Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інфокомунікацій
(повна назва)

Кафедра Інформаційно-мережної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіті другий (магістерський)

Дослідження можливостей побудови секретного каналу на базі ICMP-пакетів

(тема)

Виконав:
здобувач 2 року навчання,
групи ІМІм-23-1
Гречинський Д.М.
(прізвище, ініціали)

Спеціальність 172 Електронні комунікації та радіотехніка
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Інформаційно-мережна інженерія
(повна назва освітньої програми)

Керівник: доц. каф. ІМІ Іваненко С.А.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Безрук В.М.
(підпис) (прізвище, ініціали)

2025 р.

Не містить відомостей, заборонених до відкритого публікування

Студент _____ / Гречинський Д.М. /
(підпис) (прізвище та ініціали)

Керівник _____ / Іваненко С.А. /
(підпис) (прізвище та ініціали)

Харківський національний університет радіоелектроніки

Факультет Інфокомунікацій

Кафедра Інформаційно-мережної інженерії

Рівень вищої освіти другий (магістерський)

Спеціальність 172 Електронні комунікації та радіотехніка

(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформаційно-мережна інженерія

(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
« 28 » жовтня 2024 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Гречинському Даниїлу Миколайовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження можливостей побудови секретного каналу на базі ICMP-пакетів

затверджена наказом по університету від« 28» жовтня 2024 р. № 1148 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 21 січня 2025 р.

3. Вхідні дані до роботи Дослідити форми стеганографії та напрямки побудови стеганографічних методів. Розглянути принципи прихованого обміну даними та ключові вимоги до цифрових об'єктів-носіїв прихованої інформації. Проаналізувати існуючі методи мережевої стеганографії. Виконати дослідження методів, що базуються на використанні ICMP-пакетів для створення прихованого каналу.

4. Перелік питань, що потрібно опрацювати у роботі Вступ
1. Принципи наявного передавання інформації у телекомунікаційних мережах
2. Огляд методів мережевої стеганографії
3. Дослідження базових характеристик icmp-протоколу
4. Дослідження можливості реалізації прихованого каналу на базі пакетів ICMP
Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) слайди у форматі Power Point (назва та мета роботи, принципи побудови неявних каналів обміну даними, методи мережевої стеганографії, ICMP-стеганографія, висновки)

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Вступ	30.10.2024	виконано
2	Виконання розділу 1	05.11.2024	виконано
3	Виконання розділу 2	20.11.2024	виконано
4	Виконання розділу 3	07.12.2024	виконано
5	Виконання розділу 4	25.12.2024	виконано
6	Висновки	06.01.2025	виконано
7	Оформлення пояснювальної записки	07.01.2025	виконано

Дата видачі завдання 28 жовтня 2024 р.

здобувачеві _____
(підпис)

Керівник роботи _____ доц. Іваненко С.А.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 74 с., 29 рис., 23 джерела, 2 додатки.

Об'єкт дослідження – методи обудови прихованих каналів обміну даними на базі мережевих протоколів.

Мета роботи – дослідити можливості стеганографічних методів, які базуються на використанні поширених мережевих протоколів.

Розглядаються засади побудови методів приховування даних під час передавання та зберігання у сховищах. Виконується аналіз поширених стеганографічних методів на базі мережевих протоколів. Досліджуються алгоритми, які використовують як механізми вбудовування даних у службові та інформаційні поля пакетів поширених типів, так і зміну умов та порядку надсилання пакетів. Доводиться можливість побудови стеганографічного каналу за рахунок внесення змін у поля пакетів ICMP.

HICCUPS, LACK, ICMP, TRANSTEG, ECHO-ПАКЕТ, МЕРЕЖЕВА СТЕГАНОГРАФІЯ

THE ABSTRACT

Explanatory note: 72 p., 29 fig., 23 sources, 2 app.

The object of research is methods of setting up hidden data exchange channels based on network protocols.

The purpose of the work is to investigate the possibilities of steganograin methods, which are based on the use of common network protocols.

The fundamentals of building methods for hiding data during transmission and storage in storage are considered. An analysis of common steganographic methods based on network protocols is performed. Algorithms that use both mechanisms for embedding data into the service and information fields of packets of common types, as well as changing the conditions and order of sending packets are being studied. The possibility of building a steganographic channel by making changes to the fields of ISMR packets is proved.

HICCUPS, LACK, ICMP, TRANSTEG, ECHO-PACKET, NETWORK STEGANOGRAPHY

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

ICMP – (Internet Control Message Protocol) – мережевий протокол стеку TCP/IP;

TCP – (Transmission Control Protocol) – протокол управління передачею;

UDP – (User Datagram Protocol) – протокол користувацьких дейтаграм;

IP – (Internet Protocol) – протокол міжмережевої взаємодії;

DPI – (Deep Packet Inspection) – глибокий аналіз можливих вбудовувань у пакети;

LACK - (Lost Audio Packets Steganography) – стеганографічний метод на базі аудіо пакетів;

RSTEG - (Retransmission Steganography) – стаганографічний метод на базі ретрансляції пакетів;

HICCUPS – (HIdden Communication system for CorrUPted networkS) – мережевий стеганографічний метод;

CSMA - (Carrier Sense Multiple Access) – метод доступу до мережевого середовища;

CSMA/CD - (CSMA with Collision Detection) - метод доступу до мережевого середовища;

CSMA/CA - (CSMA with Collision Avoidance) - метод доступу до мережевого середовища;

RTCP – (Real-Time Transport Control Protocol) – протокол управління передачею у реальному часі;

RTP - (Real-Time Transport Control Protocol) - протокол передачі у реальному часі;

TTL – (Time to Live) – час життя пакету;

RTT – (Round Trip Time) – час двостороннього проходження пакетів.

ВСТУП

Питання безпеки інформаційних систем є одним з найбільш гострих у сучасному сьогоденні. Це однаково важливо як для процесів зберігання, так і для процесів обробки та передавання даних – власне, протягом усього життєвого циклу існування будь-яких даних.

Традиційно для захисту даних у файлосховищах та під час передавання каналами телекомунікаційних мереж застосовуються підходи, що фокусуються на [1]:

– забезпеченні неможливості зловмисного проникнення до середовища, де локалізовано ті чи інші дані;

– мінімізації ймовірності захоплення даних у ході передавання розподіленим середовищем;

– мінімізації ймовірності можливості використання даних зловмисником навіть за умови їх попередньої успішної крадіжки.

У першому випадку передбачається, що інформаційна система має захист від зовнішніх та внутрішніх впливів, яких базується на використанні міжмережевих екранів, систем попередження вторгнень а також передбачається наявність регламенту безпеки мережі тощо.

Реалізація другого підходу передбачає застосування механізмів аутентифікації та ідентифікації користувачів, створення захищених тунелів у середині мережі і т.д.

Нарешті, у рамках третього підходу реалізується захист даних на рівні джерела шляхом виконання відносно них тих чи інших криптографічних перетворень.

Разом з тим, для гіпотетичного зловмисника, який має у своєму розпорядженні достатній обсяг обчислювальних ресурсів та має певний специфічний досвід та знання, перелічених вище стандартних підходів до захисту даних може бути недостатньо, так як зловмиснику є відомим сам факт існування та передачі тієї чи іншої інформації.

Тобто, важливим є організувати обмін даними таким чином, щоб сам факт виконання даних операцій був неочевидним [2].

У свою чергу, неочевидний обмін даними може бути організовано у тому разі, коли сама інформація не передається безпосередньо, а спершу вбудовується

до носія, яким може бути файл чи його частина, або пакет даних, який у результаті вбудовування не втрачає своєї функціональності. За рахунок цього гарантується секретність та неочевидність передавання інформації. При цьому, зловмисник, навіть попри потенційні можливості, не матиме змоги відстежити факт передачі/прийому.

Зазначене вище свідчить про актуальність кваліфікаційної роботи.

1 ПРИНЦИПИ НАЯВНОГО ПЕРЕДАВАННЯ ІНФОРМАЦІЇ У ТЕЛЕКОМУНІКАЦІЙНИХ МЕРЕЖАХ

1.1 Поняття стеганографічної передачі даних

Сам термін «стеганографія» походить від грецьких слів «стеганос» – прихований, та «графо» – пишу. Тобто, стеганографія являє собою окрему наукову галузь, яка вирішує задачі забезпечення прихованого зберігання та передавання інформації.

При цьому, якщо криптографічні перетворення відносно даних, які захищаються від зловмисника, переводять їх до форми, у якій без наявності секретного ключа/ключів даними неможливо скористатися, стеганографія приховує саму наявність даних, які можуть бути об’єктом інтересу зловмисника [2].

На теперішній час, у рамках загальної галузі стеганографії, може бути виокремлено такі її окремі напрямки, як (рис.1.1):

– кластична стеганографія;

– комп’ютерна стеганографія;

– стеганографія цифрового типу;

– мережева стеганографія.

Рисунок 1.1 – Базові напрямки стеганографії

1.2 Напрямки стеганографії

1.2.1 Класична стеганографія

Даний напрямок стеганографії включає у себе прийоми та механізми маніпуляції відносно фізичних об’єктів, які за певних умов можуть проявлювати дані, які було у них тим чи іншим чином.

Типовим прикладом стеганографії класичного типу є невидимі чорнила, що у результаті певних впливів (нагрівання, хімічне проявлення тощо) стають візуально помітними.

1.2.2 Комп’ютерна стеганографія

У сутності, цей напрямок можна вважати продовженням класичної стеганографії, яка, при цьому, використовує особливості сучасних комп’ютерних платформ.

1.2.3 Цифрова стеганографія

В основі цифрової стаганографії також знаходяться постулати класичного напрямку, проте, головна відмінність полягає у способах їх застосування [2, 3].

Тут передбачається, що приховування даних реалізується шляхом їх розміщення у ті чи інші цифрові об’єкти.

Частіше за все об’єктами, що слугують для приховування інформації, є мультимедійні файли, такі, як – аудіо-файли, відео та фото, текстури 3D-об’єктів.

У цілому, може бути використано будь-які цифрові об’єкти, після внесення даних до яких може привести лише до незначного їх викривлення. При цьому, такі викривлення мають біти нижче умовного порогу сприйняття людини.

Фактично, методи даного напрямку вирішують завдання пошуку балансу між рівнем заповненості носія та захищеності прихованих даних, які знаходяться у зворотній залежності.

Інакше кажучи, приховане на базі методів даного напрямку повідомлення матиме високий рівень захищеності за умови незначного заповнення носія і навпаки.

Методи даної групи потенційно здатні забезпечити:

– найвищу пропускну здатність (не нижчу, ніж одиниці Мбіт/с) секретного каналу;

– найвищу захищеності приховуваних даних навіть в умовах дії аналітичних механізмів (з урахуванням того, що алгоритм реалізовано коректно, а рівень заповнення окремого цифрового носія знаходиться у межах 1-10% від доступного обсягу).

1.2.4 Мережева стеганографія

Підходи, які застосовуються у рамках методів даного напрямку, передбачають виконання змін однієї чи кількох властивостей одного або кількох мережевих протоколів відповідно до закону зміни самої інформації, що приховується [1, 2].

Окрім цього, також може бути використано взаємозв'язок між двома, або більшою кількістю різних протоколів для того, щоб забезпечити значно суттєвіше приховування даних, які передаються.

У цілому, методи даного напрямку використовують різні механізми приховування даних – як прямі, так і опосередковані, які так чи інакше зводяться до маніпуляції змістом пакету або закономірністю надсилання пакетів.

Методам даної групи властива відносно легка реалізація.

На сьогодні це досягається за рахунок використання програмних засобів управління мережевими адаптерами, або спеціалізованих програмних модулів, що дозволяють напряму взаємодіяти зі змістом пакетів.

Мережева стеганографія забезпечує можливість побудови секретних каналів, які у загальному випадку характеризуються:

– невеликим або середнім рівнем пропускної здатності (одиниці біт-одиниці Мбіт/с), що визначається використовуваними механізмами приховування;

– відносно високим рівнем захищеності даних.

1.3 Принципи побудови сучасних стеганографічних алгоритмів

Основу практично будь-яких стеганографічних алгоритмів, які може бути реалізовано у цифровому середовищі, становить механізм внесення повідомлення у той-чи інший цифровий об'єкт. У ролі такого об'єкту частіше за все може виступати [3]:

– файл;

– частина файла (пакет даних);

– цифровий пристрій зберігання (його зарезервована область, або така область, яка не використовується у штатному режимі функціонування).

Зважаючи на це, сутність процесу стеганографічного захисту може бути зведено до наступного принципу:

$$f' = S(f; m; Q), \tag{1.1}$$

де f' - цифровий об’єкт, утворений за результатами внесення повідомлення;

Q - множина опцій та параметрів механізму внесення повідомлення до цифрового об’єкту (стеганографічний ключ).

Графічно даний принцип може бути зображено за допомогою схеми, яку наводиться на рис. 1.2.

