані у його можливостях, що створює вразливості.

Розширені поля заголовків (Hop Limit, Flow Label, Destination Options Header) обробляються лише кінцевими вузлами, або взагалі ігноруються IDS, що робить їх зручними для передачі прихованої інформації.

Дослідження прихованих каналів у IPv6 має актуальне практичне значення, оскільки дає змогу виявити вразливості в новітніх реалізаціях мережевих протоколів.

Для повного розуміння значущості IPv6 важливо оцінити не лише його переваги, а й обмеження, з якими стикаються адміністратори мереж при його впровадженні.

Переваги IPv6:

– *чітка та спрощена структура заголовка*. Заголовок IPv6 спроектований так, щоб полегшити обробку маршрутами. Завдяки спрощеному дизайну він зменшує час, необхідний для аналізу кожного пакету;

– *гігантський адресний простір*. У порівнянні з IPv4, нова версія забезпечує практично необмежену кількість адрес, що дозволяє кожному пристрою мати публічну унікальну адресу.

– *покращена підтримка мультікасту*. Механізми групової адресації (multicast) реалізовані ефективніше, ніж у попередній версії, що знижує навантаження на мережу при трансляції до кількох вузлів;

– *пряма підтримка P2P-комунікацій*. Завдяки можливості глобальної адресації кожен пристрій може безпосередньо комунікувати з іншим без необхідності у NAT або проксі-серверах;

– *автоматична конфігурація адрес (SLAAC)*. Технологія Stateless Address Autoconfiguration дозволяє автоматично налаштовувати адреси без використання DHCP-серверів, що значно спрощує управління мережею;

– *вбудована підтримка шифрування через IPsec*. Протокол підтримує IPsec на рівні специфікації, забезпечуючи базові засоби захисту вже на рівні мережевого протоколу.

Недоліки IPv6:

а) мала кількість спеціалістів. Попри те, що протокол існує вже понад два десятиліття, кількість IT-фахівців, які добре володіють його налаштуванням, залишається обмеженою. Це уповільнює його впровадження, а також уповільнюється імплементація IPv6 у безпекових рішеннях, наприклад, IDS/IPS.

б) відсутність зворотної сумісності з IPv4. IPv6 не може напряму взаємодіяти з мережами IPv4, що потребує режиму Dual Stack – одночасної підтримки обох протоколів, що ускладнює архітектуру мереж;

в) складність налаштування та підтримки. Через більшу гнучкість та наявність нових функцій, адміністрування IPv6 вимагає глибших знань і ретельнішого планування;

г) фінансові витрати на впровадження. Компанії змушені оновлювати маршрутизатори, комутатори, сервери та ПЗ, що призводить до додаткових витрат часу та коштів;

д) зростання площини атак. Нові функції протоколу IPv6 створюють додаткові можливості для зловмисників. Наприклад:

1) *SLAAC-атаки (Stateless Address Autoconfiguration)* – уразливість до Router Advertisement Spoofing, коли зловмисник надсилає фальшиві RA-пакети, змушуючи клієнтів використовувати підроблений шлюз. Це дозволяє перенаправляти трафік через машину атакувальника;

2) *атаки через розширювальні заголовки* – специфічні поля, такі як Hop-by-Hop Options або Destination Options, можна використовувати для створення прихованих каналів, оскільки більшість IDS-систем не аналізує їх повноцінно;

3) *ICMPv6-атаки*. Включають спуфінг повідомлень Neighbor Solicitation/Advertisement, що дозволяє маніпулювати ARP-таблицями аналогічно до ARP-Spoofing в IPv4.

Крім того, системи виявлення вторгнень (IDS), особливо з відкритим кодом, як-от Suricata чи Snort, все ще мають обмежену базу сигнатур для IPv6, що дозволяє зловмиснику приховувати активність у трафіку.

.3 Структура заголовка IPv6

Заголовок протоколу IPv6 має довжину 40 байт і складається з кількох фіксованих полів, кожне з яких виконує специфічну функцію в обробці та маршрутизації пакетів. У таблиці нижче наведено основні поля, їх розміри, а також можливі способи використання деяких із них для прихованого передавання інформації.

Таблиця 1.2 – Структура заголовка IPv6

Поле	**Розмір, біт**	**Використання для прихованих каналів**
Версія	4	Не використовується – завжди фіксоване значення 6
Клас трафіку	8	Можливе маскування даних шляхом зміни пріоритетів
Мітка потоку	20	Може містити закодовані дані через довільні значення
Довжина корисного навантаження	16	Може використовуватися для сигналізації про обсяг даних
Наступний заголовок	8	Використовується для приховування структури заголовків
Обмеження кількості стрибків	8	Може кодувати інформацію, оскільки змінюється в кожному пакеті
Адреса джерела	128	Можливе вставлення кодованих даних у нестандартних адресах
Адреса призначення	128	Аналогічно – потенційно використовувана частина для кодування

Отже, навіть стандартні службові поля заголовка(табл.1.1), що на перший погляд слугують лише службовим цілям, можуть бути використані для прихованого передавання інформації**,** особливо якщо їх значення змінюються нестандартним або довільним чином. У наступному підрозділі розглянемо розширювальні заголовки IPv6, які відіграють не менш важливу роль у контексті безпеки та відкривають додаткові можливості для побудови прихованих каналів.

Типи адресації в IPv6. Сучасна система IPv6 передбачає три основні типи адресації, які визначають спосіб взаємодії між мережевими пристроями:

– *індивідуальна (Unicast)* – цей тип адресації забезпечує комунікацію безпосередньо з одним конкретним пристроєм на приймальній стороні. Кожен пакет даних спрямовується саме до однієї унікальної адреси, що гарантує точне і цілеспрямоване з’єднання;

– *групова (Multicast)* – даний тип дозволяє надсилати пакети одночасно всім учасникам певної групи. Пакет, адресований на multicast-адресу, розповсюджується серед усіх членів відповідної множини пристроїв, що дозволяє ефективно організовувати передачу інформації великому колу отримувачів;

– *вибіркова (Anycast)* – принцип роботи цього типу схожий на multicast, однак існує суттєва відмінність: пакет надсилається лише одному члену групи, зазвичай тому, що він розташований найближче до відправника. Таким чином, забезпечується найкоротший шлях передачі і зменшуються затримки у мережі. Ці три типи адресації забезпечують гнучкість взаємодії в IPv6-мережах.

Формат запису IPv6. Адреса протоколу IPv6 складається із восьми груп, кожна з яких містить по 16 біт (або 4 шістнадцяткові цифри), розділених двокрапкою. Цей принцип запису дещо нагадує формат IPv4, але має суттєві відмінності за структурою. Приклад типової адреси IPv6: 4000:0000:0000:0000:0BCA:AFDE:0000:8E7F.

На відміну від IPv4, адреси IPv6 значно довші, що ускладнює їх сприйняття та використання. Тому було введено механізми спрощення та скорочення запису, щоб зробити адреси більш читабельними та зручними. Наприклад, вищенаведену адресу можна скоротити до більш компактного формату 4000::0BCA:AFDE:0000:8E7F.

Скорочення полягає в тому, що послідовні групи нулів можуть бути замінені подвійною двокрапкою «::», яка позначає одну або кілька груп, що складаються виключно з нулів. Важливо пам’ятати, що ця конструкція «::» може бути застосована лише один раз в межах однієї адреси, аби уникнути неоднозначності. Крім того, кожну групу можна додатково скоротити, прибравши провідні нулі. Наприклад, якщо у групі є 4 нулі, як у «0000», її можна замінити на один нуль «0», а в інших випадках прибрати лише ті нулі, які стоять на початку.

Це додатково зменшує довжину запису, роблячи адресу компактнішою та легшою для читання, як у прикладі: 4000::BCA:FDE:0:8E7F.

Таким чином, формат запису IPv6 поєднує в собі детальність і компактність, що дозволяє ефективно керувати великою кількістю адрес у глобальній мережі.

1.4 Розширювальні заголовки IPv6

У протоколі IPv6 передбачено використання спеціальних розширювальних заголовків, які виконують додаткові функції при маршрутизації та обробці мережевого трафіку. Такі заголовки обробляються мережевими пристроями, зокрема маршрутизаторами та комутаторами, під час проходження пакетів крізь мережу. На основі інформації з цих заголовків пристрої ухвалюють рішення – передати пакет далі або ж обробити його локально. Однією з особливостей розширювальних заголовків є те, що їх довжина завжди є кратною восьми октетам (64 бітам), що забезпечує ефективність у вирівнюванні пам’яті та обробці даних у мережевих пристроях. Згідно з технічною специфікацією RFC 2460 [7], у протоколі IPv6 передбачено шість основних типів розширювальних заголовків:

- hop-by-Hop Options Header;
- routing Header;
- fragment Header;
- destination Options Header;
- authentication Header та Encapsulating Security Payload.

Ці розширювальні заголовки мають чітко визначений порядок слідування у структурі пакета(табл. 1.2).

Таблиця 1.3 – Порядок розташування розширювальних заголовків у IPv6

Порядковий номер	**Назва заголовка**
1	Основний заголовок IPv6
2	Заголовок опцій Hop-by-Hop
3	Заголовок опцій призначення (перед маршрутизацією)
4	Заголовок маршрутизації
5	Заголовок фрагментації
6	Заголовок автентифікації
7	Заголовок інкапсульованого захищеного навантаження
8	Заголовок опцій призначення (після безпеки)
9	Заголовок протоколу вищого рівня (наприклад, TCP, UDP)

Ця впорядкованість забезпечує правильну інтерпретацію структури пакета як мережевими пристроями, так і кінцевими вузлами, що вкрай важливо для підтримки безпеки, маршрутизації та сумісності між різними реалізаціями протоколу IPv6.

Hop-by-hop options header. Цей заголовок несе службову інформацію, яку має обробити кожен маршрутизатор на шляху проходження пакета. Його основне призначення – повідомлення маршрутизаторам про наявність особливих даних, які можуть вимагати спеціальної обробки. Оскільки Hop-by-Hop Options Header завжди має бути першим серед заголовків розширення, він не може розташовуватись після інших. Це визначено стандартом протоколу IPv6, адже логіка обробки передбачає проходження заголовків у послідовності. Його присутність у пакеті сигналізується значенням 0 у полі Next Header основного заголовка IPv6. Це можна побачити на рис. 1.1. Завдяки підтримці визначених і невизначених опцій, а також змінній довжині поля, цей тип заголовка може використовуватись для побудови високошвидкісних прихованих каналів. Передача інформації через Hop-by-Hop Header є складнішою для виявлення у звичайному трафіку, оскільки не всі маршрутизатори або системи аналізу глибоко обробляють ці опції.

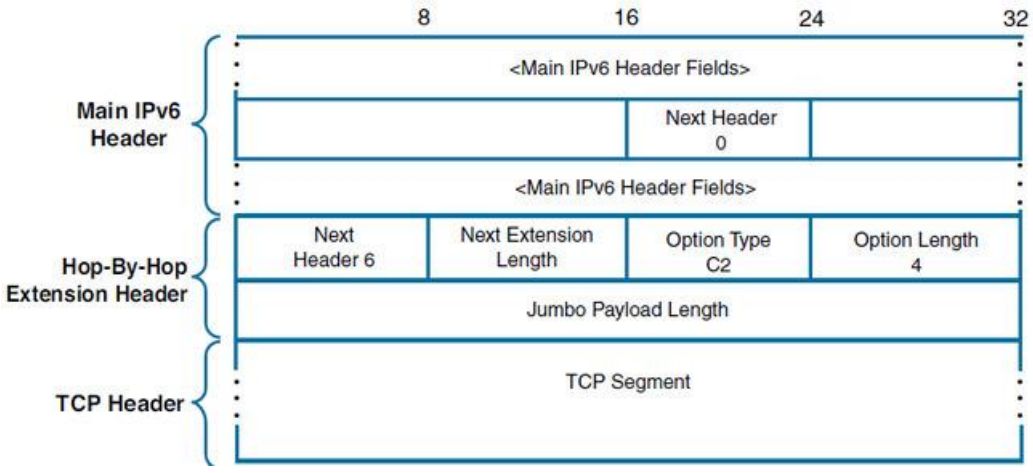


Рисунок 1.1 – Структура повного пакета IPv6

Routing Header (заголовок маршрутизації) дозволяє відправнику визначити маршрут, яким повинен слідувати IPv6-пакет до одержувача. Такий механізм

нагадує source routing у протоколі IPv4, однак реалізований більш гнучко. Цей заголовок містить список проміжних адрес, через які має пройти пакет. Під час руху мережею кожен маршрутизатор обробляє наступну адресу в списку та оновлює вміст заголовка відповідно. Це дозволяє реалізувати контроль над маршрутом передачі, що може бути як перевагою (при побудові захищених маршрутів), так і ризиком безпеки. Routing Header має кілька варіантів (типів). Найпоширеніший – Type 0, однак через відомі вразливості (наприклад, Routing Header Attack), його підтримка була знижена в сучасних реалізаціях IPv6. У контексті прихованих каналів цей заголовок потенційно може використовуватись для прихованої передачі інформації у списках маршрутів або за допомогою нестандартних типів заголовків.

Fragment Header використовується для реалізації фрагментації IPv6-пакетів. На відміну від протоколу IPv4, де фрагментація може виконуватись будь-яким маршрутизатором на шляху проходження, в IPv6 відповідальність за фрагментацію повністю покладена на відправника. Цей заголовок додається лише у випадку, якщо розмір пакета перевищує максимальний розмір передавання (MTU) для конкретного маршруту. Перед відправленням відправник виконує процедуру визначення MTU (Path MTU Discovery) і фрагментує пакет самостійно, уникаючи обробки з боку маршрутизаторів. Fragment Header містить поля, необхідні для повторної збірки пакета на приймаючій стороні: ідентифікатор, зсув фрагмента, прапорець останнього фрагмента тощо. Його наявність вказується значенням 44 у полі Next Header попереднього заголовка. У контексті прихованих каналів Fragment Header може використовуватись для передачі інформації через контроль зсувів, порядок фрагментів або навіть у змінених полях ідентифікації. Проте застосування таких методів вимагає точного контролю над пакетами і глибокого аналізу з боку приймача.

