Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:
В.о. завідувача кафедри
кібербезпеки
та захисту інформації
_____ Іван ПАРХОМЕНКО
«17» травня 2024 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи

галузь знань *12 Інформаційні технології*
(шифр і назва галузі знань)

спеціальність *125 Кібербезпека*
(код і назва спеціальності)

освітній ступень *магістр*

освітньо-наукова програма *Кібербезпека*
(назва освітньої програми)

на тему: «Метод протидії кібератакам на протокол динамічної маршрутизації OSPF»

Виконавець: студент ІІ курсу, групи КБм-21

_____ **Богдан СКЛЯР**
(підпис) (Ім'я, ПРІЗВИЩЕ)

	Ім'я, ПРІЗВИЩЕ	**Підпис**
Науковий керівник	Олександр ЛАПТЄВ	
Нормоконтроль	Юрій БАБЕНКО	

Київ 2024

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
В.о. завідувача кафедри
кібербезпеки
та захисту інформації
_____ Іван ПАРХОМЕНКО
«17» листопада 2023 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи

спеціальності *125 Кібербезпека*
(код і назва спеціальності)

освітній ступень *магістр*

Здобувача(ки) КБм-21 (група) Скляра Богдана Юрійовича (прізвище ім'я по-батькові)

Тема кваліфікаційної роботи Метод протидії кібератакам на протокол динамічної маршрутизації OSPF

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Рішення засідання кафедри кібербезпеки та захисту інформації факультету інформаційних технологій протокол № 5 від 15.11.2023 р.

2. МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень	Процес маршрутизації пакетів даних в автономній системі із використанням мережевого протоколу OSPF.
Предмет досліджень	Методи захисту протоколу OSPF та мережевого обладнання від кібератак різного типу.
Мета	Підвищення ефективності протидії кібератакам на протокол динамічної маршрутизації OSPF за рахунок розробки окремої класифікації атак і удосконаленої конфігурації OSPF-маршрутизатора.

Вихідні дані для проведення роботи	Методи захисту від кібератак системи динамічної маршрутизації на основі протоколу OSPF.

3. ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Наукова новизна	Розробка класифікації кібератак на досліджуваний протокол за рівнем впливу на автономну систему; удосконалення конфігурації OSPF-маршрутизатора за рахунок налаштування захисних механізмів.
Практична цінність	Дозволяє кваліфікувати кібератаки на протокол, оцінювати рівень їх впливу на автономну систему, розробляти конфігурації OSPF-маршрутизатора, що дає можливість підвищити стійкість пристрою до активних кібератак на систему маршрутизації.

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Робота виконана у повному обсязі відповідно до теми.

5. ЕТАПИ ВИКОНАННЯ РОБОТИ

Найменування етапів робіт	**Строки виконання робіт (початок-кінець)**
Уточнення постановки задачі	17.11.2023 – 20.12.2023
Аналіз літературних джерел	21.12.2023 – 14.02.2024
Дослідження кібератак на протокол динамічної маршрутизації OSPF	15.02.2024 – 04.03.2024
Визначення найбільш вразливих вузлів конструкції протоколу	05.03.2024 – 09.03.2024
Практична експлуатація досліджуваних вразливостей протоколу динамічної маршрутизації OSPF	10.03.2024 – 20.03.2024
Розробка класифікації кібератак на протокол динамічної маршрутизації OSPF за рівнем їх впливу на автономну систему	21.03.2024 – 01.04.2024
Розробка удосконаленої конфігурації OSPF-маршрутизатора з покращеним захистом від кібератак	02.04.2024 – 14.04.2024
Апробація роботи на науково-методичному семінарі	15.04.2024 – 26.04.2024

Найменування етапів робіт	Строки виконання робіт (початок-кінець)
Оцінка ефективності удосконаленої конфігурації	27.04.2024 – 28.04.2024
Оформлення пояснювальної записки згідно методичних рекомендацій	29.04.2024 – 11.05.2024
Оформлення презентації та отримання рецензії	12.05.2024 – 14.05.2024
Подача пакету документів на розгляд ЕК	15.05.2024 – 17.05.2024

6. РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

Економічний ефект Зниження збитків через кібератаки на системи маршрутизації компаній

Соціальний ефект Покращення системи захисту OSPF-маршрутизаторів від кібератак на підприємствах

7. ДОДАТКОВІ ВИМОГИ

Завдання видав _____ Олександр ЛАПТЄВ
(підпис) (Ім'я, ПРІЗВИЩЕ)

Завдання прийняв до виконання _____ Богдан СКЛЯР
(підпис) (Ім'я, ПРІЗВИЩЕ)

Дата видачі завдання: 17.11.2023 р.
Термін подання кваліфікаційної роботи до ЕК 17.05.2024 р.

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Метод протидії кібератакам на протокол динамічної маршрутизації OSPF»: 91 сторінка, 49 рисунків, 11 таблиць, 2 додатки і 58 літературних джерел.

Об’єкт дослідження – процес маршрутизації пакетів даних в автономній системі із використанням мережевого протоколу OSPF.

Предмет дослідження – методи захисту протоколу OSPF та мережевого обладнання від кібератак різного типу.

Метою дипломної роботи є підвищення ефективності протидії кібератакам на протокол динамічної маршрутизації OSPF за рахунок розробки окремої класифікації кібератак і удосконаленої конфігурації OSPF-маршрутизатора.

У процесі вирішення поставлених завдань у дипломній роботі були використані: методи аналізу, спостереження, індукції та моделювання; метод пошуку найкоротших шляхів між вершинами графу.

У роботі досліджено процес маршрутизації з використанням протоколу OSPF; проведено детальний аналіз його вразливостей та функціоналу безпеки; реалізовано практичну експлуатацію вразливостей; розроблено класифікацію кібератак і удосконалену конфігурацію OSPF-маршрутизатора.

Актуальність теми: більшість доступних досліджень, пов’язаних з безпекою протоколу OSPF, концентрують увагу на окремих кібератаках та можливих векторах захисту. Проведений аналіз вразливостей вищезазначеного протоколу та методів протидій ним показав слабкий розвиток комплексної ідеї захисту OSPF-маршрутизатора від кібератак різних видів.

Отже, актуальним науковим завданням, яке має як теоретичне так і практичне значення, є побудова удосконаленої конфігурації маршрутизатора

з ефективним захистом від кібератак різних видів, а також розробка класифікації існуючих кібератак за рівнем їх впливу на автономну систему.

Практичне значення роботи полягає у розробці класифікації кібератак за рівнем впливу на автономну систему і реалізації удосконаленої конфігурації пристрою маршрутизації.

Результати досліджень можуть бути впровадженні при налаштуванні системи маршрутизації на підприємстві; у навчальному процесі університету для поглиблення програми навчальної дисципліни «Безпека операційних систем на комп'ютерних мереж».

Наукова новизна дослідження полягає у:

- удосконаленні конфігурації OSPF-маршрутизатора. Запропонована конфігурація відрізняється від базової та інших існуючих тим, що в ній реалізований захист від кібератак різних типів, шляхом залучення різних функцій безпеки та їх правильної співдії;
- розробці класифікації кібератак на протокол OSPF, яка базується на рівнях їх впливу на автономну систему. Дана класифікація узагальнює, систематизує та доповнює наявну інформацію про можливі наслідки експлуатації вразливостей зловмисниками.

Одним із напрямків подальших досліджень є розширення створеної класифікації з додаванням нових типів кібератак та їх можливих наслідків для автономних систем із маршрутизацією на основі протоколу OSPF.

Ключові слова: динамічна маршрутизація, внутрішня маршрутизація, маршрутизатор, мережевий протокол, автономна система, система маршрутизації, метод протидії, вразливість, кібератака, несанкціонований доступ, криптографічна автентифікація, підробка, відносини сусідства, оголошення про стан каналу.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ 9

ВСТУП 10

РОЗДІЛ 1 АНАЛІЗ ПРОЦЕСУ МАРШРУТИЗАЦІЇ: СТАТИЧНОЇ ТА ДИНАМІЧНОЇ 13

РОЗДІЛ 2 ДОСЛІДЖЕННЯ ТЕОРЕТИЧНИХ ЗАСАД ПРОТОКОЛУ ДИНАМІЧНОЇ МАРШРУТИЗАЦІЇ OSPF 17

2.1 Історія розробки протоколу 17

2.2 Фундаментальні основи побудови 22

2.3 Структура LSA 27

2.3.1 LS age 29

2.3.2 Options 30

2.3.3 LS Type 30

2.3.4 Link State ID 31

2.3.5 Advertising Router 31

2.3.6 LS Sequence Number та Checksum 31

2.4 Механізми безпеки 32

2.5 OSPFv3 36

Висновки до розділу 2 37

РОЗДІЛ 3 РОЗРОБКА КЛАСИФІКАЦІЇ КІБЕРАТАК, ЩО ВИКОРИСТОВУЮТЬ ВРАЗЛИВОСТІ КОНСТРУКЦІЇ ПРОТОКОЛУ 38

3.1 Загальні відомості 38

3.2 Disguised LSA 40

3.3 Single Path Injection 43

3.4 Adjacency Spoofing 45

3.5 Атаки з фальсифікацією порядкового номера повідомлення 48

3.6 Реалізація атаки на автентифікацію повідомлень 49

3.7 Реалізація атаки Adjacency Spoofing 53

3.8 Класифікація кібератак за рівнем їх впливу на автономну систему маршрутизації 58

Висновки до розділу 3 60

РОЗДІЛ 4 РОЗРОБКА МЕТОДУ ПРОТИДІЇ КІБЕРАТАКАМ ТА ОЦІНКА ЙОГО ЕФЕКТИВНОСТІ 62

4.1 Розробка безпечної конфігурації OSPF-маршрутизатора 65

4.1.1 Інсталювання автентифікації пакетів даних з використанням криптографічних алгоритмів шифрування 65

4.1.2 Інсталювання захисного механізму від переповнення трафіку 69

4.1.3 Інсталювання захисного механізму перевірки часу життя пакетів даних 70

4.1.4 Інсталювання системи логування змін відносин сусідства 72

4.1.5 Інсталювання ліміту кількості оголошень про стан каналу 72

4.1.6 Інсталювання списків контролю доступу 74

4.1.7 Налаштування часових інтервалів 76

4.1.8 Налаштування привілеїв на маршрутизаторах 77

4.2 Оцінка ефективності удосконалено конфігурації 79

Висновки до розділу 4 83

ВИСНОВКИ 84

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 86

ДОДАТОК А 92

ДОДАТОК Б 97

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

IMP	–	Interface Message Processor
OSPF	–	Application Programming Interface
(D)DoS	–	(Distributed) Denial-of-Service
EGP	–	Exterior Gateway Protocol
IP	–	Internet Protocol
DR	–	Designated Router
DBR	–	Backup Designated Router
LSA	–	Link-State Advertisement
DBD	–	Data-Base Description
MD5	–	Message-Digest 5
HMAC	–	Hash-based Message Authentication Code
SHA	–	Secure Hash Algorithms
CVE	–	Common Vulnerabilities and Exploits
ID	–	Identity Document
IETF	–	Internet Engineering Task Force
IOS	–	Internetworking Operating System
AC	–	Автономна система

ВСТУП

З огляду на постійний розвиток технологій і зростаючу складність мережевих технологій та систем, забезпечення безпеки мережевих сегментів стає все більш важливим завданням для організацій незалежно від розміру та виду діяльності. Стабільність роботи автоматизованих систем починає дедалі частіше страждати через перевантаження або атаки зловмисників.

