Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)

Кафедра Безпеки інформаційних технологій
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Мережева стеганографія на основі генерації ISN номерів
(тема)

Виконала: Костенюк Т.А.
(прізвище, ініціали)
студентка 2 курсу, групи БІКСм-19-1
Спеціальність 125 Кібербезпека
(код і повна назва спеціальності)
Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма «Безпека інформаційних і комунікаційних систем»
(повна назва освітньої програми)

Керівник проф. Руженцев В.І.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Халімов Г.З.
(підпис) (прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)

Кафедра Безпеки інформаційних технологій
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 125 Кібербезпека
(код і повна назва)

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

Освітня програма «Безпека інформаційних і комунікаційних систем»
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри
_____
(підпис)
«____»_____ 20 ___ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентці *Костенюк Тамарі Анатоліївні*
(прізвище, ім'я, по батькові)

1. Тема роботи *Мережева стеганографія на основі генерації ISN номерів* затверджена наказом по університету від "22" жовтня 2020 р. № 1412Ст
2. Термін подання студенткою роботи (проекту) 11.12.2020
3. Вихідні дані до роботи (проекту) Методи мережевої стеганографії, які базуються на моделі TCP / IP; документація стеку протоколів TCP / IP
4. Перелік питань, що потрібно опрацювати в роботі (зміст пояснювальної записки)
1. Аналіз засобів прихованої передачі даних в інформаційно-телекомуні-каційних мережах
2. Аналіз сучасного стану прихованої передачі інформації в інформаційно-телекомунікаційних мережах
3. Аналіз особливостей будови мережевої моделі TCP / IP
4. Аналіз методів стеганографічної передачі даних на базі мережевої моделі TCP / IP
5. Програмна реалізація та аналіз результатів
6. Порівняльний аналіз методів мережевої стеганографії
5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій *Презентаційний матеріал у вигляді слайдів*

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів магістерської атестаційної роботи	Термін виконання етапів роботи	Примітка
1	*Отримання завдання*	*07.09.20*	
2	*Аналіз завдання, огляд літератури*	*10.09.20-22.09.20*	
3	*Аналіз технічних засобів*	*23.09.20-31.09.20*	
4	*Аналіз алгоритму*	*01.10.20-10.10.20*	
5	*Програмна реалізація*	*12.10.20-5.11.20*	
6	*Оформлення пояснювальної записки*	*10.11.20-30.11.20*	
7	*Підготовка презентації та доповіді*	*01.12.20-10.12.20*	

Дата видачі завдання _____ _____20__ р.

Студентка _____
(підпис)

Керівник роботи (проекту) _____ __проф. Руженцев В.І._____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка атестаційної роботи: 65 с., 12 рис., 5 табл., 3 дод., 16 джерел посилань.

ДАНІ, ІНФОРМАЦІЯ, МЕРЕЖА, МЕТОД, ПОЛЕ, ПОСЛІДОВНІСТЬ, ПРИХОВАНА ПЕРЕДАЧА, ТЕЛЕКОМУНІКАЦІЯ, TCP / IP.

Об'єкт дослідження – інформаційно-телекомунікаційні мережі.

Предмет дослідження – методи захисту інформації в інформаційно-телекомунікаційних мережах на базі моделі TCP / IP.

Мета роботи – дослідження та реалізація методів захисту інформації, переданої по інформаційно-телекомунікаційних мережах на базі мережевої моделі TCP / IP, що дозволяють зберегти таємність повідомлення, що відправляють.

Методи дослідження – в роботі для вирішення зазначених завдань було використано об'єктно-орієнтоване програмування.

Для досягнення поставленої мети в роботі вирішені наступні завдання:

– виконано аналіз засобів прихованої передачі даних в інформаційно-телекомунікаційних мережах;

– проведено аналіз особливостей будови мережевої моделі TCP / IP;

– виконано аналіз методів стеганографічної передачі даних на базі стека протоколів TCP / IP;

– реалізовано та перевірено на коректність програмний продукт на основі одного із алгоритмів для прихованої передачі в інформаційно-телекомунікаційних мережах на базі стека протоколів TCP / IP;

– проведено порівняльний аналіз серед де-яких методів мережевої стеганографії.

ABSTRACT

Explanatory note of attestation work: 65 pages, 12 figures, 5 tables, 3 appendices, 16 sources.

DATA, FIELD, INFORMATION, NETWORK, METHOD, SECRET TRANSMISSION, SEQUENCE, STEGANOGRAPHY, TELECOMMUNICATION, TCP / IP.

Object of research – information and telecommunication networks.

Subject of research – methods of information security in information and telecommunication networks based on the TCP / IP model.

The purpose of the work is to develop and investigate methods for protecting information transmitted over information and telecommunication networks based on the TCP / IP model, that allows to keep the confidentiality of a sending message.

Research methods – object-oriented programming was used to solve these problems.

To achieve this goal, the following tasks are solved:

– performed an analysis of applications for secret transmission of data in information and telecommunication networks;

– performed an analysis of features of a structure of network model TCP / IP;

– performed an analysis of methods of steganography for secret transmission of data based on the TCP / IP protocol stack;

– implemented and checked for correct software based on one of the algorithms for latent transmission in information and telecommunication networks based on the TCP / IP protocol stack;

– performed a comparative analysis among some methods of network steganography.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

ІТКМ – інформаційно-телекомунікаційна мережа

МС – мережева стеганографія

ОС – операційна система

П – повідомлення

СГС – стеганографічна система (стегосистема)

ШНМ – штучні нейронні мережі

ICMP – internet control message protocol

TCP / IP – transport control protocol / internet protocol

OSI – open system interconnection

RSTEG – retranslation steganography

TCP – transport control protocol.

ВСТУП

Поки існує інформація, і є необхідність в її обміну, стільки і існуватиме потреба в її прихованій передачі. Для уможливлення цього існує наука, яка займається прихованою передачею інформації і має назву стеганографія. Її використання для передачі прихованих даних само по собі не дає ідеальних показників, тому дуже часто стеганографію використовують разом із криптографією, для покращення та більш надійного захисту під час передачі таємної інформації.

Ці питання стосовно як криптографії, так і стеганографії вивчали та аналізували у різний час такі відомі дослідники, як Kutter M., Jordan F., Bossen F., К. Шеннон, Б. Шнайер, Н. А. Молдовян, А. А. Молдовян, Wojciech Mazurczyk, Milosz Smolarczyk, і інші.

На сьогодні одним із популярних видів передачі прихованої інформації є той, який використовує для цього інформаційно-телекомунікаційні мережі з використанням особливостей протоколів базової моделі мережевої взаємодії OSI. Тому разом із цим дуже поширеними стали і методи, котрі уможливлюють реалізацію даного виду передачі несанкціонованих даних. Дані методи мають назву мережева стеганографія. Відомим фактом є те, що протокол TCP є найпоширенішим в мережі інтернет. Причинами цього є відкритість його опису в [1], який існує у вигляді рекомендацій і не зобов'язує до одноманітною реалізації в різних операційних системах, а також, звичайно, надійність даного протоколу.

Проведений аналіз робіт з організації прихованих каналів в ІТКМ на базі моделі TCP / IP привів до висновку, який показує, що напрямок мережевої стеганографії не є широко вивченим. Взагалі методи мережевої стеганографії поділяються на де-кілька типів: модифікація даних в заголовках мережевих протоколів і в полях корисного навантаження пакетів, зміна структури передачі пакетів в тому чи іншому мережевому протоколі (іноді і в декількох відразу). Але

риса, яка об’єднує всі ці методи полягає в тому, що всі вони створюються із допомогою прихованих каналів передачі інформації в будь-якому відкритому каналі, в якому є якась надмірність. Отже, розробка і дослідження нових методів у даній області є актуальною темою на сьогоднішній день

Дані, які було отримано під час проведеного дослідження, дають змогу порівняти де-які методи прихованої передачі в ІТКМ, що в подальшому, враховуючи переваги та недоліки методів, має допомогти при виборі використання того чи іншого методу МС.

Ціль атестаційної роботи полягає в програмній реалізації одного із методів прихованої передачі даних в ІТКМ на базі мережевої моделі TCP / IP.

Створене програмне забезпечення дає можливість проводити прихований обмін конфіденційною інформацією за допомогою відкритих телекомунікаційних мереж на базі мережевої моделі TCP / IP.

Основний зміст роботи було опубліковано у матеріалах “Восьмої міжнародної науково-технічної конференції 2020 р” [2].

1 ЗАГАЛЬНИЙ ОПИС ПРИНЦИПУ МЕРЕЖЕВОЇ СТЕГАНОГРАФІЇ

1.1 Основні стеганографічні методи

Стеганографія (від грец. тайнопис) – це наука про приховану передачу інформації шляхом збереження в таємниці самого факту передачі.

На сьогоднішній день методи комп'ютерної стеганографії розвиваються за двома основними напрямками:

– методи, засновані на використанні спеціальних властивостей комп'ютерних форматів;

– методи, засновані на надмірності аудіо та візуальної інформації.

Нижче у Таблиці 1 наведено характеристики основних методів та їх порівняння.

Таблиця 1 –Характеристики стеганографічних методів та їх порівняння

Стеганографічні методи	Коротка характеристика методів	Недоліки	Переваги
1. Методи використання спеціальних властивостей комп'ютерних форматів даних			
1.1. Методи використання зарезервованих для розширення полів комп'ютерних форматів даних	Поля розширення є в багатьох мультимедійних форматах, вони заповнюються нульовий інформацією і не враховуються програмою	Низька ступінь скритності, передача невеликих обмежених обсягів інформації	Простота використання

Стеганографічні методи	Коротка характеристика методів	Недоліки	Переваги
1.2. Методи спецформатування текстових файлів:			
1.2.1. Методи використання відомого зміщення слів, речень, абзаців	Методи засновані на зміні положення рядків і розстановки слів у реченні, що забезпечується вставкою додаткових пробілів між	1. Слабка продуктивність методу, передача невеликих обсягів інформації 2. Низька ступінь скритності	Простота використання. Є опубліковане програмне забезпечення реалізації даного методу
1.2.2. Методи вибору певних позицій букв (нульовий шифр)	Без цензури – окремий випадок цього методу (наприклад, початкові літери кожного рядка утворюють повідомлення)		
1.2.3. Методи використання спеціальних властивостей полів форматів, які не відображаються на екрані	Методи засновані на використанні спеціальних "невидимих", прихованих полів для організації виносок і посилань (наприклад, використання чорного шрифту на чорному тлі)		
1.3. Методи приховування в невикористовуваних місцях гнучких дисків	Інформація записується в зазвичай невикористовуваних місцях ГМД (наприклад, в нульовий доріжці)	1. Слабка продуктивність методу, передача невеликих обсягів інформації 2. Низька ступінь скритності	Простота використання. Є опубліковане програмне забезпечення реалізації даного методу

Стеганографічні методи	Коротка характеристика методів	Недоліки	Переваги
1.4. Методи використання імітуючих функцій (mimic-function)	Метод заснований на генерації текстів і є узагальненням акровірша. Для таємного повідомлення генерується осмислений текст, що приховує саме повідомлення	1. Слабка продуктивність методу, передача невеликих обсягів інформації 2. Низька ступінь скритності	Результуючий текст не є підозрілим для систем моніторингу мережі
1.5. Методи видалення ідентифікуючий файл заголовка	Приховуване повідомлення шифрується і у результаті видаляється ідентифікуючий заголовок, залишаючи тільки шифровані дані. Одержувач заздалегідь знає про передачу повідомлення і має недостатній заголовок	Проблема приховування вирішується тільки частково. Необхідно заздалегідь передати частину інформації одержувачу	Простота реалізації.
2. Методи використання надмірності інформації			
2.1. Методи використання надмірності цифрових фотографії, цифрового звуку і цифрового відео	Молодші розряди цифрових відліків містять дуже мало корисної інформації. Їх заповнення додатковою інформацією практично не впливає на якість сприйняття, що і дає можливість приховування конфіденційної інформації	За рахунок введення додаткової інформації спотворюються статистичні характеристики цифрових потоків. Для зниження компрометуючих ознак потрібна корекція статистичних характеристик	Можливість прихованої передачі великого обсягу інформації. Можливість захисту авторського права, прихованого зображення товарної марки, і т.

1.2 Основні принципи мережевої стеганографії

На відміну від криптографії, яка приховує вміст секретного повідомлення, стеганографія приховує факт передачі повідомлення, який сам по собі може мати вирішальне значення. Так як в даній роботі було розглянуто лише один із типів стеганографії, а саме мережева, то доцільно буде спочатку детально розглянути базові визначення, що стосуються саме цього типу.

Так само у МС використовуються такі визначення із як і класичної стеганографії, як[3]:

– повідомлення – впроваджуване потайним чином послання, яке необхідно заховати;

– контейнер (стеганоконтейнер) – будь-який об'єкт, який використовується для таємного впровадження повідомлення;

– стеганосистема – методи і засоби, що використовуються для створення прихованого каналу для передачі інформації;

– стеганоканал – канал для передачі стеганоконтейнер.