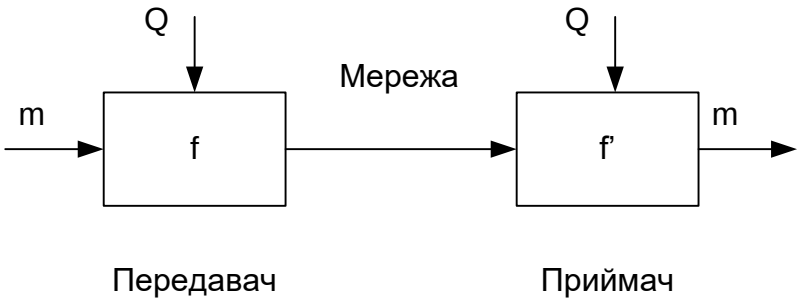

Рисунок 1.2 – Принцип стеганографічного внесення та виокремлення повідомлення з цифрового носія

Захищеність даних тут забезпечується за рахунок того, що [2]:

– зловмиснику невідомо, який саме алгоритм внесення даних до цифрового об’єкту буде використано передавачем та приймачем;

– навіть якщо зловмисник матиме відомості про використовуваний алгоритм, йому початково невідомо, який саме набір параметрів (ключ) буде використано;

– дані, приховані у межах того чи іншого цифрового об’єкту, може бути додатково піддано криптографічному перетворенню;

– зловмиснику невідомо, який саме типу цифрових об’єктів буде використовуватися для розміщення даних.

Тобто, за цих умов можливі дії зловмисника з пошуку ймовірної прихованої передачі являють собою:

– сніфінг усього можливого об’єму трафіку від цільового вузла (мається на увазі типи даних, прийнятні для розміщення усередині них прихованого контенту);

– аналіз усього захопленого трафіку на предмет виявлення ознак існування прихованих даних;

– аналіз цифрових об’єктів, які мають формальні ознаки існування прихованих даних для їх реконструкції.

Зрозуміло, що навіть в умовах, коли зловмисник має у розпорядженні значні обчислювальні потужності, пошук прихованих даних у зазначений спосіб потребує суттєвих часових ресурсів та є недоцільним – за час пошуку інформація може стати вже неактуальною.

Це зумовлено тим, що [1, 2]:

– сьогодні існує велика кількість стеганографічних алгоритмів, тому не маючи уявлення щодо того, який саме з них використовується, зловмиснику доведеться перевіряти кожен з них;

– наявність формальних ознак не гарантує існування приховуваних даних і нерідко може бути лише природньою специфічною особливістю цифрового носія;

– файли, які прийнятні для використання у ролі цифрових об’єктів для розміщення приховуваних даних, у мережі за різними оцінками складають від 82 до 90% трафіку [4].

З виразу (1.1) бачимо, що у процесі приховування повідомлення певну роль відіграє цифровий об’єкт, у який вбудовуються дані.

Визначимо, яким саме чином його властивості можуть впливати на ефективність захисту інформації.

1.4 Вимоги до цифрових об’єктів-носіїв прихованої інформації

Для ефективної реалізації процедури приховування даних з використанням того чи іншого стеганографічного алгоритму мають виконуватися наступні вимоги [2, 3]:

1. Сам механізм , що є основою алгоритму, та множина його параметрів мають бути нетривіальними.

2. Цифровий об’єкт-носій має відповідати ряду умов, серед яких:

– високий рівень надлишковості опису, тобто:

$$f(W_{max}) - f(W_{min}) \to max, \tag{1.2}$$

ця умова, по-перше, вказує на можливість використання тоги чи іншого цифрового об’єкту для розміщення приховуваної інформації m а по друге – дозволяє розглянути умову щодо відмінності об’єктів f та f';

– незначна відмінність Δf між об’єктами f та f', тобто:

$$\Delta f = f - f' = min, \tag{1.3}$$

виконання умови (1.3), у свою чергу, мінімізує ймовірність виникнення характерних ознак присутності приховуваних даних у межах цифрового об’єкту у явному вигляді;

– збереження функціональності цифрового об’єкту після виконання процедури внесення даних, що приховуються, а саме:

$$func(f) = func(f') = const. \tag{1.4}$$

Беручи до уваги перелічені вимоги щодо цифрового об’єкту-носія, можемо зробити висновок, що тією чи іншою мірою їм відповідають текстові, відео , графічні та аудіо дані.

У свою чергу, використання тих чи інших типів цифрових об’єктів, до яких реалізується внесення приховуваних даних визначає форму стеганографії.

1.5 Форми стеганографії

Найпростішою формою стеганографії у цифровому середовищі є текст. У цьому разі, закон зміни повідомлення у межах тексту може бути реалізовано за рахунок [2, 3]:

– зсуву певних слів/рядків;

– використання додаткових символів пробілу;

– зміна порядку розміщення візуально непомітних символів закінчення рядку;

– маніпулювання кількістю та положенням у тексті тих чи інших символів (деяких конкретних символів, голосних або приголосних літер, цифр тощо);

– заміни символів на візуально схожі і т.д.

Даній формі стеганографії властиве обмеження – текстові файли, які містять форматовані дані, та використовуються автоматичними обробниками, не може бути використано для розміщення приховуваних даних, так як у цьому разі умова (1.4) не буде виконуватися.

На відміну від тексту, зображення характеризуються суттєво вищим рівнем надлишковості представлення, що, як видно з умови (1.2), дозволяє виконати розміщення значної кількості даних, які необхідно приховати для передавання.

При цьому, якщо вважати що передавач та приймач використовують глобально оригінальні зображення, це автоматично гарантує виконання умови (1.4).

З зазначених причин зображення набули широкого застосування у стеганографії.

Також з причини відповідності вимогам (1.2)-(1.4), ефективною для захисту даних може вважатися аудіо стеганографія. Методи стеганографії, що належать даній формі, передбачають внесення цифрового представлення повідомлення до wav, acc, ogg чи mp3-файлу [5].

Дана форма стеганографії також може вважатися ефективною, так як типовий аудіофайл 16-бітного формату має 216 рівнів звуку, відтак – людський слух не здатен виявити відмінностей у кілька рівнів.

Разом з тим, найбільше потенційних можливостей надає стаганографія на базі відеоданих. У першу чергу, це стосується можливостей розміщення відносно великої кількості даних у межах як єдиного об’єкта (наприклад – у конкртних кадрах окремого потоку), так і у межах окремих частин такого об’єкта (в окремому кадрі).

Окрім цього, більшість відеоданих сьогодні мають супровідний аудіоряд, що надає додаткові можливості з вбудовування даних.

1.6 Обґрунтування напрямків досліджень

На сьогодні з усіх розглянутих вище напрямків стеганографії найширше застосовуються методи цифрової та мережевої стеганографії. Це зумовлено рядом факторів, таких, як:

– неможливість оперативного обміну даними на базі методів класичної стеганографії та низька місткість носіїв даних для цього випадку;

– обмеженість застосовуваності методів комп'ютерної стеганографії (для обміну даними так чи інакше буде використано методи цифрової та мережевої стеганографії);

– відносно високі рівні захищеності даних за умови використання методів мережевої та цифрової стеганографії.

При цьому, якщо у рамках методів цифрової стеганографії підходи до приховування даних базуються на використанні зазвичай деякий єдиного алгоритму (у межах певного методу), то мережева стеганографія дозволяє використовувати більше одного підходу. Більш того – передбачається можливість одночасного спільного використання різних підходів до приховування даних.

Окрім цього, реалізація методів цифрової стеганографії хоча і дозволяє забезпечити високий рівень захищеності даних та, за певних умов, високу пропускну здатність, є суттєво складнішим завданням порівняно з випадками реалізації методів мережевої стеганографії

Виходячи з цього, пропонується виконати дослідження саме методів мережевої стеганографії а отже - необхідно дослідити можливість використання мережевих протоколів, як транспорту для прямого та/або опосередкованого переносу повідомлень, які необхідно приховати у ході передавання.

Передньо можна зазначити, що прихований канал теоретично може бути побудовано на базі як протоколів, що забезпечують передавання користувацьких даних (TCP, UDP, IP, Ethernet-кадри тощо), так і сервісні протоколи [6], зокрема – ICMP і т.д.

Разом з тим слід пам'ятати, що саме корисне навантаження пакетів, які переносять користувацькі дані, може стати першочерговим об'єктом аналізу як систем, орієнтованих на виявлення стеговключень, так і DPI-систем у цілому.

Виходячи з цього, пропонується дослідити можливість використання методів мережевої стеганографії за реальних умов передавання даних – тобто, ураховуючи ймовірність існування у мережі аналітичних механізмів та існування ймовірності некоректного функціонування прихованого каналу у зв’язку з існуючим регламентом функціонування мережі.

2 ОГЛЯД МЕТОДІВ МЕРЕЖЕВОЇ СТЕГАНОГРАФІЇ

2.1 Класифікація методів мережевої стеганографії

У рамках реалізації методів мережевої стеганографії, у якості носіїв секретних даних майже завжди застосовуються мережеві протоколи еталонної моделі OSI.

Наближено можна говорити, що мережева стеганографія - це сімейство методів модифікації даних у заголовках мережевих протоколів, а також у полях корисного навантаження пакетів, зміни порядку та структури передачі пакетів у тому чи іншому мережевому протоколі (або ряду протоколів одночасно).

Так чи інакше, загальною рисою існуючих сьогодні поширених методів мережевої стеганографії є формування на їх базі неявних каналів передачі теоретично у якому завгодно відкритому каналі, що характеризується деякою надмірністю.

Методи мережевої стеганографії може бути розділено на 3 групи (рис. 2.1), а саме [7]:

1. Методи, що виконують модифікацію пакетів. Сюди належать:

– методи, які вносять зміни у зміст полів заголовків мережевих протоколів;

– методи, які модифікують інформаційну частину (корисне навантаження) пакетів - це т.з. алгоритми «водяних знаків», мовних кодеків та інших схожих стеганографічних технік;

– методи, які поєднують у собі дві попередні категорії.

2. Стеганографічні методи, які виконують зміну структури а також ті чи інші параметри передавання пакетів, а саме:

– методи, що передбачають зміну порядку надсилання пакетів;

– методи, які маніпулюють значенням інтервалу між пакетами;

– методи, які вносять спрямовані втрати пакетів за рахунок пропуску порядкових номерів у відправника.

3. Змішані (гібридні) методи стеганографії, які можуть змінювати і вміст пакетів, і час їхньої доставки, і порядок їх надходження. У рамках методів третьої групи широко використовуються два підходи:

– навмисна затримки аудіо пакетів, прикладом якого є метод LACK (Lost Audio Packets Steganography);

– ретрансляція пакетів, або RSTEG (Retransmission Steganography) [1].

Окремо також слід виділити методи, які залежно від умов надсилання приховуваних даних можуть бути реалізовані або шляхом зміни полів пакетів, або за рахунок зміни параметрів передавання послідовностей пакетів. До методів даної групи відносяться т.з. методи ICMP-стеганографії.

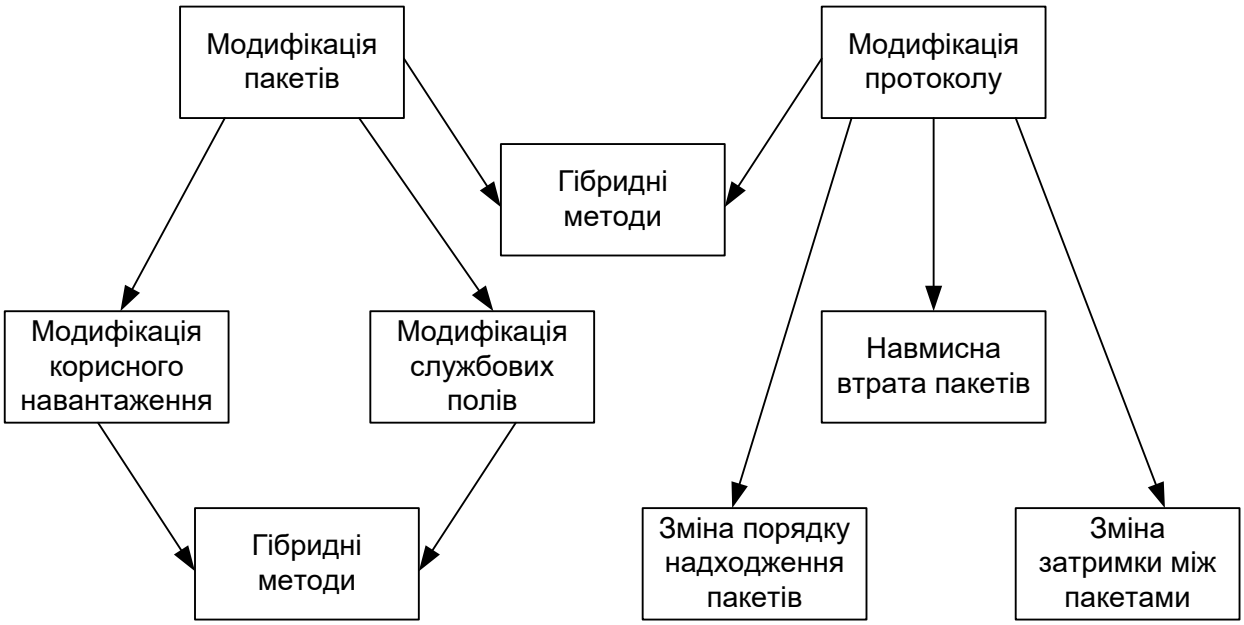

Рисунок 2.1 – Класифікація методів мережевої стеганографії

2.2 Порівняння методів мережевої стеганографії

На рисунку 2.2 приведено порівняльні характеристики різних методів мережевої стеганографії.