Однією з складових, що забезпечує збалансовану роботу приватних та корпоративних мереж є маршрутизація. Вона являє собою процес визначення найкращого шляху для пакетів даних від джерела походження до точки призначення, забезпечуючи ефективність та швидкодію. Оптимальний вибір маршруту дозволяє використовувати ресурси мережевих систем максимально ефективно.

Збої, що можливі в процесі маршрутизації, як правило, призводять до різноманітних проблем в автономних системах. При втраті зв’язку між вузлами, користувачі втрачають доступ до ресурсів та можливих послуг комп’ютерної мережі. Проте, слід зазначити, що збої можуть статись непередбачувано і раптово, а можуть бути спланованими і мати певну мету.

Асекурація захищеності системи маршрутизації безпосередньо сприяє досягненню цілей головної тріади кібербезпеки шляхом збереження конфіденційності, цілісності та доступності даних та мережевих сегментів. Неефективні налаштування безпеки дають змогу зловмиснику порушити певний або усі важливі принципи захисту шляхом виконання атак різних видів.

Актуальність. Протоколи маршрутизації використовують згідно потреб різного виду комп’ютерних мереж чи автономних систем. Одним з таких протоколів є Open Shortest Path First.

Більшість доступних досліджень, пов’язаних з безпекою протоколу OSPF, концентрують увагу на окремих атаках та можливих векторах захисту.

Проведений аналіз вразливостей вищезазначеного протоколу та методів протидій ним показав слабкий розвиток комплексної ідеї захисту OSPF-маршрутизатора від атак різних видів.

Отже, актуальним науковим завданням, яке має як теоретичне так і практичне значення, є побудова удосконаленої конфігурації маршрутизатора з ефективним захистом від атак різних видів, а також розробка класифікації існуючих кібератак за рівнем їх впливу на автономну систему.

Метою кваліфікаційної роботи є підвищення ефективності протидії кібератакам на протокол динамічної маршрутизації OSPF за рахунок розробки окремої класифікації кібератак і удосконаленої конфігурації OSPF-маршрутизатора. Для досягнення зазначеної мети кваліфікаційної роботи були сформовані окремі завдання:

- провести детальний аналіз офіційної технічної документації протоколу динамічної маршрутизації OSPF, а також вразливостей та методів протидії ним;
- визначити потенційно вразливі вузли до атак;
- розробити класифікацію кібератак на протокол за рівнем їх впливу на автономну систему;
- розробити удосконалену конфігурацію OSPF-маршрутизатора.

Об’єкт дослідження – процес маршрутизації пакетів даних в автономній системі із використанням мережевого протоколу OSPF.

Предмет дослідження – методи захисту протоколу OSPF та мережевого обладнання від кібератак різних типів.

У процесі вирішення поставлених завдань у дипломній роботі були використані: методи аналізу, спостереження, індукції та моделювання; метод пошуку найкоротших шляхів між вершинами графу.

Наукова новизна одержаних результатів:

- удосконалено конфігурацію OSPF-маршрутизатора. Запропонована конфігурація відрізняється від базової та інших існуючих тим, що в ній реалізований захист від кібератак різних

типів шляхом залучення різних функцій безпеки та їх правильної співдії;

- розроблено класифікацію кібератак на протокол OSPF, яка базується на рівнях їх впливу на автономну систему. Дана класифікація узагальнює, систематизує та доповнює наявну інформацію про можливі наслідки експлуатації вразливостей зловмисниками.

Практична цінність даної роботи полягає в:

- систематизації та узагальненні інформації про вразливості та кібератаки на протокол OSPF;
- розробці удосконаленої конфігурації OSPF-маршрутизатора, що може бути використана в комп'ютерній мережі, та, що підвищує стійкість пристрою до активних кібератак на систему маршрутизації;
- розробці класифікації кібератак на протокол, що дозволяє оцінити рівень їх впливу на автономну систему.

Результати досліджень можуть бути впровадженні при налаштуванні системи маршрутизації на підприємстві; у навчальному процесі університету для поглиблення програми навчальної дисципліни «Безпека операційних систем на комп'ютерних мереж».

Основні наукові положення і результати роботи тезисно опубліковані в збірці тез та доповідей VII Міжнародної науково-практичної конференції «Проблеми кібербезпеки інформаційно-телекомунікаційних систем» [1].

РОЗДІЛ 1
АНАЛІЗ ПРОЦЕСУ МАРШРУТИЗАЦІЇ: СТАТИЧНОЇ ТА ДИНАМІЧНОЇ

Маршрутизація являє собою процес вибору маршруту у будь-якій мережі, від телефонних комунікацій до громадського транспорту. Маршрутизація в мережах з комутацією пакетів, як Інтернет, це ніщо інше як вибір шляху для пакетів даних Інтернет-протоколу від джерела до потрібної точки в мережі. Рішення про вибір шляху приймаються спеціалізованим мережевим обладнанням, що має назву маршрутизатор. Кожен такий пристрій має у собі внутрішні таблиці маршрутизації, які використовуються при прийнятті рішення про доставку пакетів даних по мережі [1, 2].

Таблиця маршрутизації містить шляхи, якими мають бути доставлені пакети даних, до кожного місця призначення в області відповідальності даного маршрутизатора [2].

Під час отримання пакету, маршрутизатор зчитує його заголовок, щоб отримати інформацію про його кінцеве місце призначення в мережі, а далі, на основі своїх таблиць маршрутизації, приймає рішення яким саме шляхом його надіслати [2].

Вищезазначені дії виконуються відповідним апаратним забезпеченням тисячі разів в секунду з різними пакетами даних. У процесі доставлення, один пакет даних може бути маршрутизований кілька разів кількома мережевими пристроями [2].

Загалом, існує два базових методи побудови таблиць маршрутизації: статичний та динамічний [3].

Статична таблиця маршрутизації створюється, підтримується оновлюється системним адміністратором або адміністратором мережі. Статичні маршрути до кожної кінцевої точки відповідної мережі мають бути прописані в таблицях маршрутизації кожного маршрутизатора для

безперебійності підключення. Даний метод забезпечує досить детальний контроль над процесом доставлення пакетів даних по мережі, однак, стає пропорційно менш практичним зі збільшенням розмірів мережі [3].

Значною перевагою даного методу є ефективне використання оперативної пам’яті та можливостей процесору маршрутизатором [3].

Однак, зміни топології автономної системи призводять до помилок у маршрутизації і вимагають ручного втручання спеціалістів у галузі мережевих технологій. Адже в автономній системі з налаштованою маршрутизацією лише статичним методом, жоден маршрутизатор не має змоги обрати інший або кращий маршрут у випадку непередбачуваних відмов або навіть передбачуваних змін [3].

Динамічна таблиця маршрутизації, в свою чергу, створюється, підтримується та оновлюється протоколом маршрутизації, що налаштований на відповідному апаратному забезпеченні [3].

Протоколи динамічної маршрутизації діляться на два типи: внутрішні та зовнішні. Протоколи внутрішньої маршрутизації використовуються для доставки пакетів всередині автономної системи, а зовнішньої поза її межами [3]. Більш детально з класифікацією протоколів динамічної маршрутизації можна ознайомитись на рисунку 1.1 [4].

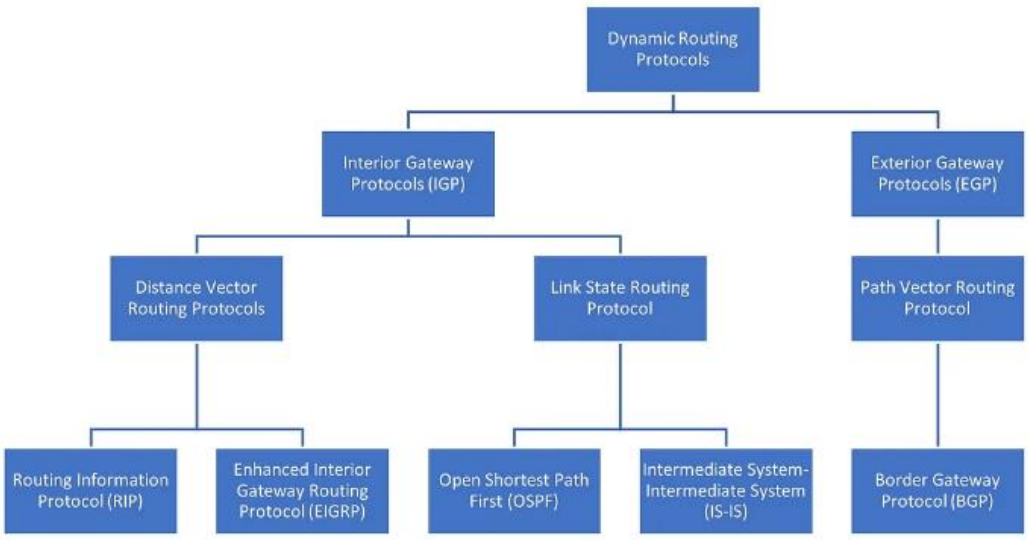

Рисунок 1.1 – Класифікація протоколів динамічної маршрутизації

У випадку використання динамічної таблиці маршрутизації, маршрутизатори поширюють інформації про доступність своїх інтерфейсів, через що використання можливостей процесору та оперативної пам’яті зростає. Проте, на відміну від статичного методу, протоколи маршрутизації, налаштовані на мережевому обладнанні, можуть вибрати як інший, так і кращий маршрут у випадку змін в мережевій інфраструктурі АС [5].

Рішення про те, який саме тип маршрутизації обрати у тій чи іншій автономній системі приймається, беручи до уваги основні переваги та недоліки кожного з них. Детальніше можна ознайомитись у таблиці 1.1.