Головна відміна риса МС полягає утому, що це такий вид стеганографії, в якому в якості носіїв секретних даних використовуються мережеві протоколи еталонної моделі OSI – мережевої моделі взаємодії відкритих систем. У загальному вигляді мережева стеганографія є сімейством методів по модифікації даних в заголовках мережевих протоколах і в полях корисного навантаження пакетів, зміни структури передачі пакетів і гібридних методів в тому чи іншому мережевому протоколі (більш детальніше буде розглянуто пізніше).

Спільною рисою всіх методів МС є створення з їх допомогою прихованих каналів передачі інформації в будь-якому відкритому каналі, в якому є якась надмірність[4].

1.3 Класифікація мережевої стеганографії

Методи МС можна розділити на три групи (Рисунок 1) [5]:

– методи стеганографії, суть яких в зміні даних в полях заголовків мережевих протоколах і в полях корисного навантаження пакетів.

Плюсом даного методу є передача без змін інформації від відправника до одержувача, але це також обмежує кількість посилається інформації. Стеганографія на основі даного методу є легко реалізовується, має непогану пропускну здатність, так як можна послати безліч IP-пакетів з внесеними змінами і низьку вартість за рахунок застосування полів, що не порушують функціоналу пакету. Якщо вибрати маршрут так, щоб під час шляху пакет не була фрагментована, то даний метод може мати цілком широке застосування за рахунок вказаним них переваг. З недоліків слід виділити те, що передані дані містяться у відкритому вигляді і можуть бути легко прочитані спостерігачем (хоча можна посилити захист, використовуючи додатково криптографію);

– методи стеганографії, в яких змінюється структура передачі пакетів, наприклад, змінюється порядок передачі пакетів або навмисне введення втрат пакетів при їх передачі;

– змішані (гібридні) методи стеганографії – при їх застосуванні змінюються вміст пакетів, терміни доставки пакетів і порядок їх передачі.

Недоліком таких методів є те, що реалізація таких методів занадто складна, але може бути не можлива в межах деяких операційних систем.

Кожен з цих методів поділяється ще на кілька груп; наприклад, методи модифікації пакетів включають в себе три різні способи:

– методи зміни даних в полях заголовків протоколу: вони засновані на модифікації полів заголовків IP, TCP, SCTP і так далі;

– методи модифікації корисного навантаження пакета; в цьому випадку застосовуються різні алгоритми водяних знаків, мовних кодеків і інших стеганографічних технік по прихованню даних;

– методи змішаних технік.

Методи модифікації структури передачі пакетів включають в себе три напрямки:

– методи, в яких змінюється порядок послідовності пакетів;

– методи, що змінюють затримку між пакетами;

– методи, суть яких полягає у введенні навмисної втрати пакетів шляхом пропуску порядкових номерів у відправника.

Змішані (гібридні) методи стеганографії використовують два підходи: методи втрати аудіо пакетів (LACK) і ретрансляція пакетів (RSTEG)[6].

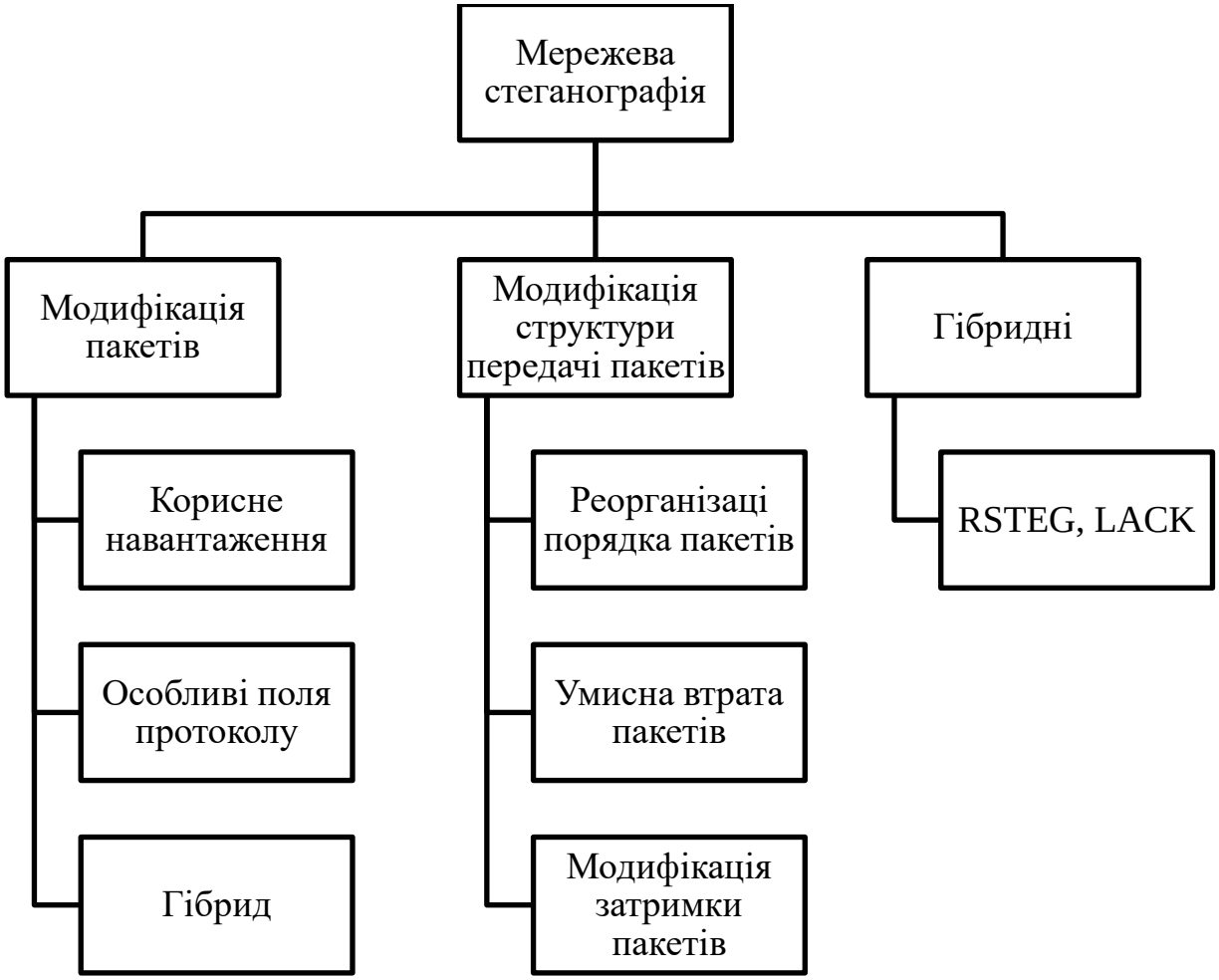

Рисунок 1 – Схема класифікації методів мережевої стеганографії

1.4 Постановка задачі

Хоча методи стеганографії відомі вже багато століть, в наш час, завдяки розвитку комп'ютерної техніки і телекомунікаційних технологій, отримав стрімкий розвиток новий вид прихованої передачі даних – МС. Із розвитком і

вдосконаленням технологій передачі інформації комп'ютерними мережами, з'являються нові різноманітні методи непомітної передачі інформації. Це відкриває великі перспективи для тих, хто хоче непомітно передавати повідомлення через будь-які кордони і створює небезпеку для установ, що займаються захистом інформації від несанкціонованого витоку.

Метою роботи є висвітлення існуючих на сьогодні методів мережевої стеганографії, їх принципів роботи та пошук можливих областей використання різних методів прихованої передачі на основі стеку протоколів TCP/IP, в залежності від виявлених переваг та недоліків цих методів. Враховуючи той факт, що МС є досить не повністю дослідженою областю, то має сенс розгляд одного із обраних методів, а саме на основі генерації IS-номерів у SYN-пакетах. Наша мета полягає також у практичній реалізації цього методу а також у подальшому аналізі отриманих практичних результатів, при тестуванні розробленого ПО. Окрім цього, також виявлення переваг та недоліків даного методу.

2 АНАЛІЗ ПРИНЦИПУ ФУНКЦІОНУВАННЯ СТЕКУ ПРОТОКОЛІВ TCP / IP

2.1 Загальне представлення мережевої моделі TCP / IP

TCP / IP – мережева модель передачі даних, представлених в цифровому вигляді. Модель описує спосіб передачі даних від джерела інформації до одержувача. У моделі передбачається проходження інформації через чотири рівні (прикладний. транспортний, міжмережевий, канальний), кожен з яких описується правилом (протоколом передачі)[7]. Набори правил, що вирішують завдання з передачі даних, складають стек протоколів передачі даних, на яких базується Інтернет[8, 9, 1].

Стек протоколів TCP / IP – набір мережевих протоколів, на яких базується Інтернет. Зазвичай в стеці TCP / IP верхні 3 рівня (прикладної, уявлення і сеансовий) моделі OSI об'єднують в один – прикладний. Оскільки в такому стеку не передбачено уніфікований протокол передачі даних, функції з визначення типу даних передаються з додатком[10]. Як можемо бачити із Таблиці 2.1 відмінність мережевої моделі TCP / IP від OSI 7 полягає в кількості рівнів, у еталонної моделі їх сім, в моделі стека протоколів їх чотири[3].

Таблиця 2.1 – Порівняння рівнів мережевої моделей OSI та TCP/IP

<table>
<tr><th>OSI</th><th colspan="2">TCP/IP</th></tr>
<tr><td>7. Прикладний</td><td rowspan="3">HTTP, FTP, Telnet, SMTP, DNS (RIP, що працює поверх UDP, і BGP, що працює поверх TCP, є частиною мережевого рівня), LDAP, RTP</td><td rowspan="3">4. Прикладний</td></tr>
<tr><td>6. Уявлення</td></tr>
<tr><td>5. Сеансовий</td></tr>
</table>

OSI	TCP/IP	
4. Транспортний	TCP, UDP, SCTP, DCCP (протоколи маршрутизації, подібні OSPF, що працюють поверх IP, є частиною мережевого рівня)	3. Транспортний
3. Мережевий	IP (допоміжні протоколи, типу ICMP і IGMP, але є частиною мережевого рівня; ARP не працює поверх IP)	2. Мережевий
2. Канальний	Ethernet, Token Ring, і подібні	1. Канальний
1. Фізичний		

Розглянемо далі більш детально, рівні моделі TCP / IP та їх особливості.

2.2 Прикладний рівень стеку TCP / IP

Прикладний рівень стека TCP / IP відповідає трьом верхнім рівням моделі OSI [3]: прикладного, уявлення і сеансовому. Він об'єднує послуги, що надаються системою користувальницьким додаткам. За довгі роки застосування в мережах різних країн і організацій стек TCP / IP нагромадив велику кількість протоколів і служб прикладного рівня [11]. До них відносяться такі поширені протоколи, як протокол передачі файлів FTP (File Transfer Protocol), протокол емуляції терміналу telnet, простий протокол передачі пошти SMTP (Simple Mail Transfer Protocol), протокол передачі гіпертексту HTTP (Hypertext Transfer Protocol) і багато інших . Протоколи прикладного рівня розгортаються на хостах.

HTTP (HyperText Transfer Protocol – протокол передачі гіпертексту) – символьно-орієнтований клієнт-серверний протокол прикладного рівня без збереження стану, який використовується сервісом World Wide Web.

Основним об'єктом маніпуляції в HTTP є ресурс, на який вказує URI (Uniform Resource Identifier – унікальний ідентифікатор ресурсу) в запиті клієнта.

Основними ресурсами є що зберігаються на сервері файли, але ними можуть бути і інші логічні (напр. Каталог на сервері) або абстрактні об'єкти (напр. ISBN). Протокол HTTP дозволяє вказати спосіб представлення (кодування) одного і того ж ресурсу за різними параметрами: mime-типу, мови і т. Д. Завдяки цій можливості клієнт і веб-сервер можуть обмінюватися двійковими даними, хоча даний протокол є текстовим.

Структура протоколу визначає, що кожне HTTP-повідомлення складається з трьох частин , які передаються в наступному порядку:

– стартовий рядок (англ. Starting line) – визначає тип повідомлення;

– заголовки (англ. Headers) – характеризують тіло повідомлення, параметри передачі та інші відомості;

– тіло повідомлення (англ. Message Body) – безпосередньо дані повідомлення. Обов'язково повинно відділятися від заголовків порожнім рядком.

FTP (англ. File Transfer Protocol – протокол передачі файлів) – протокол, призначений для передачі файлів в комп'ютерних мережах. FTP дозволяє підключатися до серверів цього протоколу і переглядати вміст каталогів, завантажувати файли з сервера або на сервер.

Формально це щось на зразок підключення до якоїсь папці, яка знаходиться на іншому комп'ютері/сервері, використовуючи мережу або інтернет. У разі, якщо передача файлу була перервана з яких-небудь причин, протокол передбачає кошти для докачки файлу, що буває дуже зручно при передачі великих файлів.