Destination Options Header. Цей тип розширеного заголовка використовується для передачі специфічної інформації, призначеної виключно для обробки кінцевими вузлами, до яких направлено IPv6-пакет. На відміну від заголовків, які аналізуються всіма проміжними вузлами на маршруті, заголовок

опцій призначення ігнорується маршрутизаторами і аналізується лише тими вузлами, для яких пакет є адресованим. Згідно з таблиці 1.2, даний заголовок може з’являтися у стеку заголовків IPv6 двічі:

– безпосередньо після заголовка Hop-by-hop, якщо опції призначені для обробки маршрутизатором або першим кінцевим вузлом на маршруті;

– у завершальній частині стеку, коли інформація повинна бути отримана безпосередньо адресатом пакета.

Формат цього заголовка аналогічний заголовку Hop-by-hop options, що забезпечує структурну уніфікацію та спрощує процес парсингу в системах аналізу пакетів. Завдяки своїй гнучкості та можливості додавати довільні параметри, цей заголовок також може бути використаний як потенційний носій прихованих каналів. Враховуючи його обробку лише кінцевими вузлами, зловмисники можуть використовувати цей простір для передачі конфіденційної або маскованої інформації, не викликаючи підозр з боку проміжної інфраструктури мережі. Таким чином, заголовок опцій призначення не лише виконує роль корисного розширення функціональності IPv6, але й є потенційно вразливою частиною протоколу з точки зору інформаційної безпеки.

Authentication Header. Це один з основних компонентів протоколу IPsec, призначений для забезпечення цілісності даних та аутентифікації джерела переданого пакета. Він дозволяє перевірити, що дані не були змінені під час передачі, та підтвердити, що вони надійшли саме від легітимного відправника. На відміну від ESP, заголовок AH не виконує шифрування вмісту, але обчислює спеціальну перевірочну величину – ICV (Integrity Check Value), яка формується на основі:

– основного заголовка IPv6;

– інших розширених заголовків (що йдуть до AH);

– та частково – вмісту пакета.

Данний механізм не потребує встановлення з’єднання (connectionless), тому є ефективним у середовищах із великою кількістю коротких сесій. AH ідентифікується значенням 51 у полі Next Header.

Encapsulating Security Payload (ESP) є одним із основних компонентів протоколу IPsec і призначений для забезпечення конфіденційності, цілісності та автентифікації даних, які передаються мережею. На відміну від заголовка автентифікації (AH), ESP не тільки перевіряє достовірність, а й шифрує частину пакета, включаючи дані вищого рівня (наприклад, TCP, UDP). ESP забезпечує захист зв'язку між кінцевими вузлами (end-to-end) або між шлюзами (gateway-to-gateway) у випадку VPN-з'єднань. Цей заголовок ідентифікується значенням 50 у полі Next Header. Через шифрування важко відслідковувати структуру вкладених даних, що створює умови для маскування.

1.5 Протокол управління повідомленнями Інтернету для IPv6

Цей протокол є невід'ємною частиною IPv6. Він використовується для індикації помилок у процесі комунікації або під час обробки відповідних пакетів. Протокол має фундаментальне значення, оскільки робота IPv6 вимагає застосування повідомлень із сучасного ICMPv6 для різних важливих процесів, як описано у RFC 4443 [3]. Він містить нові функції, характерні для IPv6, наприклад, виявлення MTU на шляху (MTU Path Discovery) та виявлення сусідів (Neighbor Discovery) [8]. Протокол є досить простим, що робить його потенційною мішенню для різних зловживань та атак. У полі Next Header він позначений значенням 58.

Структура протоколу ICMPv6 згідно з рисунком 1.2 має наступний вигляд [3]:

– *тип (8 біт)* – містить інформацію про тип повідомлення. Повідомлення про помилки мають значення в діапазоні від 0 до 127, а інформаційні повідомлення – від 128 до 255;

– *код (8 біт)* – залежить від типу повідомлення і використовується для уточнення додаткових деталей повідомлення;

– *поле контролю (checksum)* – призначене для перевірки цілісності повідомлення.

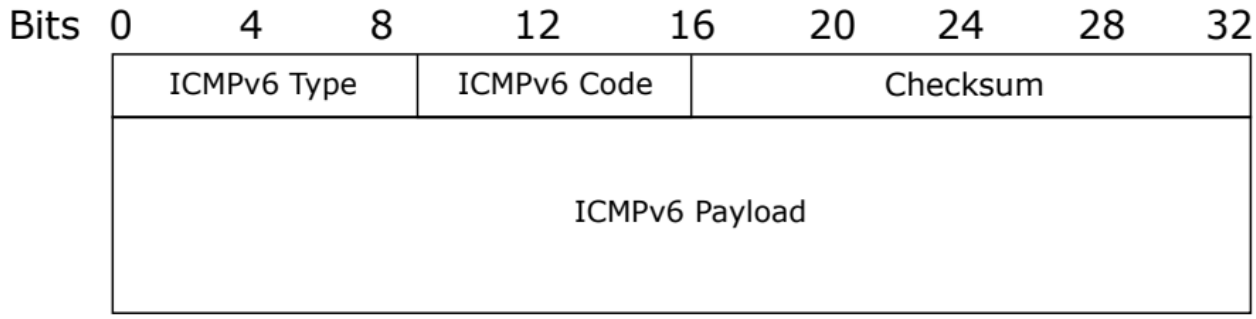


Рисунок 1.2 – Структура заголовка протоколу ICMPv6

2 ПРИХОВАНІ КАНАЛИ В IPV6

2.1 Основи побудови та класифікація прихованих каналів

При аналізі важливо оцінити обидві сторони безпеки комп'ютерних систем. Вони можуть бути публічними або прихованими. Публічний канал – це канал, призначений для авторизованої передачі даних. Прихований канал – призначений для передачі прихованої інформації, яка невидима для інших осіб і призначена лише тій, хто має її отримати. Проблема полягає в тому, що під час звичайної комунікації важко зафіксувати будь-які зміни, наприклад, у протоколі IPv6. Їх можна поділити на канали збереження та синхронізації [6]. Канал збереження – це прямий запис у сховище одним процесом і його безпосереднє зчитування іншим. Синхронізаційний канал включає сигнальні механізми, як-от процесор або час, тому зміна часу відгуку, спостережена іншим процесом, може передати інформацію.

Приховані канали, які містять приховану інформацію для сторонніх осіб, не призначені для обмеження звичайного доступу до інформації в процесі комунікації, а для забезпечення того, щоб вони не привертали уваги користувачів, а також для гарантії безпеки самих прихованих даних [9]. Прихований канал можна створити за допомогою керуючих бітів мережевих протоколів, які містяться у заголовках. Керуюча інформація у протоколі IPv6 поділяється на фіксовані поля, які є обов'язковими, та розширені заголовки, які є необов'язковими. Більшість прихованих каналів можна створити за допомогою розширених заголовків [10].

Hop Limit працює подібно до поля заголовка IPv4 TTL. Воно не дозволяє безкінечне циклювання пакетів між вузлами. Зазвичай встановлюється на 64 або 128 вузлів, що має бути достатньо для передачі пакетів у будь-якому зв'язку.

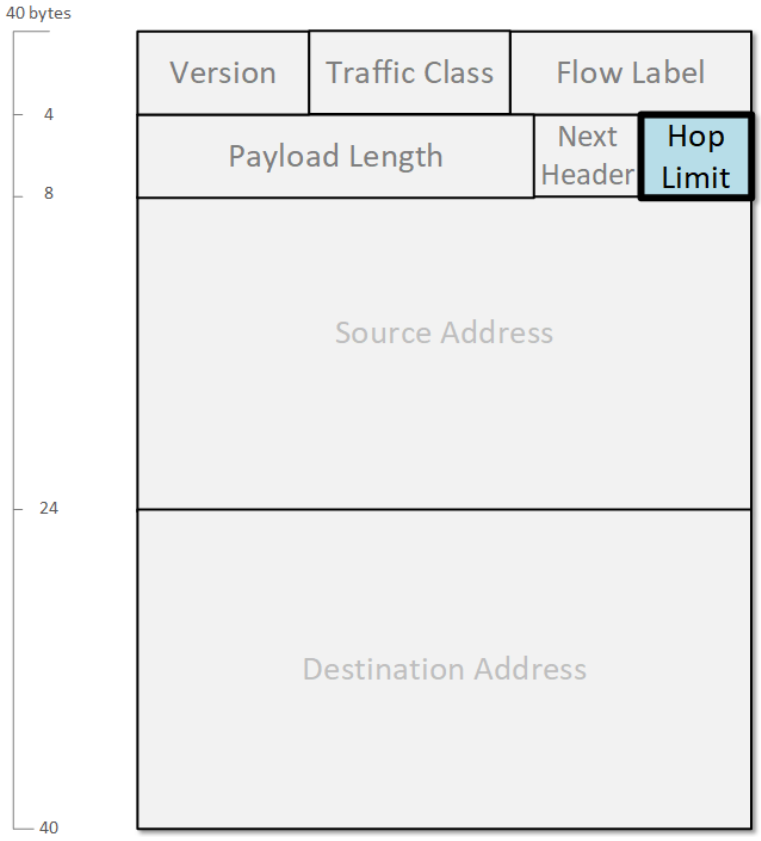


Рисунок 2.1 – Базова структура IPv6-заголовка з виділеним полем Hop Limit

Існує кілька причин, які свідчать про те, що використання цього поля для прихованого каналу є вигідним [11]:

– *різноманітність самого поля*, яке вважається нормальним. Це означає, що якщо змінити початкове значення на інше, яке ми хочемо, це вважається нормальним поведінкою. Поле складається з 8 біт (тобто максимальне значення 255), що є максимальною кількістю вузлів, через які може пройти пакет. Через часті зміни маршрутизуючої інформації в мережі зміни цього поля не будуть викликати підозр;

– *поле можна використовувати* для пошуку найближчого сервера у групі інших серверів, які надають однакові послуги. Hop Limit відсилає запит до найближчого сервера зі значенням 1 і чекає на відповідь. Якщо відповіді немає, відправляється наступний запит зі значенням на 1 більшим, ніж попереднє, і так далі, доки не буде отримано потрібну відповідь. Таким чином хост може спростити пошук, щоб не навантажувати віддалені сервери.

One-bit-per-packet channel. Основна ідея полягає в кодуванні інформації всередині цього поля. Метод полягає у заміні останнього біта поля одним бітом інформації. Цей метод має свої переваги і недоліки [11]:

– *перевага* полягає в тому, що прихований канал важче виявити у цільовому вузлі;

– *недоліком* є те, що якщо змінити значення останнього біта на потрібне, це може вплинути на коректність каналу, оскільки, наприклад, щоб передати одиницю, для більшої коректності каналу потрібно обрати відповідне число між 0 та 255, щоб це виглядало більш природньо. Але цього зробити неможливо, бо змінюється лише останній біт.

*Flow Label***.** Це поле використовується вузлом для ідентифікації IPv6-протоколу. Нульові значення Flow Label позначають пакети, які не були позначені. Класифікатори пакетів можуть використовувати трійцю: Flow Label, адресу відправника і отримувача пакета для ідентифікації потоку, до якого належить конкретний IPv6-пакет [12]. За даними [13], проміжні вузли не повинні змінювати значення Flow Label. Це значення має бути рівномірно розподілене, щоб мінімізувати ймовірність його вгадування. При використанні UDP-протоколу потік, значення Flow Label якого містить приховану інформацію, зміна ненульових значень у окремих UDP-пакетах може викликати підозру. Якщо ж поширювати Flow Label у випадкових пакетах, це буде досить складно виявити.

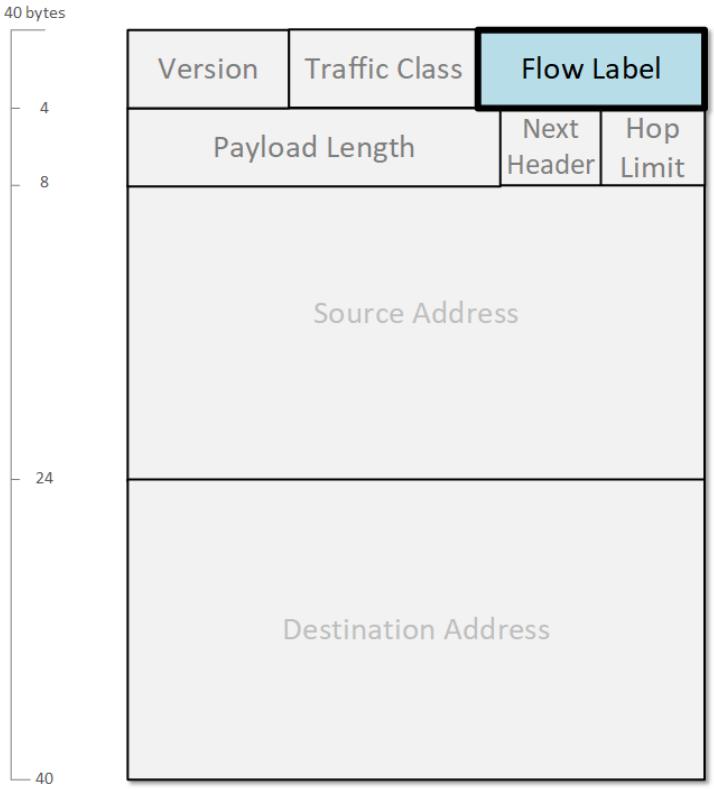


Рисунок 2.2 – Базова структура IPv6-заголовка з виділеним полем Flow Label

За даними публікації про фільтрування ICMPv6[14], реалізація Flow Label у різних операційних системах може бути недосконалою, тому перевірка його поведінки може підвищити загальний рівень безпеки. Це поле заголовка IPv6 у реальних умовах використовується рідко, тому є сенс мати можливість маніпулювати ним і передавати інформацію.

Destination Options extension Header. У списку порядку заголовків зустрічається двічі, обробляється цільовим вузлом. Інформація, яка міститься в DOH, кодується у форматі Type-Length-Value. При неправильному використанні опцій заголовка це може стати прихованим каналом для передачі необхідної інформації отримувачу.

Розширений заголовок DOH у протоколі IPv6 містить низку механізмів, що можуть бути використані для створення прихованих каналів передачі інформації. Це свідчить про те, що на сьогоднішній день даний заголовок залишається предметом активних досліджень і подальшого вдосконалення. Основною метою таких досліджень є виявлення та усунення потенційних вразливостей, які

дозволяють зловмисникам використовувати DOH для прихованої комунікації, що робить її одним із ключових напрямків для підвищення безпеки в мережах IPv6.