Таблиця 1.1

Порівняльний аналіз методів маршрутизації

Тип	Переваги	Недоліки
Статична маршрутизація	Мінімальне навантаження на можливості процесора та оперативної пам’яті	Необхідність мануально вносити зміни в таблиці маршрутизації при змінах в топології АС
	Відсутні витрати на пропускну здатність (оновлення не розсилаються між маршрутизаторами)	Відсутність динамічної відмовостійкості, у випадках непередбачуваного виходу з ладу певного вузла мережі
	Максимально деталізований контроль пакетів даних під час маршрутизації	Непрактична у мережах з великою топологією
Динамічна маршрутизація	Легкість та простота в налаштуванні у великих автономних системах	Зменшення пропускної здатності через поширення інформації про маршрутизацію між усіма пристроями в автономній системі
	Можливість вибору іншого, або кращого, маршруту при змінах в інфраструктурі	Налаштовані протоколи маршрутизації споживають певну частку потужності процесора та оперативної пам’яті
	Навантаження на вузли балансується у процесі маршрутизації	Рішення про вибір «найкращого» маршруту для передачі пакетів даних приймає протокол маршрутизації, а не адміністратор мережі

Висновки до розділу 1

У цьому розділі було розглянуто поняття маршрутизації, її методи і функції. Окремо було проведено порівняльний аналіз методів маршрутизації та виокремлення їх сильних та слабких сторін.

Процес вибору шляху пакетів даних від джерела походження до точки призначення в мережі і є маршрутизацією. Обирання шляху відбувається за допомогою таблиць маршрутизації, які можуть бути статичними на динамічними, звідки і походять назви методів маршрутизації

Статична маршрутизація більш ефективно використовує ресурси апаратного забезпечення та мінімізує витрати на пропускну здатність каналу зв'язку. Однак, даний метод непристосований до змін топології, а також до автономних систем з великою кількістю кінцевих пристроїв. Таблиці маршрутизації в такому випадку необхідно модифікувати мануально.

Динамічна маршрутизація в свою чергу надає можливість автоматичної зміни маршруту на інший у випадку позаштатної ситуації. Даний метод досить легко налаштовується у великих автономних системах через поширення інформації про маршрутизацію між усіма пристроями топології. Проте, через цей факт динамічний метод маршрутизації є досить ресурсоємним.

Наступний розділ містить детальний аналіз офіційних технічних специфікацій протоколу OSPF, історію його розробки, а також оголошень про стан каналу.

РОЗДІЛ 2
ДОСЛІДЖЕННЯ ТЕОРЕТИЧНИХ ЗАСАД ПРОТОКОЛУ ДИНАМІЧНОЇ МАРШРУТИЗАЦІЇ OSPF

2.1 Історія розробки протоколу

Перші згадки про початок розвитку маршрутизації трафіку припадає на 70-ті роки 20-го століття. ARPANET була першою мережею, що використовувала комутацію пакетів для обміну даними між пристроями. Першочергово, ця мережа розроблялась, як канал резервного зв'язку з високою швидкодією та зручність, для військових об'єктів у випадку ядерної війни, а власне створена була на замовлення для Міністерства Оборони Сполучених Штатів Америки [6].

Також, на цей період припадає створення перших маршрутизаторів, які назвали IMP. Дане апаратне забезпечення розроблялось спеціалізовано під ARPANET і мало основну функцію – керування маршрутизацією даних [6]. Однак, нічого спільного у дизайні годі й шукати.

Розробка протоколу динамічної маршрутизації OSPF почалась в 1987 році. Open Shortest Path First був одним із перших протоколів, що був повністю розроблений IETF [6]. Простими словами, це організація стандартизації Інтернету, яка несе відповідальність за технічні стандарти, що становлять набір протоколів Інтернету. Офіційного списку членів організації або вимог до них не існує. Всі учасники організації залучені на волонтерських основах.

Робоча група, що займалась розробкою протоколу OSPF досі існує, а технічна документація протоколу все ще доповнюється новою інформацією. Дослідження процесу розробки OSPF дало змогу більш детально зрозуміти процес появи нового функціоналу протоколу. З розвитком технологій, та мережі Інтернет зокрема, OSPF також отримував свої доповнення. Певні функції, такі як Point-to-Multi-Point інтерфейс, не планувались при початковій

розробці, але були впроваджені постфактум, а деякий функціонал так і не були спроектовані, хоча й планувались. Детальну хронологію розвитку протоколу динамічної маршрутизації OSPF можна побачити на рисунку 2.1 [6].

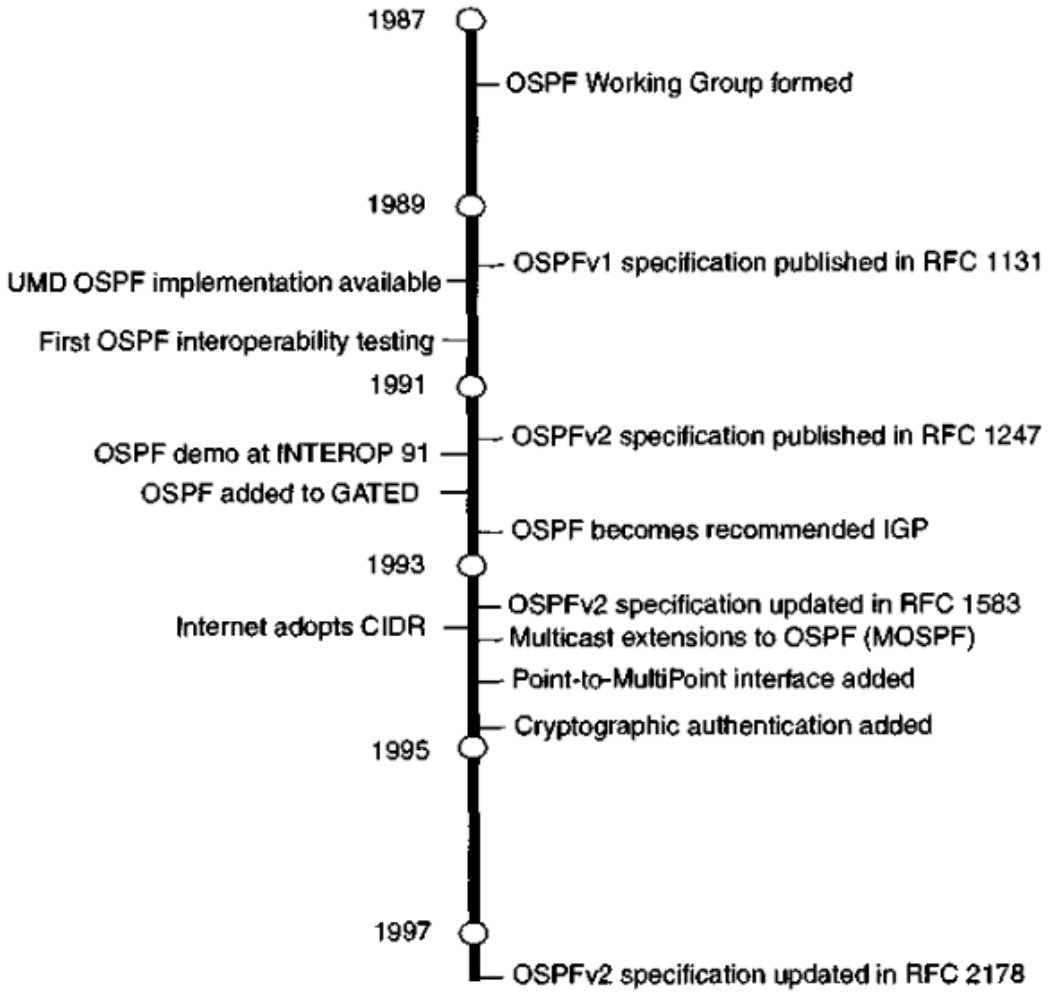

Рисунок 2.1. – Хронологія розвитку протоколу OSPF

Для розуміння першочергової мети групи розробки протоколу OSPF, треба пригадати, що собою являла мережа Internet в 1987 році. В ті часи це була в більшій мірі академічна та дослідницька мережа, яка фінансувалась урядом Сполучених Штатів Америки, де для маршрутизації всередині автономних систем використовувався протокол RIP, а для комунікації між ними використовувався EGP. Хоч обидва протоколи мали велику кількість недоліків, було прийняте рішення почати розвивати заміну саме протоколу

RIP. Основною причиною стала думка, що заміна Routing Information Protocol матиме ширше застосування, як в мережі Internet, так і в комерційних мережах TCP/IP [6].

Таким чином, основною вимогою до нового протоколу маршрутизації, який би використовувався в межах автономних систем, стала більша ефективність, ніж RIP. Групі розробки з IETF було поставлено задачу оптимізації споживання ресурсів нового протоколу, але при цьому збільшити швидкість пошуку іншого маршруту для пакетів даних у випадку зміни топології мережі, наприклад у випадку відмови маршрутизатора, або комутатора [6]. Іншими функціональними вимогами були встановлені [6]:

- більша змістовність метрики маршруту, яка б дала змогу прибрати обмеження на діаметр мереж;
- балансування по рівнозначним маршрутам, що дало б змогу реалізувати можливість знаходження декількох кращих маршрутів до призначеного маршрутизатора, за умови їх наявності;
- ієрархія маршрутизації, яка б дала можливість будувати великі домени маршрутизації;
- розподіл зовнішніх та внутрішніх маршрутів. Автономні системи, які використовували RIP, мали великі проблеми з визначенням типу маршруту. Протокол не бачив різниці між цими двома типами маршрутів;
- підтримка гнучкої схеми поділу на підмережі всередині автономних систем, задля більшої оптимізації використання адресних просторів;
- можливість контролювати додавання нових маршрутизаторів до домену OSPF, що підвищило б показники безпеки нового протоколу.

Перша версія технічної специфікації протоколу OSPFv1 була оприлюднена в жовтні 1989 року під назвою RFC 1131. Початкові практичні реалізації протоколу одразу допомогли виявити проблеми. Розробники

помітили, що OSPFv1 не завжди видаляв інформацію з таблиць маршрутизації, що могло призводити до перевантаження апаратного забезпечення, або до проблем зі створенням нових маршрутів. Цей недолік разом з рядом інших помічених технічних помилок змусив команду розробки переглянути специфікації OSPFv1 [6].

Оновлений варіант технічних специфікацій світ побачив у липні 1991 року під назвою RFC 1247. Майже одразу після виходу даної версії, протокол був розгорнутий в мережі Інтернет, тому команда розробки вже була обмежена у внесенні змін. Основною метою під час додавання нового тепер була підтримка працездатності робочої версії протоколу [6].

Після цього, можна вважати, що заміна для протоколу RIP була готова. Розробники протоколу OSPF взяли до уваги всі вимоги, які були висунуті [6]. З основними відмінностями протоколів RIP та OSPF можна ознайомитись в таблиці 2.1.

Таблиця 2.1.