FTP є одним з найстаріших прикладних протоколів, що з'явилися задовго до HTTP, в 1971 році. Він і сьогодні широко використовується для розповсюдження програмного забезпечення і передачі файлів.

Telnet (англ. TErminaL NETwork) – протокол прикладного рівня, який використовується для реалізації двонаправленого інтерактивного текстового інтерфейсу в мережі через віртуальний термінал. Дані від користувача перемішуються з керуючими командами TelNet в восьмібітних байт-орієнтовані дані, що передаються по протоколу TCP. TelNet був розроблений в 1969 році.

Першою версією була RFC 15, далі розширена в RFC 854 і далі стандартизована в один з перших інтернет-стандартів IETF STD 8. Telnet забезпечує доступ до командного рядка операційної системи на віддаленому хості, підтримуючи більшість видів мережевого обладнання та операційних систем з утилітою конфігурації, проте через серйозні проблеми в безпеці використання Telnet у відкритій мережі (Інтернет), все частіше для цих цілей використовують протокол SSH. Однак, Telnet часто використовують також і для посилання на програмне забезпечення, що містить клієнтську частину протоколу, так як клієнтські програми Telnet доступні практично для всіх комп'ютерних платформ.

SMTP (Simple Mail Transfer Protocol – простий протокол передачі пошти) – стандартний, розроблений спеціально для поштових систем протокол, який поштова служба використовує, як засіб передачі повідомлення. Як і більшість інших протоколів прикладного рівня, SMTP реалізується несиметричними взаємодіючими частинами: SMTP-клієнтом і SMTP-сервером. Важливо відзначити, що цей протокол орієнтований на передачу даних у напрямку від клієнта до сервера. отже, SMTP-клієнт працює на стороні відправника, а SMTP-сервер – на стороні одержувача. SMTP-сервер повинен постійно бути в режимі підключення, чекаючи запитів з боку SMTP-клієнта.

Логіка роботи протоколу SMTP дійсно є досить простий (як це і випливає з його назви). Після того як, застосовуючи графічний інтерфейс свого поштового клієнта, користувач клацає на значку, ініціюванні відправку повідомлення, SMTP-клієнт надсилає запит на встановлення TCP-з'єднання на порт 25 (це призначений порт SMTP-сервера). Якщо сервер готовий, то він посилає свої ідентифікаційні дані, зокрема своє DNS-ім'я. Потім клієнт передає серверу адреси (імена) відправника і одержувача. Якщо ім'я одержувача відповідає очікуваному, то після отримання адрес сервер дає згоду на встановлення TCP-з'єднання, і в рамках цього надійного логічного каналу відбувається передача повідомлення. Використовуючи одне TCP-з'єднання, клієнт може передати кілька повідомлень, випереджаючи кожне з них зазначенням адрес відправника і одержувача. Після завершення передачі TCP- і SMTP-з'єднання розриваються.

Якщо на початку сеансу зв'язку SMTP-сервер виявився не готовий, то він посилає відповідне повідомлення клієнту, у відповідь той знову посилає запит, намагаючись заново встановити з'єднання. Якщо сервер не може доставити повідомлення, то він передає звіт про помилку відправнику повідомлення та розриває з'єднання. Після того як передача повідомлення благополучно закінчується, надіслане повідомлення буде зберігатись в буфері на сервері.

DNS (англ. Domain Name System – система доменних імен) – комп'ютерна розподілена система для отримання інформації про домени. Найчастіше використовується для отримання IP-адреси по імені хоста (комп'ютера або пристрою), отримання інформації про маршрутизації пошти і/або обслуговуючих вузлах для протоколів в домені (SRV-запис).

Розподілена база даних DNS підтримується за допомогою ієрархії DNS-серверів, що взаємодіють за певним протоколом.

Основою DNS є уявлення про ієрархічну структуру імені та зонах. Кожен сервер, який відповідає за ім'я, може передати відповідальність за подальшу частину домену іншого сервера (з адміністративної точки зору – інший організації або людині), що дозволяє покласти відповідальність за актуальність інформації на сервери різних організацій (людей), що відповідають тільки за свою частину доменного імені.

Починаючи з 2010 року в систему DNS впроваджуються засоби перевірки цілісності переданих даних, звані DNS Security Extensions (DNSSEC). Передані дані не шифруються, але їх достовірність перевіряється криптографічними способами. Впроваджуваний стандарт DANE забезпечує передачу засобами DNS достовірної криптографічного інформації (сертифікатів), які використовуються для встановлення безпечних і захищених з'єднань

DNS має наступні характеристики:

– розподіленість адміністрування. Відповідальність за різні частини ієрархічної структури несуть різні люди або організації;

– розподіленість зберігання інформації. Кожен вузол мережі в обов'язковому порядку повинен зберігати тільки ті дані, які входять в його зону відповідальності, і (можливо) адреси кореневих DNS-серверів;

– кешування інформації. Вузол може зберігати деяку кількість даних не зі своєї зони відповідальності для зменшення навантаження на мережу;

– ієрархічна структура, в якій всі вузли об'єднані в дерево, і кожен вузол може або самостійно визначати роботу нижчестоящих вузлів, або делегувати (передавати) їх іншим вузлам;

– резервування. За зберігання та обслуговування своїх вузлів (зон) відповідають (зазвичай) декілька серверів, розділені як фізично, так і логічно, що забезпечує збереження даних і продовження роботи навіть у разі збою одного з вузлів.

2.3 Мережевий рівень стеку TCP / IP

Протоколи мережевого рівня TCP/IP забезпечують взаємодію мереж різної архітектури тощо[11]. Основним протоколом мережевого рівня технології TCP/IP є міжмережевий протокол IP та його допоміжні протоколи: адресний протокол ARP реверсний адресний протокол RARP (Reverse ARP); протокол діагностичних повідомлень ICMP (Internet Control Message Protocol), який надсилає повідомлення вузлам мережі про помилки на маршруті, які виникають при передачі пакетів тощо.

Протокол ICMP не є протоколом орієнтованим на з'єднання, тобто при втраті пакету ICMP не буде робити ніяких спроб по його відновленню.

Пакет ICMP складається з наступних частин (Таблиця 2.2) [8]:

– заголовок: перший байт визначає тип пакету, другий – код операції, третій і четвертий є контрольну суму, де-які дані заголовку;

– область даних з довжиною залежною від типу пакета і його функції.

Таблиця 2.2 – Складові частини ICMP-пакету

Біти	0-7	8-15	16-31
ICMP –заголовок (8 байт)	Тип пакету	Код операції	Контрольна сума
	Дані заголовка		
ICMP – область даних (по-різному)	Корисне навантаження		

Головне завдання мережевого протоколу IP – це маршрутизація пакетів даних між різнотипними комп'ютерними мережами. Для розв'язання цього завдання протокол IP підтримує IP-адресацію мереж та вузлів, використовує таблицю маршрутизації пакетів, виконує, за необхідності, фрагментацію та дефрагментацію цих пакетів.

Нижче на Таблиці 2.3 можливо детально побачити складові IPv4. Отже, він складається із з даних верхнього рівня та IP-заголовку. За специфікацією протоколу [9], пакет має бути не більший за 65535:

– версія (Version) – 4-бітове поле, що описує використовувану версію протоколу IP. Всі пристрої зобов'язані використовувати протокол IP однієї версії, пристрій що використовує іншу версію буде відкидати пакети;

– довжина IP-заголовку (IP header Length – HLEN) – 4-бітове поле, що описує довжину заголовку пакету в 32-бітових блоках. Це значення – це повна довжина заголовку з врахуванням двох полів змінної довжини;

– тип обслуговування (Type of Service – TOS) – 8-бітове поле, що вказує на ступінь важливості інформації, що привласнена протоколом верхнього рівня;

– загальна довжина (Total Length) – 16-бітове поле, що описує довжину пакету в байтах, із заголовком та даними включно. Для того щоб вирахувати довжину блока даних, потрібно від повної довжини відняти значення поля HLEN;

– ідентифікація (Identification) – 16-бітове поле, що зберігає ціле число, яке описує даний пакет. Це число являє собою послідовний номер;

– прапорці (Flags) – 3-бітове поле, в якому два молодших біта контролюють фрагментацію пакетів. Перший біт визначає чи було пакет фрагментовано, а другий чи є цей пакет останнім фрагментом в серії фрагментів;

– зміщення фрагментації (Fragment Offset) – 13-бітове поле, що допомагає зібрати разом фрагменти пакетів. Це поле дозволяє використовувати 16 бітів в сумі для прапорів фрагментації;

– час життя (Time-to-Live – TTL) – 8-бітове поле – лічильник, в якому зберігаються послідовно зменшуване значення кількості пройдених вузлів (роутерів, що їх ще іноді в цьому випадку називають хопами (hops)) на шляху до місця призначення. У випадку коли лічильник пройдених хопів дорівнюватиме нулю – пакет відкидається, таким чином унеможливлюється нескінченна циклічна пересилка пакетів;

– протокол (Protocol) – 8-бітове поле, що вказує на те, який протокол верхнього рівня отримає пакет, після завершення обробки IP-протоколом. Наприклад TCP або UDP;

– контрольна сума заголовку (Header Checksum) – 16-бітове поле, що допомагає перевірити цілісність заголовку пакету;

– IP-адреса відправника (Source IP address) (адресант, сорс, відправник) – 32-бітове поле, що зберігає IP-адресу вузла-відправника;

– IP-адреса отримувача (Destination IP adress) (адресат, дест, отримувач) – 32-бітове поле, що зберігає адресу вузла призначення (отримувача) ;

– опції (Options) – поле змінної довжини, що дозволяє протоколу IP реалізувати підтримку різних опцій, зокрема засобів безпеки;

– підкладка (Padding) – поле, що використовується для вставки додаткових нулів, для гарантування кратності IP-заголовку 32 бітам;

– дані (Data) – поле змінної довжини (64 Кбіт макс.), що зберігає інформації для верхніх рівнів.

Таблиця 2.3 – Складові частини IP-пакету

<table>
<tr><td>Біти 0-3</td><td>4-7</td><td>8-15</td><td>16-18</td><td>19-23</td><td>24-31</td></tr>
<tr><td>Версія (Version)</td><td>HLEN (IP header length)</td><td>Тип обслуговування (TOS)</td><td colspan="3">Загальна довжина (Total Length)</td></tr>
<tr><td colspan="3">Ідентифікатор (Identification)</td><td>Прапорці (Flags)</td><td colspan="2">Зміщення фрагментації (Fragment Offset)</td></tr>
<tr><td colspan="2">Час життя (TTL)</td><td>Протокол (Protocol)</td><td colspan="3">Контрольна сума заголовку</td></tr>
<tr><td colspan="6">IP-адреса відправника (Source IP address)</td></tr>
<tr><td colspan="6">IP-адреса отримувача (Destination IP address)</td></tr>
<tr><td colspan="5">Опції (Options)</td><td>Додаток (Padding)</td></tr>
<tr><td colspan="6">Дані (65535 мінус заголовок) (Data)</td></tr>
</table>

2.4 Транспортний рівень стеку TCP / IP

Третій рівень моделі TCP / IP має таку ж назву, як і в моделі OSI – Транспортний рівень, але в моделі OSI цей рівень йде четвертим. TCP (Протокол керування передачею) – разом із протоколом IP є стрижневим протоколом Інтернету, який дав назву моделі TCP / IP [1]. На відміну від іншого поширеного протоколу транспортного рівня UDP, TCP забезпечує надійну передачу даних від хоста-відправника до хоста-отримувача, для цього встановлюється логічний зв'язок між хостами. Таким чином TCP належить до класу протоколів зі встановленим з'єднанням.

TCP-сегмент складається із TCP-заголовка і поля Дані (Data), яке називають сегментом даних або пейлодом або SDU [1].

Стандартний розмір TCP-заголовка – 20 байт, але з використанням опцій розмір може зростати до 60 байт. Як правило, опціями хости обмінюються на етапі встановлення з'єднання.

Розмір сегменту даних (поля даних) визначається опцією MSS (Максимальний розмір сегменту, Maximum segment size) на етапі встановлення з'єднання. Якщо обміну опціями не відбулося, то розмір сегменту даних встановлюється за замовчуванням 536 байт. Розмір сегменту даних тісно пов'язаний з MTU (Максимальний блок передачі). Фактично MSS дорівнює MTU з відніманням розміру IP- і TCP-заголовків[10].