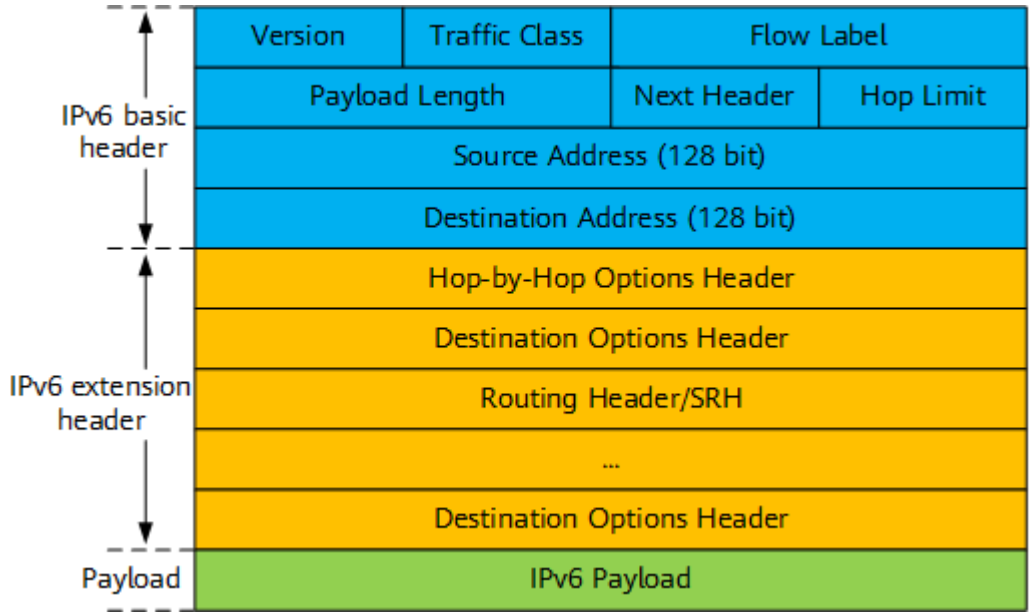


Рисунок 2.3 – Загальна структура IPv6-пакета з базовим і розширеними заголовками

Створення прихованого каналу за допомогою Unknown Option у заголовку DOH**.** Відправник, в нашому випадку потенційний зловмисник, має можливість вставити в поле типу «Option» довільні дані або надіслати «Unknown Option», починаючи з двох нулів. Якщо проміжний вузол не розпізнає цю опцію, він її пропускає і продовжує обробку пакету [7][10]. Це створює основу для формування прихованого каналу передачі даних. Для цієї мети можна використовувати діапазони, які не були зарезервовані Інтернет-організацією IANA [16]. Якщо пакет, що містить реальну IP-адресу та невідому опцію IANA у заголовку, потрапляє до цільової операційної системи, вона проігнорує цю опцію, передасть пакет на вищий рівень обробки і не розглядатиме його як підозрілий. Таким чином, ця методика прихованого каналу дозволяє непомітно передавати інформацію протоколам вищого рівня, мінімізуючи ризик виявлення.

Хоча головним фокусом дослідження є приховані канали у заголовках IPv6, важливо врахувати, що деякі службові протоколи, тісно пов’язані з IPv6, також

можуть використовуватись для прихованої передачі даних. Одним із таких є ICMPv6 – невід'ємна складова стеку IPv6, яка відповідає за обробку помилок, перенаправлення, перевірку досяжності тощо.

Поля в деяких типах ICMPv6-повідомлень, зокрема Destination Unreachable, Time Exceeded або Redirect, можуть містити невикористані або зарезервовані байти, що відкриває можливості для створення прихованих каналів. Хоча це не є частиною основного заголовка IPv6, такі методи прямо залежать від специфікації та логіки IPv6, тому доцільно розглянути їх у межах загального аналізу.

Нижче наведено типи повідомлень протоколу ICMPv6, які можуть бути використані як приховані канали для передачі інформації. Відправник вставляє секретне повідомлення у поле заголовка ICMPv6 і надсилає його отримувачу.

Повідомлення про помилку «Destination unreachable» у протоколі ICMPv6 передаються між односторонніми адресами. Таке повідомлення може бути створене будь-яким вузлом під час взаємного обміну, якщо пакет не може бути доставлений з причин, окрім перевищення часу життя пакету (TTL). Ці повідомлення відіграють важливу роль у прискоренні циклу передачі пакетів [17]. У повідомленні є спеціальне резервоване поле довжиною 4 байти. Відправник прихованого повідомлення може вставити у це поле свої дані, що забезпечує приховану пропускну здатність 32 біти на пакет [9]. Системи IDS часто ідентифікують такі пакети як підозрілі, оскільки зміни у резервованому полі зустрічаються дуже рідко.

Time exceeded. Повідомлення цього типу може виникати у двох випадках, описаних у RFC 4890 [18]:

– *код 0* генерується будь-яким вузлом на шляху проходження пакету і надсилається від цього вузла до цілі між унікастними адресами, коли значення Hop Limit у вузлі досягає нуля;

– *код 1* генерується у випадку, якщо всі сегменти фрагментованого повідомлення не були отримані кінцевими вузлами (end-to-end unicast) протягом часу повторного збирання.

Повідомлення з кодом 0 може використовуватися під час встановлення з’єднання, якщо шлях до певної цілі включає велику кількість переходів між вузлами. Недоліком цього методу є те, що обсяг інформації, який можна передати, є дуже обмеженим. Оскільки передача відбувається у двійковій формі, один символ повідомлення може відповідати приблизно 4-5 пакетам. Тому цей прихований канал підходить лише для передачі невеликого обсягу даних.

Neighbor discovery redirect. Це повідомлення також містить поле довжиною 4 байти, у яке можна вставити дані зі скритою пропускною здатністю 32 біти на пакет. Воно використовується для функцій автоконфігурації IPv6-адрес, виявлення сусідів або маршрутизаторів, управління префіксами, різноманітністю адрес, виявлення дублікатів та підтримки інформації про досяжність активних сусідів. Якщо обсяг інформації перевищує 4 байти, дані розбиваються на кілька пакетів відповідно до їх розміру. Наприклад, для передачі 12 байтів відправляються 3 пакети по 4 байти кожен. Передача може відбуватися з невеликими випадковими часовими інтервалами, щоб система IDS не розпізнала такі пакети як підозрілі.

2.2 Вплив прихованих каналів на безпеку інформаційних систем

Приховані канали в мережевих протоколах становлять серйозну загрозу для цілісності та конфіденційності даних в інформаційних системах. Їх здатність передавати дані, непомітні для звичайних засобів контролю, дозволяє зловмисникам обходити політики безпеки, системи моніторингу та фаєрволи. У протоколі IPv6 для створення прихованих каналів можуть використовуватись поля заголовка, які зазвичай не перевіряються маршрутизаторами або IDS-системами. Це знижує ймовірність виявлення прихованої передачі інформації. Наприклад, поля Hop Limit, Flow Label або Destination Options Header можуть містити дані, які не змінюють логіку маршрутизації, але переносять секретні повідомлення.

Основні загрози, пов’язані з такими каналами, включають:

– утилізацію корпоративних мереж для нелегальної передачі даних;

– витік конфіденційної інформації поза межі контрольованого середовища;

– обхід систем виявлення атак (IDS), які не аналізують менш відомі поля заголовків;

Використання для командно-контрольного (C&C) зв’язку в інфраструктурі ботнетів. Зловмисники можуть комбінувати приховані канали з методами шифрування, що ще більше ускладнює виявлення. Такі комунікації часто мають низьку пропускну здатність, але цього достатньо для передавання команд, ключів шифрування або невеликих порцій даних.

З боку захисту важливо впроваджувати поглиблену інспекцію трафіку, створювати спеціальні сигнатури для IDS/IPS та розробляти алгоритми аномального аналізу, здатні виявляти ознаки нетипової активності в полях заголовків IPv6. Таким чином, приховані канали у протоколах нового покоління мають значний потенціал впливу на інформаційну безпеку й вимагають комплексного підходу до виявлення та нейтралізації.

3 АРХІТЕКТУРА ТА РЕАЛІЗАЦІЯ СЕРЕДОВИЩА

3.1 Структури відправника та приймача

Файл covert.py виконує функцію формування та надсилання IPv6-пакетів із вбудованими прихованими повідомленнями.

Основні компоненти:

а) Функція parseArgs(): зчитує параметри запуску, такі як:

1) -c – тип каналу (наприклад, hoplimit, flowlabel, doh);
2) -i – мережевий інтерфейс (наприклад, eth0);
3) IP/MAC адреси (-sip, -dip, -smac, -dmac);
4) -m – повідомлення для прихованої передачі.

б) словник каналів (channels) асоціює назви з відповідними модулями;

в) метод main() ініціює запуск обраного методу кодування;

г) README.md містить приклади запуску з поясненням кожного параметра.

Файл decoder.py відповідає за приймання пакетів та вилучення з них прихованої інформації. Ключові елементи:

- scapy: використовується метод sniff() для перехоплення трафіку;
- функція parseArgs(): дозволяє обрати інтерфейс, тип каналу, тривалість захоплення;
- модуль декодування (наприклад, hop_limit_decoder.py) залежить від обраного типу каналу;
- словник декодерів обирає відповідний метод обробки;
- інтерактивний режим дозволяє запускати без повного набору ключів;
- README.md надає інструкції щодо запуску.

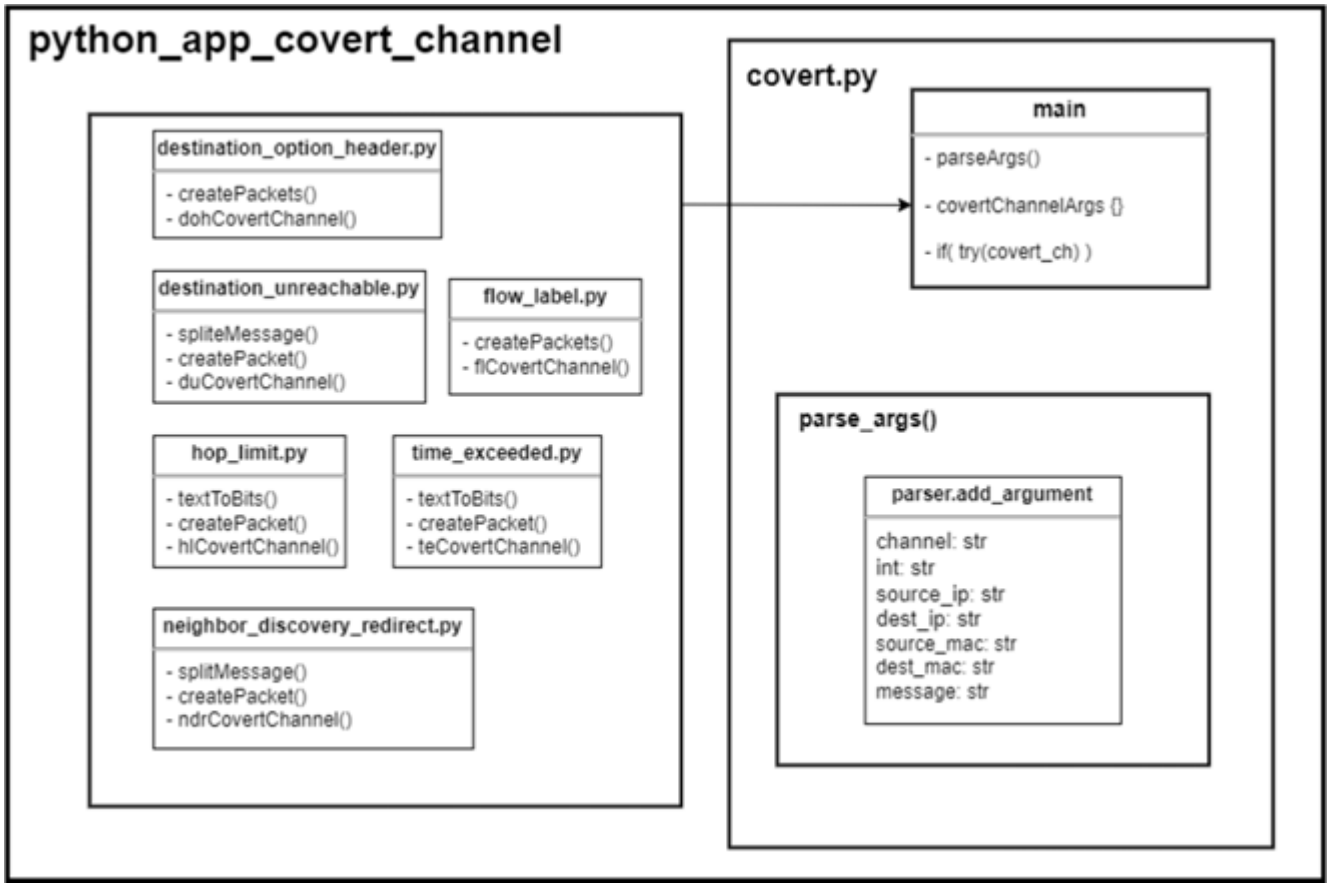


Рисунок 3.1 – Спрощена структура програми covert.py (відправник)