Основні відмінності протоколів

RIP	OSPF
Ґрунтується на відстеженні вектору відстані	Ґрунтується на відстеженні стану каналів
Для побудови маршруту використовується алгоритм Белмана-Форда	Для побудови маршруту використовується алгоритм Дейкстри
Маршрутизатори надсилають оновлені таблиці маршрутизації сусіднім кожні 30 секунд	Маршрутизатори періодично генерують та надсилають список своїх сусідів (кожні 1-2 години)
Метрика: кількість хопів; найбільша кількість: 15	Метрика: «вартість» використання відповідного інтерфейсу
Таблиці маршрутизації надсилаються лише у сусідні вузли автономної системи	Таблиці маршрутизації розповсюджуються по всій топології мережі

З того моменту було випущено ще понад 20 офіційних документів від IETF, пов’язаних з протоколом динамічної маршрутизації OSPF. Основні документи, окрім вже описаних, більш детально перелічені в таблиці 2.2 [7].

Таблиця 2.2

Основні документації протоколу

Номер публікації	Офіційна назва	Рік написання
RFC 2328	OSPF Version 2	1998
RFC 3101	The OSPF Not-So-Stubby Area Option	2003
RFC 3630	Traffic Engineering Extensions to OSPF Version 2	2003
RFC 3623	Graceful OSPF Restart	2003
RFC 5340	OSPF for IPv6	2008
RFC 5709	OSPFv2 HMAC-SHA Cryptographic Authentication	2009

Згідно з вищезазначеними даними, в 2008 році було випущено документ RFC 5340 «OSPF for IPv6». Цей варіант технічної документації представив наступну версію протоколу OSPFv3. Основною функцією цієї версії є маршрутизація префіксів IPv6. Проте, незважаючи на те, що адреси в IPv6 було розширено до стандартизованих 128 біт, ідентифікація доменів або роутерів все ще виконується з використанням 32-бітних чисел [8].

За інформацією, наявною у відкритих джерелах, за весь час активного використання протоколу налічується 61 вразливість. Більш детально з

розподілом, що ґрунтується на році дослідження вразливості, можна ознайомитись на рисунку 2.2 [9].

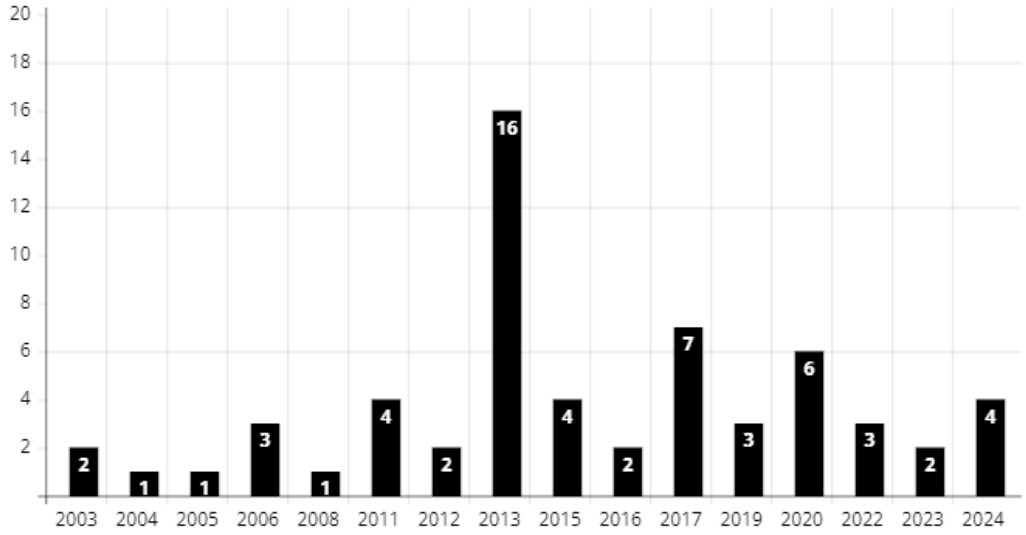

Рисунок 2.2 – Статистика документації вразливостей протоколу OSPF

Відповідно до знайденої інформації, найбільша кількість вразливостей була задокументована в 2013 році [9]. Аналіз даних вразливостей наведений у розділі 3.

2.2 Фундаментальні основи побудови

Протокол Open Shortest Path First – це протокол динамічної маршрутизації, що відноситься до сімейства протоколів внутрішньої маршрутизації, та ґрунтується на алгоритмі відстеження стану каналу. Оскільки, він належить до категорії протоколів внутрішньої маршрутизації, область його дії, зазвичай, обмежується однією автономною системою [10].

Алгоритм відстеження стану каналу дозволяє маршрутизаторам надсилати інформацію про поточну топологію своїм найближчим сусідам. Ці оголошення розсилаються так, що кожен пристрій маршрутизації автономної системи має повне розуміння її топології. Використовуючи дані схеми розташування, протокол динамічної маршрутизації будує маршрути для

доставлення пакетів даних, задіюючи для цього алгоритм Дейкстри. Його суть полягає в знаходженні найкоротшого шляху від потрібної вершини графа до всіх інших існуючих (Додаток А) [11]. Тобто, умовно, топологія будь-якої автономної системи при побудові таблиць маршрутизації перетворюється на неорієнтований граф, в якому кожна вершина – пристрій маршрутизації, а шлях для пересилання пакету даних від маршрутизатора А до маршрутизатора В, які лежать в межах однієї автономної системи, у звичайній ситуації, буде будуватись шляхом пошуку найкращого маршруту напряму, або через інші вершини графа [12].

Протокол має усі переваги, описані при порівнянні статичного та динамічного типу маршрутизації. Однак, варто зазначити, що технологія відстеження стану каналу, яка надає детальне розуміння топології автономної системи, дає змогу будувати маршрути відповідні певним критеріям. Наприклад, одним із критеріїв може бути обмеженість в обчислювальних ресурсах апаратного забезпечення або в кількості оброблюваних маршрутів одним маршрутизатором задля покращення швидкості та якості маршрутизації пакетів та якості обслуговування загалом [13].

З іншого боку, потенціал масштабування спричиненого збільшенням кількості пристроїв маршрутизації не є сильною стороною протоколу динамічної маршрутизації OSPF. Більша кількість маршрутизаторів спричиняє більшу частоту оновлень схеми автономної системи. Зростає частота – зростає й кількість часу на пошук та вирахування маршрутів в межах топології. При досягненні певної максимальної позначки в допустимій кількості апаратних засобів, система маршрутизації стає непридатною до використання. Маршрутизатори один за одним починають раптово перезавантажуватись, адже їм не вистачає обчислювальних здатностей для одночасної обробки великої кількості оновлень схем автономної системи та обрахування маршрутів. Ніяких попереджень при цьому не отримується. Компанія Cisco, що спеціалізується на розробці програмного та апаратного забезпечення, рекомендує обмежитись п’ятдесятьма маршрутизаторами в

одному домені [13]. Саме цей недолік і визначив долю протоколу OSPF, як протоколу внутрішньої маршрутизації.

Задля покращення швидкодії в процесі пошуку, обрахування та вибору маршрутів та при розсиланні оголошень про стан каналу в автономних системах з різними підходами до організації мереж з множинним доступом окремо обирається виділений маршрутизатор (DR) та резервний виділений маршрутизатор (BDR) [14].

Основною функцією виділеного маршрутизатора є керування процесом розсилання оголошень про стан каналу в межах топології окремої автономної системи. Місце розташування виділеного маршрутизатора може бути довільним, в межах схеми, однак, кожен інший пристрій маршрутизації налагоджує відносини сусідства з ним. При виявленні змін або збоїв в маршрутизації мережі, маршрутизатор, що це помітив, надсилає відповідне оголошення на виділений. Зі свого боку виділений маршрутизатор розповсюджує отриману інформацію усім пристроям в автономній системі. Обирається даний підтип пристрою за найбільшим пріоритетом, який в свою чергу може бути встановлений як вручну системним адміністратором, або адміністратором мережі, так і може бути успадкований в залежності від наданої маршрутизатору IP-адреси [14].

Резервний виділений маршрутизатор виконує ту саму роль, але лише в разі збоїв чи пошкоджень виділеного. Він також має відносини сусідства з кожним пристроєм маршрутизації в автономній системі, тому час недоступності мережі в екстрених ситуаціях мінімізується [14].

При налаштуванні OSPF в певній автономній системі, вона може бути поділена на окремі області для більшої зручності. Кожна область має свої бази даних інформації про стан каналів та окремі графи для швидкої побудови маршрутів. Однак, у випадку розділення топології, маршрутизатори можуть виконувати свої окремі функції, одночасно з основними. З класифікацією цих пристроїв за технічними специфікаціями RFC 2328 можна ознайомитись у таблиці 2.3 [14].

Таблиця 2.3

Класифікація маршрутизаторів протоколу OSPF

Номер	Тип маршрутизатора	Опис та функції
1	Внутрішній	Маршрутизатор, кожен активний інтерфейс якого має підключення всередині лише одної відповідної області
2	Граничний маршрутизатор області	Маршрутизатор, який має активні інтерфейси у двох, або більше областях; накопичує та передає інформацію про топології зон до яких має відносини у магістраль
3	Магістральний маршрутизатор	Маршрутизатор, який має активний інтерфейс у магістральній області; розповсюджує інформацію про топологію областей
4	Граничний маршрутизатор автономної системи	Маршрутизатор, який ділиться даними з іншими пристроями маршрутизації, що відносяться до інших автономних систем

За умови поділення автономної системи на області, маршрутизатор має інформацію про стан каналів інших пристроїв лише цієї області. Для доставлення пакетів даних між областями обов’язково має бути створена так звана магістраль [14].

Магістраль, або Area 0 – це область, що відповідає за розповсюдження інформації про топологію інших немагістральних областей і містить усі граничні маршрутизатори інших областей. При доставленні пакету даних поміж двома немагістральними областями використовується інформація отримана від магістрального пристрою маршрутизації [14]. Цей шлях можна розбити на три окремі секції [14]:

- шлях всередині однієї області до її граничного маршрутизатора;
- магістральний шлях від одного граничного маршрутизатора до іншого;
- шлях всередині потрібної області до пункту призначення.

Можливо, це може виглядати складно, однак, на практиці це спрощує процес, адже навіть в такому випадку маршрут будується за допомогою алгоритму Дейкстри, бо магістральних маршрутизаторів може бути декілька.

Кожен граничний маршрутизатор області за замовчуванням має зв’язок з магістраллю. Усі граничні пристрої маршрутизації області збирають топології немагістральних областей до яких вони мають підключення і передають цю інформацію у магістраль, а також приймають у відповідь звідти таку ж інформацію про області, прямого зв’язку з якими немає [14].