Заголовок складається з наступних частин:

– порт джерела – 0-15 біти. Порт джерела (Source port) ідентифікує номер TCP-порту, з якого відправляється сегмент;

– порт призначення – 16-31 біти. Порт призначення (Destination port) ідентифікує номер TCP-порту, на який відправляється сегмент;

– номер послідовності – 32-63 біти. Номер послідовності (Sequence number) є числом, що відображає номер першого байту в сегменті надісланих даних від хоста-відправника до хоста-отримувача. Це число є акумулювальним, тобто поточний номер послідовності є сумою номеру послідовності попереднього сегменту і кількості даних (в байтах) відправлених у ньому. Використовується для відстеження кількості та правильної послідовності отриманих сегментів даних;

– номер підтвердження – 64-95 біти. Номер підтвердження (Acknowledgment number) фактично є запитом від хоста отримувача на надіслання нового сегменту даних починаючи зі вказаного номера. З іншого боку, коли хост відправник отримує це П, він переконується, що всі сегменти даних з номерами послідовності меншими за номер підтвердження були успішно прийняті отримувачем;

– зміщення даних – 96-99 біти. Зміщення даних 4-бітний номер, який визначає розмір TCP-заголовка в 32-бітових словах. Мінімальний розмір

становить 5 (0101) слів, а максимальний – 15 (1111), що є відповідно 20 і 60 байт. Фактично визначає розмір поля Опції (Options) від 0 до 40 байт;

– зарезервовано – 100-102 біти, зарезервовані для майбутнього використання і повинні містити нулі (000);

– прапорці (керуючі біти) – це поле містить бітові прапорці, з яких шість основних описані в RFC 793 з 106 по 111 біт, два прапорці додані до заголовка в RFC 3168, розміщуються в 104 і 105 бітах заголовка, в 103 біті знаходиться експериментальний прапорець згідно з RFC 3540. Прапорці вважається встановленими, якщо їх бітове значення є 1.

Основні прапорці:

а) 106 URG – Важливість (Urgent), вказує, що TCP-сегмент містить важливі дані. Коли до хоста-отримувача надходить сегмент зі встановленим прапорцем URG, TCP відправляє важливі дані з цього сегменту, які знаходяться завдяки полю покажчик важливості до відповідного протоколу верхнього рівня минаючи чергу і без перевірки успішності надходження попередніх сегментів;

б) 107 ACK – Підтвердження (Acknowledge) успішності отримання TCP-сегменту;

в) 108 PSH – Просування (Push), також як і прапорець URG, вказує, на пріоритетність TCP-сегменту. Хост-відправник позачергово надсилає цей сегмент даних через IP-мережу. За аналогією з прапорцем URG, PSH інструктує хост-отримувач, що сегмент даних має бути негайно переданий до прикладного рівня (кінцевого споживача даних);

г) 109 RST – Обривання (Reset) вказує, хосту-отримувачу негайно скинути з'єднання без подальшої взаємодії. Така ситуація наступає у разі, якщо сервер (хост-відправник) не надає послуги визначеного сервісу;

д) 110 SYN – Синхронізація (Synchronize) використовується для встановлення з'єднання між хостами при так званому триходовому рукостисканні;

е) 111 FIN – Фініш (Finish) вказує на завершення з'єднання;

– розмір вікна – 112-127 біти. Розмір вікна (англ. Window Size) визначає кількість байтів даних, які відправник може надіслати до того, як отримає підтвердження (запит на новий сегмент) від хоста-отримувача. На практиці це означає, що хост-відправник може надсилати певну кількість сегментів даних без отримання підтвердження від хоста-отримувача. Розмір вікна TCP вираховує на основі максимальної пропускної здатності (англ. bandwidth) лінії зв'язку між хостами (фактично це є пропускна здатність відрізка шляху з її найгіршим значенням) та загальній затримці (англ. latency) (часу потрібному на доставку сегмента) на всьому шляху;

– контрольна сума – 128-143 біти. Контрольна сума (Checksum) розраховується на основі усього TCP-сегменту включно із заголовком та важливих полів IP-пакету: IP-адрес хостів відправника та отримувача, номеру протоколу (TCP має номер 6) та загального розміру IP-пакету. Контрольна сума забезпечує можливість перевірки цілісності надісланих даних;

– покажчик важливості – 144-159 біти. Покажчик важливості (Urgent pointer). Поле береться до уваги тільки в разі встановленого прапорця URG, та містить значення зміщення відносно номеру послідовності сегменту. Фактично це число вказує на позицію в TCP-сегменті де закінчуються важливі дані.;

– опції – 160-479 біти. Опції (Options) необов'язкове поле, розмір якого визначається в залежності від значення поля зміщення даних та є кратним 8 (одному байту). Кожна опція в свою чергу складається з 3-х полів: Номер (kind) – 1 байт, Довжина (length, вказує на загальний розмір опції в байтах) – 1 байт, Дані (data) в залежності від поля довжина. Опції використовується для обміну додаткових параметрів між хостами з метою покращення функціонування протоколу TCP. Частіше за все це поле включає наступні опції:

а) MSS (Максимальний розмір сегменту, Maximum segment size), номер – 2, довжина – 4. Опція максимальний розмір сегменту визначає максимальний розмір поля Дані в TCP-сегменті тобто кількість даних які можуть бути поміщені в один сегмент при їх передачі між хостами;

б) масштабування вікна (Window scale),номер – 3, довжина – 3, слугує для збільшення значення TCP-вікна, максимальне значення цієї опції є 14;

в) вибіркові підтвердження (Selective Acknowledgments, SACK), RFC 2018, номер – 2, довжина – від 4 байт – верхня межа варіюється, як правило містить у собі два 2-х байтних поля даних. Мета її введення є покращення ефективності роботи TCP, як відомо TCP для передачі сегментів даних використовує протокол IP, який є протоколом без встановлення з'єднання, тобто доставка пакетів з TCP-сегментами не є гарантованим, допускається, що частина IP-пакетів може бути втрачена. В свою чергу TCP забезпечує надійне відправляння даних, що базується на механізмі надсилання номерів підтвердження (acknowledgment number). Якщо якийсь сегмент від відправника до отримувача не надійшов у встановлений час то ініціюється повторна передача починаючи зі втраченого сегменту, навіть якщо TCP-сегменти з номерами послідовності більшими за номер втраченого сегменту були успішно отримані. Механізм вибіркового підтвердження дозволяє ретранслювати лише втрачені сегменти даних, чим суттєво покращує ефективність роботи TCP;

– мітки часу (Timestamps), номер – 8, довжина – 10, містить у собі два 4-х байтних поля Значення мітки часу (Timestamp Value) та Exo-відповідь мітки часу (Timestamp Echo Reply). Як правило хости обмінюються значеннями міток часу на етапі встановлення з'єднання. За допомогою міток часу TCP визначає скільки потрібно часу на доставку сегментів між хостами. На основі цих значень встановлюються TCP-таймери відповідальні на стороні хоста-відправника за повторну передачу даних, якщо підтвердження отримання не надійшло у встановлений час, а у разі використання опції вибіркового підтвердження хост-отримувач самостійно ініціює запит на повторну передачу конкретного сегменту даних.

Якщо деякий простір поля Опції лишається незаповненим то він заповнюється спеціальною опцією NOP (No-Operation, нічого не робити).

Тепер можемо розглянути більш детально етапи TCP-сесії.

Головною задачею протоколу, як випливає з його назви: “протокол керування передачею”, є контроль надійної передачі даних між хостами, для забезпечення цього між хостами встановлюється логічне з'єднання, яке зветься TCP-сесією. Протокол TCP працює у форматі архітектури клієнт-сервер. Хост який надсилає запит на отримання сервісу є клієнтом, той хто відповідає на запит зветься сервером. Хости зв'язуються один з одним за TCP-портами, на стороні клієнта це, як правило, динамічний порт, а на стороні сервера це загальновідомий або зареєстрований порт, номер якого відповідає протоколу прикладного рівня. Номери портів та значення інших параметрів заносяться до заголовка TCP-сегменту.

Тут під терміном сервер (server) розуміють комп'ютер під управлінням ОС, який має доступ до IP-мережі та надає послуги одного чи декількох сервісів прикладного рівня. В свою чергу на сервері-комп'ютері встановлені спеціальні сервер-програми, які забезпечують роботу протоколів прикладного рівня.

Важливо розуміти, що протягом TCP-сесії дані надсилаються в обох напрямках, як від сервера до клієнта так і від клієнта до сервера, тобто створюються два потоки даних. Причому не завжди більший потік даних прямує від сервера до клієнта.

Кожна окрема сесія роботи протоколу TCP може бути поділена на три фази:

– Встановлення з'єднання (триходове рукостискання).

Необхідною умовою для встановлення TCP-сесії є відкритий доступ до програмного сокету процесу на сервері, що відповідає за роботу протоколу прикладного рівня, який мовою TCP зветься портом. За такої умови стан TCP-сесії на стороні сервера є LISTEN (слухати). Тобто сервер слухає, якийсь конкретний TCP-порт і очікує отримати запит від клієнта на надання послуг, що відповідають номеру цього порту. Початковий стан TCP-сесії на стороні клієнта є CLOSED.

Для встановлення з'єднання протокол TCP використовує триходове рукостискання, назване так за кількістю П між хостами:

1) клієнт формує TCP-заголовок: у поле порт джерела заносить свій номер TCP-порта, як правило динамічний, у поле порт призначення номер порту протоколу прикладного рівня, послуги якого хоче отримати, в поле номер послідовності сегменту довільне значення та встановлює прапорець SYN. Сформований таким чином TCP-сегмент відправляється серверу. TCP-сесія на стороні клієнта переходить у стан SYN-SENT.

2) сервер отримує TCP-сегмент від клієнта зі встановленим прапорцем SYN та у відповідь формує TCP-заголовок: у поле порт джерела заносить свій номер TCP-порта, у поле порт призначення номер порту клієнта. Додає до отриманого від клієнта номера послідовності сегменту 1 і поміщає отримане число до номеру підтвердження, вносить свій власний початковий номер послідовності сегменту та відправляє сегмент до клієнта з прапорцями SYN та ACK. TCP-сесія на стороні сервера переходить у стан SYN-RECEIVED.

3) після отримання TCP-сегмента зі встановленими прапорцями ACK та SYN клієнт переходить у стан ESTABLISHED та відповідає на запит сервера про синхронізацію шляхом додавання 1 до номера отриманої послідовності сегменту та поміщає це число до номера підтвердження. Далі клієнт надсилає таким чином сформований сегмент до сервера з прапорцем ACK.

Після отримання сервером TCP-сегмента з прапорцем ACK стан TCP-сесії на його стороні стає також ESTABLISHED, разом з чим розпочинається передача даних.

Також на етапі встановлення з'єднання між хостами, як правило відбувається обмін опціями, тобто TCP-параметрами, які впливають на ефективність передачі даних.

– Передача даних.

– Закінчення з'єднання.

Ініціатором закінчення з'єднання може бути, як клієнт так і сервер. Для закінчення TCP-сесії використовується так зване чотириходове рукостискання:

1) ініціатор розірвання з'єднання направляє своєму партнеру TCP-сегмент зі встановленим прапорцем FIN. TCP-сесія ініціатора переходить зі стану ESTABLISHED у стан FIN-WAIT-1;

2) хост-отримувач приймає FIN від ініціатора та посилає у відповідь TCP-сегмент зі встановленим прапорцем ACK. TCP-сесія отримувача переходить зі стану ESTABLISHED у стан CLOSE-WAIT. З набуттям хостом цього стану TCP припиняє отримувати нові запити, на передачу даних, від відповідного протоколу верхнього рівня та встановлює таймер на завершення попередніх запитів. Ініціатор отримує ACK та переходить у стан FIN-WAIT-2;

3) після закінчення оброблення всіх запитів протоколів верхнього рівня хост-отримувач переходить у стан LAST-ACK та відправляє ініціатору TCP-сегмент зі встановленим прапорцем FIN;

4) ініціатор приймає FIN від отримувача та посилає у відповідь TCP-сегмент зі встановленим прапорцем ACK. Отримувач приймає ACK та переходить у стан CLOSED.

Ініціатор розірвання з'єднання чекає протягом подвійного часу від MSL (максимального життя сегмента, maximum segment lifetime), щоб переконатися, що посланий ACK був отриманий та також переходить у стан CLOSED.

2.5 Канальний рівень стеку TCP / IP

Нагадаємо, що нижні рівні моделі OSI (канальний і фізичний) реалізують безліч функцій доступу до середовища передачі, формування кадрів, узгодження величин електричних сигналів, кодування і синхронізації, а також деякі інші. Всі ці вельми конкретні функції складають суть таких протоколів обміну даними, як Ethernet, PPP і багатьох інших.

У нижнього рівня стека TCP / IP завдання істотно простіше – він відповідає тільки за організацію взаємодії з підмережами різних технологій, що входять в складену мережу. TCP / IP розглядає будь-яку підмережу, що входить в складену

мережу, як засіб транспортування пакетів між двома сусідніми маршрутизаторами.

Завдання організації інтерфейсу між технологією TCP / IP і будь-який інший технологією проміжної мережі спрощено можна звести до двох завдань:

– упаковка (інкапсуляція) IP-пакета в одиницю переданих даних проміжної мережі;

– перетворення мережевих адрес в адреси технології даної проміжної мережі.