Дані характеристики було отримано у наслідок проведення ряжу досліджень [8, 9].

Тут ряд поширених методів стеганографії було порівняно за:

– пропускною здатністю методу;

– складністю виявлення прихованого повідомлення;

– собівартістю реалізації методу;

– складністю методу.

У свою чергу, ранжування методів виконується у відносних одиницях, тобто, без відображення кількісних характеристик. Чим більшим на діаграмі є той чи інший параметр відповідного методу, тим вищою є його характеристика.

Як видно з рисунку 2.2, найвищі показники пропускної здатності забезпечує метод TranSteg, тоді як найвищу захищеність даних (складність виявлення) має метод HICCUPS.

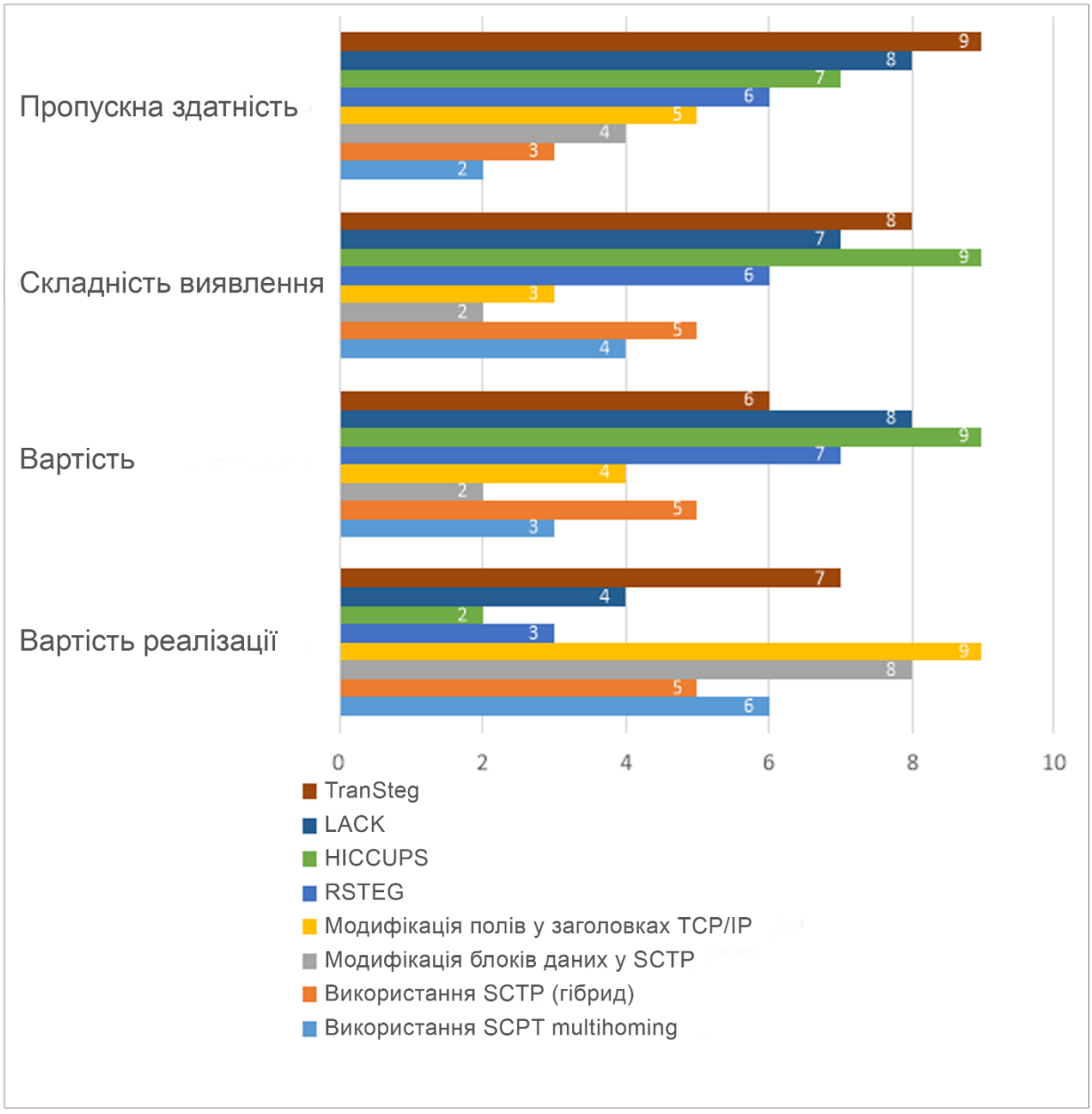

Рисунок 2.2 – Порівняння поширених методів мережевої стеганографії

У свою чергу, метод HICCUPS також відзначається найнижчою складністю реалізації, а за показником пропускної здатності знаходиться на третьому місці у рейтингу, що дає змогу розглядати його як потенційно найбільш ефективного.

Також слід розуміти те, що усі показники мають між собою взаємозв'язки. Зокрема, такий показник, як, складність виявлення, прямим чином залежить від якості реалізації методу, тобто — від складності реалізації.

2.3 Поширені методи мережевої стеганографії

2.3.1 HICCUPS

Метод HICCUPS базується на використанні недосконалостей середовища передачі, а саме – наявність у каналі шумів та перешкод, що являють собою природні фактори спотворення даних [10].

Застосовується метод у мережах з розділюваним середовищем передачі даних, базова технологія яких передбачає наявність тих чи інших механізмів доступу до середовища. Це, наприклад, може бути CSMA (Carrier Sense Multiple Access), CSMA/CD (CSMA with Collision Detection) або CSMA/CA (CSMA with Collision Avoidance) чи Token Bus.

При цьому, усі перелічені механізми доступу мають загальну властивість – можливість прослуховувати та «чути» усі кадри з даними, які передаються середовищем.

Обов'язковою умовою для успішного перехоплення кадрів мережевим адаптером терміналу є доступ до фізичного середовища.

Для випадку дротової мережі — за допомогою кабельних з'єднань, для бездротових — шляхом взаємодії з радіопередаючим та приймальний обладнанням на відповідній відстані та частоті.

У свою чергу, «прослуховування» усього масиву кадрів даних у загальному розподіленому середовищі, а також можливість надсилання пошкоджених кадрів, що містять неправильні значеннями кодів корекції є найважливішими мережевими функціями HICCUPS.

Наприклад, мережі бездротового типу можуть використовувати радіопередачу зі змінною частотою бітових помилок (BER). Це дає змогу реалізації ін'єкцій «штучних» пошкоджених кадрів.

Іншими словами, можна сказати, що новизна HICCUPS полягає у:

– використанні середовища захищеної інфокомунікаційної мережі, де передбачено наявність криптографічних механізмів, для побудови стеганографічної системи;

– використанні алгоритму розподілу пропускної здатності для реалізації стеганографічного механізму, заснованого на пошкоджених кадрах.

Система HICCUPS є ефективною у середовищах, що мають такі властивості, як [10]:

– у розподіленому середовищі є можливість перехоплення кадрів;

– використовується один з відомих методів ініціалізації алгоритму шифрування, це можуть бути, наприклад вектори ініціалізації;

– присутні механізми забезпечення цілісності для зашифрованих кадрів (наприклад, циклічний надлишковий код CRC, одностороння хеш-функція тощо).

Попри усе зазначене, критичною то обов'язковою є лише перша вимога з зазначених.

При цьому, у мережі, що має перелічені властивості, може бути побудовано три приховані канали передавання даних у кадрі MAC, а саме:

– HDC1 або канал, побудований на базі векторів ініціалізації шифру;

– HDC2 або канал, який будується на базі MAC-адрес (наприклад, MAC-адреси призначення та джерела);

– HDC3: або канал, утворений з використанням значень механізму цілісності (зокрема, контрольної суми).

У випадку реалізації прихованих каналів на базі HICCUPS у мережах, де криптографічні механізми не використовуються, може бути застосовано лише HDC2 та HDC3.

Необхідно також зазначити що значна кількість мереж, побудованих на базі дротових технологій, не підтримують безпеку на рівні MAC, на відміну від бездротових.

Загальна схема HICCUPS, як показано на рисунку 2.3, використовує 3 режими, а саме – режим ініціалізації, базовий режим та режим пошкоджених кадрів.

Так, у ході ініціалізації системи усі робочі станції, які включено до складу групи, що виконує прихований обмін даними, спершу реалізують процедуру встановлення секретного ключа для шифрів, вбудованих у стеганографічну систему.

Дана система система не є обмеженою одноадресним режимом розсилки (1: 1 - один відправник одному одержувачу), використовується також режим

багатоадресної розсилки (1: N - один до багатьох, M: N - багато для багатьох) та режим широкомовної розсилки.

При цьому, рішення HICCUPS передбачає можливість реалізації будь-яких процедур додавання до груп (групи), ключової угоди та/або протоколу обміну ключами [2].

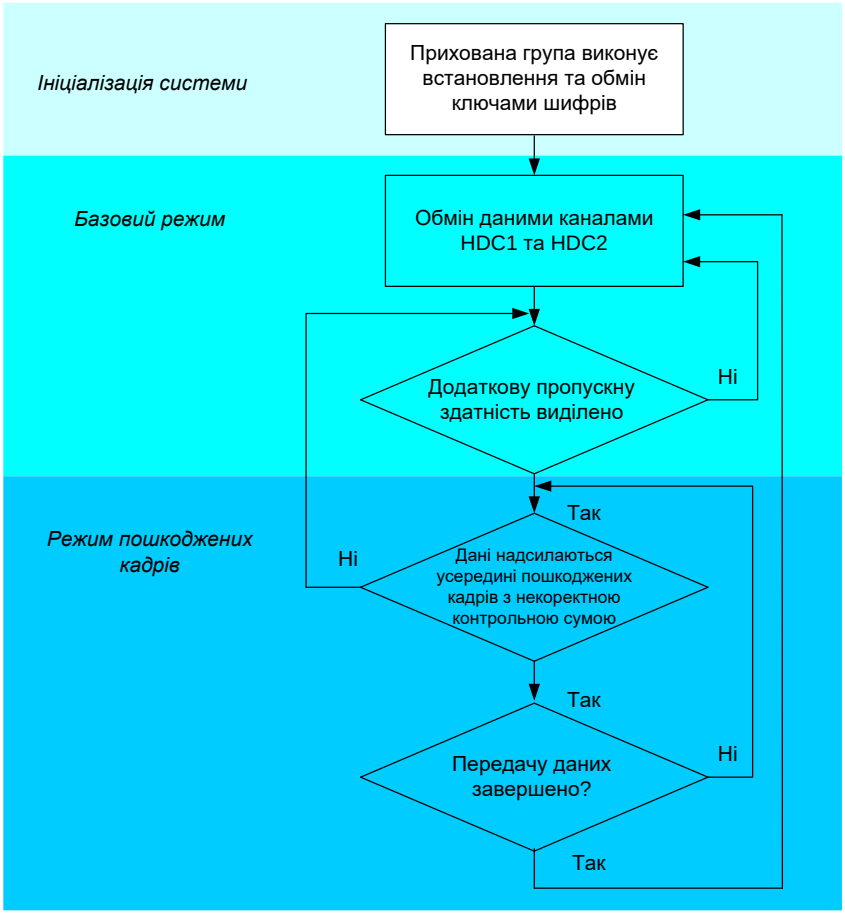

Рисунок 2.3 – Загальна схема HICCUPS

Режимом роботи HICCUPS за замовчуванням вважається обмін даними на базі векторів ініціалізації шифру (HDC1) а також MAC-адрес (HDC2). Прихованим каналам зв'язку, утвореним подібним чином, властива низька пропускна здатність, яка у середньому складає порядку 1% від доступного обсягу кадру.

Виходячи з цього, такі канали можуть використовуватися для обміну контрольними повідомленнями між секретними робочими станціями та для передавання даних з низькою бітовою швидкістю [10].

При цьому, для масиву даних, які передаються з використанням HDC1 чи HDC2, термінали, які належать до прихованої групи, переходять до режиму пошкоджених кадрів, який, у свою чергу, надає додаткову пропускну здатність.