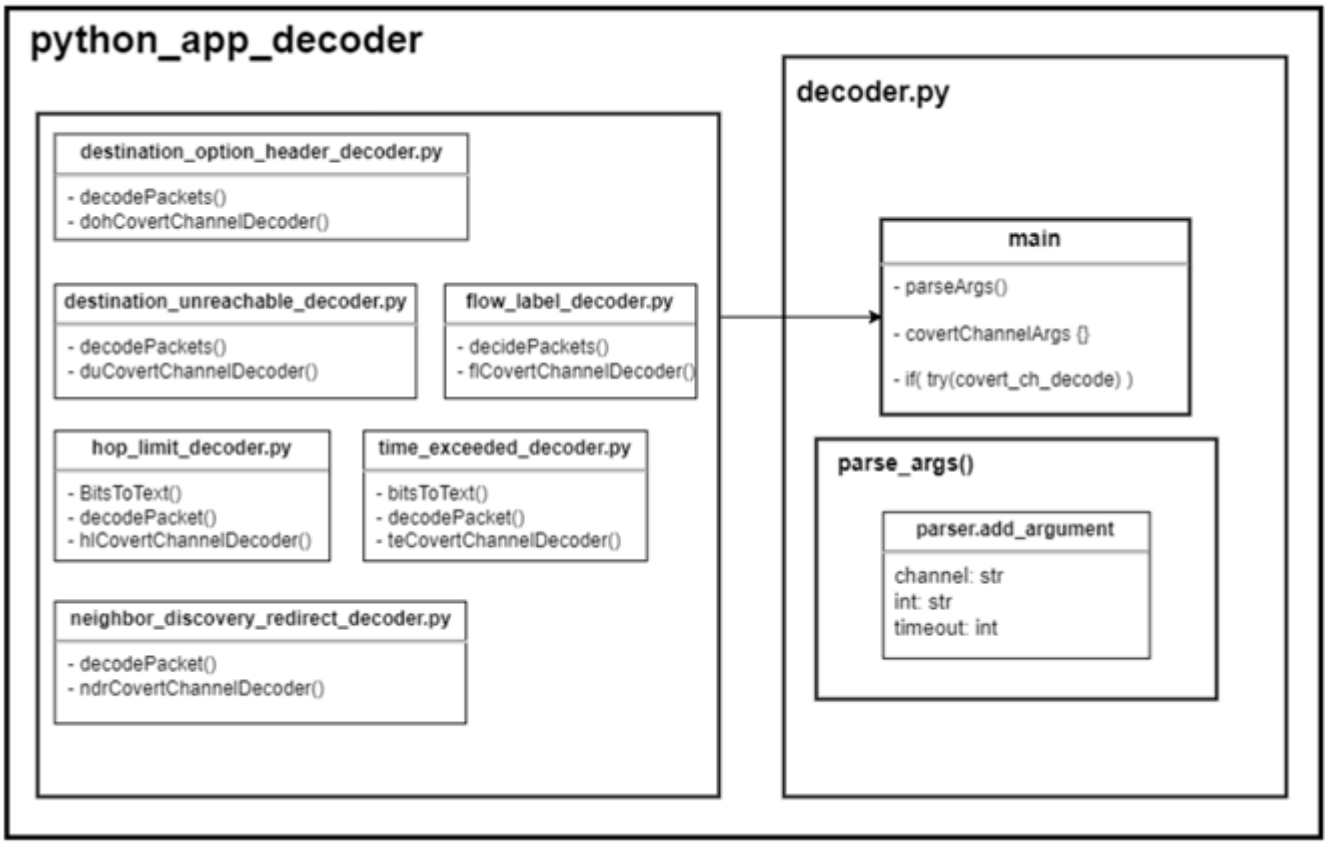


Рисунок 3.2 – Спрощена структура програми decoder.py (приймач)

Для запуску програмного забезпечення, розробленого в рамках цієї роботи, необхідно використовувати інтерфейс командного рядка. Відправник формує пакети з прихованими даними відповідно до заданих параметрів, а приймач перехоплює трафік і здійснює декодування. Загальна логіка взаємодії між програмами covert.py та decoder.py представлена нижче.

```
┌──(kali㉿kali)-[~/Desktop/ss]
└─$ sudo python3 covert.py -c hoplimit -d 2001:db8::1 -m "Hi"
```

Рисунок 3.3 – Приклад запуску відправника (Hop Limit)

```
┌──(kali㉿kali)-[~/Desktop/ss]
└─$ sudo python3 decoder.py -c hoplimit -i eth0 -t 10
```

Рисунок 3.4 – Приклад запуску приймача (Hop Limit)

```
┌──(kali㉿kali)-[~/Desktop/ss]
└─$ sudo python3 covert.py -c flowlabel -d 2001:db8::1 -m "Secret"
sudo python3 decoder.py -c flowlabel -i eth0 -t 10
```

Рисунок 3.5 – Запуск із використанням Flow Label

У наведеному прикладі(рис.3.5):

– параметр -c вказує на тип каналу (hoplimit, flowlabel, doh);
– -d – IPv6-адреса одержувача;
– -m – повідомлення для передачі;
– -i – мережевий інтерфейс;
– -t – тривалість захоплення трафіку (у секундах).

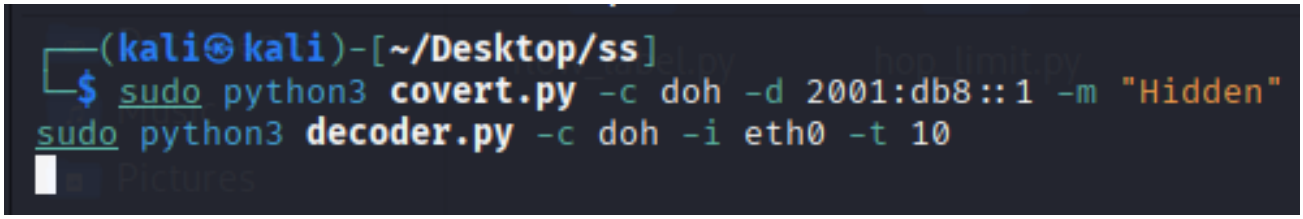


Рисунок 3.6 – Запуск із використанням Destination Options Header (DOH)

Ці команди забезпечують повний цикл: від генерації пакетів із прихованими даними до їх аналізу на приймальній стороні.

3.2 Використані інструменти, платформи та застосунки

Kali Linux – це відкрита Linux-дистрибуція, що базується на Debian і призначена для виконання різноманітних завдань у сфері інформаційної безпеки. Серед основних застосувань – пенетраційне тестування, безпековий аналіз, комп'ютерна криміналістика та реверс-інжиніринг [19]. Для реалізації завдань у цій роботі застосовується версія Kali 2023.3, інстальована на двох віртуальних машинах.

Для створення мережевої інфраструктури використано два віртуальні маршрутизатори Cisco, які надають можливість передачі пакетів поза межі локальної мережі. Вони також дозволяють організувати власну мережу, до якої Kali Linux автоматично підключається, отримуючи IPv6-адресу за допомогою технології SLAAC (Stateless Address Autoconfiguration). Обидва маршрутизатори виступають у ролі шлюзу за замовчуванням на боці відправника та отримувача. Використовується програмне забезпечення Cisco CSR1000V версії 15.4(1)S2.

Wireshark – провідний у світі аналізатор мережевих протоколів, що постійно вдосконалюється завдяки внескам експертів із різних куточків світу. Програма є спадкоємцем проекту, що розпочався у 1998 році [20]. Wireshark дозволяє захоплювати мережеві пакети, у тому числі й ті, які використовуються для прихованої передачі інформації, що робить його незамінним інструментом для

виявлення помилок, які було б надзвичайно важко ідентифікувати без спеціального програмного забезпечення. У цій роботі застосовується версія Wireshark 4.0.7. Основні переваги Wireshark [20]:

– можливість живого захоплення трафіку та аналізу офлайн;

– підтримка розшифровки багатьох протоколів, включно з IPsec, ISAKMP, Kerberos, SNMPv3, SSL/TLS, WEP, WPA/WPA2;

– наявність правил фільтрації, що забезпечують швидкий та інтуїтивний пошук необхідних пакетів.

Також використовується Python для розробки основного програмного забезпечення, що оперує бібліотекою Scapy, а також для реалізації скритого каналу передачі інформації. Конкретно застосовується версія Python 3.11.4 для створення відповідного застосунку, що досліджує та керує скритими каналами.

Suricata є потужним відкритим програмним забезпеченням для аналізу мережевого трафіку та виявлення загроз, яке широко застосовується у багатьох організаціях як приватного, так і державного секторів [22]. У рамках цієї роботи цей інструмент необхідний для глибокого аналізу підозрілого мережевого трафіку, що виникає під час процесу комунікації. Suricata дає змогу задавати власні правила, які відстежують та ідентифікують певні пакети в режимі реального часу. Налаштування та конфігурація правил у цьому програмному забезпеченні залишаються на стандартному рівні. Як показано на рис. 3.1, на стороні приймача застосовується версія Suricata 7.0.2.



Рисунок 3.7 – Приклад запуску IDS Suricata в операційній системі Kali Linux

Scapy – потужна інтерактивна бібліотека для роботи з мережевими пакетами, розроблена на мові Python. Вона дозволяє створювати, змінювати, підробляти або декодувати пакети великої кількості різних протоколів, а також порівнювати запити з відповідями [23]. Ця бібліотека є ключовим інструментом для виконання поставлених завдань у цій роботі. За допомогою функціоналу Scapy ми можемо як детально аналізувати, так і генерувати необхідні пакети IPv6 для подальшого дослідження.

3.3 Реалізація прихованих каналів у протоколі IPv6

Згідно з рисунком 3.8, зображено приклад віртуального середовища, що складається з трьох логічних підмереж: VMnet2, VMnet0 та VMnet3, об'єднаних через маршрутизатори Cisco CSR1000v, які реалізують маршрутизацію IPv6.

VMnet2 – Зона відправника (вузол-джерело трафіку):

а) тут знаходиться машина Kali Linux**,** яка виконує роль відправника IPv6-пакетів із прихованим каналом.;

б) машина підключена до мережі через інтерфейс eth0 та використовує протокол SLAAC для автоматичної конфігурації IPv6-адреси: 2001:db8:aacc:0:e4f6:91ee:bb0e:7db6 /64;

в) трафік передається через інтерфейс GigabitEthernet2 маршрутизатора R1 у мережу VMnet0;

г) у пакетах містяться модифіковані поля (наприклад, Hop Limit, Flow Label або DOH), які несуть приховану інформацію**.**

VMnet0 – Транзитна зона (внутрішня мережа маршрутизаторів):

д) у цій зоні працює маршрутизатор R1, який приймає IPv6-пакети з мережі VMnet2;

дані передаються далі через інтерфейс GigabitEthernet1 маршрутизатора R2.

ж) маршрутизатори ідентифікуються за MAC-адресами інтерфейсів G1 і G2, наприклад:

1) R1 → 00:0c:29:8a:7a:10

2) R2 → 00:0c:29:3e:e1:6d

VMnet3 – Зона приймача (вузол-аналізатор):

а) пакети надходять на Kali Linux, який підключений до мережі через eth0 та має адресу: 2001:db8:bbcc:0:cb64:306c:e05d:53e7 /64;

б) перед доставкою на Kali Linux, трафік аналізується системою Suricata IDS, яка розміщена між R2 та кінцевим вузлом;

в) Suricata здійснює перевірку:

1) якщо пакет визначається як підозрілий (наприклад, має невідому опцію у DOH або дивне значення Hop Limit), він видаляється;

2) інакше, пакет передається Kali Linux для подальшої обробки та дешифрування прихованої інформації;

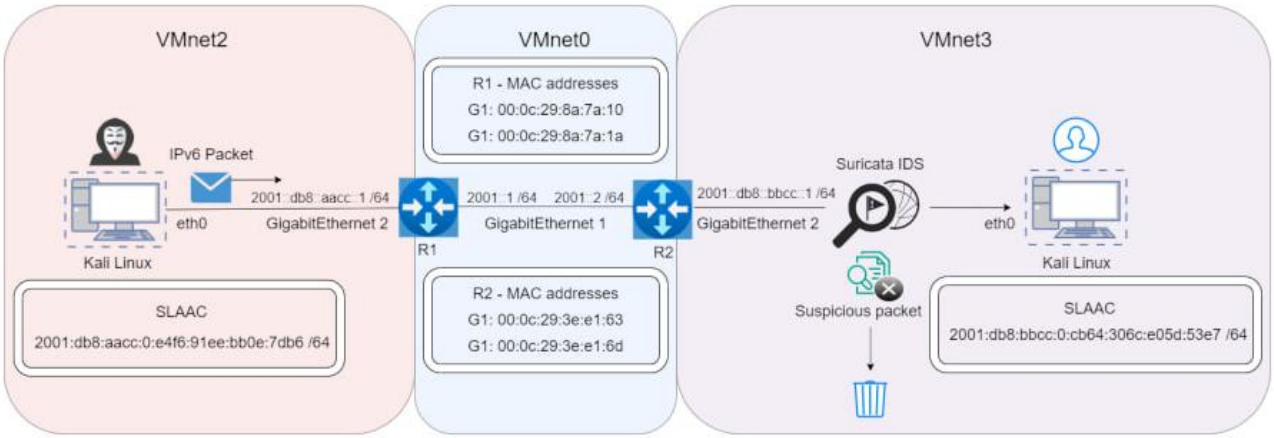


Рисунок 3.8 – Процес комунікації

Практична реалізація прихованих каналів у протоколі. Метою є демонстрація можливості вбудувати інформацію у службові поля пакетів IPv6 для прихованої передачі даних. Для цього було розроблено два окремі програмні модулі:

– відправник, який модифікує службові поля пакету;

– приймач, який прослуховує мережу та витягує приховану інформацію.

Передача даних здійснюється трьома способами:

– через поле Hop Limit – модифікація останнього біта для передачі одного біта на пакет;

– через поле Flow Label – використання 20 біт для кодування кількох бітів одночасно;

– через Destination Options Header (DOH) – вставка опцій із довільними даними.

Hop Limit є 8-бітним полем у заголовку IPv6, яке зменшується при проходженні пакета через кожен маршрутизатор. Його основна функція – обмеження часу життя пакета в мережі, подібно до TTL в IPv4. Метод прихованої передачі даних реалізується шляхом модифікації значення Hop Limit на рівні одного біта. Наприклад:

– значення 64 умовно відповідає біту «0»;

– значення 65 – біту «1».

Таким чином, передаючи послідовність пакетів із визначеним значенням Hop Limit, можна кодувати довільне бінарне повідомлення.