Такий гнучкий підхід спрощує вирішення проблеми розширення набору підтримуваних технологій. З появою нової популярної технології вона швидко включається в стек TCP / IP шляхом розробки відповідного стандарту, що визначає метод інкапсуляції IP-пакетів в її кадри (наприклад, специфікація RFC 1577, що визначає роботу протоколу IP через мережі ATM, з'явилася в 1994 році незабаром після прийняття основних стандартів ATM). Так як для кожної знову з'являється технології розробляються власні інтерфейсів засоби, функції цього рівня не можна визначити раз і назавжди, і саме тому нижній рівень стека TCP / IP не регламентується.

Протокол Ethernet – відкритий промисловий мережевий стандарт, який підтримує неявний обмін повідомленнями (обмін повідомленнями введення / виведення в реальному часі), явний обмін (обмін повідомленнями) або обидва і використовує широко поширені комерційні чіпи зв'язку Ethernet і фізичні носії. Оскільки технологія Ethernet використовується з середини 1970-их і широко практикується в усьому світі, то продукти Ethernet підтримує велику кількість постачальників.

Token Ring – протокол передачі даних в локальної обчислювальної мережі з топологією кільця і з маркерним доступом. Станції в локальної обчислювальної мережі Token Ring логічно організовані в кільцеву топологію, з даними, переданими послідовно від однієї станції в кільці до іншої. Token Ring використовує спеціальний трьохбайтовий блок даних, званий маркером, який так

само переміщається по кільцю. Володіння маркером надає його власнику право передавати дані.

Якщо у станції, що володіє маркером, є інформація для передачі, вона захоплює маркер, змінює у нього один біт (в результаті чого маркер перетворюється в послідовність «початок блоку даних»), доповнює інформацією, яку він хоче передати, і відсилає цю інформацію до наступної станції кільцевої мережі. Коли інформаційний блок циркулює по кільцю, маркер в мережі відсутня (якщо тільки кільце не забезпечує раннього звільнення маркера – early token release), тому інші станції, які бажають передати інформацію, змушені очікувати. Якщо забезпечується раннє вивільнення маркера, то новий маркер може бути випущений після завершення передачі блоку даних.

3 АНАЛІЗ МЕТОДУ СТЕГАНОГРАФІЧНОЇ ПЕРЕДАЧІ ДАНИХ В ІНФОРМАЦІЙНО-ТЕЛЕКОМУНІКАЦІЙНІЙ МЕРЕЖІ НА ОСНВІ ГЕНЕРУВАННЯ ISN

3.1 Метод на основі генерації ISN

3.1.1 Теоретична складова методу

Одним із представників методів МС із модифікацією пакетів є метод на основі ISN-генерації. Його реалізація із практичного зводиться до двох завдань [12]:

1) формування пакету з необхідним вмістом в полях заголовка і корисного навантаження, що відрізняється від стандартних пакетів, що генеруються ОС;

2) виділення сформованого пакета із загального потоку трафіку приймаючою стороною і розпізнавання поміщених в нього стеганографічних П.

Треба також зазначити де-які випадки, в яких же випадках змінюються / не ISN[9]:

– передача одного FIN пакета = +1;

– передача одного SYN пакета = +1;

– передача одного ACK пакета = 0;

– передача одного SYN / ACK пакета = +1;

– передача одного FIN / ACK пакета = +1;

– зміна за 1 секунду = +128,000;

– встановлення одного з'єднання = +64,000.

В основу запропонованого методу покладено механізм генерації початкового номера послідовності ISN кожного TCP з’єднання.

Відповідно [1], генерація номера заснована на поточному (можливо, фіктивному) 32-бітовому значенні часу, в якому молодший біт інкрементується приблизно кожні 4 мікросекунди. Насправді, значення ISN обчислюється в різних операційних системах по-різному, але в загальному випадку це значення є, свого роду, тимчасовим штампом і відповідає значенню функції, аргументом

якої є поточне значення машинного часу (апаратних годин конкретної операційної системи), тобто ISN = F(t). Отже, значення ISN, для стороннього спостерігача, при першому спостереженні, є випадковим. При аналізі достатньої вибірки значень F(t), стає можливим обчислити закономірність (апроксимувати функцію).

У заголовку TCP-фрагмента, значення ISN записується в поле довжиною 32 біта. Дані, що переносяться TCP-фрагментом, представляються у вигляді послідовності байтів. Номер першого переданого байта даних відповідає (ISN + 1).

Використовуючи в якості стеганоконтейнеру поля «Номер послідовності» TCP-фрагмента необхідно (Рисунок 3.1)[14]:

1. обчислити 8-розрядне значення для кодування;
2. побайтно заповнити 32-розрядний регістр;
3. перетворити отримане 32-розрядне значення в номер ISN;
4. розмістити номер ISN у відповідному полі заголовка SYN-фрагмента протоколу TCP;
5. розмістити опорний текст в якості корисного навантаження протоколу TCP;
6. передати стеганокотейнер по відкритому каналу передачі даних;
7. на приймальній стороні витягти стего в зворотній послідовності.

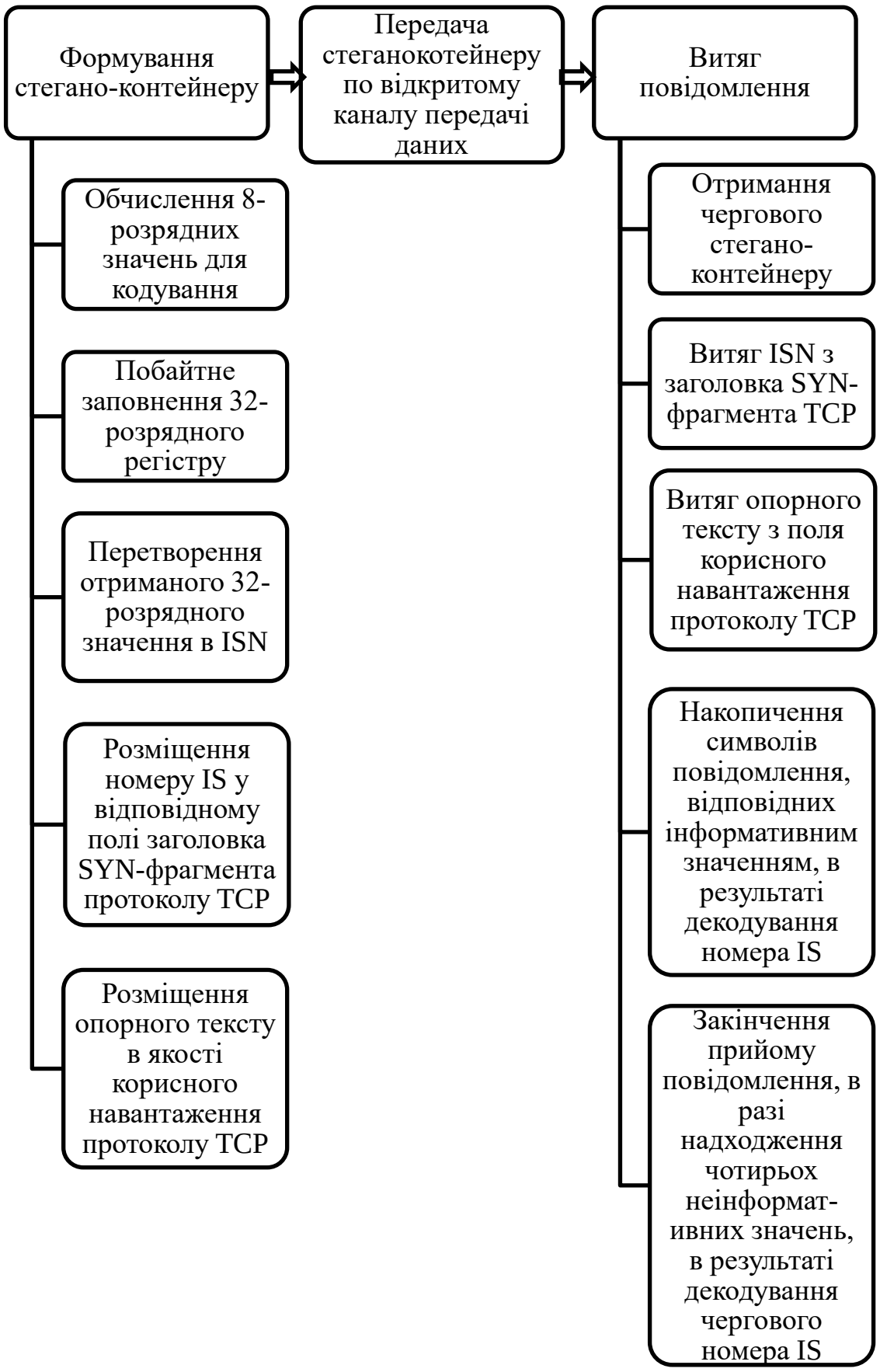

Рисунок 3.1 – Структурна схема моделі обробки TCP – з'єднань для стеганографічної передачі даних

3.1.2 Аналіз прикладу МС для практичної реалізації

Приклад створення стеганоконтейнеру для слова «leto» має наступний вигляд: «l» – 108_{10} – $6C_{16}$ – 0110 1100_2, «e» – 101_{10} – 65_{16} – 0110 0101_2, «t» – 116_{10} – 74_{16} – 0111 0100_2, «o» – 111_{10} – $6F_{16}$ – 0110 1111_2. Відповідно до логіки роботи описуваного методу, далі необхідно згенерувати потрібний номер ISN. У стеці протоколів TCP / IP заповнення заголовків і полів даних проводиться в порядку «Від старшого до молодшого». Потрібний початковий номер послідовності має вигляд: $01101100011001010111010001101111_2$ = 1818588271_{10}. На Рисунку 3.2 відображена схема встановлення TCP-з'єднання із зазначенням даних ISN та ACK для прикладу, який було наведено вище.

Звідси випливає, що при оголошенні такого ISN, перший байт TCP даних, що передаються, буде мати номер $(1818588271 + 1)_{10}$.

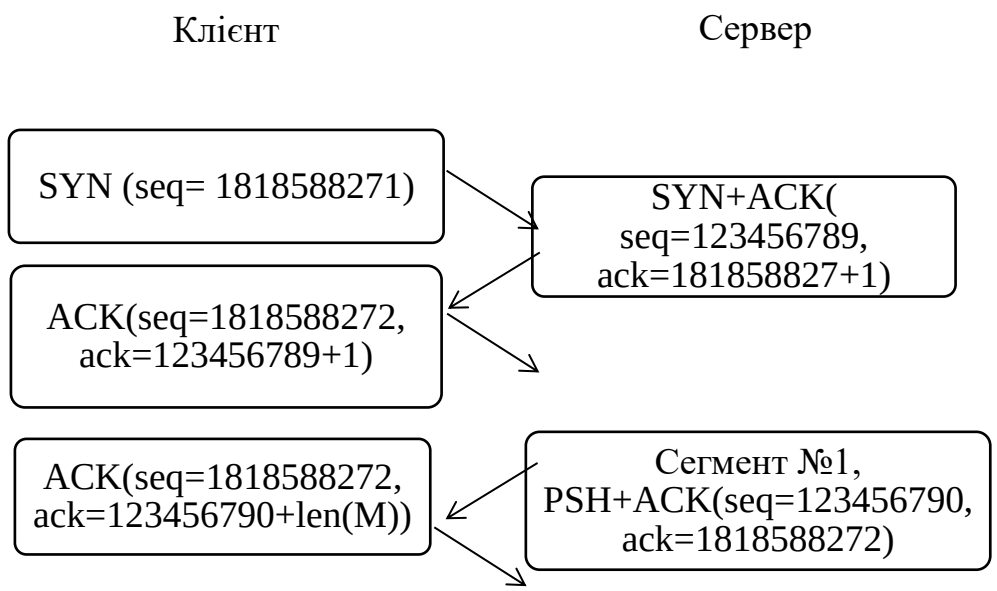

Рисунок 3.2 – Схема встановлення TCP-з'єднання із значеннями із прикладу

3.2 Метод на основі ICMP-інкапсуляції

Для сучасного мережевого обладнання та операційних систем, що використовують стек TCP / IP, підтримка протоколу ICMP обов'язкова [1], що є

його безперечною перевагою при використанні для передачі прихованих повідомлень. З цього факту випливають ще кілька переваг стеганографії на основі ICMP-інкапсуляції [15]. Одним з них є періодична розсилка ICMP-повідомлень вузлами мережі, яка відбувається при виявленні помилок в IP-адреси пакетів, відсутності маршрутів до запитуваних мереж і багатьох інших випадках, що призводить до великої кількості ICMP-пакетів, що містять службову мережеву інформацію, серед яких можна приховати пакети з будь-якої іншою інформацією.

Інша особливість протоколу ICMP, що відрізняє його від інших службових протоколів, і визначається довгою історією розвитку і великою кількістю виконуваних функцій – складність внутрішньої структури пакетів[8], яка призводить до наявності великого числа можливих поєднань полів заголовка і корисного навантаження.