Граничні маршрутизатори автономної системи мають інформацію про топологію інших і розповсюджують її по всій автономній системі, до якої вони мають відношення [14]. Таким чином, у випадку наявності зв’язку одного з маршрутизаторів з іншими автономними системами, будь-який пристрій маршрутизації зможе за потреби отримати маршрут та здійснити передачу пакету даних.

Для збору потрібної інформації в межах області та загалом для обміну інформацією про стан каналів маршрутизатори мають бути у відносинах сусідства [14].

Відносини сусідства – це зв’язок між двома сусідніми пристроями маршрутизації, що налагоджується з метою обміну та синхронізації їх баз даних про стан каналів. Для встановлення відповідного зв’язку протокол динамічної маршрутизації OSPF використовує певні типи пакетів, детальніше ознайомитись з якими можна на таблиці 2.4 [14].

Таблиця 2.4

Типи OSPF-пакетів

Тип	Назва	Функція
1	Hello	Вивлення нових сусідів
2	DB Description	Зміст бази даних про стан каналів
3	Link-State Request	Запит та завантаження бази даних
4	Link-State Update	Оновлення бази даних
5	Link-State Ack	Підтвердження оновлення бази; підвердження одержання пакету Link-State Update

Більш детально з процесом встановлення відносин сусідства можна ознайомитись на рисунку 2.3.

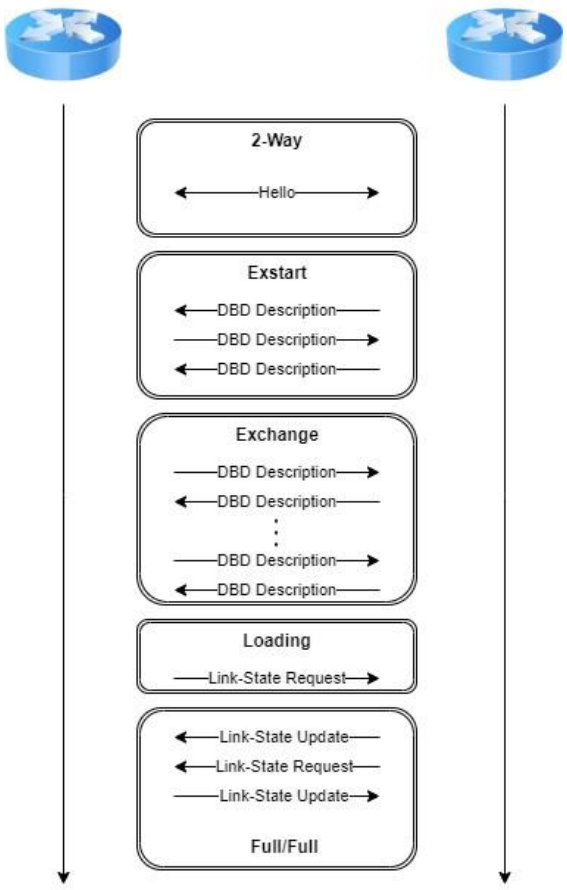

Рисунок 2.3 – Процес встановлення сусідства між двома маршрутизаторами

2.3 Структура LSA

Кожен пристрій маршрутизації автономної системи генерує та розсилає як мінімум одне оголошення про стан каналу (LSA, link-state advertisement). Дане оголошення містить інформацію про активні інтерфейси маршрутизатора, активні відносини сусідства, а також значення метрики при використанні того чи іншого активного інтерфейсу. Метрика в даному випадку визначає навантаження, що теоретично можливе при пересиланні пакетів

даних з використанням заданого інтерфейсу. Як вже було зазначено в порівнянні протоколів RIP та OSPF, в якості метрики у даному протоколі динамічної маршрутизації використовується вартість. Маршрут, котрий матиме меншу вартість, буде більш пріоритетним, що логічно. Вартість інтерфейсу є обернено пропорційним значенням до його пропускної здатності, тобто чим більша пропускна здатність, тим менше значення метрики [14]. Вартість вираховується за наступною формулою [15]:

Вартість: Задана пропускна здатність/пропускна здатність інтерфейсу.

Задана пропускна здатність за замовчуванням завжди рівна 10^8 біт/с [15]. Для більшого розуміння нижче наведена таблиця 2.5 зі значеннями метрики протоколу OSPF для різних типів інтерфейсів.

Таблиця 2.5

Значення вартості для різних типів інтерфейсів

Тип інтерфейсу	Задана пропускна здатність (біт/с)	Пропускна здатність інтерфейсу за замовчуванням (біт/с)	Вартість
10 Gig Ethernet 10 Гбіт/с	10^8	10^10	1
Gigabit Ethernet 1 Гбіт/с	10^8	10^9	1
Fast Ethernet 100 Мбіт/с	10^8	10^8	1
Ethernet 10 Мбіт/с	10^8	10^7	10
Serial 1544 Кбіт/с	10^8	1 544 000	64
Serial 128 Кбіт/с	10^8	128 000	781
Serial 64 Кбіт/с	10^8	64 000	1562

Використовуючи отримані в оголошенні про стан каналу дані, кожен маршрутизатор будує відповідні таблиці маршрутизації. При виявленні змін в

топології мережі або збоїв, виділений маршрутизатор розсилає оновлені оголошення та маршрути перераховуються. Варто зазначити, що забезпечується можливість зберігання та підтримка декількох маршрутів для пакетів даних, у випадку однаковості значення метрики інтерфейсу.

Кожне оголошення про стан каналу містить свій заголовок та основне поле, що містить всю необхідну інформацію для маршрутизації пакетів даних. Зі структурою заголовку оголошення про стан каналу можна ознайомитись на рисунку 2.4 [16].

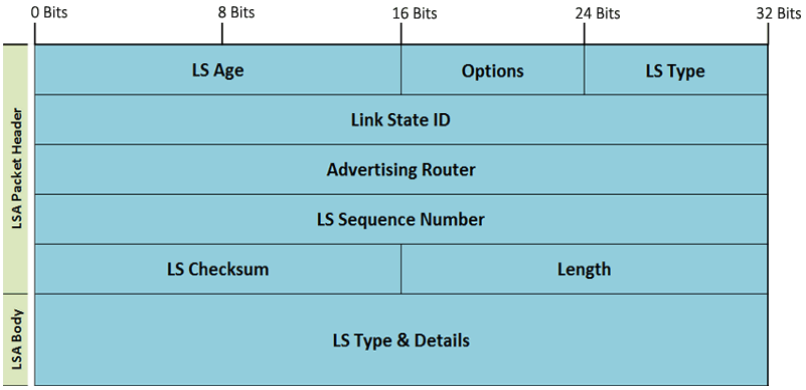

Рисунок 2.4 – Структура заголовку оголошення про стан каналу

Як можна побачити, будова заголовку є досить комплексною, адже вона містить 8 окремих комірок інформації.

2.3.1 LS age

Дане поле містить час, що пройшов від створення оголошення, у форматі 16-бітного цілого числа. Значення даної комірки ніколи не перевищить значення MaxAge для оголошення про стан каналу. Відповідно до технічних документацій протоколу OSPF, значення MaxAge по замовчуванню становить 3600 секунд, що еквівалентно одній годині. Якщо значення віку оголошення перевищить цей час, оголошення про стан каналу видаляється з бази даних маршрутизатора. У випадку отримання пристроєм маршрутизації двох оголошень з однаковим порядковим номером, першочергово перевіряється

саме це поле. Оголошення з меншим часом від створення зберігається, а інше видаляється, що дозволяє не засмічувати базу даних маршрутизатора [17].

2.3.2 Options

Дане поле містить додаткову інформацію, пов’язану з характеристиками та можливостями створеного оголошення. Воно може містити дані про: тип сервісу, підтримку групової передачі, тип області [17].

2.3.3 LS Type

Дане поле визначає формат та функцію створеного оголошення про стан каналу. Різні типи оголошень мають різні назви та функції. З основними типами таких оголошень можна ознайомитись у таблиці 2.6 [18].

Таблиця 2.6

Типи оголошень про стан каналу

Тип оголошення LSA	Опис
1	Даний тип має назву LSA маршрутизатора. Містить інформацію про стан інтерфейсів відповідного маршрутизатора. Поширюється в межах однієї області.
2	Даний тип має назву LSA мережі. Містить інформацію про перелік маршрутизаторів у області, де поширюється.
3	Даний тип має назву сумарне LSA мережі. Містить інформацію про маршрутизацію всередині області. Генерується граничним або резервним граничним маршрутизатором
4	Даний тип має назву сумарне LSA граничних маршрутизаторів. Містить усю потрібну інформацію про граничні маршрутизатори.
5	Даний тип має назву LSA зовнішніх автономних систем. Містить інформацію про маршрути поза межами даної автономної системи.

2.3.4 Link State ID

Дане поле визначає фрагмент домену маршрутизації, що описаний в оголошенні про стан каналу. У залежності від типу оголошення, ідентифікатор стану приймає значення описані у таблиці 2.7 [19].

Таблиця 2.7

Типи ідентифікаторів

Тип оголошення LSA	Опис ідентифікатору
1	Ідентифікатор приймає значення ідентифікатора маршрутизатора, що створив повідомлення
2	Ідентифікатор приймає значення IP-адреси граничного маршрутизатора області
3	Ідентифікатор приймає значення IP-адреси мережі призначення
4	Ідентифікатор приймає значення ідентифікатора маршрутизатора, що описаний в оголошенні як граничний
5	Ідентифікатор приймає значення IP-адреси мережі призначення

2.3.5 Advertising Router

Дане поле зазначає ідентифікатор маршрутизатора, що створив відповідне оголошення про стан каналу [20].

2.3.6 LS Sequence Number та Checksum

Порядковий номер оголошення має вигляд невід’ємного цілого 32-бітного числа. Використовується для ідентифікації старих та дублікатних оголошень. Чим більший порядковий номер, тим новіше оголошення отримує маршрутизатор [20].

Контрольна сума включає в себе всю інформацію з оголошення, окрім поля віку. Основною функцією даного поля є ідентифікація пошкоджень даних [20]. Зазвичай, це може статись у процесі доставлення або зберігання в пам’яті пристрою маршрутизації.

2.4 Механізми безпеки

Хоч протокол динамічної маршрутизації OSPF був створений досить давно, його технічні специфікації описують певні механізми, що можуть підвищити стан захищеності системи маршрутизації автономної системи. Це робить протокол певною мірою надійним і стійким до збоїв під час кібератак зловмисників.

Розробники протоколу з IETF заклали можливість доступу до пакетів даних маршрутизаторам лише проходячи процедуру автентифікації. Якщо пристрій проходить дану процедуру успішно, він автоматично стає довіреним і може брати участь в процесі маршрутизації пакетів даних. Кожен пакет протоколу OSPF має спеціальні поля автентифікації та типу автентифікації довжиною 32 та 16 біт відповідно. Обидва поля, за умови налаштування, складають механізм автентифікації пакетів [21]. Більш детально з заголовком OSPF пакету можна ознайомитись на рисунку 2.5 [22].