Для реалізації цього підходу було створено Python-скрипт hop_limit.py, який формує та відправляє IPv6-пакети зі змінним значенням поля Hop Limit відповідно до вхідного повідомлення у бітовому представленні.

```
from scapy.all import IPv6, send
import time


destination = "2001:db8::1"


binary_data = "101101"

print("[*] Передача даних через Hop Limit:")

for bit in binary_data:
    hop_value = 65 if bit == "1" else 64
    pkt = IPv6(dst=destination, hlim=hop_value)
    send(pkt)
    print(f"[+] Відправлено біт {bit} (Hop Limit = {hop_value})")
    time.sleep(0.5)
```

Рисунок 3.9 – Скрипт hop_limit.py для передавання бітів через Hop Limit у IPv6

Поле Flow Label у заголовку IPv6 є 20-бітним і призначене для ідентифікації потоків даних, що потребують однакового оброблення – наприклад, з боку маршрутизаторів або QoS-механізмів. У реальних мережах це поле рідко використовується повноцінно, що відкриває можливість застосування його для побудови прихованих каналів.

Суть методу полягає в тому, що значення Flow Label штучно встановлюється таким чином, щоби закодувати певну бітову послідовність. На відміну від Hop Limit, яке змінюється на кожному маршрутизаторі, Flow Label зберігає свою цілісність при проходженні маршрутом, що робить його більш надійним з точки зору доставки прихованого повідомлення.

```
from scapy.all import IPv6, send
import time

destination = "2001:db8::1"

binary_data = "101010"

print("[*] Передача даних через поле Flow Label:")

for i in range(len(binary_data)):

    flow_bits = int(binary_data[i]) << (19 - i)

    pkt = IPv6(dst=destination, fl=flow_bits) / "Hidden data"
    send(pkt)

    print(f"[+] Відправлено біт {binary_data[i]} (Flow Label = {flow_bits})")
    time.sleep(0.5)
```

Рисунок 3.10 – Скрипт flow_label.py з кодуванням бітів у Flow Label

Останній зі способів прихованого передавання даних у протоколі IPv6, який було використано, полягає у використанні заголовка опцій призначення (Destination Options Header, DOH). Цей заголовок дозволяє вбудовувати спеціальні параметри, які не аналізуються проміжними маршрутизаторами, а лише обробляються кінцевими вузлами, що робить його привабливим для побудови прихованих каналів.

Зловмисник може створити такий DOH-заголовок, що міститиме невідомий тип опції (Unknown Option) з довільними даними. Згідно з RFC 8200, формат опції дозволяє використовувати довільні 8-бітні ідентифікатори (Option Type) та вміст (Option Data), який не впливає на маршрутизацію, але може бути прочитаний спеціальною програмою на приймальній стороні.

```
from scapy.all import *

doh_option = HBHOptUnknown(
    otype=0b00000000,     # Користувацький тип опції (можна змінити)
    optdata=b"secret123"  # Закодоване повідомлення
)

pkt = IPv6(dst="2001:db8::1") / IPv6ExtHdrDestOpt(options=[doh_option]) / ICMPv6EchoRequest()

send(pkt)
```

Рисунок 3.11 – Скрипт створення прихованого каналу через Destination Options Header (DOH)

3.4 Аналіз із використанням Suricata IDS

З метою виявлення потенційних прихованих каналів було використано Suricata IDS. В результаті запуску системи з конфігурацією /etc/suricata/suricata.yaml та моніторингом інтерфейсу eth0, вдалося зафіксувати тривожне повідомлення:DOH з Unknown Option був помічений лише при ручному аналізі. [**] [1:2200088:2] SURICATA IPv6 DSTOPTS unknown option – свідчить про здатність IDS виявити змінені або нестандартні пакети з DOH-опціями.

```
┌──(root㉿kali)-[/home/kali/Desktop/python_app_covert_channel]
└─# python covert.py -i eth0 -sip 2001:db8:aacc:0:e4f6:91ee:bb0e:7db6 -dip 2001:db8:bbcc:0:cb64:306c:e05d:53e7 -smac 00:0c:29:74:69:11 -dmac
00:0c:29:8a:7a:1a -m "qwe" -c du
Packet Sent!
```

Рисунок 3.12 – Запуск Suricata в Kali Linux

У процесі моніторингу за допомогою правил, активних за замовчуванням, Suricata успішно виявила пакет із невідомою опцією у Destination Options Header, згенерований скриптом covert.py.

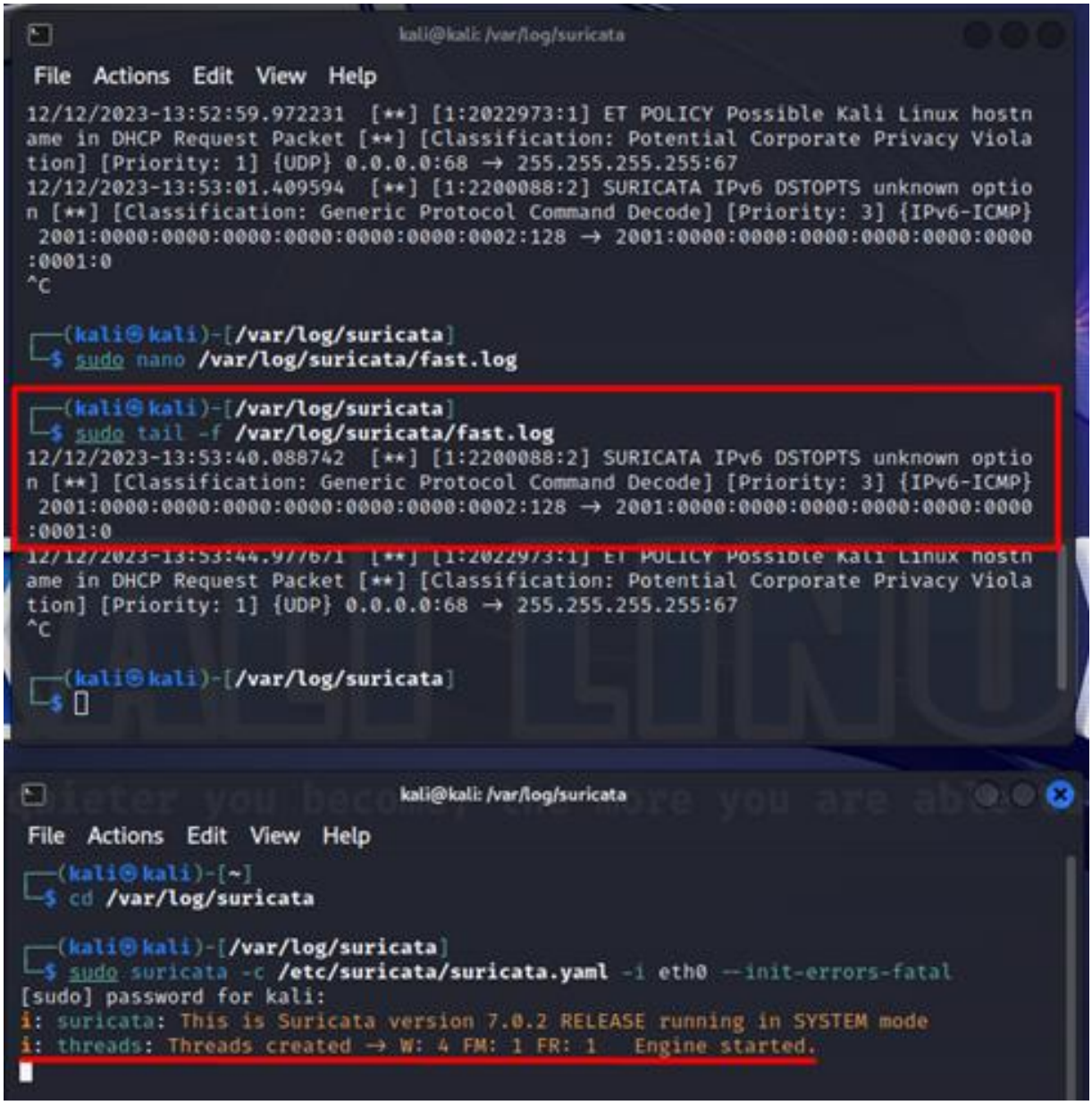


Рисунок 3.13 – Приклад виявлення пакетів із прихованим каналом у системі IDS Suricata

3.5 Виявлення трафіку через Wireshark

Під час захоплення трафіку було зафіксовано передачу пакету з прихованим повідомленням через кожен тип каналу. На рис. 3.14, 3.15 і 3.16 показано приклади перехоплених пакетів у Wireshark.

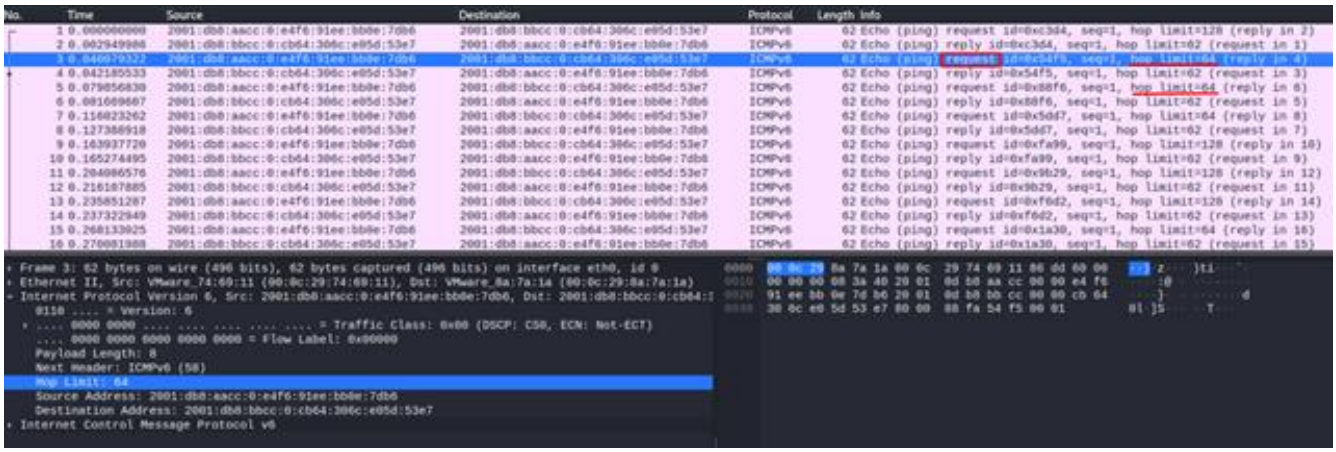


Рисунок 3.14 – Передача бітів через поле Hop Limit (TTL: 64/65)

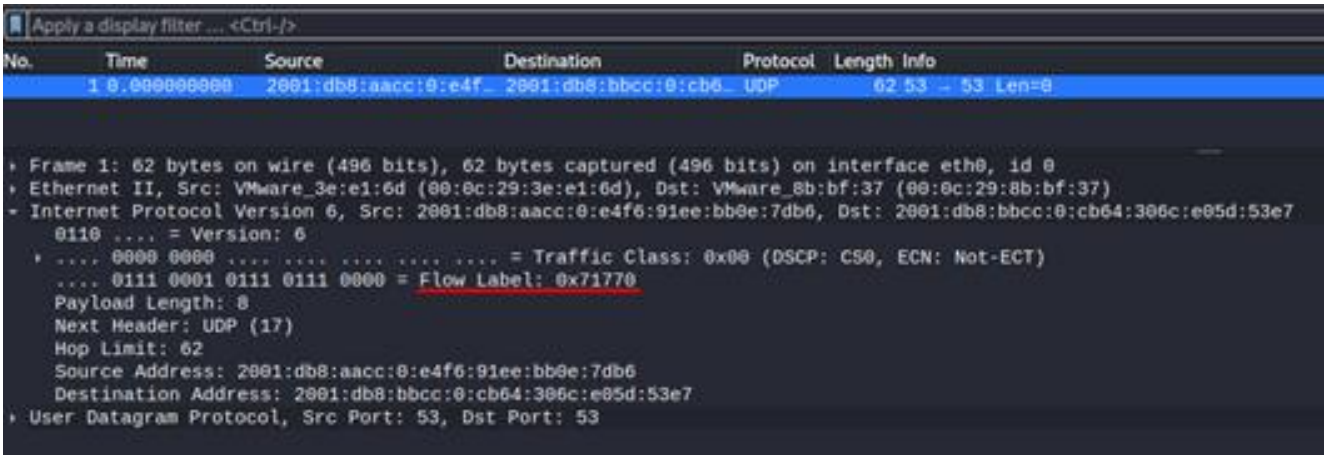


Рисунок 3.15 – Використання Flow Label для кодування бітів

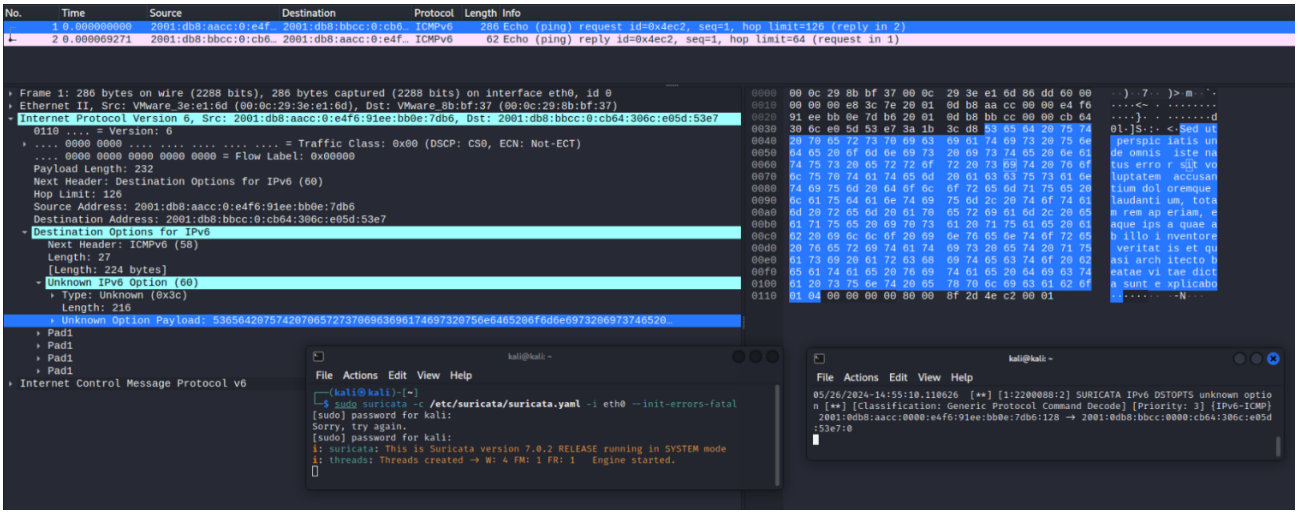


Рисунок 3.16 – Передача повідомлення через Destination Options Header (DOH)