Отже, практична реалізація даного методу зводиться до двох задач[12]:

– формування пакету з необхідним вмістом в полях заголовка і корисного навантаження, що відрізняється від стандартних пакетів, що генеруються операційною системою;

– виділення сформованого пакета із загального потоку трафіку приймаючою стороною і розпізнавання вбудованих в нього стеганографічних повідомлень.

На початковому етапі створюється ICMP-пакет. Для нього необхідно задати значення полів заголовка, описаних нижче.

TypeCode – числовий ідентифікатор типу повідомлення. Правильне заповнення цього поля потрібно тільки в випадку використання протоколу ICMP за призначенням, а для прихованої передачі даних воно не має ніякого значення, тому може бути встановлено, наприклад, в EchoRequest (тип 8, код 0) і EchoResponce (тип 0, код 0).

Sequence – номер пакета в послідовності, який необхідний для його ідентифікації в задачах, для яких потрібно відправка декількох пакетів. Дане поле необхідно використовувати при відправці довгого повідомлення, яке

розбивається на кілька коротших, кожне з яких поміщається в окремий ICMP-пакет зі своїм номером у послідовності, що задається цілим числом від 0 до 216 – 1.

Data – поле для запису службових даних, яке при реалізації стеганографічної системи буде містити в собі інформацію, передану приховано.

3.3 Метод RSTEG

Одним із представників гібридної МС є метод RSTEG. Він заснований на роботі надійного протоколу транспортного рівня TCP, зокрема на таймерах повторної передачі [6].

Він передбачає передачу прихованих даних усередині пакетів TCP \ IP протоколу, що відправляються нібито для виправлення невдало переданих даних[1].

Загальну роботу алгоритму можна уявити наступним чином. Існує приймач і відправник П. Все починається з легального обміну даними, який може тривати як завгодно довго. Але в певний момент часу, відправник вирішує передати стеганограму. В цьому випадку його дії наступні: він відправляє сегмент з легальними даними, на що відправник не відповідає сегментом з підтвердженням, внаслідок чого на стороні відправника закінчується таймер RTO, і він змушений повторно передати дані . У цей момент, до ретрансляції, в легальні дані вбудовується стеганограма, і даний сегмент відправляється. На що приймач знову не відповідає підтвердженням. Після чого у відправника вдруге закінчується RTO, і він ретранслює сегмент уже з початковими легальними даними, на що відправник відповідає підтвердженням. Загальна схема зображена нижче на Рисунку 3.3.

У зв'язку з тим що алгоритми RSTEG змінюють структуру передачі пакетів, а саме: навмисно не відповідає сегменту з встановленим бітом ACK на отримані дані, він тим самим змінює значення RTO. Дана умова може бути використано для стеганоаналізу.

Отже, метод RSTEG добре підходить для TCP / IP, і при розумному рівні навмисних ретрансляцій даний метод не повинен викликати підозр у спостерігача. Але даний метод досить складно реалізувати, особливо ті його алгоритми, які засновані на перехопленні і виправленні пакетів звичайних користувачів. Через різке зростання їх частоти ретранслюються пакетів або виникнення незвичайних затримок при передачі, стеганограми можуть викликати підозри у стороннього спостерігача.

Рисунок 3.3 – Схема загального принципу функціонування методу RSTEG

3.4 Аналіз методів виявлення каналу прихованої передачі в інформаційно-телекомунікаційній мережах

Існує два типи канали для створення несанкціонованої передачі інформації: за часом, із пам’яттю [5].

Для реалізації прихованого каналу першого типу необхідно реалізувати процедуру внесення змін в пакет даних. Даний тип реалізується відносно складніше, ніж більшість прихованих каналів передачі інформації із пам’яттю. Але звідси випливає його перевага, яка полягає в тому, що даний канал складно виявити без використання спеціальних методів. Нижче наведено такі методи виявлення прихованих каналів передачі інформації в ІТК мережах [16]:

– простий аналіз трафіку (наявність будь-яких підозрілих даних у полях пакетів, які одразу кидаються в очі, та інші);

– ймовірністно-статистичний аналіз. Він же в свою чергу поділяється на:

а) приховані канали із пам’яттю (згідно стандарту RFC–793 поле SN TCP-пакета повинне мати випадкове значення. У разі використання поля SN для кодування інформації у прихованому каналі, за допомогою ймовірністно-статистичного аналізу можливо виявити факт того, що випадкові значення не є таким);

б) приховані канали за часом (ймовірністно-статистичний аналіз затримок між пакетами дає змогу визначити інтервали часу для кодування логічної одиниці і логічного нуля поміж нормального розподілу значень часових затримок. У разі накопичення достатньо великого обсягу статистичної інформації детектуванню підлягають також приховані канали із застосуванням механізму сортування пакетів, оскільки велика ентропія значень часових затримок не є нормою в реальній мережі).

– система “процес-подія” (Process Query System, PQS, аналізує мережеву активність процесів в системі і формує зв'язки “процес-подія“. PQS складається із декількох елементів: потік подій, що спостерігається системою, сукупність моделей, які описують потенційний процес-генератор подій, алгоритми

відстеження, що аналізують події і визначають який процес їх створив, ядро PQS, що об'єднує попередні елементи і дає ймовірністну оцінку потоку подій);

– ШНМ (можуть використовуватися для виявлення багатьох типів прихований каналів, недоліком їх застосування є необхідність попереднього навчання та налаштування);

– нормалізатори трафіку (нормалізатори трафіку в процесі роботи виправляють поля заголовків IP- та TCP-пакетів у відповідності із стандартом, поля, які мають бути порожніми нормалізатор занулює. Всі некоректні пакети повинні відкидатися. Для протидії прихованим каналам за часом нормалізатор трафіку може вносити випадкові часові затримки для пакетів. Нормалізатори трафіку можуть використовуватись спільно з іншими методами виявлення прихованих каналів для більш ефективної роботи);

– поведінковий аналіз (метод поведінкового аналізу передбачає аналіз профілю використання мережі. Поведінковий аналіз проводиться в режимі реального часу і дозволяє виявляти незвичну поведінку в мережі (незвичний трафік, його характер і обсяг). Виявлення аномалій засноване на порівнянні поточного характеру трафіку із стандартним сценарієм, що заздалегідь був визначений як нормальний. Наприклад, аномалією є багаторазова відправка пакету з одним і тим самим SN. Перевагою поведінкового аналізу є робота аналізатора в режимі реального часу та незалежність роботи аналізатора від конкретної реалізації того чи іншого прихованого каналу. Недоліком даного методу є необхідність розробки системи правил для роботи аналізатора та необхідність попереднього дослідження мережевого середовища з метою складання стандартного сценарію).

Розглянувши актуальні методи виявлення прихованих каналів передачі інформації, можна зробити висновок, що жоден із методів не може забезпечити гарантованого захисту в мережі від несанкціонованого витоку інформації, але гібридна система, що об'єднує в собі статистичний метод, нормалізацію трафіку та одну із систем із здатністю самонавчання, може забезпечити прийнятний показник виявлення та низький рівень хибних спрацювань.

4 РОЗРОБКА ПРОГРАМИ СТЕГАНОГРАФІЧНОЇ ПЕРЕДАЧІ ДАНИХ В ІНФОРМАЦІЙНО-ТЕЛЕКОМУНІКАЦІЙНІЙ МЕРЕЖІ ЗА ДОПОМОГОЮ ПРОТОКОЛУ TCP / IP

4.1 Технології розробки

Для можливості коректної реалізації програмного забезпечення а також його тестування необхідна можливість замикання протоколу TCP / IP на себе (loopback). Отже, тому перевагу було віддано такій ОС як Ubuntu Linux 19.04 (x64). Окрім цього нам знадобився інтерпретатор Python 2.7 для Linux, а також бібліотека “Scapy” для можливості маніпулювання мережевими пакетами на мові програмування Python. Для відстеження результатів було використано програму для аналізу мережевих пакетів Wireshark.

4.2 Архітектура

Дана програма складається із трьох програмних модулів, які являють собою імітацію клієнт-серверного додатку, спілкування яких відбувається на базі стеку протоколів TCP\IP, а також сніферу, налаштованого безпосередньо на необхідну нам задачу, а також двох текстових файлів, ‘text.txt’, ’message.txt’’, необхідних для зчитування та запису інформації відповідно.

Перший модуль Client містить клас TcpSession_client, який поєднує в собі 10 наступних методів, та функцію int main(), код яких представлено у Додатку А:

def __init__(self, target),

де target – масив даних, який визначає користувач. Масив має містити IP та порт серверу.

Даний метод ініціалізує початкові загальні дані зразкового пакету, але з початку зчитує дані із текстового файлу ‘text.txt’ та перетворює їх до вигляду, необхідного для їх передачі стеганографічним методом;

def start(self),

метод призначений для ініціації зв’язку із сервером. Процес виконується із за допомогою функції def send_syn(self)(див. нижче);

def handle_recv(self, pkt),

де pkt – пакет отриманий від сервера.

Метод приймає та обробляє пакет в залежності від отриманих даних та прапору та відсилає відповідну відповідь на сторону сервера;

def send_syn(self),

Метод формує та відправляє SYN-пакет, у даному випадку серверу. У якості значення, яке повертається використовується результат функції для обробки відповіді на відправлений пакет def handle_recv(self, pkt)(див. вище);

def send_synack_ack(self, pkt),

де pkt – SYN+ACK-пакет, який було відправлено сервером клієнту, як відповідь на запит SYN-пакета.

Метод, використовуючи отриману відповідь, формує та відправляє свою відповідь на отриманий SYN+ACK-пакет, а саме – ACK-пакет;

def send_fin(self),

метод формує та відправляє FIN-пакет, у даному випадку серверу. У якості значення, яке повертається використовується результат функції для обробки відповіді на відправлений пакет def handle_recv(self, pkt)(див. вище);

def send_finack(self, pkt),

де pkt – FIN-пакет, який було відправлено сервером клієнту.

Метод, використовуючи отриманий пакет, формує та відправляє свою відповідь на отриманий FIN-пакет, а саме – FIN+ACK-пакет;

def send_ack(self, pkt),

де pkt – PSH+ACK-пакет, який було відправлено сервером клієнту.

Метод, використовуючи отриманий пакет, формує та відправляє свою відповідь на отриманий PSH+ACK –пакет, а саме – ACK-пакет;

def send_finack_ack(self, pkt),

де pkt – FIN+ACK-пакет, який було відправлено сервером клієнту як відповідь на запит FIN-пакета.

Метод, використовуючи отриману відповідь, формує та відправляє свою відповідь на отриманий FIN +ACK-пакет, а саме – ACK-пакет;

def _sniff(self),

метод, який відстежує пакети, які надходять від серверу до клієнта. Так, як даний програмний продукт являє собою демонстраційний варіант, то в методі вже заздалегідь встановлено необхідний момент обробки стеганопакетів.

int main(),

головна функція модулю Client.py, яка виконує обробку результатів виконання методів та яка організовує їх роботу.

В загальній формі діаграма даного класу представлена на Рисунку 4.1

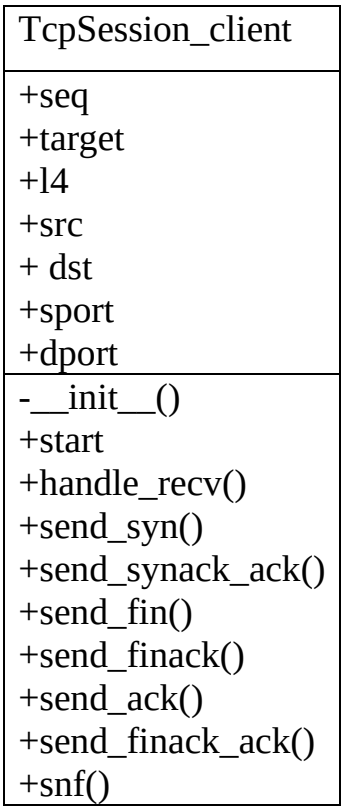

Рисунок 4.1 – Діаграма класу TcpSession_client

Другий модуль Server.py , код якого представлено у Додатоку Б, містить головні загальну реалізацію для sockets для встановлення TCP-зв’язку із модулем Client та відправки йому просто повідомлень. Після відправки усіх даних з’єднання із модулем Client закривається. Розглянемо більш детально функції даного модулю.

socket.socket(socket. AF_INET, socket.SOCK_STREAM),

де AF_INET – вказує сімейство протоколів створюваного сокета (у даному випадку – мережевий протокол IPv4),

SOCK_STREAM – вказує на тип сокету (у даному випадку – потоковий сокет).

Метод призначений для створення кінцевої точки з'єднання;

s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1),

де SOL_SOCKET – вказує на рівень сокету,

SO_REUSEADDR – параметр використовується для доступу до значень прапорів (для повторного використовувати пору, навіть якщо минулий процес завершився невдало).

Метод маніпулює прапорами, встановленими на сокеті;

s.bind(ip, int(port))),

де ip – ip-адреса серверу,

port – номер порта серверу.