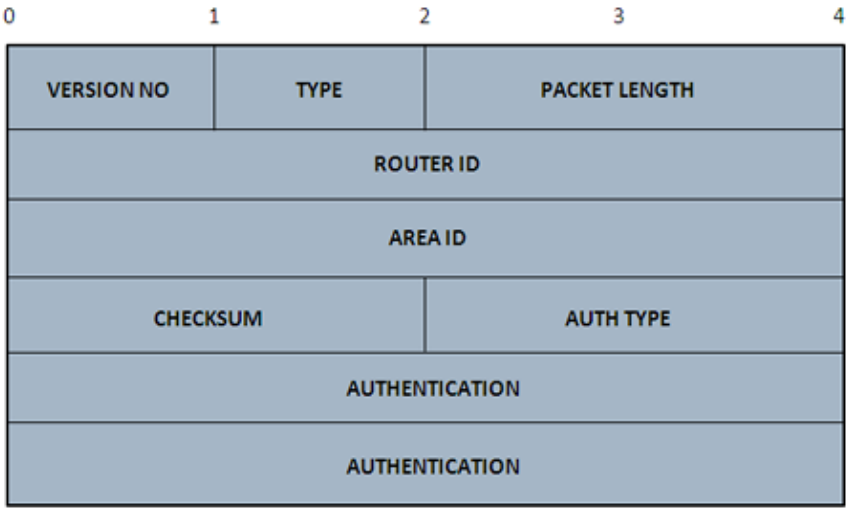

Рисунок 2.5 – Заголовок OSPF-пакету

Тип автентифікації, зазвичай налаштовується напряму на інтерфейсі. Однак, маршрутизатори певних виробників дають змогу налаштування для всіх інтерфейсів одночасно. Дане поле заголовку може мати наступні значення [23]:

- 0 – нульова автентифікація (відсутня);
- 1 – автентифікація з використанням незашифрованого паролю;
- 2 – автентифікація з використанням шифруванням паролів;

Звісно, найбільш стійкою до розкриття вважається автентифікація з криптографічним шифруванням паролів. Для цього налаштовується однаковий ключ для всіх OSPF-маршрутизаторів в автономній системі. Маршрутизатори в свою чергу генерують 32-бітні повідомлення з ключем і необхідною інформацією в якості вхідних даних і відправляє його до пунктів призначення. У випадку перехоплення такого пакету даних, зловмисники не матимуть змоги відновити ключ, адже згенеровані повідомлення являють собою односторонню хеш-функцію MD5 [24]. При спробі його підробити, зашифроване повідомлення буде понівечене, в порівнянні з оригіналом і в такому випадку зловмисники не матимуть змогу надурити маршрутизатор шляхом підміни оригінального повідомлення на модифіковане розшифроване повідомлення [25].

На даний момент, для забезпечення високого рівня безпеки маршрутизації, можна вибрати лише два варіанти алгоритму шифрування. Це алгоритми MD5 та HMAC-SHA. Однак, навіть у випадку використання криптографічної автентифікації повідомлень, автономна система лишається незахищеною від можливості перегляду та аналізу трафіку, хоч і зашифрованого.

Роботу протоколу динамічної маршрутизації OSPF супроводжує досить надійний та швидкий механізм розсилання оголошення про стан каналу, основна мета якого – забезпечення постійної синхронізації в кожній області автономної системи. При отриманні маршрутизатором оголошення про стан

каналу, він має назад у відповідь надіслати відповідний пакет LS Ack, що підтвердить отримання ним даного оголошення. Це допомагає виявляти втручання в процес маршрутизації, наприклад додавання зловмисного пристрою між двома вже існуючими. Також, у порівнянні з векторними протоколами маршрутизації, таким як RIP, бази даних, у даному випадку стану каналів маршрутизаторів, є розподіленими і синхронізованими. Тобто, допоки існуватимуть альтернативні шляхи, маршрутизатори матимуть змогу отримувати та надсилати оголошення незалежно від відмови певного вузла чи втручань зловмисників [25]. Також варто зазначити що оголошення про стан каналу розсилаються лише по маршрутизаторам-сусідам. Це сильно зменшує ризики компрометації та модифікації інформації на відміну від випадку розсилання таких оголошень одним маршрутизатором по всій топології автономної системи. Пристрої маршрутизації при роботі з протоколом OSPF додають до своїх баз даних лише згенеровану інформацію маршрутизатором-сусідом. Даного типу інформаційна незалежність допомагає швидше зрозуміти який з маршрутизаторів поширює неправильну інформацію у випадку можливої компрометації.

Ієрархічний механізм маршрутизації також виконує свою роль, як механізму безпеки, хоч і був спочатку спроектований для вирішення проблеми протоколу маршрутизації RIP – завеликі розміри таблиць маршрутизації. Використаний принцип вирішив вищезазначений недолік, а також оптимізував використання обчислювальних ресурсів маршрутизаторів. Більше того, це й покращує складову безпеки автономної системи. Пристрої однієї області напряму нічого не знають про пристрої іншої, а мають зв’язок лише через свої граничні маршрутизатори [26]. Тобто позаштатні ситуації в одній області прямим чином не впливають на маршрутизацію в іншій, а у випадку втручання зловмисника вся топологія автономної системи розкрита також не буде.

Інший принцип роботи протоколу, що також підвищує його захищеність від ворожих втручань, поміж спеціалістів отримав ім’я «fight-back». Суть його

роботи полягає у фільтрації створених оголошень про стан каналу маршрутизаторами. Одержання маршрутизаторами самостворених оголошень є абсолютно нормальним явищем, адже вони теж зберігаються в базах даних пристроїв. Механізм спрацьовує в моменті отримання маршрутизатором оголошення про стан каналу, де наявна сфальсифікована інформація про нього ж. В такому випадку, такий маршрутизатор одразу створює новий пакет даних та надсилає своїм сусідам для виправлення помилкової інформації та анулює пошкоджене оголошення [27]. Наприклад: маємо просту автономну систему з трьох пристроїв маршрутизації R1, R2 та R3. Уявімо, що зловмисник видає себе за маршрутизатор R1 та надсилає маршрутизатору R2 повідомлення з інформацією «Маршрутизатор R3 з IP-адресою X більше не існує». Маршрутизатор R2 на правах відносин сусідства передає цю ж інформацію на R3. R3 виявивши пошкоджений, а в нашому прикладі сфальсифікований, пакет шляхом перевірки полей Advertising Router та Router ID одразу створить нове оголошення про стан каналу з порядковим номером більшим на одиницю і поверне його на R2. Робота даного механізму відбила велику кількість різноманітних кібератак на автономні системи з підробкою оголошень про стан каналів.

Функція регулювання створення оголошень про стан каналу також додає протоколу незахищеності. Її суть полягає в затримці між створенням оголошень про стан каналу у 5 секунд. Цей механізм було створено для більш послідовної та стабільної роботи маршрутизатора під час активації процесу створення нового такого пакету даних. Однак, у випадку отримання маршрутизатором такого оголошення, що спровокує роботу механізму «fight-back», у зловмисника буде 5 секунд на те, щоб підкинути йому вже трохи виправлений підроблений пакет даних. В такому випадку, старий пакет видалиться, але новий пакет, що маршрутизатор буде розсилати по топології автономної системи, буде вже також зараженим [28].

2.5 OSPFv3

OSPFv2 – версія протоколу для IPv4, а OSPFv3 у свою чергу – версія протоколу для IPv6.

OSPFv3 також є протоколом динамічної маршрутизації, що ґрунтується на відстеженні стану каналу. Повна мапа топології тут також будується шляхом обміну оголошеннями про стан каналу між пристроями маршрутизації [29].

Хоч дана версія і створена для перенаправлення префіксів IPv6, певні схожості з попередньою версію все-таки також лишились [29]:

- типи пакетів;
- типи інтерфейсів;
- механізм знаходження сусідніх пристроїв;
- процедура розсилання оголошень про стан каналу.

Однак, окрім невеликої кількості схожостей, наявна набагато більша кількість відмінностей, детальніше ознайомитись з якими можна в таблиці 2.8 [29].

Таблиця 2.8

Відмінності між OSPFv2 та OSPFv3

№	OSPFv2	OSPFv3
1	Розмір заголовку пакету становить 24 байти	Розмір заголовку пакету становить 16 байтів
2	7 типів оголошень про стан каналу	9 типів оголошень про стан каналу
3	Існування лише одного екземпляру для посилання	Існування багатьох екземплярів для посилання
4	Працює в підмережах, а не в посиланнях	Працює на посиланнях, а не в підмережах
5	Потребує маску підмережі для формування відносин сусідства	Не потребує маску підмережі для формування відносин сусідства
6	MD5/HMAC-SHA використовуються для криптографічної автентифікації повідомлень	IPSec використовується для криптографічної автентифікації повідомлень

Висновки до розділу 2

Даний розділ містить детальний аналіз офіційних технічних специфікацій протоколу OSPF різних версій, його функціоналу захисту, а також оголошень про стан каналу та їх заголовків.

OSPF – протокол динамічної маршрутизації, що ґрунтується на відстеженні стану каналу. Маршрути для пакетів даних будуються за допомогою алгоритму Дейкстри, а основною структурною одиницею протоколу є оголошення про стан каналу різних типів, які поширюються між пристроями з побудованими відносинами сусідства.

Функціонал безпеки протоколу базується на надійному механізмі розсилання оголошень про стан каналу по топології автономної системи; механізмі фільтрації створених оголошень про стан каналу; механізмі затримки між створенням нових оголошень; механізмі криптографічної автентифікації повідомлень з різними алгоритмами шифрування.

Третя версія протоколу створена спеціально під IPv6 і має як схожості, так і відмінності з другою. У активному використанні наразі є обидві версії протоколів динамічної маршрутизації.

У наступному розділі проведено детальний аналіз кібератак, що використовують вразливості конструкції протоколу; експлуатацію вразливості криптографічної автентифікації повідомлень; розроблено класифікацію кібератак на протокол OSPF, за рівнем їх впливу на автономну систему.

РОЗДІЛ 3
РОЗРОБКА КЛАСИФІКАЦІЇ КІБЕРАТАК, ЩО ВИКОРИСТОВУЮТЬ ВРАЗЛИВОСТІ КОНСТРУКЦІЇ ПРОТОКОЛУ

3.1 Загальні відомості

Як вже було вказано в підрозділі історії створення протоколу OSPF, на сьогодні офіційно задокументовано 61 вразливість, а найбільша їх кількість припадає на 2013 рік.

Для більшого розуміння, необхідно одразу визначити вразливі місця в побудові протоколу, через які можуть бути спричинені злодіяння.