Разом з тим, у даному режимі (HDC3) надсилання приховуваного повідомлення здійснюється усередині корисного навантаження кадрів, до яких навмисно вписуються некоректні контрольні суми. Означений режим здатен забезпечити майже 100% пропускну здатність на протязі деякого часового відрізку.

У свою чергу, термінали, що не входять до прихованої групи, відкидають кадри, які мають некоректні контрольні суми.

Мережеві адаптери, які, у свою чергу, функціонують у режимі пошкоджених кадрів, мають виконувати моніторинг усіх кадрів мережі у середовищі передачі.

Функціональними частинами системи HICCUPS є:

1. Для режиму 1 - мережеві адаптери, призначені, наприклад, для IEEE 802.11b, IEEE 802.11g і т.д.; такі пристрої повинні мати можливість здійснювати контроль HDC1-HDC3 та корисне навантаження даних у кадрі MAC.

2. Для режиму 2 - система керування для контролю HDC1-HDC3, а також даних у кадрі MAC.

Реалізація системи у режимі P2 може бути як програмною так і апаратною.

У цьому режимі система має виконувати такі функції, як:

– з’єднання з до прихованою групою;

– відключення від прихованої групи;

– функціонал забезпечення інтерфейсу для верхнього мережевого рівня, що спрямовано на надання можливості управління HDC1-HDC3 та даними у кадрі MAC.

Якщо для стеганографічної системи тим чи іншим чином передбачено додаткове використання криптографічних модулів, додатковими функціями у цьому разі є:

– угода про ключі;

– обмін ключами;

– оновлення ключа;

– шифрування та дешифрування.

Після досліджень існуючого ринку мережевих адаптерів (включно з IEEE 802.11 та IEEE 802.3), розробники HICCUPS не виявили таких, що за

замовчуванням забезпечують інтерфейс, який дозволяв би створити кадр із заданим CRC.

Виходячи з цього, повнофункціональна реалізація HICCUPS можлива або за умови перепрограмування існуючих мережевих адаптерів, або за умови апаратної реалізації власних.

Водночас, ряд достатньо поширених мережевих адаптерів стандарту IEEE 802.11 (у першу чергу, це стосується адаптерів на базі Prism II) мають вбудований режим контролю передачі, тому використовуючи їх, може бути забезпечено перехоплення усього трафіку, включаючи умовно-пошкоджені кадри.

2.3.2 Стеганографія на базі VoIP-потоку

Одним з найбільш поширених прикладів методів даного типу є метод LACK [11, 12], принцип функціонування якого ілюструє схема, яку зображено на рисунку 2.4.

Спочатку передавач виконує вибір з потоку RTP одного пакету, після чого корисне навантаження такого пакету (у даному випадку це - аудіодані) замінюється на масив біт повідомлення (1). Далі цей пакет передається у мережу з деякою затримкою (2).

При цьому, якщо пакет з перевищеною затримкою отримує приймач, який не належить прихованій групі вузлів, тобто, такий, що не підозрює про стеганографічну процедуру, даний пакет відкидається (3).

Дана процедура є наслідком того, що для «звичайних» приймачів приховані дані є «невидимими» [11].

Разом з тим у випадках, коли приймач знає про приховане повідомлення, він зчитує пакет з затримкою і тим самим отримує з нього приховані дані (4).

Так як приховані дані, які вносяться до пакетів, що далі надсилаються з затримкою, надсилаються одержувачу чи одержувачам, які попередньо поінформовані про дану процедуру, ніяких додаткових пакетів не генеруються [8, 12].

Виходячи зі специфіки методу LACK, факт передавання прихованих даних на його базі зазвичай важко виконати. Зумовлено це тим, що втрата пакетів в IP-мережах є штатним явищем», тому спрямовані втрати пакетів, які ініціюються LACK, нелегко виявити, коли їх обсяг суттєво не перевищує статистики втрат пакетів за умови відсутності прихованої передачі.

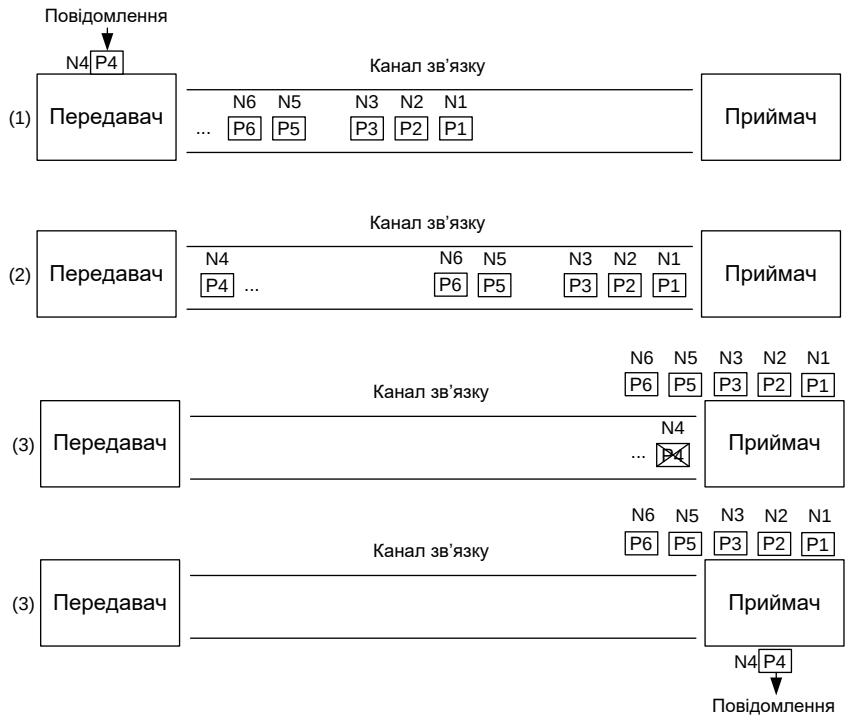

Рисунок 2.4 – Принцип функціонування методу LACK

У свою чергу, до методів стегоаналізу, які потенціно здатні здійснювати виявлення LACK-стеганографії, належать [8, 9]:

1. Статистичний аналіз втрачених пакетів для викликів у підмережі. Означений тип стегоаналізу може бути реалізовано за допомогою т.з. «пасивного наглядача», яким може бути у т.ч. будь-який мережевий вузол.

У цьому разі аналіз виконується на базі інформації, яку включено до звітів протоколу керування транспортним режимом (RTCP) у режимі реального часу. Це дані щодо:

– сукупної кількості втрачених пакетів між користувачами під час їхнього обміну;

– порядкових номерів пакетів потоку RTP.

Тобто, якщо для деяких спостережуваних викликів кількість втрачених пакетів є вищою за середню (або вище певного встановленого порогу), це може розцінюватися ознакою ймовірної LACK-передачі у мережі. Це саме стосується аномалій у порядкових номерах пакетів потоку RTP [13].

2. Статистичний аналіз на базі тривалості VoIP-дзвінків. Даний підхід може бути використано у випадках, якщо для певної підмережі відомий розподіл ймовірності тривалості викликів.

Тоді статистичний стегоаналіз може бути виконаний для виявлення аномальних джерел VoIP – тобто, таких, що не відповідають існуючому розподілу.

Аномальними у даному випадку вважаються джерела, тривалість викликів яких перевищує існуючий розподіл, оскільки виклики LACK можуть бути довшими порівняно з викликами без LACK, що є результатом введення стеганографічних даних.

3. «Активний наглядач», який виконує аналіз усіх існуючих у мережі RTP-потоків, зокрема - ідентифікатор джерела синхронізації та поля: порядковий номер та тчасову мітку з заголовку RTP) [11].

У рамках даного способу виявлення ознак LACK ідентифікуються пакети, які мають надмірну затримку а відтак - не можуть бути використані для відновлення аудіоданих.

Активний наглядач може виконувати обнулення полів даних таких пакетів, або просто відкидати їх. Отже, у цьому разі доцільно говорити не про виявлення, а про активні протидію надсиланню прихованих даних.

Разом з тим, даному підходу властива потенційна проблема, зумовлена тим, що деяку частину пакетів зі значною затримкою, які видаляються, може бути використано для відновлення розмови.

Це зумовлено тим, що розмір буфера тремтіння приймача активному наглядачеві є апріорі невідомий.

При цьому, коли активний наглядач відкидає усі пакети з затримкою вище певного рівня, він потенційно може відкинути пакети, які може бути застосовано для відновлення аудіоданих.

Тобто, якість розмови може при цьому суттєво знизитись, так як активний наглядач маніпулює пакетами як «стеганографічних» дзвінків, так і звичайних.

У свою чергу, оскільки методи ICMP-стеганографії, залежно від умов реалізації, можуть охоплюють більшість існуючих підходів до побудови прихованого каналу, їхнє дослідження доцільно виконувати більш детально. Для цього спершу виконаємо дослідження характеристик та можливостей ICMP-протоколу.

2.4 Висновки за розділом

Досліджено ряд принципів та методів мережевої стеганографії. У ході цього було виявлено, що:

1. Головними механізмами розміщення даних повідомлення, які необхідно надсилати мережею у неявному вигляді, для випадку методів мережевої стеганографії є:

– маніпуляція даними службових на інформаційних полів у межах мережевих пакетів поширених типів;

– маніпуляція порядком та умовами надсилання пакетів;

– поєднання обох зазначених підходів.

2. Одними з поширених мережевих стеганографічних методів є:

– система HICCUPS, де залежно від умов обміну даними може бути використано один або кілька з можливих режимів (HDC1, HDC2 або HDC3 відповідно);

– метод LACK, що передбачає розміщення приховуваної інфрмації у пакетах аудіо-даних VoIP-потоку;

– група методів ICMP-стеганографії.

3 ДОСЛІДЖЕННЯ БАЗОВИХ ХАРАКТЕРИСТИК ICMP-ПРОТОКОЛУ

3.1 Призначення протоколу ICMP

Головним завданням протоколу межмережевих керуючих з’єднань, або ICMP (Internet Control Message Protocol), є управління мережею [6].

Для цього у рамках протоколу визначається певний перелік різних керуючих повідомлень.

Протокол входить до стеку TCP/IP.

Самі ICMP-повідомлення формується на базі IP-пакетів, які генерують ICMP-відповідь.

При цьому, протокол IP виконує інкапсуляцію відповідне ICMP-повідомлення з новим заголовком IP та надсилає далі одержані пакети.

Важливим тут є те, що кожне повідомлення ICMP інкапсулюється в окремо узятий IP-пакет.

У свою чергу, усі повідомлення ICMP умовно може бути розподілено на два типи - повідомлення про помилки, а також повідомлення виду запит - відповідь.

Так, зокрема, на базі даного протоколу вузол-відправник може бути проінформовано про те, що:

– та чи інша мережа є недоступною;

– вузол-приймач є недоступним;

– для доставки пакетів необхідно активувати фрагментацію (за умови, що попередньо активовано параметр DF (Don’t fragment) тощо.

У свою чергу, прийнявши те чи інше ICMP-повідомлення, кінцевий вузол отримує відомості про те, що у ході обміну даними існує певна помилка, яку слід виправити.

Водночас, заходи з виправлення помилок вже не регламентуються протоколом ICMP.

Протокол ICMP належить до мережевого рівня моделі OSI (рис.3.1), хоча, при цьому, частково може виконувати функції, властиві протоколам транспортного рівня.

Як вже було зазначено вище, для передавання каналами попередньо виконується інкапсуляція ICMP-пакетів у пакети IP.

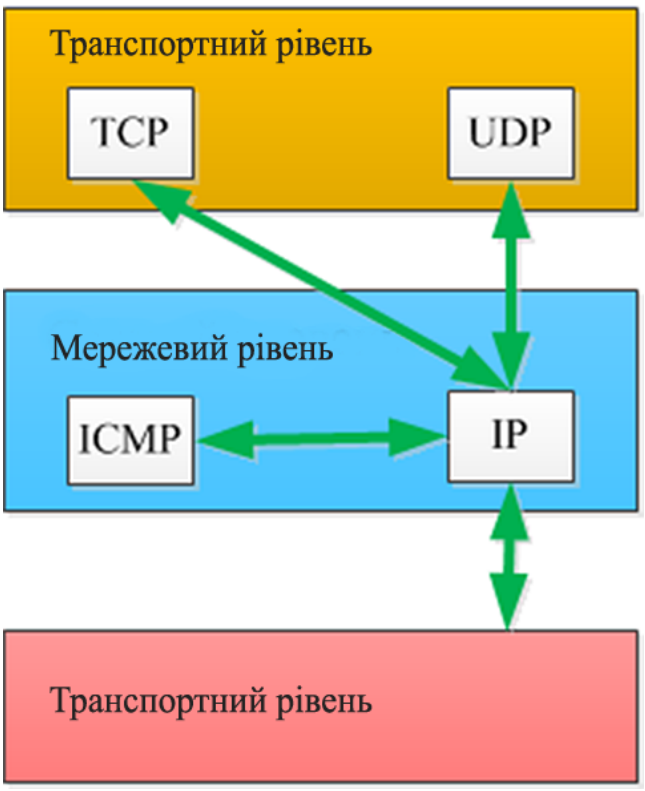

Рисунок 3.1 – Протокол ICMP у моделі OSI

Виходячи з цього, можемо зазначити, що фактично, має місце схема, як показано на рисунку 3.2.