Метод ініціалізує ip-адрес і порт, прив’язує сокет до адреси, при цьому перевіряється, чи не зайнятий порт іншою програмою;

s.listen(),

метод підготовлює сокет, що прив’язується, до прийняття вхідного з’єднання;

accept(),

метод приймає запит на встановлення з’єднання від віддаленого хосту;

conn.send(),

даний метод відправляє звичайне повідомлення data, яке заздалегідь було визначено користувачем;

conn.close(),

метод відправляє повідомлення про закриття з’єднання із клієнтом.

Третій модуль Sniff.py , код якого представлено у Додатку В, містить реалізацію сніферу, котрий приймає лише SYN-пакети, виділяє інформацію, яка знаходиться у полі IS, перетворює цю інформацію до стану необхідного для уможливлення читання секретного повідомлення, та відкриває, у крайньому випадку створює, файл ‘message.txt’ і записує цю інформацію до нього.

Загальну схему принципу реалізації звичайної TCP-сесії та сесії із використанням МС на основі методу генерування ISN можливо побачити на Рисунку 4.2 та Рисунку 4.3 відповідно.

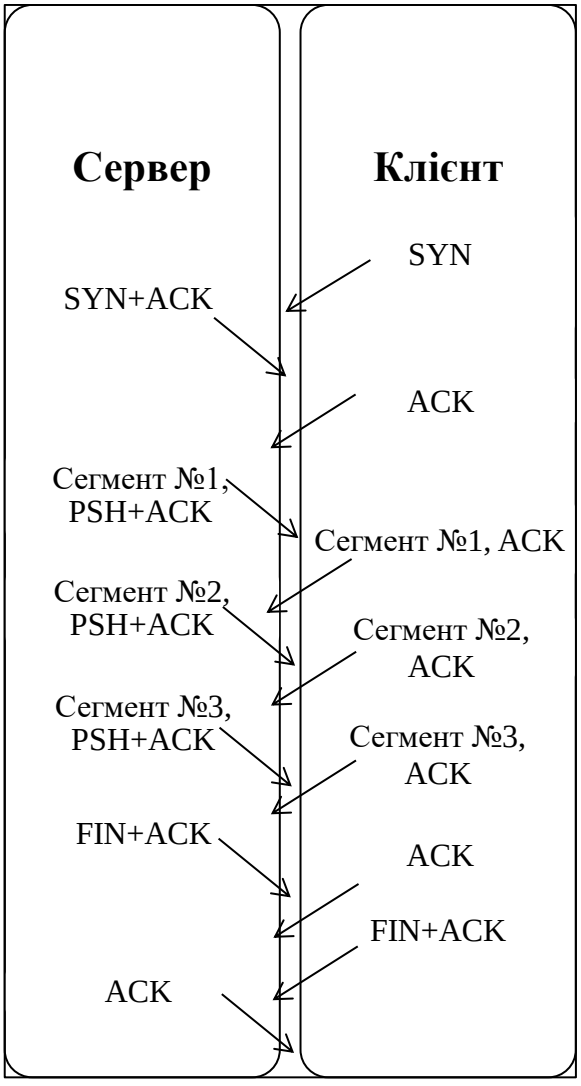

Рисунок 4.2 – Вигляд порядку звичайної TCP-сесії

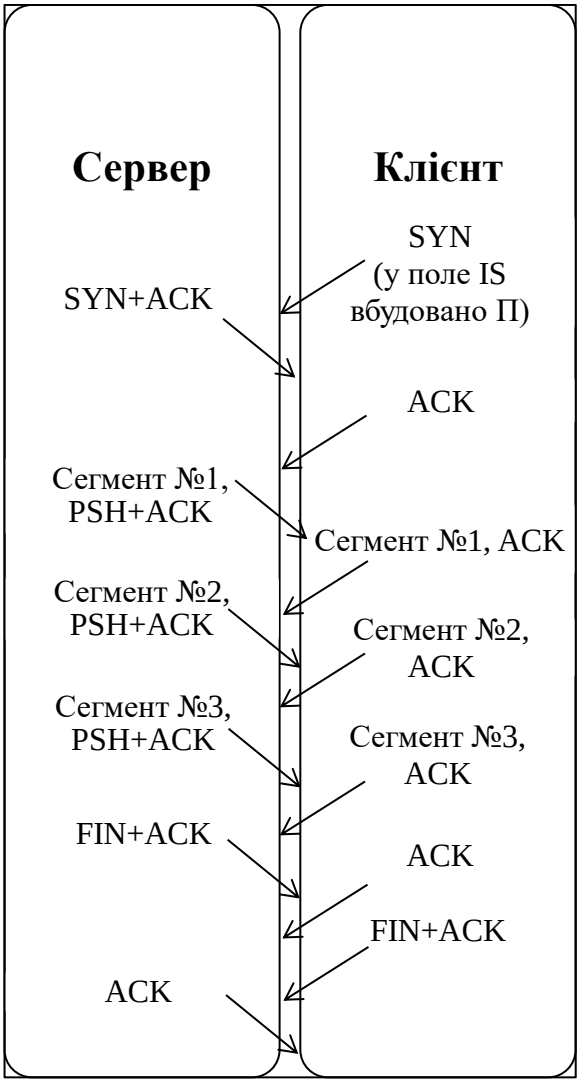

Рисунок 4.3 – Вигляд порядку TCP-сесії із використанням МС

Завдяки рисункам, які описують покрокову роботу TCP-сесії із використанням обраного методу МС (Рисунок 4.3) а також діаграми класу (Рисунок 4.1) ми побудували діаграму послідовностей (Рисунок 4.4), яка дає можливість чітко відстежити принцип роботи програми.

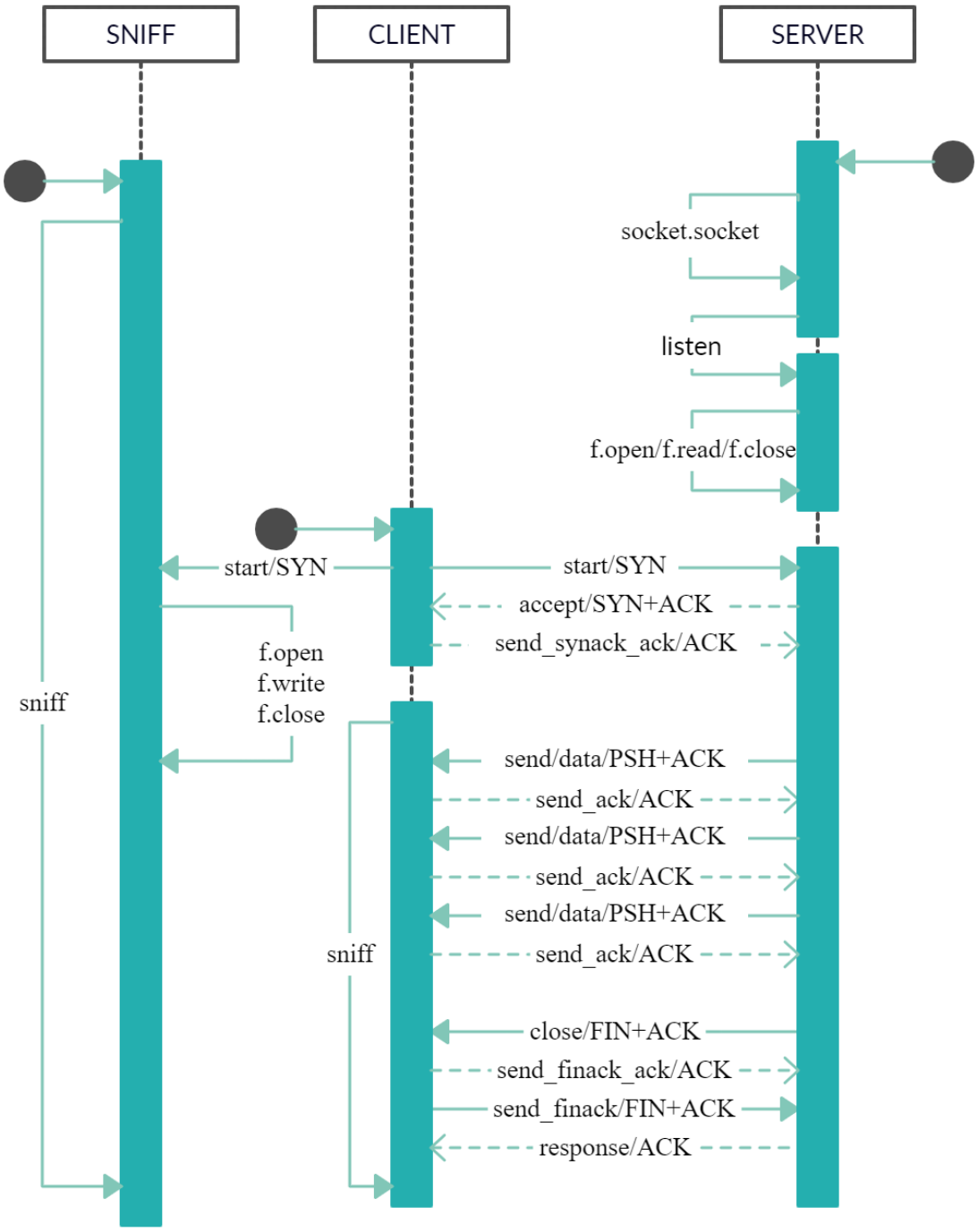

Рисунок 4.4 – Діаграма послідовностей під час TCP-сесії із використанням методу генерації ISN

4.3 Тестування

Для тестування програми та детального відстеження передачі пакетів було використано додаток “Wireshark” а також наш програмний модуль Sniff (Додаток В). Під час проведених тестів були отримані наступні результати (Рисунок 4.5) та (Рисунок 4.6). На даному малюнку можливо побачити результати передачі, використовуючи метод та схему МС, які було зазначено раніше (Рисунок 3.1).

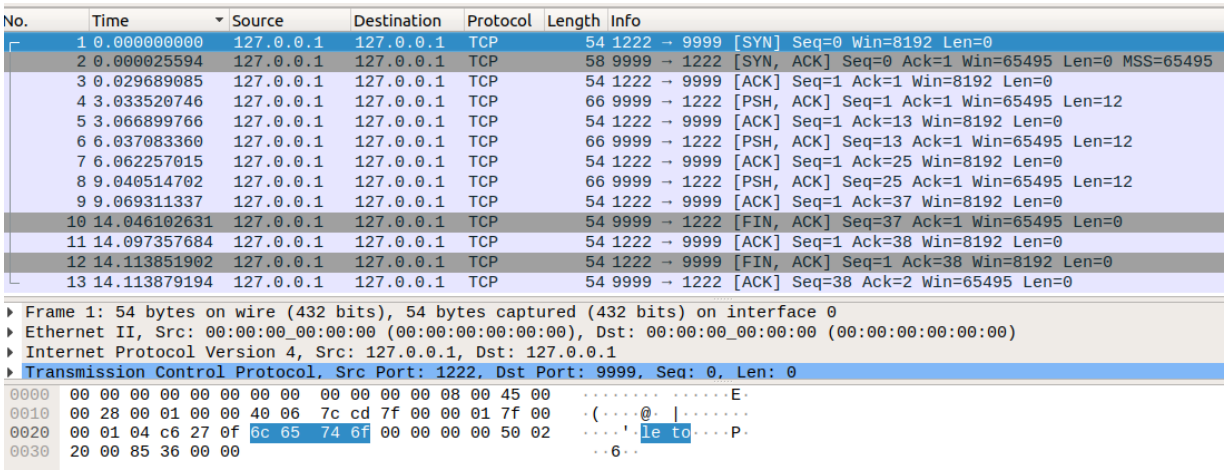

Рисунок 4.5 – Передача пакетів у “Wireshark” під час виконання програми, загальний вид

Із за особливостей програмного продукту “Wireshark”, значення seq у таблиці дорівнює нулю, але якщо відкрити детальний опис SYN-пакету (Рисунок 4.6) , то можливо побачити реальні значення відправленого пакету (дужкою позначено місця початку значень TCP-частини).

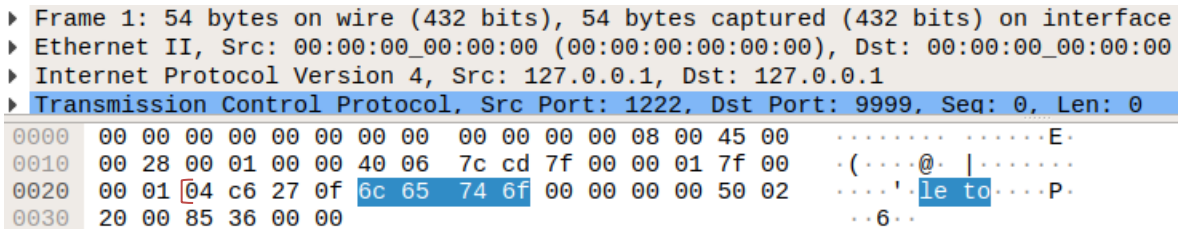

Рисунок 4.6 – Передача пакетів у “Wireshark” під час виконання програми, детальний опис пакету

На Рисунку 4.6 ми маємо змогу побачити SYN-пакет, який ми формуємо на стороні клієнта, який вже містить наше П у полі seq, яке ми заздалегідь зчитали із зазначеного файлу та перетворили до вигляду необхідного для його стеганографічної передачі.