Також зображено запуск скриптів:

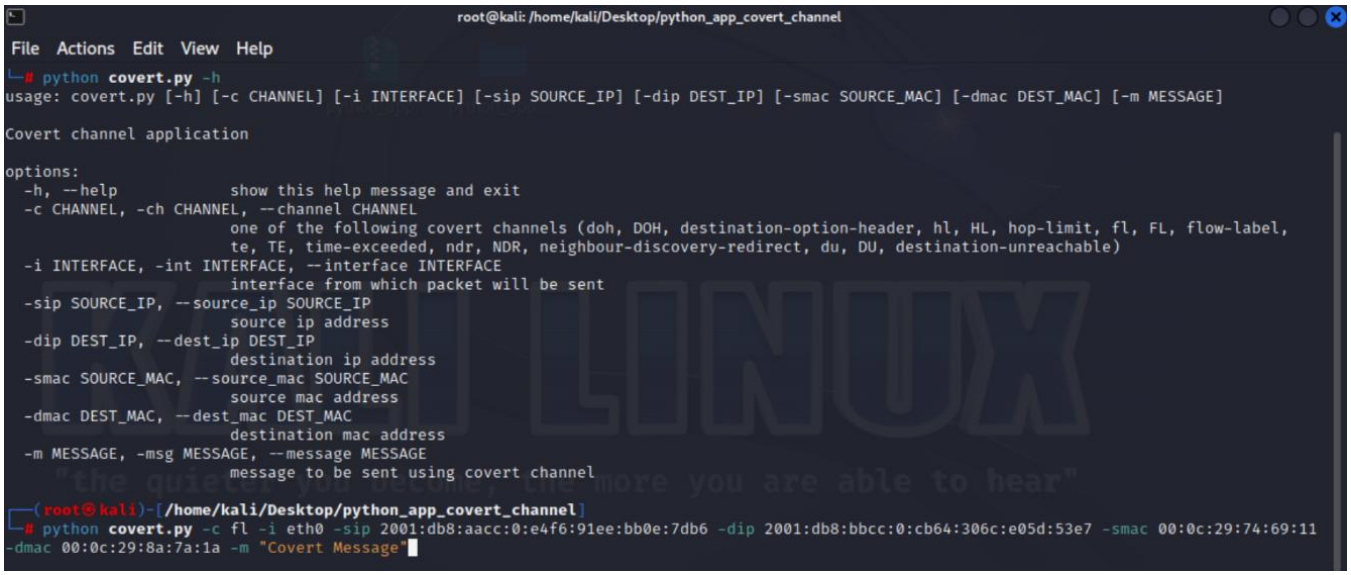


Рисунок 3.17 – Використання covert.py з аргументами

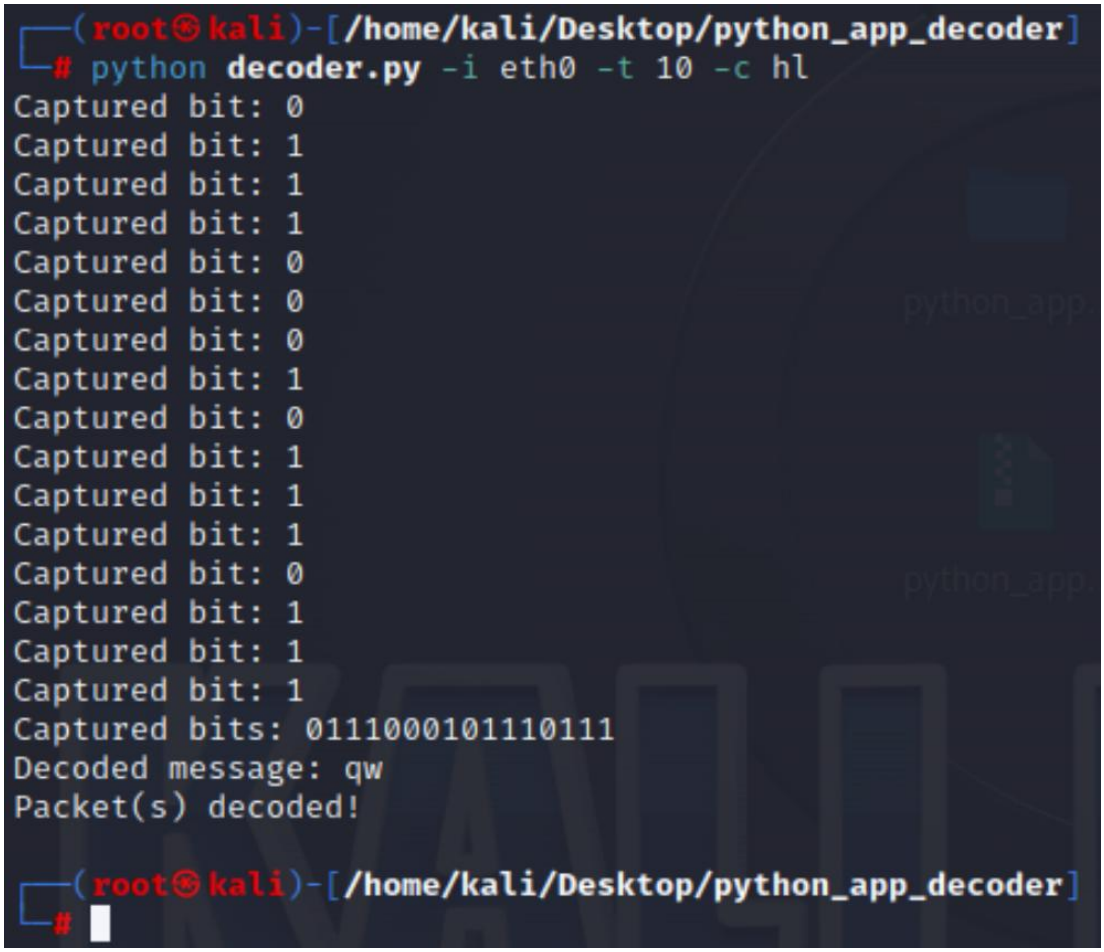


Рисунок 3.18 – Результат decoder.py (Hop Limit)

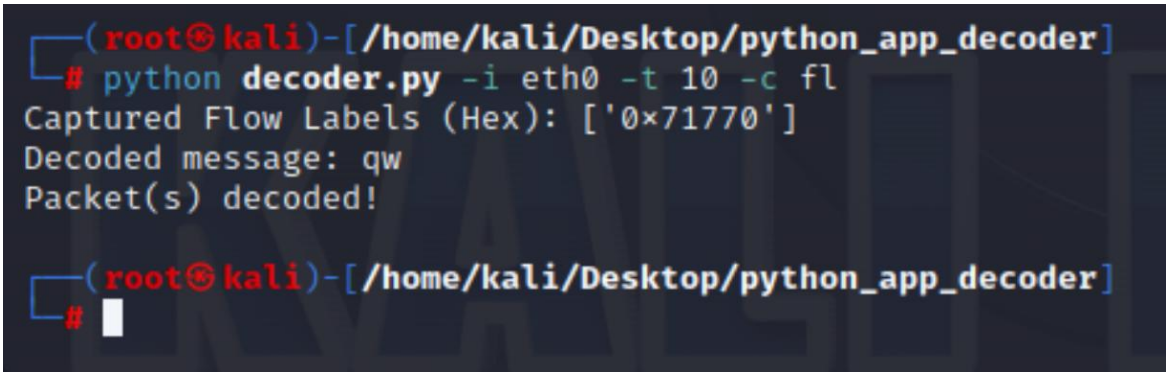


Рисунок 3.19 – Результат decoder.py (Flow Label)

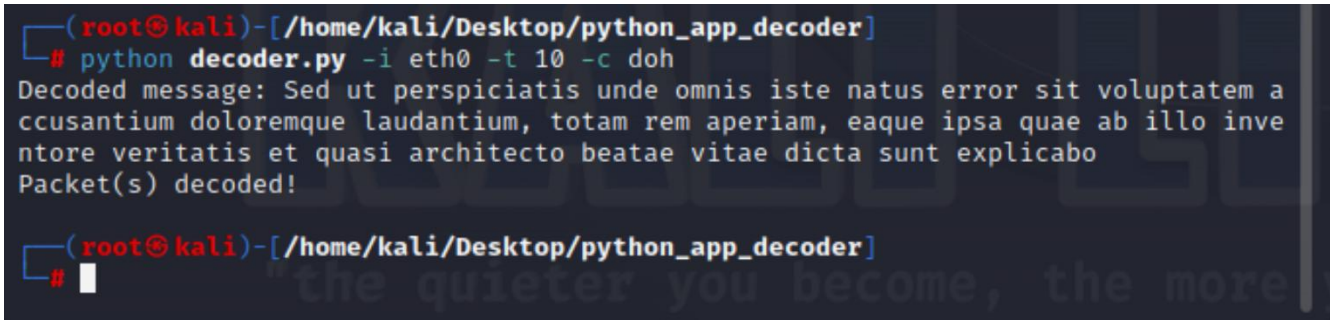


Рисунок 3.20 – Результат decoder.py (DOH)

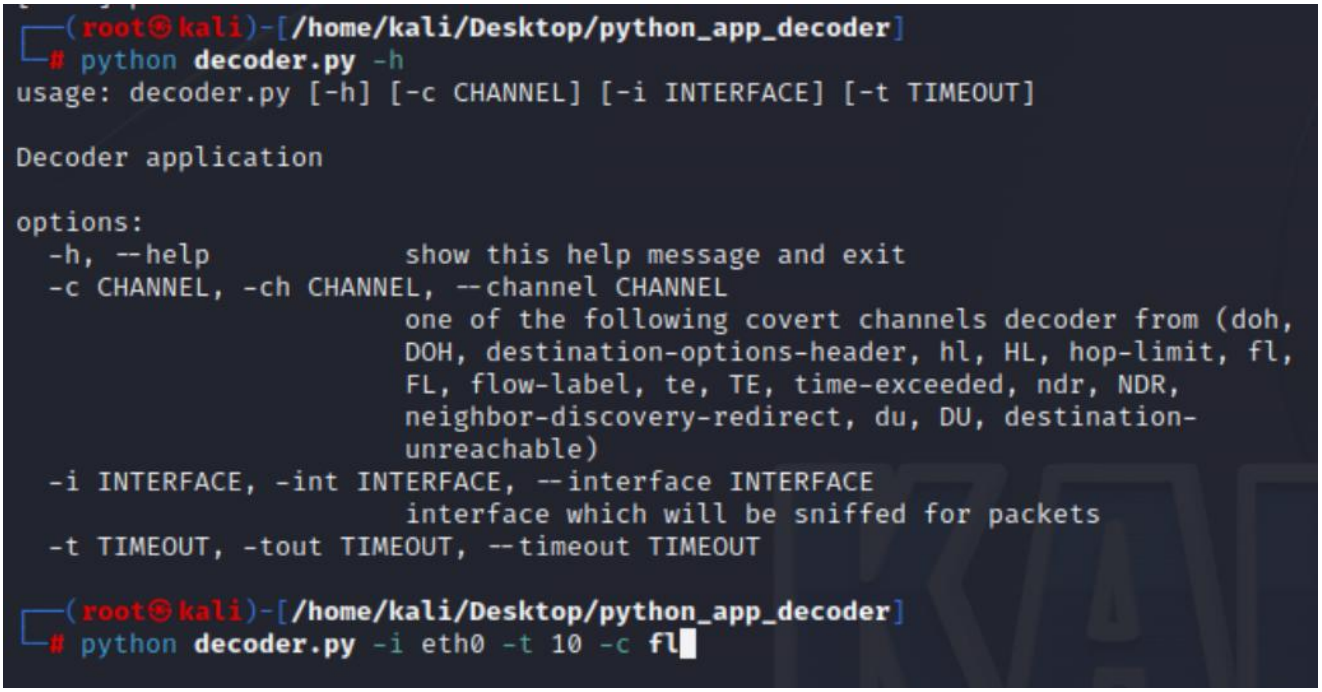


Рисунок 3.21 – Виявлення DOH через /var/log/suricata/fast.log

3.6 Оцінка результатів надійності каналів та порівняння

В рамках експерименту було протестовано три типи прихованих каналів. Передача повідомлень здійснювалась у віртуальному середовищі Kali Linux, а зчитування – через decoder.py.

Таблиця 3.1 – Порівняльна таблиця надійності прихованих каналів у IPv6

Тип каналу	**Максимальна довжина повідомлення**	**Надійність передачі**	**Швидкість (біт/пакет)**
Hop Limit	до 32 біт (рядок ~4 символи)	Висока	1 біт / пакет
Flow Label	до 20 біт за раз	Середня	20 біт / пакет
DOH	>100 байт	Висока	змінна (вся опція)

Загалом, приховані канали, реалізовані в межах експерименту, продемонстрували різну ефективність залежно від обраного методу. Канал на основі поля Hop Limit виявився найбільш стабільним і надійним. Завдяки тому, що значення Hop Limit, як правило, не змінюється маршрутизаторами та легко піддається контролю, передача даних через модифікацію останнього біта цього поля відбувалася точно та без спотворень. Однак основним недоліком цього методу є його низька пропускна здатність – лише 1 біт на пакет, що суттєво обмежує швидкість передавання повідомлень.

Flow Label, навпаки, дозволив збільшити обсяг даних, які можуть бути вкладені в один пакет. Його 20-бітна структура дає змогу кодувати більші обсяги інформації за один сеанс передачі. Але варто зазначити, що цей канал виявився менш стабільним: у деяких випадках значення поля Flow Label було змінено під час маршрутизації або проігноровано мережевими пристроями, що призводило до втрат або спотворень переданої інформації. Надійність у порівнянні з Hop Limit була дещо нижчою, хоча швидкість передачі значно вища.

Найбільш гнучким і ємним виявився канал, реалізований через Destination Options Header (DOH). Завдяки можливості вкладати в опції довільну кількість байтів, цей метод забезпечив передавання повноцінних текстових повідомлень і навіть бінарних структур. Пакети з DOH успішно доходили до кінцевого вузла без втрат, за умови коректного оброблення розширених заголовків. Водночас, через нестандартну структуру пакету, такий канал має підвищену помітність для систем виявлення атак (IDS), зокрема Suricata, яка зафіксувала аномалії при передачі DOH-опцій з невідомим типом.