Рисунок 3.2 – Співвідношення між IP та ICMP-пакетами

Зрозуміло, що функціонал сповіщення щодо помилок є надзвичайно важливим, так як безпосередньо сам протокол IP не передбачає власних механізмів, які здатні інформувати вузли у разі виникнення тих чи інших мережевих помилок.

З одного боку, ICMP – не єдиний протокол, який інформує вузли щодо виникнення тих чи інших помилок [6].

Проте, з іншого боку, наприклад, протокол TCP, а також протоколи прикладного уровня OSI механізми інформування про помилки застосовують

або після того, коли мережеве з’єднання вже було встановлено, або на самій стадії встановлення.

Окрім цього у випадку, коли на мережевому рівні існує та чи інша проблема, пакет не може бути трансльовано до вищих рівнів - транспортного та прикладного і, відповідно, «верхні» протоколи не матимуть відомості про наявність та суть проблеми. Відповідно, ці протоколи не зможуть відповідним чином відреагувати на її наявність.

Далі, для ілюстрації роботи ICMP, розглянемо ряд прикладів.

3.2 Приклади роботи ICMP

Уявімо, що перед початком передавання інформації, у налаштування IP-пакетів на маршрутизатору з тієї чи іншої причини було увімкнено параметр DF.

При цьому, на шляху проходження пакетів зустрічається один з проміжних мережевих вузлів, якому для обробки пакету необхідно виконати фрагментацію.

У той же час, оскільки попередньо було увімкнено DF, даний мережевий вузол не зможе виконати обробку запиту а відтак - не зможе спрямувати пакет на транспортний рівень для подальшого опрацювання. Таким чином, нефрагментований пакет буде знищено.

У цьому разі, без інформування протоколом ICMP про існуючу проблему, вузол, який є джерелом пакетів, продовжує надсилати пакети з забороною на фрагментацію.

У свою чергу, користувачі сервісу, якому належать пакети (у т.ч. додатки прикладного рівня), по перше, не отримують тих чи інших послуг а по-друге – не отримують відомостей про наявність проблеми та її сутність, що не дасть змоги її виправити [6].

У розглянутому ж випадку вузол-приймач спочатку виконує процедуру видалення нефрагментованого пакету, після чого далі виконує формування спеціалізованого пакету ICMP-відповіді. У такому пакеті, вказується на те, що для можливості передавання даних необхідно виконати активацію фрагментації пакетів.

Схематично даний процес може бути проілюстровано рис.3.3.

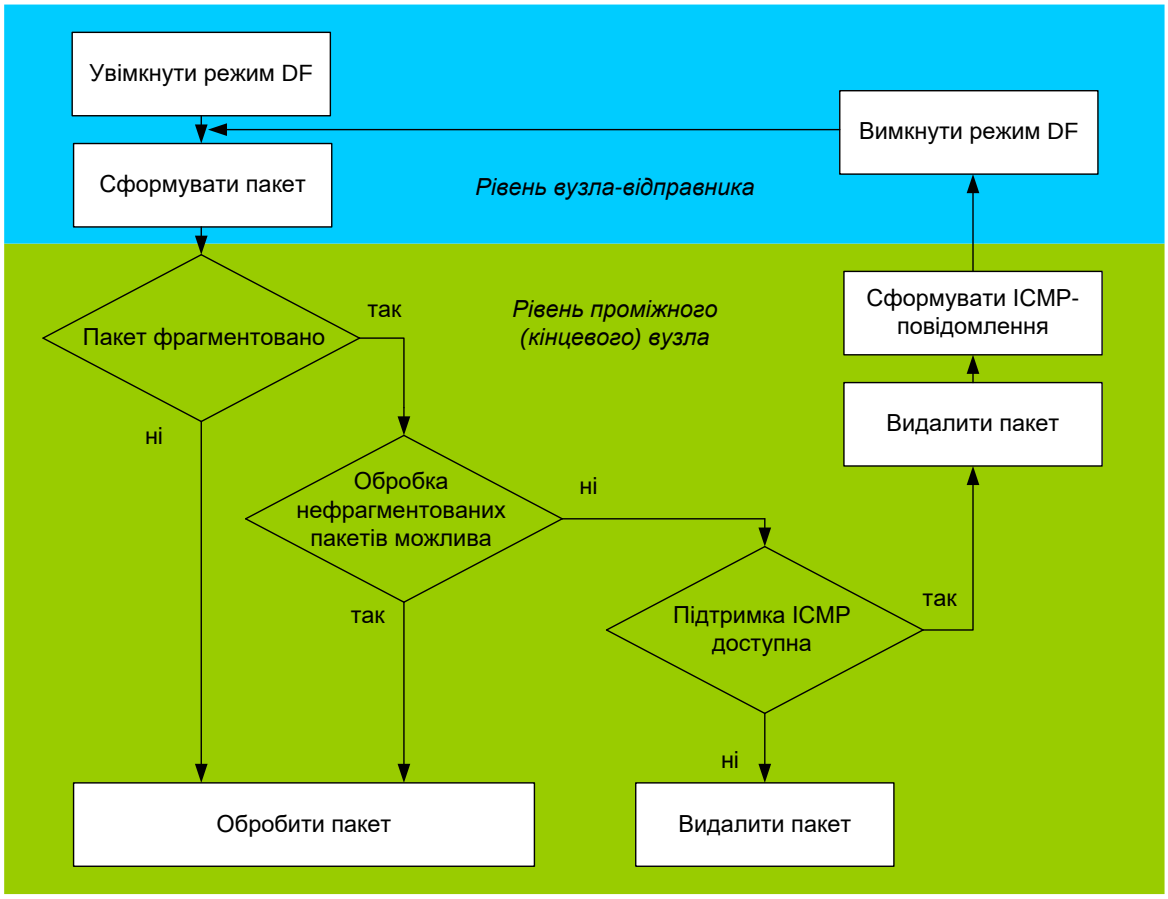

Рисунок 3.3 – Приклад використання ICMP для випадку управління режимом фрагментації пакетів

Розглянемо інший приклад.

З користувацької точки зору процес відвідування будь-якої веб-сторінки зводиться до внесення відповідної адреси у браузер, після чого її зміст буде завантажено на екран. Узагальнена схема перебігу зазначеного процесу зводиться до наступного:

1. Користувацький термінал формує пакети та спрямовує їх до маршрутизатору.

2. Маршрутизатор надсилає сформовані пакети інтернет-провайдеру.

3. Здійснюється звернення до відповідного ресурсу з запитом на завантаження змісту сторінки.

При цьому, процес перенаправлення пакетів від одного вузла до іншого функціонує на третьому, тобто, мережевому рівні OSI.

Далі уявімо, що користувацький маршрутизатор не має відомостей про те, куди слід спрямовувати пакети (це може зумовлюватися рядом причин). За таких умов пакети буде видалено.

У свою чергу, тут знову таки буде задіяно ICMP.

Так, видаливши пакет, маршрутизатор сформує повідомлення про те, що мережа є недоступною, надішле створене повідомлення до клієнтського терміналу.

Після цього у браузері клієнта буде відображено відповідне повідомлення, яке проінформує про те, що з'єднання з зовнішньою мережею відсутнє.

Розглянуті приклади свідчать про те, що навіть на мережевому рівні необхідно мати механізми контролю помилок.

Таким чином, це вказує на глобальність та природність використання ICMP у мережі.

Інакше кажучи, та чи інша кількість ICMP-пакетів, які надсилаються каналами мережі, не сприймаються як аномалія а відтак, такі пакети може бути використано для неявного обміну даними у мережі.

Для того, щоб визначити, яким чином ICMP-пакети може бути використано для створення секретних каналів, розглянемо попередньо формати цих пакетів детальніше.

3.3 Структура пакету ICMP

Загальну структура пакету зображено на рисунку 3.4 [1, 6].

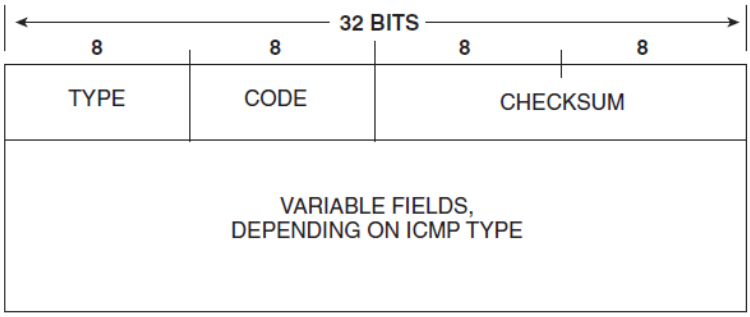

Рисунок 3.4 – Структура пакету ICMP

На схемі, наведеній рис. 3.4, поле Type визначає тип пакету.

Разом з тим, більшість типів пакетів може бути виокремлено у специфічні підтипи, для чого додатково застосовується також службове поле Code. Відповідно, Checksum – поле контрольної суми пакету, яке використовується для контролю його цілісності.

Усі можливі типи та підтипи пакетів ICMP визначено у рекомендації RFC 1700 [14].

Наприклад, у випадку, коли Type=3, а Code=6, це відповідає сповіщенню відносно того, що запитувана мережа є невідомою (Destination Network Unknown).

Коли ж Type=8, а Code=0, це відповідає ECHO-пакету, при Type=11, а Code=0 маємо сповіщення про перевищення TTL у ході передавання (Time to Live Exceeded in Transmit) і т.д.

Таким чином, зчитуючи значення Type та Code, вузол-приймач розуміє, який саме тип сповіщення йому надіслано.

У свою чергу, поле даних міститиме у собі різну інформацію залежно від того, які саме значення прийматимуть Type та Code.

Приклад змісту ICMP-пакету у ході його аналізу з Wireshark наводиться на рисунку 3.5.

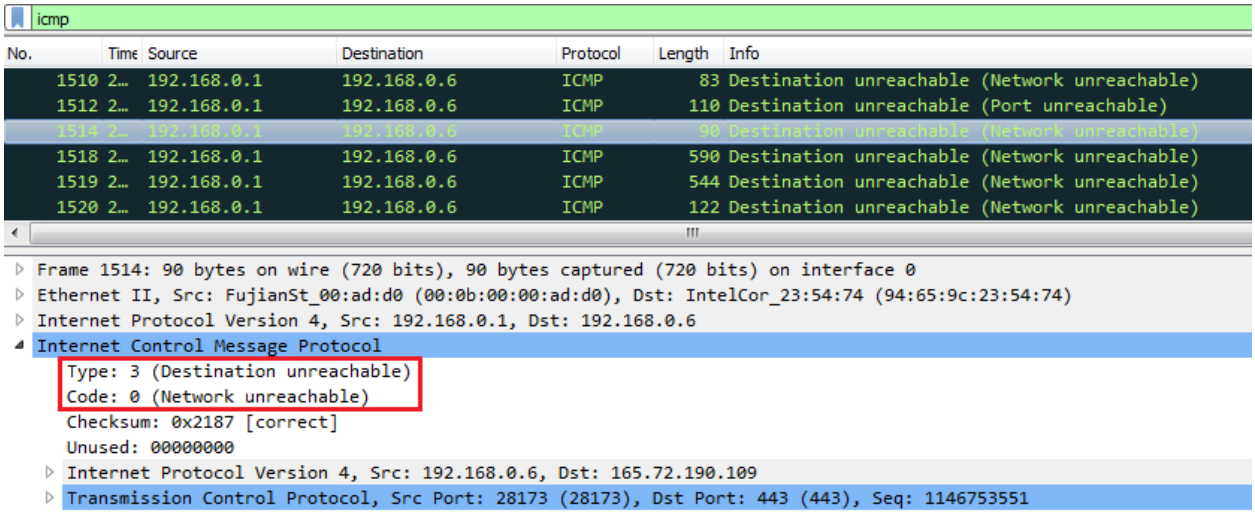

Рисунок 3.5 – Приклад змісту ICMP-пакету у ході його аналізу з використанням Wireshark

З рисунку 3.5 видно, що даний пакет є сповіщенням про те, що вузол призначення не може бути розпізнано, так як не може бути розпізнано мережу призначення.

Також вказуються вузол-джерело та вузол призначення, тип протоколу а також контрольна сума, яка у даному випадку є вірною.

4 ДОСЛІДЖЕННЯ МОЖЛИВОСТІ РЕАЛІЗАЦІЇ ПРИХОВАНОГО КАНАЛУ НА БАЗІ ПАКЕТІВ ICMP

4.1 Аналіз можливості використання службових полів ICMP-пакетів для розміщення прихованого повідомлення

У цілому, у явному вигляді частіше за все фігурують такі типи ICMP-пакетів, як [6, 14]:

– ECHO-запит. Саме ці пакети генерує утиліта ping для перевірки досяжності вузла у мережі;

– ECHO-відповідь. Пакети даного типу надсилають вузли у відповідь на ECHO –запити;

– час життя пакету (TTL) вичерпано. Такі пакети, у свою чергу, використовує утиліта traceroute для того, щоб визначити проміжні вузли мережі.