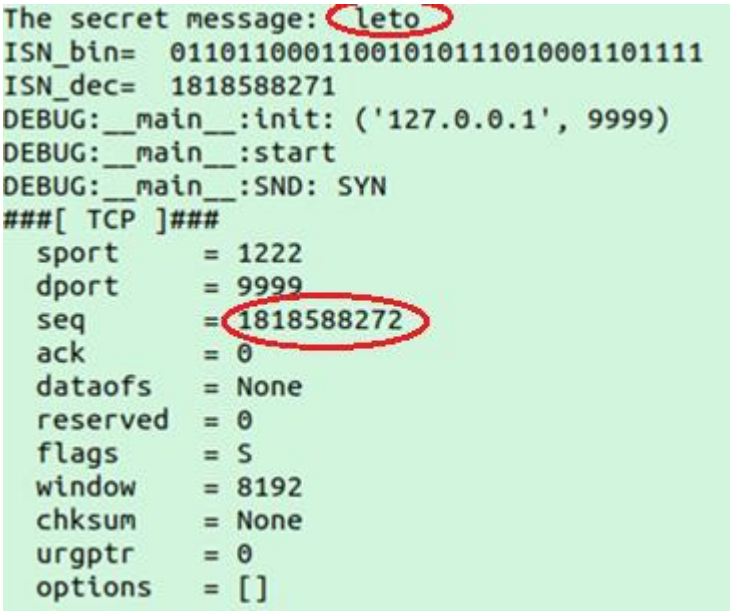

Рисунок 4.6 – Результат роботи модулю “Client.py”, вигляд сформованого SYN-пакету

Наступний Рисунок 4.7 дає нам можливість побачити результат, які ми отримаємо на стороні серверу, тобто отримуємо наш ISN та перетворюємо на текст. Отже, П, яке ми відправили, та яке отримали, як кінцевий результат повністю збігаються.

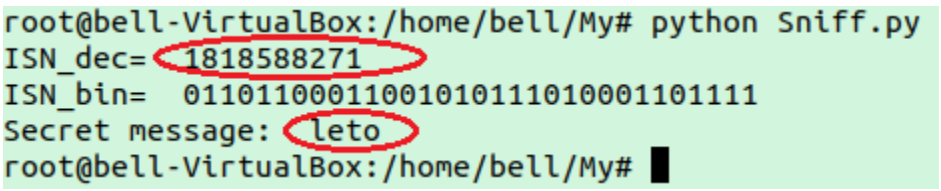

Рисунок 4.7 – Результат роботи модулю “Sniff.py”

Важливо також зазначити, що обране для стеганоконтейнеру поле, має містити не більше ніж 32 біти. Так як програма є демонстраційної, то ми

заздалегідь взяли до уваги цей факт, тому і повідомлення підготували, враховуючи ці особливості. В подальшому, для повноцінного застосування розробленого ПО, треба додати функцію, яка б розділяла подане П на частини не більше 32 біт, і відповідно вбудовувала би ці частини у кожен SYN-пакет нової сесії.

4.4 Порівняння результатів методу на основі генерації ISN із результатами інших методів МС

Маючи результати із нашого власноруч розробленого ПО а також результати із раніше розробленого ПО та опираючись на теоретичні результати відкритих джерел стосовно інших методів МС, маємо можливість зробити аналіз де-яких методів МС, для виявлення найкращого для перспективи його подальшого використання.

Отже, для нашого порівняння ми обрали наступні методи МС:

– метод на основі генерації ISN;

– на основі ICMP-інкапсуляції;

– метод RSTEG.

Для наглядного порівняння і визначення переваг ті недоліків кожного із методів, звернемося до побудованої Таблиці 4.1.

Треба також зазначити, що так як ми робимо порівняння алгоритмів тільки відносно один одного, відповідно оцінки де-яких характеристик також не є абсолютними, а тільки відносними у рамках тих алгоритмів, що ми розглядаємо.

Отже, після аналізу даних у таблиці можна прийти до висновку, що не існує будь-якої ідеальної стеганосистеми, і вибір того чи іншого стеганометоду буде залежати від умов та цілей використання МС.

Таблиця 4.1 – Порівняння обраних методів МС

Метод МС Критерій порівняння	На основі генерації IS номерів	На основі ICMP-інкапсуляції	RSTEG
Поля, що ви користуються у ролі стеганоконтейнеру	поле seq у SYN-пакетах	де-які поля заголовку ICMP-пакету, та поле даних	де-які поля заголовку TCP/IP-пакету, та поле даних
Пропуска здатність за одне відправлення	32 біти	залежить від типу ICMP-повідомлення, достатньо велика	залежить, від обраних у якості стеганоконтейнеру полів, достатньо велика
Вірогідність розкриття	мінімальна	середня	середня
Методи стеганоаналізу, які можуть бути використані для розкриття	ймовірністно-статистичний аналіз, ШНМ	простий аналіз трафіку, ймовірністно-статистичний аналіз, ШНМ, поведінковий аналіз	простий аналіз трафіку, ймовірністно-статистичний аналіз, ШНМ, поведінковий аналіз
Переваги	Відносно висока складність розкриття	Відносна простота реалізації, велика пропускна здатність	Велика пропускна здатність

Метод МС / Критерій порівняння	На основі генерації IS номерів	На основі ICMP-інкапсуляції	RSTEG
Недоліки	Відносна складність реалізації	Відносно низька складність розкриття; наявність великої кількості методів, для здійснення стеганоаналізу	Складність реалізації; відносно низька складність розкриття; наявність великої кількості методів, для здійснення стеганоаналізу

ВИСНОВКИ

У ході атестаційної роботи було розглянуто різні методи побудування прихованого каналу передачі даних в інформаційно-телекомунікаційній мережі, що базуються на використанні особливостей протоколів мережевої моделі TCP / IP. Окрім цього, ми провели аналіз даних методів, який слугує в якості теоретичної допомоги при реалізації одного із методів. Для цього було потрібно детально опанувати теоретичну базу мережевої моделі TCP / IP, щоб зрозуміти деталі та слабкі місця, які ми змогли би використати не тільки за для головної цілі, тобто реалізація стеганоканалу, а і для подальшого спрощення можливого стеганоаналізу. Отже, як ми з’ясували, методи мережевої стеганографії не є досить глибоко опановані і широко відомими для звичайних користувачів. Як наслідок, ми маємо таку ж ситуацію стосовно методів стеганоаналізу. Вище приведені факти дають перевагу стосовно даного виду прихованої передачі даних.

Особливу увагу в даній роботі було приділено методу генерації IS номерів – це метод MC, головна суть якого полягає у наступному. Зазвичай, відповідно до [1], порядковий номер у SYN-пакету, з якого починається зв'язок між учасниками TCP-з’єднання, має генеруватися випадково, враховуючи рекомендації, які надано у приведеному стандарті. Для стороннього ж спостерігача дане поле є абсолютно рандомним. У цьому і є головна ідея даного методу. Тобто, у обране поле можливо підставити своє таємне повідомлення, яке заздалегідь необхідно перетворити до відповідного вигляду.

Базуючись на приведених вище теоретичних даних було створено ПО, яке і реалізує обраний метод MC. Для реалізації нам знадобилось програмне забезпечення “Linux” і такий інструмент для створення та взаємодії із мережевими пакетам, як “Scapy”. Усі приведені вище засоби є загальнодоступними, що дає змогу кожному створити подібний засіб. Стосовно програми також треба зазначити, що ми заздалегідь передбачаємо, що таємне

повідомлення, яке необхідно відправити, не є більш ніж 32 біти. При подальшому покращенні програми можливо вирішити дану особливість для більш зручного використання. Окрім цього, було наведено результати перевірки коректності роботи реалізованої програми.

Проведені дослідження функціонування даного методу показали, що головним недоліком залишається то й факт, що обране поле не може приймати значення, більші ніж 32-біти. Тому проблематичним залишається факт передачі великих повідомлень. Стосовно стеганоаналізу даного методу, то тут майже єдиним варіантом є використовування ймовірністно-статистичний аналізу, але він вимагає велику кількість місця для зберігання усіх SYN-пакетів, для виявлення в подальшому необхідної закономірності, або факту її відсутності. Це також впливає на зниження вірогідності розкриття, використовуючи даний метод МС для прихованої передачі інформації.

У роботі також розглянуто та проаналізовано інші методи з боку їх основних характеристик і деталей реалізації. Приведені матеріали можуть бути використані в якості бази для розробки інших програмних засобів на основі цих методів.

Отже, ми приходимо до висновку, що не може існувати ідеального методу прихованої передачі інформації, тому вся суть полягає у компромісі. Із врахуванням умов передачі та цілей, ми можемо обрати оптимальний метод МС. Як приклад, ми можемо надати переваг більш надійному методу, але пожертвувати пропускною здатністю, і навпаки.

ПЕРЕЛІК ПОСИЛАНЬ

1. Postel J. Transmission Control Protocol – DARPA Internet Program Protocol Specification [Текст] / J. Postel // Internet RFC 791. – California: USC/Information Sciences Institute – 1981.

2. Костенюк Т.А. Метод прихованої передачі даних в інформаційно-телекомунікаційній мережі на основі генерації ISN TCP-з'єднань [Текст] / Костенюк Т.А., Руженцев В.І. // Проблеми Інформатизації : міжнар. наук.-техн. конф., 26-27 листоп. 2020 р.: тези доповідей – Харків, 2020.

3. TCP/IP vs. OSI: в чем разница между двумя моделями? [Електронний ресурс] // FS – 25.12.2019 – Режим доступу: https://community.fs.com/ru/blog/tcpip-vs-osi-whats-the-difference-between-the-two-models.html.

4. Пескова О. Ю. Применение сетевой стеганографии для защиты данных, передаваемых по открытым каналам Интернет [Текст] / О. Ю. Пескова, Г. Ю. Халабурда. – СПб.: МПСС, 2012 – С. 348.

5. Белкина Т. А. Аналитический обзор применения сетевой стеганографии для решения задач информационной безопасности [Електронний ресурс] / Т. А. Белкина //Молодой ученый. – Казань – 2018. – №11. – С. 36-44. – Режим доступу: https://moluch.ru/archive/197/48821.

6. Mazurczyk W. Retransmission steganography and its detection [Текст] / W. Mazurczyk, M. Smolarczyk, K. Szczypiorski // Soft Computing. – Springer-Verlag, 2011. – С. 505–515.

7. Олифер В. Г. Компьютерные сети. Принципы, технологии, протоколы [Текст] / В. Г. Олифер, Олифер Н. А. – СПб.: Питер, 2017 – №5 – С. 992.

8. Postel J. Internet Control Message Protocol – DARPA Internet Program Protocol Specification [Текст] / J. Postel // Internet RFC 792. – California: USC/Information Sciences Institute – 1981.

9. Postel J. Internet Protocol – DARPA Internet Program Protocol Specification [Текст] / J. Postel // Internet RFC 791. – California: USC/Information Sciences Institute – 1981.

10. Микитишин А. Г. Комп'ютерні мережі [Текст] / А. Г. Микитишин М. М. Митник, П. Д. Стухляк, В. В. Пасічник. – Львів: Магнолія 2006, 2013. – 256 с.

11. Модель стека протоколов TCP/IP и ее особенности [Електронний ресурс] // ZametkiNaPolyah.ru. – 12.06.2018 – Режим доступу: https://zametkina polyah.ru /kompyut ernye-seti/stek-protokolov-tcp-ip.html.

12. Стеганография на базе стека протоколов TCP / .IP. Часть 1 [Електронний ресурс] // securitylab – 18.04.2020 – Режим доступу: https://www.securitylab.ru /analytics/485917.php?R=1.

13. Рубан И. В. Метод стеганографической передачи данных в информационно-телекоммуникационных сетях на основе генерации ISN tcp-соединений / И. В. Рубан, А. О. Смирнов // Системи обробки інформації. – Харків: ХУПС, 2015. – № 9 (134). – С. 99–101.

14. Модель обработки tcp-соединений для стеганографической передачи данных в информационно-телекоммуникационных сетях / И. В. Рубан, А. О. Смирнов // Сучасні інформаційні технології у сфері безпеки та оборони. – Харків: ХУПС, 2015. – № 3 – С. 908–112.

15. Орлов В.В. Активная стеганография в сетях TCP / IP / В.В. Орлов, А. П. Алексеев // Инфокоммуникационные технологии. – 2009. – № 2. – С. 73-78.

16. Юдін О. К. Виявлення прихованих каналів передачі інформації на базі методів стеганоаналізу [Текст] / О. К. Юдін, Я. А. Симониченко // Наукоємні технології. – 2016. – №4 (32). – С. 389-394.