Таким чином, Hop Limit підходить для надійної, але повільної передачі, Flow Label – для швидкої, проте менш стабільної, а DOH – для гнучкої та об’ємної, але потенційно виявленої передачі. У залежності від сценарію використання, кожен з каналів може бути оптимальним за певних умов.

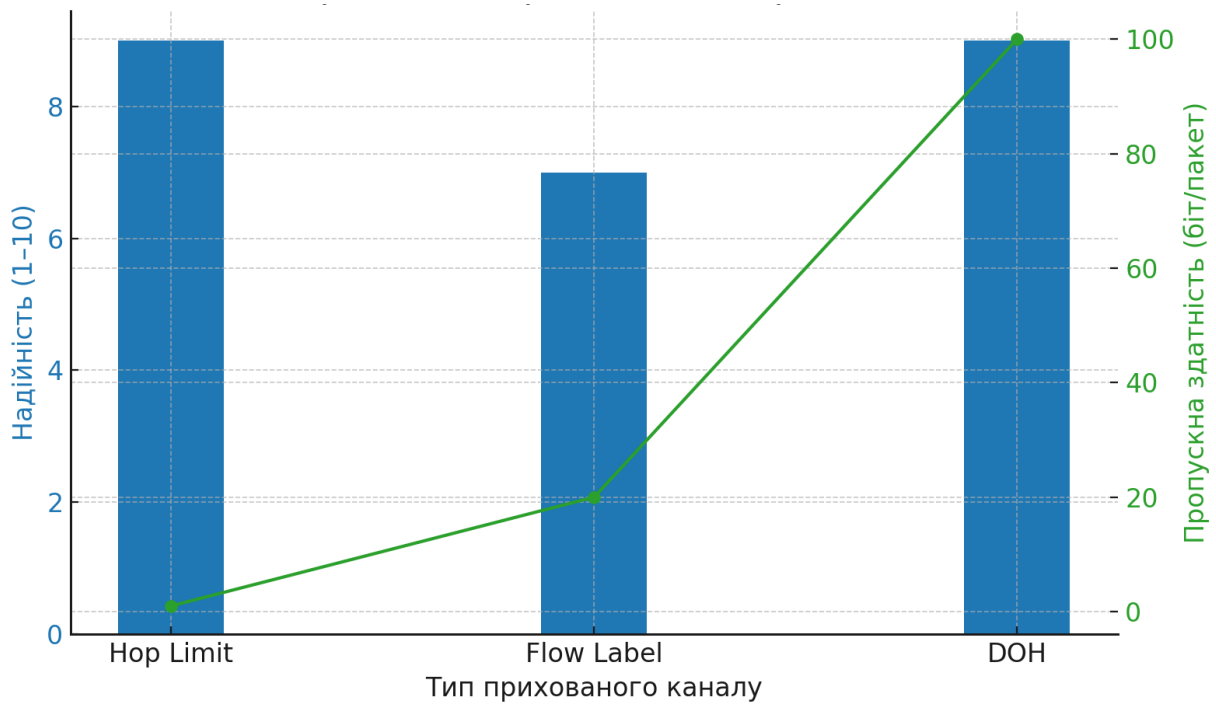


Рисунок 3.22 – Графік порівняння ефективності прихованих каналів

На графіку(рис.4.1) зображено співвідношення між надійністю (оціненою за шкалою 1–10) та пропускною здатністю (у бітах на пакет) для трьох типів каналів: Hop Limit, Flow Label і Destination Options Header (DOH). Видно, що Hop Limit забезпечує високу надійність при низькій швидкості, DOH – найвищу швидкість при високій вразливості до виявлення, а Flow Label займає проміжну позицію.

Рекомендації для виявлення прихованих каналів у IPv6 слід:

– створювати кастомні сигнатури в IDS;
– аналізувати аномалії в полях заголовків;
– логувати змінні заголовки для подальшого порівняння;
– IDS має бути налаштована для аналізу розширювальних заголовків IPv6;
– DOH потрібно перевіряти навіть без відомих опцій;

– використання Wireshark з фільтрами ipv6.hlim, ipv6.fl, ipv6.dstopt дозволяє ефективно відслідковувати підозрілий трафік.

Дослідження підтвердило, що специфікація IPv6, попри свою гнучкість, може бути використана для прихованої передачі даних. Аналіз з використанням Scapy, Wireshark і Suricata показав, що більшість IDS-систем не розпізнають подібні канали без спеціального налаштування. Це створює нові виклики для фахівців з безпеки, яким слід адаптувати політики моніторингу до розширених заголовків та поведінкових аномалій у трафіку IPv6.

Результати експерименту демонструють, що IPv6 дозволяє реалізовувати приховані канали без зміни основного функціоналу протоколу. Для їх виявлення необхідна глибока інспекція пакетів та спеціалізовані сигнатури.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було проведено глибоке дослідження прихованих каналів у протоколі IPv6, яке охопило як теоретичні, так і практичні аспекти їх функціонування, виявлення та протидії. IPv6, як сучасний мережевий протокол, має значні переваги над своїм попередником, зокрема розширений адресний простір, підтримку розширюваних заголовків, удосконалене управління маршрутизацією та автоконфігурацією. Проте ці можливості також відкривають простір для потенційних загроз, зокрема прихованої передачі даних через службові поля заголовків.

У межах цієї роботи було виконано аналіз заголовків IPv6 та ICMPv6, виявлено поля, що можуть бути використані як носії прихованої інформації – Hop Limit, Flow Label, Destination Options Header створено приклади прихованої передачі інформації з використанням мови Python та бібліотеки Scapy. Розгорнуто тестове середовище на базі Kali Linux та Cisco CSR1000v з подальшим аналізом трафіку через Wireshark і Suricata IDS. Оцінено здатність IDS-систем виявляти нетипові ознаки прихованих каналів у IPv6, в тому числі з використанням власних сигнатур.

Отримані результати дозволили зробити висновки. Поле Hop Limit забезпечує високий рівень прихованості, оскільки рідко сприймається як аномальне, що ускладнює його виявлення. Поле Flow Label має обмежену пропускну здатність, але також може бути використане для прихованого каналу з мінімальними ризиками. Заголовок DOH (Destination Options Header) забезпечує найвищу пропускну здатність, проте й найвищий ризик детектування при активному моніторингу.

Практичне значення роботи полягає у демонстрації того, що навіть стандартні поля IPv6 можуть бути використані для прихованої комунікації. Це створює нові виклики для систем виявлення загроз (IDS/IPS) та потребує оновлення політик безпеки в організаціях.

У рамках роботи запропоновано підходи до виявлення прихованих каналів за ознаками нетипових параметрів заголовків, модифікації існуючих IDS-сигнатур для врахування нестандартних сценаріїв у IPv6, удосконалення моніторингових процедур при переході на IPv6.

Таким чином, дослідження довело наявність нових ризиків, притаманних архітектурі IPv6, та необхідність їх урахування під час впровадження цього протоколу у корпоративні мережі. Подальші дослідження доцільно спрямувати на розширення спектру аналізованих полів, інтеграцію AI-рішень для виявлення прихованих каналів та розробку практичних захисних механізмів у середовищі IPv6.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Wikipedia contributors. (2023). IPv6. Wikipedia. https://en.wikipedia.org/wiki/IPv6
2. Gupta, M., & Conta, A. (2006). Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification. IETF. https://datatracker.ietf.org/doc/html/rfc4443#section-2
3. Hinden, B., & Deering, S. (1998). Internet Protocol, Version 6 (IPv6) Specification. IETF. https://datatracker.ietf.org/doc/html/rfc2460#section-4
4. Juniper Networks, Inc. (2023). IPv6 Flow-Based Processing Overview. https://rb.gy/ylcs3b
5. Lucena, N. B., Lewandowski, G., & Chapin, S. J. (2006). Covert channels in IPv6. In Danezis, G., & Martin, D. (Eds.), Privacy Enhancing Technologies. PET 2005. Lecture Notes in Computer Science, vol 3856. Springer. https://doi.org/10.1007/11767831_10
6. Mavani, M., & Ragha, L. (2014). Covert channel in IPv6 Destination Option Extension header. In IEEE International Conference on Computer Science and Information Technology Advances (CSCITA) (pp. 219–224). https://doi.org/10.1109/CSCITA.2014.6839262
7. Qu, H., Su, P., & Feng, D. (2004). A typical noisy covert channel in the IP protocol. In IEEE Carnahan Conference on Security Technology (pp. 189–192). https://doi.org/10.1109/CCST.2004.1405390
8. Amante, S., Carpenter, B., Jiang, S., & Boucadair, M. (2011). IPv6 Flow Label Specification. IETF. https://datatracker.ietf.org/doc/html/rfc6437#section-2
9. Amante, S., Carpenter, B., Jiang, S., & Boucadair, M. (2011). Covert Channel Risk in IPv6. IETF. https://datatracker.ietf.org/doc/html/rfc6437#section-6.1
10. Caviglione, L., Mazurczyk, W., & Wendzel, S. (2021). Code augmentation for detecting covert channels targeting the IPv6 flow label. In 2021 IEEE International

Conference on Network Softwarization (NetSoft) (pp. 450–456). https://doi.org/10.1109/NetSoft51509.2021.9492661

11. Internet Assigned Numbers Authority (IANA). Internet Protocol Version 6 Parameters. https://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xml

12. Berger, J., Klein, A., & Pinkas, B. (2020). Flaw Label: Exploiting IPv6 flow label. In IEEE Symposium on Security and Privacy (SP) (pp. 1259–1276). https://doi.org/10.1109/SP40000.2020.00075

13. Internet Assigned Numbers Authority (IANA). Internet Protocol Version 6 Parameters. https://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xml

14. Davies, E. B., & Mohácsi, J. (2007). Recommendations for Filtering ICMPv6 Messages in Firewalls. IETF. https://datatracker.ietf.org/doc/html/rfc4890#appendix-A.1

15. Davies, E. B., & Mohácsi, J. (2007). Time Exceeded Error Message. IETF. https://datatracker.ietf.org/doc/html/rfc4890#appendix-A.3

16. Offensive Security. (n.d.). Kali Linux: The most advanced Penetration Testing Distribution. https://www.kali.org

17. Wireshark Foundation. (n.d.). About Wireshark. https://www.wireshark.org/about.html

18. Python Software Foundation. (n.d.). Python Programming Language. https://www.python.org

19. OISF. (n.d.). Suricata IDS: Observe. Protect. Adapt. https://suricata.io

20. Scapy. (n.d.). Scapy: Packet crafting for Python. https://scapy.netForemski, P., Plonka, D., & Berger, A. (2016). Entropy/IP: Uncovering structure in IPv6 addresses. *arXiv preprint arXiv:1606.04327*. https://arxiv.org/abs/1606.04327

21. Caviglione, L., & Mazurczyk, W. (2022). IPv6CC: IPv6 covert channels for testing networks against stegomalware and data exfiltration. *SoftwareX, 20*, 100900. https://doi.org/10.1016/j.softx.2022.100900

22. Foremski, P., Plonka, D., & Berger, A. (2016). Entropy/IP: Uncovering structure in IPv6 addresses. *arXiv preprint arXiv:1606.04327*. https://arxiv.org/abs/1606.04327

23. Mazurczyk, W., Powojski, K., & Caviglione, L. (2019). IPv6 covert channels in the wild. In *Proceedings of the Third Central European Cybersecurity Conference* (pp. 1–6). ACM. https://doi.org/10.1145/3360664.3360674

24. Rahman, M. W. U., Lin, Y.-Z., Weeks, C., Ruddell, D., Gabriellini, J., Hayes, B., Hariri, S., & Ziegler Jr, E. V. (2025). AI/ML based detection and categorization of covert communication in IPv6 network. *arXiv preprint arXiv:2501.10627*. https://arxiv.org/abs/2501.10627

25. Saidi, S. J., Gasser, O., & Smaragdakis, G. (2022). One bad apple can spoil your IPv6 privacy. *arXiv preprint arXiv:2203.08946*. https://arxiv.org/abs/2203.08946

26. Wendzel, S., Zander, S., Fechner, B., & Herdin, C. (2014). A pattern-based survey and categorization of network covert channel techniques. *arXiv preprint arXiv:1406.2901*. https://arxiv.org/abs/1406.2901

27. Wei, M., Rong, C., Liang, E., & Zhuang, Y. (2022). An intrusion detection mechanism for IPv6-based wireless sensor networks. *International Journal of Distributed Sensor Networks, 18*(5), 15501329221077922. https://doi.org/10.1177/15501329221077922

28. Zuppelli, M., & Caviglione, L. (2021). DICCh-D: Detecting IPv6-based covert channels using deep neural networks. In *Proceedings of the 2021 International Conference on Availability, Reliability and Security* (pp. 1–8). Springer. https://doi.org/10.1007/978-3-031-20977-2_4

Додаток А

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

(Презентація)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій
Кафедра Комп'ютерної інженерії

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА

на тему: «Конструкція прихованих каналів у протоколі IPv6 і їх вплив у інформаційній системі»

Виконав студент групи КІД-42
Демченко Артем Сергійович

Керівник кваліфікаційної роботи:
Бученко Ігор Анатолійович, старший викладач кафедри КІ

Київ – 2025

Мета роботи – Дослідити механізми прихованих каналів у IPv6, їх реалізацію та вплив на безпеку.