При цьому, очевидно, що пакети ECHO-запитів та ECHO-відповіді є поширені у мережі а факт їхньої появи – штатне явище, що не викликає підозри. Тому саме пакети даного типу може бути використано для передачі приховуваних даних.

Розглянемо зміст та структуру пакету ECHO-запиту, як показано на рис. 4.1.

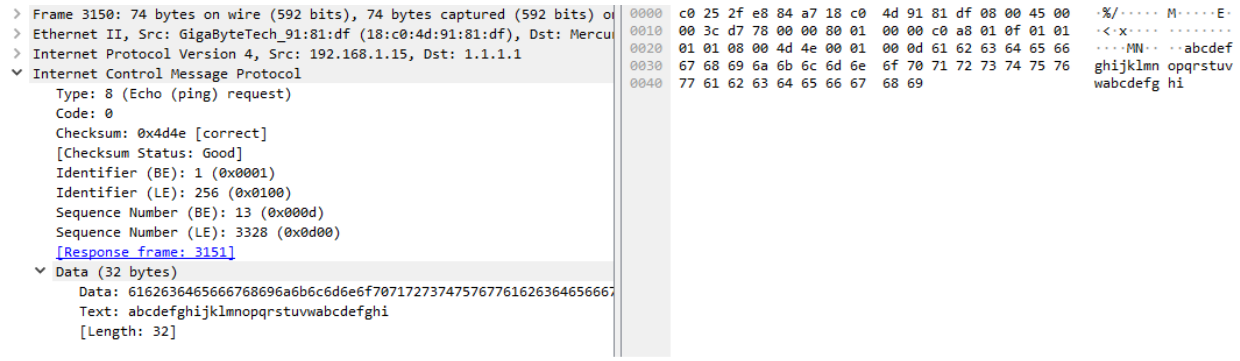

Рисунок 4.1 – Приклад змісту ICMP-пакету ECHO-запиту у ході його аналізу з використанням Wireshark

З рисунку 4.1 бачимо, що теоретично нам може бути доступними ряд полів, а саме:

– тип: 8 (ECHO). Дане поле змінити не можемо;

– код: 0. Зміна цього поля також неможлива;

– поле контрольної суми. Значення цього поля обчислюється;

– ідентифікатор. У сутності, це номер, який можна інтерпретувати як «сесію» пінгів;

– номер пакету у послідовності, це є порядковим номером пакету у межах «сесії».

– поле даних.

При цьому, на перший погляд включення секретного повідомлення до змісту поля даних є найбільш очевидним та простим рішенням. Переваги такого способу наступні [15]:

– у рамках одного ICMP-пакету может бути розмішено до 65535 байт даних (якщо не брати до уваги заголовки IP та ICMP) відповідно до обмежень для IP-пакетів;

– користувач має у розпорядженні вже існуючий доступний інструментарій для прихованої передачі, що надається утилітою ping. Так, аргумент -p даної утиліти дає змогу прямо вказувати дані для наступної передачі безпосередньо в пакетах ECHO -запитів, тоді як аргумент -s дозволяє вказати розмір таких даних.

З іншого боку, включенню секретного повідомлення безпосередньо у поле даних ICMP-пакету властивий ряд суттєвих недоліків, зокрема [16]:

– факт передавання повідомлення без використання «зашумлюючого» наповнення поряд з ним та без шифрування є очевидним;

– у разі максимального заповнення пакети досить легко виявити;

– у різних версіях ICMP-тунелів передбачено використання т.з. «магічних» чисел для того, щоб відокремити звичайні «пінги» від корисного навантаження.

Окрім цього, даний спосіб організації каналу є актуальним лише в умовах, коли як протокол TCP, так і UDP з тих чи інших причин не може бути використано, а відносно ICMP, у свою чергу, не застосовано суттєвих обмежень використання. Тому розглянемо інші способи внесення даних до ICMP-пакетів.

Для цього ще раз проаналізуємо поля пакету, які можемо змінювати, наприклад [16]:

– контрольна сума. Як вже говорилося, значення цього поля обчислюється. Отже, змінюючи значення інших полів пакету, можемо впливати на зміст цього поля;

– поле ідентифікатору. Тут маємо число, на представлення якого зарезервовано 16-біт. У межах однієї сесії це число є однаковим для усього масиву пакетів;

– номер пакету у послідовності. Починається з 1 та поступово збільшується з кожним новим пакетом.

Разом з тим, слід пам’ятати, що значення ідентифікатору ICMP також може змінюватися. Відбувається це у разі проходження пакету через NAT [6].

Отже, зрозуміло, що у цьому разі зміна ідентифікатору автоматично веде до зміни величини контрольної суми.

Звідси виходить, що використовувати або поле контрольної суми, або поле ідентифікатору у чистому вигляді неможливо, а сам підхід потребує доробки.

У свою чергу, як ми бачимо, здійснюється збільшення на 1 номеру пакету у послідовності. І якщо маніпулювання зі значенням даного поля може бути не поміченим на рівні мережевого обладнання, то на рівні дампу трафіку подібні зміни будуть відстежуватися.

Тобто, формально використання даних полів пакету є недоцільним. Тому виконаємо більш глибокий аналіз поля даних ICMP-пакету.

4.2 Аналіз можливості використання поля даних ICMP-пакету для вбудовування повідомлень

Зміст блоку даних пакету ECHO-запиту за замовчування, під час запуску утиліти ping є стандартним та залежить від середовища, де виконується дана операція. Зокрема [15, 16]:

– для випадку платформи Windows розмір пакету за замовчуванням складає 32 байти. При цьому, у поле даних циклічно записуються символи англійського алфавіту (виключаючи символи Y та Z), таким чином повністю заповнюючи доступний простір;

– для платформ macOS та більшості збірок Linux у випадку, коли ping виконується з використанням пакету iputils, розмір пакету дорівнює 56 байт. При цьому, зміст і-го байту корисного навантаження розраховується як .

При цьому також додатково перші 8 або 16 байт, з огляду на розрядність операційної системи, буде займати т.з. структура timeval.

Важливим тут є те, що утиліта ping, що входить до набору iputils, на початку ICMP-даних зберігає часову мітку щодо відправки пакету.

При цьому, оскільки пакет ECHO-відповіді обов'язково має містити дані з ECHO-запиту, за цих умов час кругової затримки, або round trip time (RTT) може бути розраховано у вигляді різниці поточного часу і часу , який, у свою чергу, записано у поле даних ECHO-відповіді, тобто [6]:

$$RTT = t_{curr} - t_{answ}, \tag{4.1}$$

Даний підхід, звичайно, сприяє спрощенню розробки, оскільки у цьому разі не потрібно зберігати час відправки ECHO-запитів.

Окрім цього, даний прийом відомий ще з першого коміту [17] iputils.

Тому існуючі аналізатори трафіку, у т.ч., Wireshark, сприймають часову мітку не як елемент поля даних, а як інформацію, що відноситься до заголовку ICMP-пакету, можна бачити з рисунку 4.2.

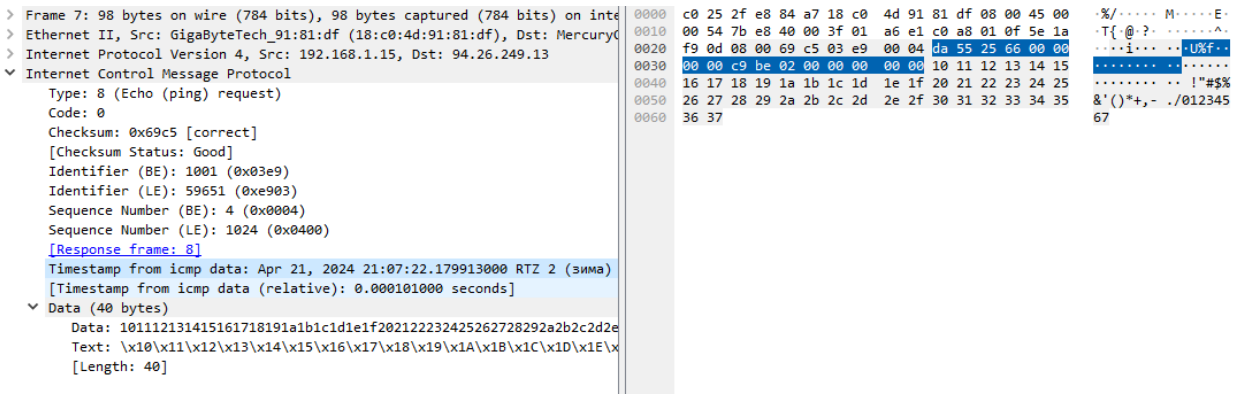

Рисунок 4.2 – Особливості інтерпретації Wireshark часової мітки у полі даних

Тобто, як можна бачити з рисунку 4.2, Wireshark було винесено мітку часу до заголовку ICMP-пакету, при цьому її розмір було виокремлено з блоку Data, хоча, при чьому, рекомендація RFC 792, зі свого боку, не регламентує подібних дій.

Теоретично дану обставину може бути використано для рішення наших завдань.

Проте, попередньо, виконаємо розгляд структури даних timeval, яку, у свою чергу, було описано у рамках системного виклику gettimeofday (рисунок 4.3).

Тут, як можемо бачити, поле tv_sec містить у собі кількість секунд з початку епохи, у ролі чого взято відмітку від 1 січня 1970 року [17].

Водночас, tv_usec містить у собі додаткові дані у мікросекундах, що сприяють точності.

Разом з тим, міжнародна система одиниць (СІ) стверджує, що одна секунда містить у собі мільйон мікросекунд.

Відтак, це може вказувати на те, що у даних щодо мікросекунд може зберігатися до двох байт даних.

У свою чергу, це може бути реалізовано наступним чином [16]:

```
struct timeval* t = (struct timeval*)icmp_packet.payload;
gettimeofday(t, NULL);
t->tv_usec = (t->tv_usec - t->tv_usec & 0xFFFF) + data;
```

```
/* Опис у документації */
struct timeval {
    time_t      tv_sec;     /* seconds */
    suseconds_t tv_usec;    /* microseconds */
};

/* Опис у sys/time.h */
/* A time value that is accurate to the nearest
   microsecond but also has a range of years.  */
struct timeval
{
#ifdef __USE_TIME_BITS64
  __time64_t tv_sec;     /* Seconds.  */
  __suseconds64_t tv_usec;  /* Microseconds.  */
#else
  __time_t tv_sec;       /* Seconds.  */
  __suseconds_t tv_usec; /* Microseconds.  */
#endif
```

Рисунок 4.3 – Структура даних timeval

Два молодших байта тут вносять похибку у часову мітку і таким чином можуть внести додатковий час до 65 мс.

Наприклад, наступним чином:

```
Timepoint: 2024-04-24 14:00:00.000000
Data: 0xFFFF, 65535 decimal.
Packet timepoint: 2024-04-24 14:00:00.065535
Packen sent: 2024-04-24 14:00:00.001000
Packet intercepted by a sniffer: 2024-04-24 14:00:00.002000
```

Тобто, як видно з наведеного лістингу, пакет має часову мітку «з майбутнього».

Разом з тим, від'ємний відносний час Wireshark не вважає аномалією, тому пакет з такими атрибутами не маркується як помилковий та відносно нього системою не виноситься попереджень, як можна бачити з рисунку 2.7.

Отже, факт існування пакетів «з майбутнього» системою сприймається як повністю штатна ситуація (рис. 4.4).

Зокрема, такі випадки можуть виникати у наслідок того, що або відправник, або приймач, або кожен з учасників обміну даними матиме некоректно налаштований час.

З іншого боку, за наявності пакетів з часовими мітками, які змінюються від майбутнього до минулого часу може викликати обгрунтовані підозри. Окрім цього, факт передавання тексту латиницею може бути виявлено простим переглядом змісту у шістнадцятеричному редакторі.

Таким чином, до переваг розглянутого способу внесення даних у ICMP-пакет можемо віднести [16, 17]:

– відсутність у пакеті «магічних» констант та будь-яких ознак, що можуть вказувати на передавання даних;

– забезпечується можливість маскування даних, які не повторюються, за рахунок особливостей мікросекундної точності представлення часу.

З іншого боку, мінусами способу є:

– необхідність відстежування та уникнення появи пакетів «з майбутнього»;

– фіксована локація даних, що приховуються. У нашому випадку це — байти 0x33 а також 0x34;

– низька швидкість надсилання даних (для розглянутого випадку – 2 байти за секунду.

При цьому, обмеженість локації длдя розміщення даних можна вважати одним з найсуттєвіших недоліків способу, так як це суттєвим чином спрощує процедуру стегоаналізу.