Першочерговим ризиком є найпростіша компрометація маршрутизатора. Наслідки можуть бути абсолютно різні: від зупинки роботи пристрою, що можливо призведе до колапсу в маршрутизації, до розповсюдження цим пристроєм завідомо неправильної інформації [30].

Принцип роботи механізму «fight-back» вже відомий. Однак, за певної послідовності дій це може спричинити відмову в обслуговуванні (DoS). Лише одне фальшиве оголошення про стан каналу може спричинити безліч оновлених пакетів даних, що розсилатимуться по всій топології автономної системи. Хоч, така реакція є очікуваною та ефективною, але при спробі здійснити атаку відмови в обслуговуванні на автономну систему це лише допоможе адже ще більше завантажить маршрутизатори. Також, у разі, якщо сфальсифіковане оголошення про стан каналу не доходить до маршрутизатора, який його створив – механізм не спрацює. Статись це може у випадку захоплення порушником магістрального маршрутизатора, що поєднує дві області. Він матиме можливість розповсюдити фальшиве оголошення по одній області, нібито від маршрутизатора з іншої області досягаючи ефекту обману [30].

Наступним вузлом роботи, що може спричинити ризик, є криптографічна автентифікація повідомлень. За умови її невикористання, зловмисники можуть розсилати фальшиві повідомлення про стан каналу, які здатні призводити до помилок в маршрутизації або пристроях. Використання простої автентифікації покращує ситуацію не дуже сильно. У 64-бітне поле автентифікації у заголовку пакета записаний простий пароль, а повідомлення передається у вигляді незашифрованого тексту. В такому випадку, при перехваті такого повідомлення, зловмисник спокійно дізнається пароль автентифікації повідомлень і матиме змогу самостійно поширювати будь-які пакети даних іншим маршрутизаторам. Увімкнена криптографічна автентифікація захищає автономну систему від зловмисних повідомлень які йдуть з-поза її меж. Однак, зловмисники також мають можливість здійснити атаку відмови в обслуговуванні шляхом додавання до пакетів даних завідомо невірних криптографічних ключів автентифікації, що спричинить виділення маршрутизатором великого обсягу обчислювальних ресурсів на відкидання таких пакетів. В результаті даних дій можливі як затримки в доставленні повідомлень маршрутизаторами, так і помилкове їх неприйняття, що може результувати у збій відносин сусідства, перерахування маршрутів і зміну таблиць маршрутизації [30].

Механізм перевірки часу від створення пакету також може бути вразливим до зловмисних дій. Навіть за умови використання криптографічної автентифікації повідомлень, поле Age не шифрується. Тобто, при перехопленні такого пакету даних, зловмисник має змогу змінити значення поле Age на значення MaxAge. В результаті, оголошення про стан каналу просто не прийметься маршрутизатором до бази даних, що спричинить нестачу інформації про маршрутизатор, який породив цей пакет даних [30].

Повертаючись до вразливостей, що були задокументовані та опубліковані у відкритих джерелах, треба наголосити, що стосуються вони як протоколу, так і різних пристроїв маршрутизації різних виробників. Найвідоміша публікація має назву CVE-2013-0149, суть якої полягає в

неналежній перевірці оголошень про стан каналу першого типу, до того як починається робота з базою даних, що спричиняло відмову в обслуговуванні або перехоплення конфіденційної інформації. Дана вразливість охоплює певні версії програмного забезпечення пристроїв маршрутизації Cisco [31]. Інші 11 публікацій, які теж були опубліковані в 2013 році, мають таку саму вразливість, однак стосуються вони іншого програмного забезпечення, а значить й інших виробників маршрутизаторів. Це і D-Link, і HP, і IBM. Дані вразливості були оцінені Національним інститутом стандартів і технологій оцінками CVSS 5.2-8.5. За класифікацією, весь цей перелік можна назвати вразливостями середнього та високого рівня [32].

Однією з останніх задокументованих вразливостей є CVE-2024-20313. Дана вразливість стосується програмного забезпечення Cisco IOS XE, а суть її полягає в неправильному підтвердженні пакетів оновлень протоколу, які оброблюються пристроєм маршрутизації. Це може дати можливість неавтентифікованому зловмиснику спричинити атаку типу DDoS, шляхом неочікуваного перезавантаження ураженого пристрою [33]. Захист від даної атаки буде описаний в розділі 4.

3.2 Disguised LSA

Вперше про атаку даного типу стало відомо ще в 1992 році. Текст експлуатації вразливості був опублікований у книзі «Interconnections: Bridges and Routers» автора Радії Перлман, американської програмістки та мережевої інжерки [34].

Вразливість для виконання атаки походить з третьої секції офіційної технічної документації протоколу OSPF RFC 2328. Відповідно до цього документу два оголошення про стан каналу вважаються ідентичними у разі співпадіння полів [35]:

- Sequence Number;
- Checksum;

- Age (+/- 15 хвилин);

Однак, проблемою є те, що тіло оголошення не переглядається взагалі. Навіть у випадку різниці в фактичному часі життя пакету в 15 хвилин, оголошення все одно вважатимуться однаковими.

Експлуатація вразливості полягає в розповсюдженні оголошення про стан каналу з завідомо помилковою інформацією від імені маршрутизатора-жертви, встановлюючи значення вищезазначених полей такими ж як і у дійсного оголошення. Через це атака й отримала таку назву, що в перекладі означає «замаскований LSA». В момент отримання маршрутизатором-жертвою замаскованого оголошення, механізм «fight-back» не спрацює. Причиною є те, що маршрутизатор вважатиме це оголошення дублікатом свого останнього валідного пакету даних. Однак, всі інші пристрої топології також вважатимуть його лише копією справжнього оголошення, тому не додаватимуть це до своїх баз даних. Схематично атака зображена на рисунках 3.1 та 3.2 [35].

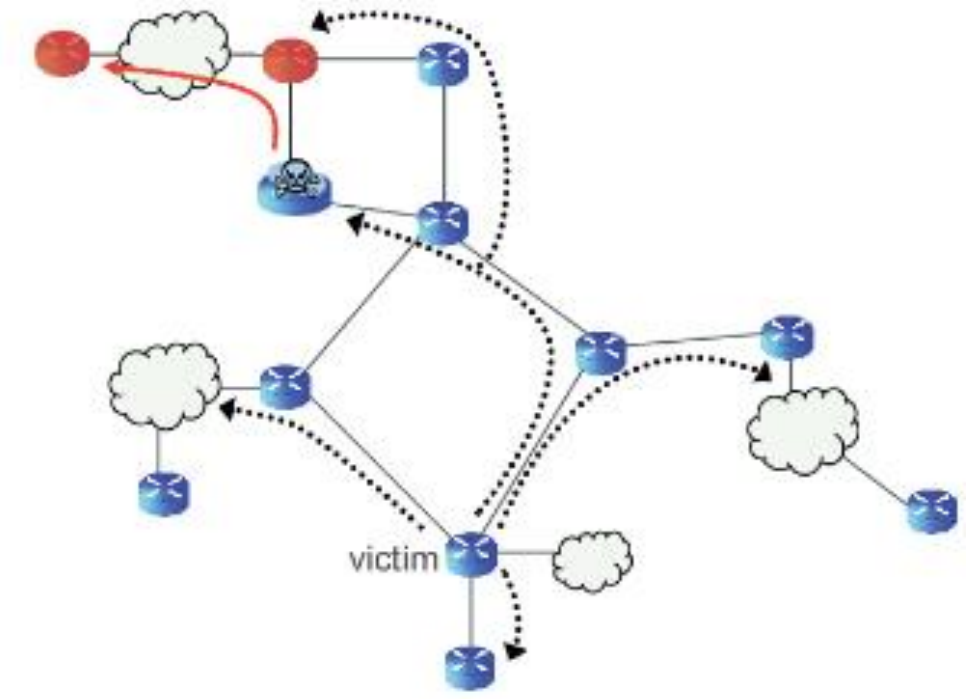

Рисунок 3.1 – Маршрут пакетів даних до атаки

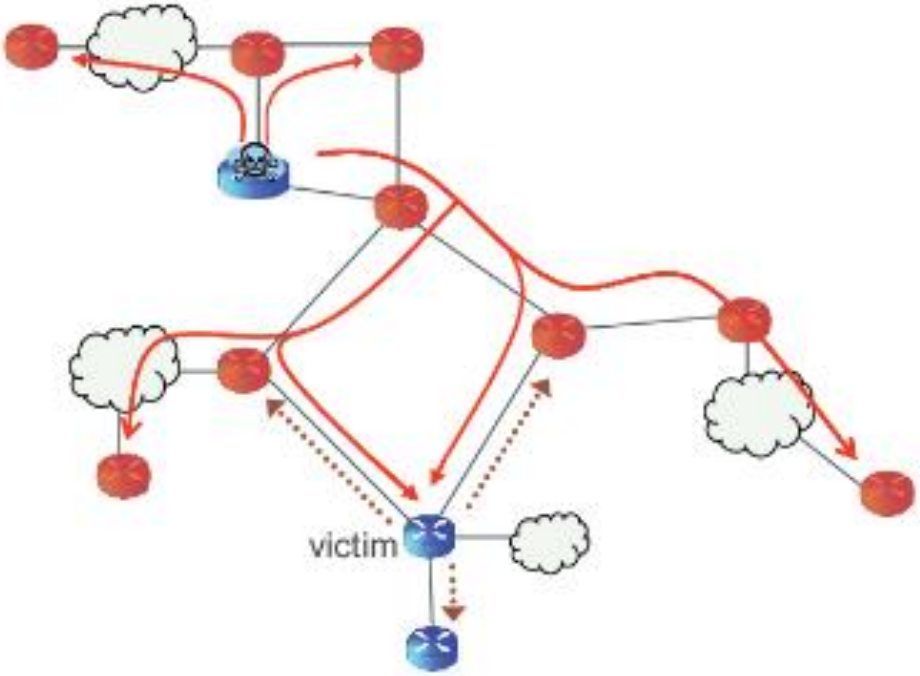

Рисунок 3.2 – Результат реалізації атаки

Більш цікавим підходом до реалізації є оголошення зловмисником замаскованого оголошення одразу після створення дійсного оголошення про стан каналу. З використанням даної техніки, такий пакет від зловмисника може розповсюдитись по топології автономної системи швидше за оригінал, і саме він буде записаний до баз даних про стан каналу усіх маршрутизаторів. Оригінальний пакет даних, дістався до вже заражених маршрутизаторів, буде відкинутий, так як вважатиметься повторним. Задля розуміння, зловмисник перед створенням такого оголошення може спровокувати роботу механізму «fight-back» для більш точного і вдалого початку розповсюдження фейкового пакету даних [36].