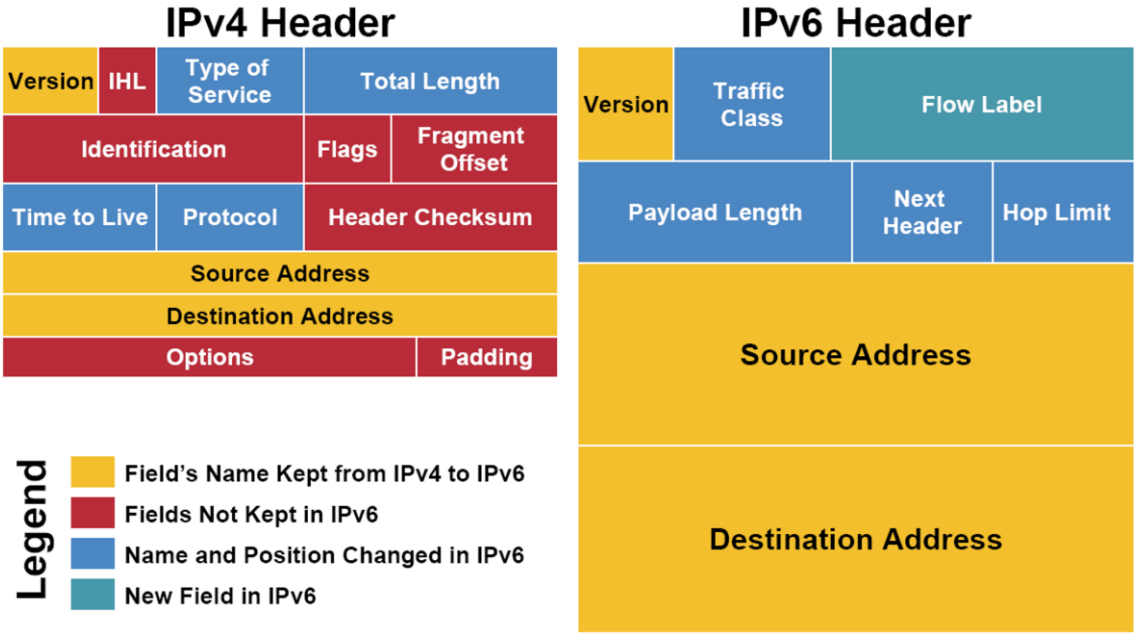
Об`єкт дослідження – Мережеві протоколи передачі даних, зокрема IPv6.

Предмет дослідження – Методи створення прихованих каналів у заголовках IPv6.

Актуальність обраної теми

Протокол IPv6 активно впроваджується в сучасних мережах, однак залишається недостатньо вивченим з погляду інформаційної безпеки. Завдяки наявності великої кількості службових полів і розширювальних заголовків, IPv6 створює сприятливі умови для побудови прихованих каналів зв'язку, які важко виявити традиційними засобами захисту. Більшість систем виявлення атак, зокрема з відкритим кодом, мають обмежену підтримку IPv6, що дозволяє зловмисникам передавати інформацію приховано. Багато адміністраторів зосереджені на безпеці IPv4 і не приділяють достатньої уваги аналізу трафіку нового покоління. Це створює додаткові вектори атак і підвищує ризики витоку даних або обходу захисту. Саме тому тема дослідження прихованих каналів у протоколі IPv6 є надзвичайно актуальною і потребує детального аналізу та розробки ефективних методів виявлення.

Порівняння службових полів IPv4/IPv6

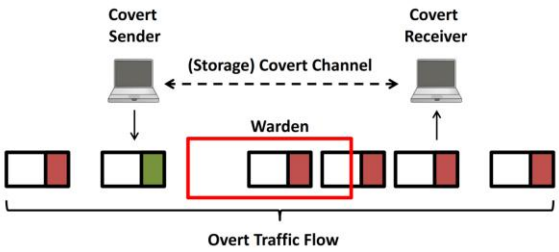


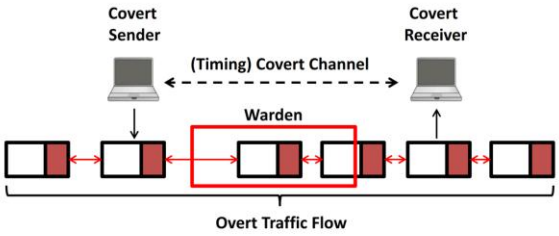
Прихований канал

Прихований канал — це метод прихованого передавання інформації, який порушує політику безпеки системи, використовуючи нестандартні засоби передачі, зокрема службові або резервні поля протоколів.

Storage Covert Channel: приховані біти в полях пакетів (наприклад, Hop Limit, Flow Label)

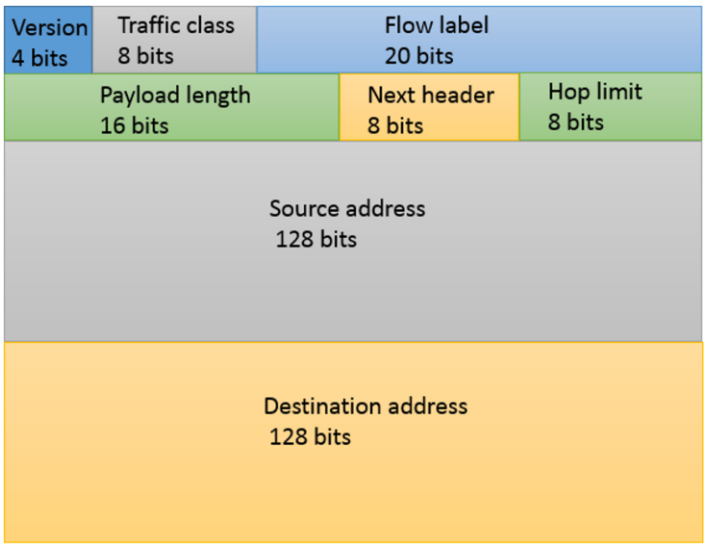
Timing Covert Channel: прихована інформація у вигляді затримок між пакетами





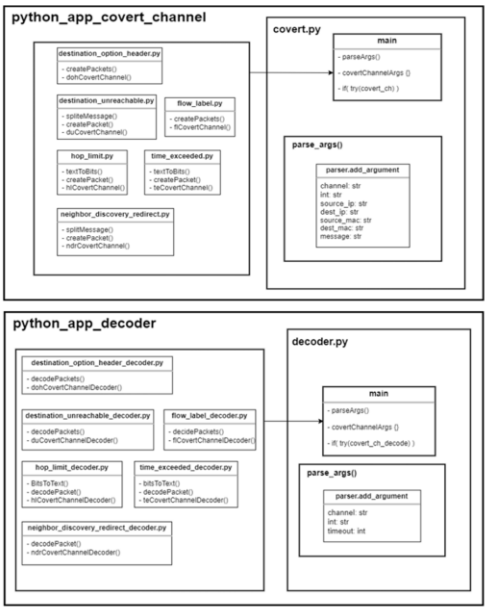
Поля IPv6

1. *Hop Limit.* Змінюється по ходу маршруту
2. *Flow Label.* Може містити цілі бінарні послідовності
3. *Destination Options Header.* Підтримує довільні опції
4. *ICMPv6.* Часто проходять мережеві фільтри.



Архітектура експериментального стенду

1. Відправник (Kali Linux + Scapy). Маршрутизатор (Cisco CSR1000v) Приймач / Аналізатор (Kali Linux + Wireshark, Suricata);
2. *файл covert.py* виконує функцію формування та надсилання IPv6-пакетів із вбудованими прихованими повідомленнями;
3. *файл decoder.py* відповідає за приймання пакетів та вилучення з них прихованої інформації.



Реалізація Hop Limit каналу

1. вхідні дані кодуються у вигляді бітів (наприклад, "101101");
2. для кожного біта формується IPv6-пакет
3. Пакети надсилаються через мережу на приймача

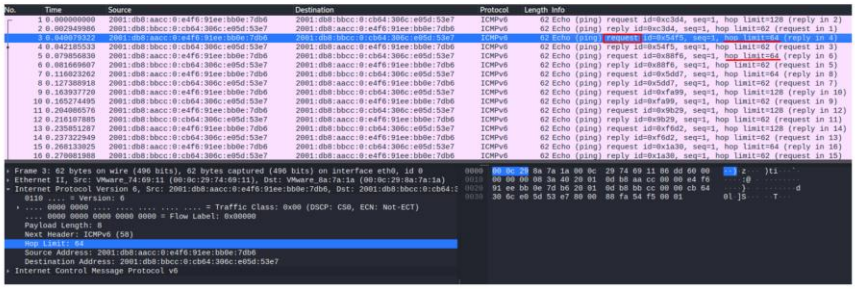
```
from scapy.all import IPv6, send
import time


destination = "2001:db8::1"


binary_data = "101101"

print("[*] Передача даних через Hop Limit:")

for bit in binary_data:
    hop_value = 65 if bit == "1" else 64
    pkt = IPv6(dst=destination, hlim=hop_value)
    send(pkt)
    print(f"[+] Відправлено біт {bit} (Hop Limit = {hop_value})")
    time.sleep(0.5)
```



Реалізація Flow Label / DOH

```
from scapy.all import IPv6, send
import time

destination = "2001:db8::1"

binary_data = "101010"

print("[*] Передача даних через поле Flow Label:")

for i in range(len(binary_data)):

    flow_bits = int(binary_data[i]) << (19 - i)

    pkt = IPv6(dst=destination, fl=flow_bits) / "Hidden data"
    send(pkt)

    print(f"[+] Відправлено біт {binary_data[i]} (Flow Label = {flow_bits})")
    time.sleep(0.5)
```

Рисунок 3.10 – Скрипт flow_label.py з кодуванням бітів у Flow Label

```
from scapy.all import *

doh_option = HBHOptUnknown(
    otype=0b00000000,      # Користувацький тип опції (можна змінити)
    optdata=b"secret123"  # Закодоване повідомлення
)

pkt = IPv6(dst="2001:db8::1") / IPv6ExtHdrDestOpt(options=[doh_option]) / ICMPv6EchoRequest()

send(pkt)
```

Рисунок 3.11 – Скрипт створення прихованого каналу через Destination Options Header (DOH)

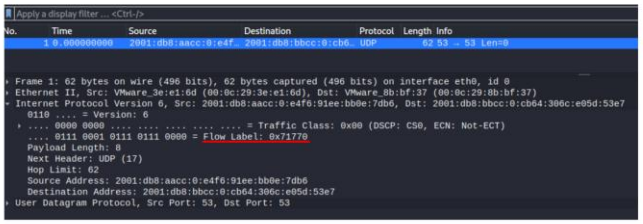


Рисунок 3.15 – Використання Flow Label для кодування бітів

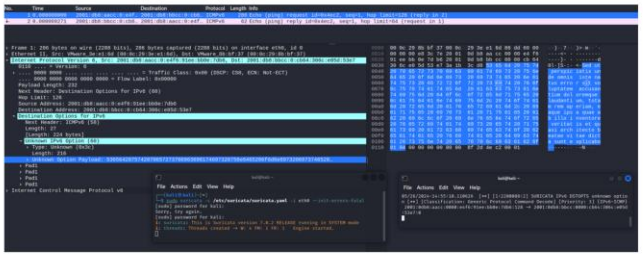
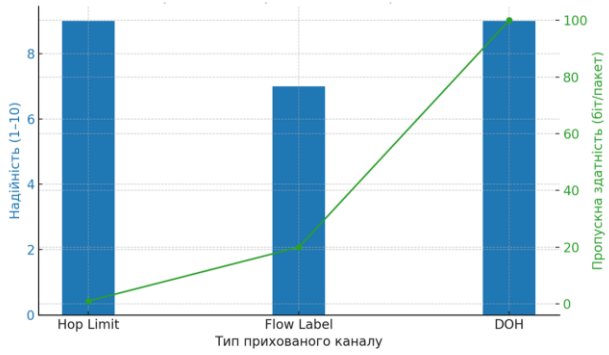


Рисунок 3.16 – Передача повідомлення через Destination Options Header (DOH)

Порівняльний аналіз каналів

Поле	Стійкість до виявлення	Пропускна здатність	Складність реалізації
Hop Limit	Висока	Низька	Низька
Flow Label	Середня	Середня	Середня
DOH	Низька	Висока	Висока



Етапи розробки та впровадження проєкту:

1. **Аналітичний етап.** Вивчення архітектури протоколу IPv6. Аналіз методів створення прихованих каналів у мережевих протоколах. Огляд існуючих досліджень та інструментів виявлення прихованих каналів;

2. **проєктування.** Визначення полів IPv6, які можуть бути використані для прихованих каналів (Flow Label, Hop Limit, DOH, ICMPv6). Розробка методів кодування/декодування інформації в цих полях. Вибір програмних інструментів: Scapy, Wireshark, Suricata, Kali Linux;

3. **реалізація.** Створення та відправлення IPv6-пакетів з прихованими даними. Використання Scapy для інжекції спеціальних значень у заголовках. Аналіз трафіку за допомогою Wireshark. Спостереження за реакцією систем виявлення атак (Suricata);

4. **тестування та аналіз результатів.** Перевірка ефективності передачі даних прихованим каналом. Вимірювання швидкості, помітності та надійності. Виявлення слабких місць у стандартних засобах захисту. Порівняльний аналіз різних методів;

ВИСНОВКИ

Дослідивши приховані канали у протоколі IPv6, яке охопило як теоретичні, так і практичні аспекти їх функціонування, виявлення та протидії можна дійти висновку, що IPv6, як сучасний мережевий протокол, має значні переваги над своїм попередником, зокрема розширений адресний простір, підтримку розширюваних заголовків, удосконалене управління маршрутизацією та автоконфігурацією.

У межах цієї роботи було виконано аналіз заголовків IPv6 та ICMPv6, виявлено поля, що можуть бути використані як носії прихованої інформації — Hop Limit, Flow Label, Destination Options Header створено приклади прихованої передачі інформації з використанням мови Python та бібліотеки Scapy. Розгорнуто тестове середовище на базі Kali Linux та Cisco CSR1000v з подальшим аналізом трафіку через Wireshark і Suricata IDS. Оцінено здатність IDS-систем виявляти нетипові ознаки прихованих каналів у IPv6, в тому числі з використанням власних сигнатур.

ПРОДОВЖЕННЯ ВИСНОВКІВ

Отримані результати дозволили зробити висновки. Поле Hop Limit забезпечує високий рівень прихованості, оскільки рідко сприймається як аномальне, що ускладнює його виявлення. Поле Flow Label має обмежену пропускну здатність, але також може бути використане для прихованого каналу з мінімальними ризиками. Заголовок DOH (Destination Options Header) забезпечує найвищу пропускну здатність, проте й найвищий ризик детектування при активному моніторингу.

Практичне значення роботи полягає у демонстрації того, що навіть стандартні поля IPv6 можуть бути використані для прихованої комунікації. Це створює нові виклики для систем виявлення загроз (IDS/IPS) та потребує оновлення політик безпеки в організаціях.