Разом з тим, у випадку передавання масивів однакових даних – нульових символів чи одиниць – зовні процес буде сприйматися, як надсилання ECHO-запитів з проміжком в одну секунду а також з точністю до мікросекунди і без жодних погрішностей, що є потенційно підозрілим [15].

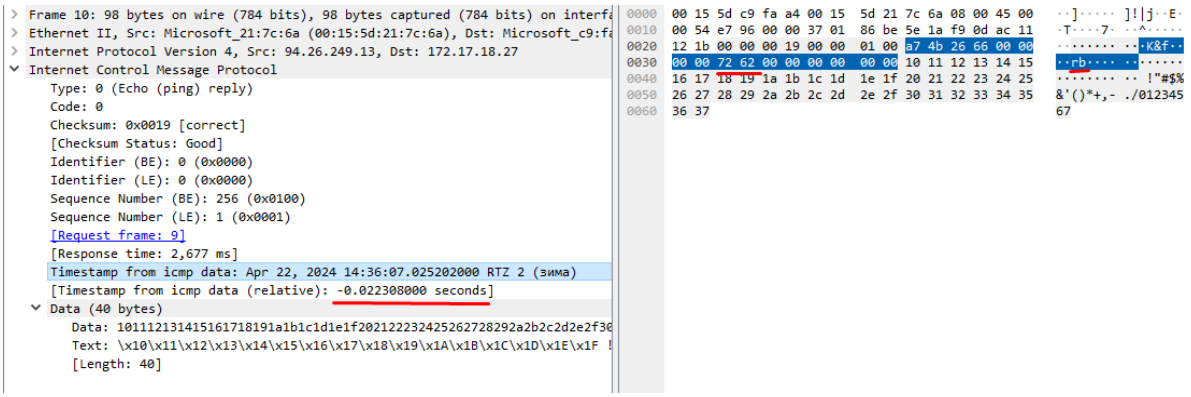

Рисунок 4.4 – Аналіз пакету «з майбутнього» у Wireshark(ссілку надо віше)

Отже, підхід на базі маніпулювання часовими мітками у полі даних ICMP-пакетів потенційно не забезпечує високого рівня захищеності секретного повідомлення, яке передається у даний спосіб.

Далі проаналізуємо можливість побудови прихованого каналу, використовуючи неявні параметри ICMP.

4.3 Використання неявних параметрів ICMP для побудови прихованого каналу

Одним з неявних параметрів ICMP, який є важливим для функціонування протоколу, але жодним чином не фігурує у заголовках, є інтервал слідування пакетів, так як надсилання ECHO-запитів здійснюється з деяким інтервалом [15].

Відтак, теоретично зміст приховуваного повідомлення може бути закодовано за допомогою маніпулювання величинами інтервалів надходження пакетів.

При цьому, для утиліти ping значення інтервалу між сусідніми ECHO-запитами за замовчуванням складає 1 секунду. Отже. виконаємо додавання

деякого обсягу мілісекунд, який буде рівним, наприклад, змісту байту, який необхідно переслати [16].

Водночас, з тієї причини, що мережа за замовчуванням є ненадійним середовищем, виконуємо налаштування часової мітки ICMP-пакету згідно з попередньо обчисленим значенням і відповідно до цього часу далі надсилаємо пакет приймачеві.

Фактично для цього вносимо ряд опцій у структуру timeval, як показано далі [17]:

```
/* додаємо байт даних у вигляді мілісекунд */
tv.tv_usec += ((uint8_t)message[i]) * 1000;
/* додаємо секунду та можливий переніс з мікросекунд */
tv.tv_sec += 1 + tv.tv_usec / 1000000;
/* усуваємо переповнення, якщо воно виникає */
tv.tv_usec = tv.tv_usec % 1000000;
/* опціонально: виконуємо модифікацію мікросекунд, щоб надати їм природній вигляд */
tv.tv_usec = ((tv.tv_usec / 1000) * 1000) + (rand() % 1000);
```

У результаті таких дій байт даних являтиме собою різницю міток часу між двома сусідніми пакетами.

Окрім цього, постійно зазнають змін три байти мікросекунд, тим самим створюючи хибну ознаку.

Це може бути корисно у тому випадку, коли аналітик намагається виявити вбудовані дані безпосередньо у сам пакет ICMP.

Важливо, що додаткова затримка, внесена штучно для зміни тривалості інтервалу між пакетами, формально не має ознак, які б вказували на наявність аномалії.

Це зумовлено тим, що навіть пакети ECHO-запитів стандартного змісту надсилаються до мережі не чітко щосекунди, а можуть мати затримку тривалістю до трьох мілісекунд.

Звичайно, пакети ECHO-запитів, що слугують для непрямого передавання секретних повідомлень, можуть мати помітно порівняно більший рівень затримки.

Водночас, виявити факт маніпуляцій з інтервалом пакетів можливо лише маючи у розпорядженні дамп «штатного» трафіку.

Розглянутому способу передавання секретних повідомлень властиві такі переваги [16]:

– факт вбудовування даних у пакет відсутній, пакет при цьому не містить ознак модифікації;

– ефективне виявлення прихованої передачі у розглянутий спосіб гарантується лише попереднім збором і наступним переглядом логів щодо передуючих обмінів ICMP-пакетами від того ж самого вузла.

Дане завдання є надзвичайно ресурсоємким, вимагає великих часових та обчислювальних ресурсів.

Разом з тим, недоліками способу можна вважати:

– низька швидкість обміну даними – 1 Байт/с.

– втрата одного пакету зумовлює втрату двох байт повідомлення.

Таким чином, розглянутий спосіб створення прихованого каналу на базі протоколу ICMP, з одного боку, не забезпечує високої швидкості передачі повідомлень. З іншого боку, коли протокол ICMP у мережі не має обмежень, забезпечується захищеність передаваних повідомлень, виявлення яких не є тривіальним завданням.

4.4 Умови та особливості використання ICMP-каналу, побудованого шляхом маніпуляції інтервалами надсилання пакетів

За великим рахунком, для успішного використання прихованого каналу на базі ICMP-пакетів, без обмежень, додатково має бути вирішено ряд завдань, а саме:

– завдання підвищення ймовірності успішної доставки пакету;

– односпрямованість даних;

– диференціація «штатних» пакетів ECHO-запиту та пакетів, які несуть корисне навантаження.

4.4.1 Підвищення ймовірності успішної доставки пакету

Протокол ICMP не надає гарантій доставки ECHO-пакету а отже – пакет може бути як втрачено у ході передавання, так і прийнято у невірному порядку.

Проте, за певних умов (таких, наприклад, як відносна стабільність ділянки мережі, де ведеться передавання даних) проблема втрати пакету не є суттєвою.

Так, у рамках дослідження на адресу дзеркала Debian у Португалії було надіслано 3000 ECHO-запитів. У цьому випадку значення RTT для цього випадку складало лише порядку 310-360 мс, при цьому жоден пакет не було втрачено.

У свою чергу, забезпечення коректного порядку слідування пакетів може бути досягнуто, використовуючи службове поле «номер у последовательности» заголовку пакету ICMP.

Проблему ж ймовірної втрати пакетів може бути вирішено застосуванням надлишкового кодування.

4.4.2 Односпрямованість даних

Підхід, який було розглянуто у пункті 4.3, дає можливість створення односпрямованого стеганографічного каналу зв'язку, що зумовлюється наступними факторами:

– у штатному режимі функціонування ICMP, приймач копіює дані з ECHO-запиту до ECHO-відповіді. При цьому, у разі модифікації даних ECHO-запиту, такий пакет буде сприйматися, як пакет з помилкою, а сама його наявність викликатиме підозру;

– використання затримки між надсиланням пакетів ECHO-відповідей може бути сумнівним з причин наявності джитеру.

4.4.3 Диференціація «штатних» пакетів ECHO-запиту та пакетів, які несуть корисне навантаження

Необхідність виокремлення пакетів, що несуть корисне навантаження з усього масиву ICMP-пакетів, на надходять до вузла, на перший погляд є очевидною.

Разом з тим, як свідчить дослідження, для розпізнавання «корисних» пакетів, які призначені вузлу, передавачеві достатньо виконати надсилання кількох десятків ECHO-запитів і далі підтримувати коректне поле номеру у послідовності [15].

4.5 Побудова прихованого каналу на базі ICMP за умови модифікації підходу щодо використання полів пакету відповідно до умов його проходження

Припустимо, що необхідно налагодити канал прихованого передавання даних у рамках ідеології ICMP-стеганографії за умови, що:

– з тих чи інших причин маніпуляції довжиною інтервалу між пакетами неможливі;

– на ділянці між передавачем та приймачем передбачено NAT.

У цьому випадку прихований канал може бути створено на базі змін даних службових та інформаційних полів ECHO-пакетів, якщо ураховуються умови проходження пакету між передавачем та приймачем [16].

Як зазначалося раніше, ICMP-пакети інкапсулюються до пакетів IP. Тому спочатку розглянемо структуру IP-пакету (рисунок 4.5).

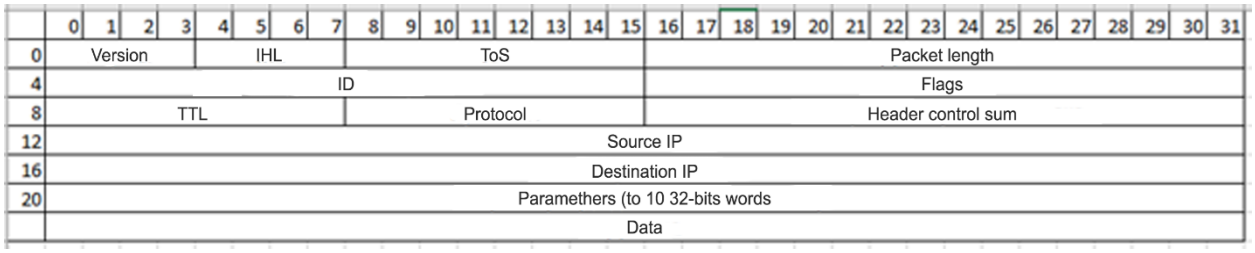

Рисунок 4.5 – Структура пакету IP

Формально полями, внесення змін до яких не вносить суттєвої зміни до пакету, це:

– IHL, значення у якому може змінюватися у діапазоні від 5 до 15;

– поле ToS, що використовується для встановлення приорітетності трафіку та повідомлень про колізії без відкидання пакетів. За замовчуванням значення цього поля рівне 0. Теоретично сюди може бути вписано до байту даних;

– поле довжина пакету, у свою чергу, може бути використано для передавання чисел від 20 до 65535;

– у полі TTL теоретично може передаватися до 7 біт даних. Для цього необхідно знати кількість хопів до приймача та брати це до уваги.

Ураховуючи це, спробуємо внести зміни до полів пакету, скориставшись можливостями фреймворку Scapy [18] у середовищі Python [19].

Побудова прихованого каналу шляхом маніпулювання змістом полів на базі Scapy

Для дослідження можливості побудови прихованого каналу необхідно мати у розпорядженні 2 ПК з інстальованим середовищем Python та фреймворком Scapy.

При цьому, у межах цих ПК необхідно створити передаючий та прийомний модулі, скориставшись функціоналом Scapy.

У свою чергу, створюємо передаючий модуль sender.py, який виконуватиме надсилання ICMP-пакетів спершу без вбудованих повідомлень:

```
from scapy.all import *
# створюємо пакет для 192.168.1.55 с icmp-type 8 (echo-request)
pkt = IP(src="192.168.1.54", dst="192.168.1.55") / ICMP(type = 8)
# Надсилаємо пакет
sr1(pkt)
```

При цьому, Scapy попередньо, перед самим надсиланням пакету, самостійно заповнить інші поля пакету значеннями за замовчуванням, а також виконає розрахунок контрольної суми [16].

Далі, на прийомному боці, створюється додаток-приймач listener.py, функціонал якого зводиться до прийому та виведення на консоль усі пакети ICMP, які надходять:

```
from scapy.all import *

# налаштування прослуховування пакетів
#   filter -- лише icmp
#   timeout -- прослухування лише 10 секунд
#   count – очікується не більш, ніж 100 пакетів
#   iface – лише у межах інтерфейсу eth1


packets = sniff(filter = "icmp", timeout = 10, count = 100, iface
= "eth1")

# перевірка за усіма отриманими пакетами
for pkt in packets:
    # необхідні лише отримані пакети echo-request
    if pkt[ICMP].type != 8:
        continue
    # виведення у зручному вигляді
    pkt.show()
```

Приклад виведення даних на консоль додатком listener.py наводиться на рисунку 4.6.

Тепер спробуємо заповнити поле ідентифікатору (ID) з заголовку IP-пакету. Внесемо до зазначеного поля символи «A» та «B», для чого слугує наступний фрагмент коду:

```
payload = ord("A") * 0x100 + ord("B")
pkt = IP(src="10.0.0.1", dst="192.168.1.55", id = payload) / ICMP(type = 8)
```

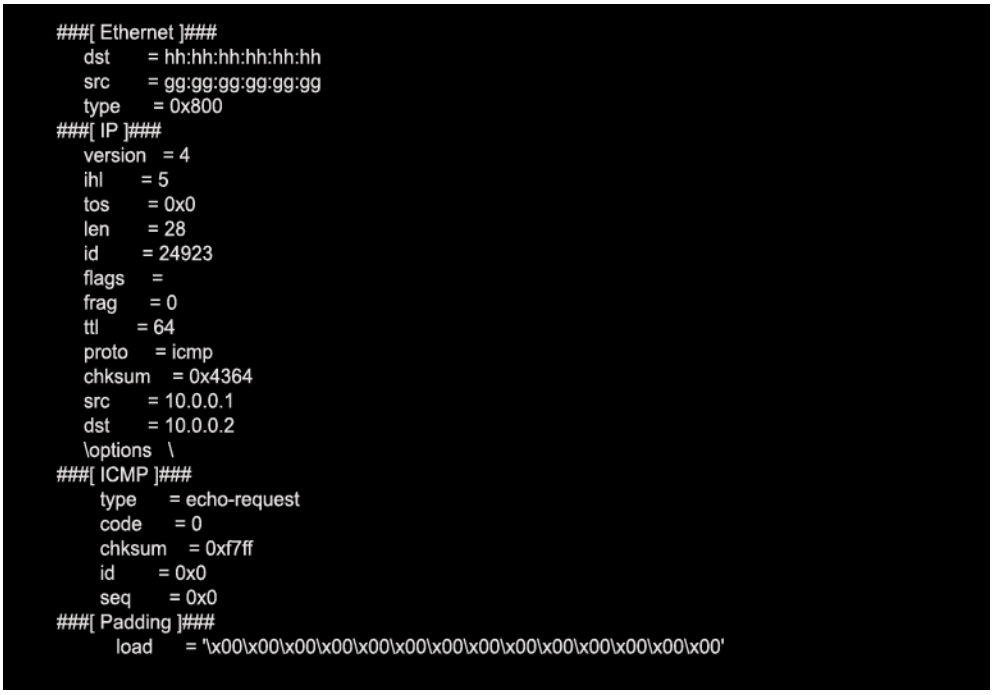

Рисунок 4.6 – Приклад виведення даних щодо отриманого пакету додатком listener.py

Окрім цього, беручи до уваги відсутність ряду обмежень в умовах надсилання пакетів, як було зазначено раніше, у заголовку самого пакету ICMP можемо використати аналогічне поле, куди також може бути завантажено 2 байти.

Далі відповідним чином внесемо зміни до коду додатку-приймача для виведення на консоль одержаних даних:

```
from scapy.all import *
import sys

packets = sniff(filter="icmp", timeout = 10, count = 100, iface="eth0")
```

```
for pkt in packets:
    if pkt[ICMP].type != 8:
        continue
    # розділюємо 2 символи
    a, b = divmod(pkt[IP].id, 0x100)
    sys.stdout.write(chr(a))
    sys.stdout.write(chr(b))
    sys.stdout.flush()
```

У зазначений спосіб в умовах обмежень на обмін пакетами у мережі може бути заповнено практично кожне поле, яке попередньо було визнано придатним для цієї операції.

У цілому, процедура передавання даних, реалізована у розглянутий вище спосіб, за спрощених умов обміну даними може вважатися неочевидною.

Разом з тим, за цих умов дані може бути приховано за рахунок розміщення у полі контрольної суми, що сприятиме збільшенню ступеню їх захищеності.

Так, згідно з RFC1071, контрольна сума розглядається як результат побітової інверсії арифметичної суми.

Припустимо, що для деякого заголовку необхідно обчислити контрольну суму.

Під час розрахунків виконується обнулення поля checksum:

```
4500 003c 000a 0000 8001 [checksum] c0a8 000d c0a8 000d
```

Спершу складаються усі 16-бітні слова, при цьому запам’ятовуючи перенесення зі старшого розряду:

```
4500 + 003c + 000a + 0000 + 8001 + [checksum=0000] + c0a8 + 000d +
c0a8 + 000e = (2) 46b2
```

Далі результат складається з переносами:

```
46b2 + 2 = 46b4
```

Після цього виконується інверсія:

```
~(46b4) = b94b
```

Отже, b94b —контрольна сума для нашого випадку. У свою чергу, перевірку може бути виконано шляхом підстановки отриманих даних у заголовок з подальшим виконанням попередньо виконаних дій. При цьому, якщо у результаті отримуємо FFFF, це буде свідчити про коректність знайденої контрольної суми:

```
1. 4500 + 003c + 000a + 0000 + 8001 + [checksum=b94b] + c0a8 + 000d
+ c0a8 + 000e = (2) FFFD
2. FFFD + 2  = FFFF
```

Відомо, що значення контрольної суми пакету зазнає змін у ході проходження вузлів у мережі, оскільки змін зазнає також величина TTL.

Окрім цього, при проходженні NAT у пакеті виконується заміна адреси відправника, що також впливає на величину контрольної суми. А також зменшується значення TTL при досягнення вузла-приймача.

Важливим тут є те, що розрядність ідентифікатору відповідає розрядності поля контрольної суми. Даний факт дозволяє здійснювати вплив на контрольную сумму і виконувати її зміни на будь-які значення з можливого діапазону.

Оскільки величину контрольної суми (корисне навантаження) буде розраховано лише після проходження фінального вузла маршруту, у ході розрахунку необхідно урахувати усі параметри, які може бути змінено за час проходження маршруту.

Розглянемо алгоритм розрахунку «ідентифікатора», здатного забезпечити необхідну контрольну суму:

1. Налаштування пакету як для умов проходження фінального вузла (IP, TTL і т.д.)

2. Запис корисного навантаження до ідентифікатору.

3. Розрахунок контрольної суми.

4. Запис результату обчислення до ідентифікатору пакету, який надсилається.

У свою чергу, функція, яка з урахуванням кількості хопів, IP-адрес за NAT'ом а також двох байт корисного навантаження (повідомлення або його частини) формуватиме пакет:

```
# src – адреса відправника
# src_nat - адреса відправника за NAT
# dst - адреса призначення
# dttl – кількість вузлів на шляху до адреси призначення
# a, b – пайти приховуваного повідомлення
def send_stegano(src, src_nat, dst, dttl, a, b):
    # формування корисного навантаження з двох байт
    payload = ord(a)*0x100 + ord(b)
    # створення стану пакету під час проходження фінального вузла
маршруту
    pkt = IP(dst=dst, src=src_nat, ttl=64-dttl, id = payload) /
ICMP(type=8)
    # обчислення chksum з використанням Scapy
    pkt = IP(raw(pkt))
    # підготовка пакету до відправки
    pkt[IP].src = src
    pkt[IP].ttl = 64
    pkt[IP].id = pkt[IP].chksum
    # зтирання поля chksum, для подальшого розрахунку Scapy    del
pkt[IP].chksum
    # наступне обчислення Scapy значення контрольної суми
    pkt = IP(raw(pkt))
    # відправка пакету та очікування відповіді
    sr1(pkt)
```

ВИСНОВКИ

У відповідності до завдання на кваліфікаційну роботу, було виконано дослідження методів мережевої стеганографії, зокрема:

– визначено форми стеганографії та розглянуто класифікацію стеганографічних методів;

– розглянуто ключові вимоги до носіїв приховуваної інформації а також схему функціонування стеганографічної системи;

– виконано огляд методів мережевої стеганографії на прикладі систем HICCUPS, LACK та групи методів, які отримали умовну назву ICMP-стегонографії;

При цьому, виявлено, що:

1. Досліджені методи мережевої стеганографії реалізуються за рахунок:

– зміни змісту службових полів або полів даних мережевих пакетів;

– маніпулювання умовами надходження пакетів (зміна порядку надсилання, інтервалів і т.д.);

– внесення у потік даних навмисно пошкоджених пакетів;

– комбінування кількох або усіх зазначених вище підходів у рамках одного методу.

2. Підходи, орієнтовані на використання у мережі пакетів даних одного типу не є універсальними, так як у деяких випадках обмін даними на базі пакетів даних того чи іншого типу може бути не передбачено регламентом мережі.

3. Для умов реальної мережі, де можливість реалізації прихованої передачі на засадах існуючих методів (HICCUPS, LACK та подібні) обмежено (обмін пакетами аудіо відсутній за регламентом, пакети з помилками та черговість їх надходження перевіряються аналітичної системою), у найпростішому випадку може бути використано один з методів ICMP-стегонографії, а саме – маніпулювання довжиною інтервалу між сусідніми пакетами.

4. У рамках ICMP-стеганографії для умов. зазначених у попередньому пункті, прихований канал може бути побудовано шляхом внесення даних у службові поля пакетів, наприклад – поле ідентифікатору. Даний підхід потребує урахування кількості проміжних вузлів на ділянці між приймачем і передавачем, існування чи відсутності NAT у мережі а також з урахуванням усього переліченого – парерахунку контрольної суми для того, щоб модифікований пакет не було марковано як зіпсований та прийнято вузлом призначення.

Отже, усі пункти завдання кваліфікаційної роботи виконано у повні мірі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Шаньгін В.Ф. Захист інформації в розподілених корпоративних мережах і системах [Текст]: підручник / В.Ф. Шаньгин, А.В. Соколов. – М.: ДМК Пресс, 2002. – 656 с.
2. Fridrich Y. Steganography in Digital Media: Principles, Algorithms and Applicaticks. Cambridge Press, 2010. 462.
3. Конахович Г. Ф., Пузыренко А. Ю. Компьютерная стеганография. Теория и практика. - К.: МК-Пресс, 2006. - 288 с
4. Camouflage Home Page – Hide your Files! [Електронний ресурс] – Режим доступу: http://camouflage.unfiction.com.
5. audio-steganography · GitHub Topics · GitHub [Електронний ресурс] – Режим доступу: https://github.com/topics/audio-steganography?ysclid=m2bpz6c0fq423561912
6. Олифер В.Г., Олифер Н.А., Компьютерные сети. Принципы, технологии, протоколы (4-ое изд.) / В.Г. Олифер, Н.А. Олифер - Питер, 2010. - 943 стр
7. Network Steganography and its Techniques: A Survey [Електронний ресурс] – Режим доступу: https://www.ijcaonline.org/archives/volume174/number2/singh-2017-ijca-915319.pdf
8. Wojciech Mazurczyk, Paweł Szaga, Krzysztof Szczypiorski. Retransmission steganography and its detection. [Електронний ресурс] - Режим доступу: http://cygnus.tele.pw.edu.pl/~wmazurczyk/art/RSTEG.pdf
9. Wojciech Mazurczyk, Paweł Szaga, Krzysztof Szczypiorski. Using Transcoding for Hidden Communication in IP Telephony. [Електронний ресурс] - Режим доступу: http://arxiv.org/pdf/1111.1250v1.pdf.
10. K. Szczypiorski — HICCUPS: Hidden Communication System for Corrupted Networks — [Електронний ресурс] - Режим доступу: https://arxiv.org/ftp/arxiv/papers/1102/1102.0023.pdf
11. Mazurczyk W., Szczypiorski K. Steganography of VoIP streams.: In: MeersmanR, TariZ (eds) Springer-Verlag, 2009. [Електронний ресурс] - Режим доступу: http://home.elka.pw.edu.pl/~wmazurcz/moja/art/OTM_StegVoIP_2008.pdf
12. Stewart R., Ed. Stream Control Transmission Protocol. -RFC 4960–Request for Comments: 4960, 2007 [Електронний ресурс] - Режим доступу: http://tools.ietf.org/html/rfc4960.

13. W. Frączek, W. Mazurczyk, K. Szczypiorski. Stream Control Transmission Protocol Steganography. [Електронний ресурс] - Режим доступу: http://arxiv.org/abs/1006.0247.
14. 14. RFC 1700: Assigned Numbers. [Електронний ресурс] - Режим доступу: https://www.rfc-editor.org/rfc/rfc1700.html.
15. Steganography in TCP-IP Networks.pdf | K. Szczypiorski - Academia.edu [Електронний ресурс] - Режим доступу: https://www.academia.edu/22606031/Steganography_in_TCP_IP_Networks_pdf
16. Ciobanu, Radu-Ioan & Tirsa, Mihai-Ovidiu & Lupu, Raluca & Stan, Sonia & Andreica, Mugurel. (2011). SCONeP: Steganography and Cryptography approach for UDP and ICMP. 1-6. 10.1109/RoEduNet.2011.5993700.
17. GitHub - iputils/iputils at 33370345c7d8c217b51c13b0e2864b64b53d5f96 [Електронний ресурс] - Режим доступу: https://github.com/iputils/iputils/blob/33370345c7d8c217b51c13b0e2864b64b53d5f96.
18. Scapy [Електронний ресурс] - Режим доступу: https://scapy.net.
19. Python [Електронний ресурс] - Режим доступу: https://www.python.org/.