Для експлуатації вразливості та виконання атаки зловмиснику залишається лише заповнити три поля замаскованого оголошення ідентично до оригінального. Знову у гру вступають прогалини в офіційній документації протоколу, а саме:

- поле Age встановлюється зі значенням «0», так як маршрутизатори приймають оголошення з проміжком створення

у 15 хвилин, а так як сфальсифіковане оголошення буде надсилатись майже одночасно – цього не буде помічено;

- поле порядкового номеру будь-якого нового оголошення, згідно з офіційними документами, буде більшим на одиницю;
- поле Checksum вираховується мануально за кілька секунд з допомогою калькулятора і перегляду 2-3 попередніх оголошення про стан каналу.

3.3 Single Path Injection

Даний тип атаки досить схожий на атаку Disguised LSA, так як теж пов’язаний з розсиланням фальсифікованих оголошень. Але на відміну від вищезгаданої атаки, для реалізації даного типу нападу на мережеву інфраструктуру достатньо лише одного фейкового оголошення про стан каналу для зміни таблиць маршрутизації всіх пристроїв області, або навіть автономної системи [37].

Вразливість для виконання даного типу атаки була знайдена в 13 секції офіційної технічної документації протоколу OSPF, в якому описана процедура обробки отриманих пакетів Link-State Ack маршрутизаторами. Пристрій має ігнорувати поточний пакет підтвердження и розглядати вже наступний, у випадку коли пакет даних не має у собі екземпляра для повторної передачі маршрутизатору-сусіду [37]. Під час виникнення такої ситуації пристрій маршрутизації просто видалить такий пакет підтвердження.

Була знайдена можливість розповсюджувати помилкові оголошення про стан каналу через проміжний пристрій маршрутизації до пристрою-цілі без провокування механізму «fight-back».

Для більшої наочності, приклад топології в якій може статись така атака зображений на рисунку 3.3.

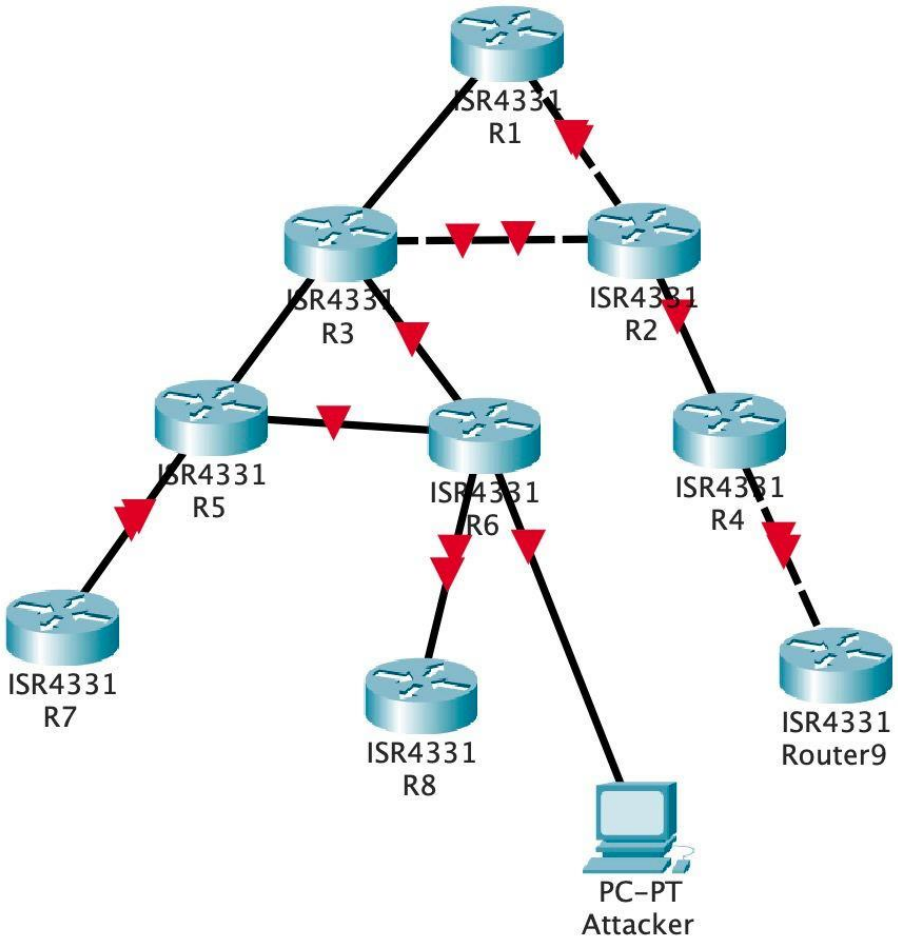

Рисунок 3.3 – Приклад топології для атаки

Відповідно до даної топології, проміжним маршрутизатором може виступати лише маршрутизатор R6 і передати сфальсифіковані дані пристрою R8. Для правильного створення потрібного оголошення зловмиснику необхідно вказати IP-адресу та Router ID маршрутизатора R6, як адресу джерела та ідентифікатор відповідно. При отриманні такого пакету даних, пристрій R8 вважатиме таке оголошення про стан каналу відправленим з R6 і збереже його до своєї бази даних. Після чого, свої активні з'єднання R6 передає всім іншим маршрутизаторам топології. Використовуючи дану особливість, у зловмисника є можливість обійти механізм «fight-back», навіть у випадку отримання пристроєм R6 пакету Link-State Ack, що включатиме сфальсифіковане оголошення про стан каналу від маршрутизатора R8 [37].

Коли злодій надсилає підробне оголошення про стан каналу з ідентифікаційною інформацією проміжного маршрутизатора, перелік повторної передачі стану каналів не матиме жодної інформації про такий пакет [37]. Причина досить проста – пристрій його не створював. Саме тому, коли такий проміжний маршрутизатор отримає пакет даних Link-State Ack, що міститиме це фальшиве оголошення, він просто його відкине і не запустить механізм «fight-back».

Однак, для виконання атаки даного типу має обов’язково виконуватись одна умова: між проміжним пристроєм та маршрутизатором-жертвою має бути лише один шлях передачі даних [37]. З огляду на дану умову, можна зробити висновок, що пар пристроїв маршрутизації навіть на оглядовій топології більш ніж достатньо, це R5-R7, R4-R9, R2-R4. Тобто зловмиснику для підготовки потрібно лише переконатись, що така пара існує.

Дана атака може стати підґрунтям для подальших атак на підмережі з метою перехоплення трафіку або переливання трафіку в так звану «чорну діру». Метод перенаправлення трафіку автономної системи на нульовий маршрут використовується для запобігання досягнення цільової точки шкідливих пакетів даних під час DDoS атаки. Проте, у випадку його застосування без потреби та відома адміністраторів мережі, система маршрутизації просто не доставлятиме необхідні пакети даних від відправника до отримувача [37].

3.4 Adjacency Spoofing

При виконанні шлюзом ролі OSPF-маршрутизатора в автономній системі, він динамічно знаходить своїх сусідів і періодично передає їм пакети типу «Hello» по своїм активним каналам зв’язку. Це повідомлення містить дані про всі маршрутизатори та їх ідентифікатори.

У момент, коли зловмисник отримує доступ до шлюзу певної автономної системи, першочерговою його задачею є перехоплення пакетів

типу «Hello», що розсилаються, з метою отримання інформації про ідентифікатори пристроїв маршрутизації усієї топології, номери областей, інтервали життя оголошень про стан каналу, тип автентифікації пакетів, що використовується в процесі маршрутизації [37]. Всі ці параметри можуть допомогти злодію створити фальсифікований «Hello»-пакет, однак йому варто врахувати, що ідентифікатор маршрутизатора-жертви, який був обраний, має бути більшим, аніж той, що має шлюз.

У багатьох випадках шлюз ще й являє собою виділений маршрутизатор. Атакувальник може здійснити спробу встановлення відносин сусідства з ним шляхом надсилання фальшивого пакету типу «Hello», видаючи себе за фактично сусідній маршрутизатор за мапою топології. Після отримання такого повідомлення, шлюз у відповідь відправить опис бази даних. В цей же час між зловмисником і шлюзом відбувається обмін оголошеннями про стан третього типу за допомогою повідомлень DBD. Після такого обміну, маршрутизатор-жертва, що виконує роль шлюзу, виконує запит нових оголошень про стан каналу і отримує сфальсиковані від зловмисника. Ну, і останнім кроком, це підробне оголошення розповсюджується по всій топології області або автономної системи, в разі її нерозбиття. Процес також зображено на рисунку 3.4 [37].

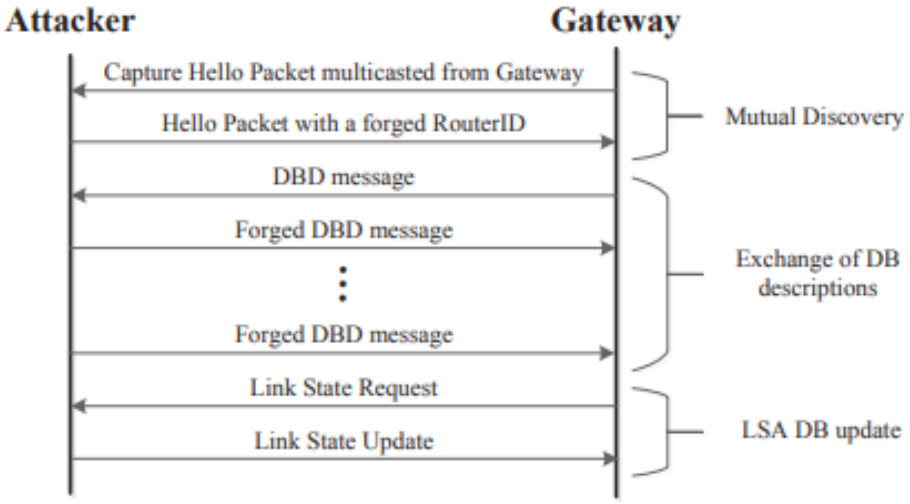

Рисунок 3.4 – Процес встановлення відносин сусідства із зловмисником

Після виконання даного типу атаки, маршрутизатор-шлюз сприйматиме зловмисника як сусіда, що відкриває дуже великі можливості для останнього [37]. Він матиме можливість розповсюджувати будь-які фейкові повідомлення і змінювати маршрути трафіку за своїм бажанням, адже кожен маршрутизатор, який отримає підробне оголошення про стан каналу, стовідсотково змінить свою таблицю маршрутизації.

На відміну від атаки типу Remote False Adjacency, де зловмисник намагається видати себе за фантомний маршрутизатор, тут у злодія є можливість видати себе справжнім маршрутизатором автономної системи. Приклад схеми даної атаки зображений на рисунку 3.5 [38].

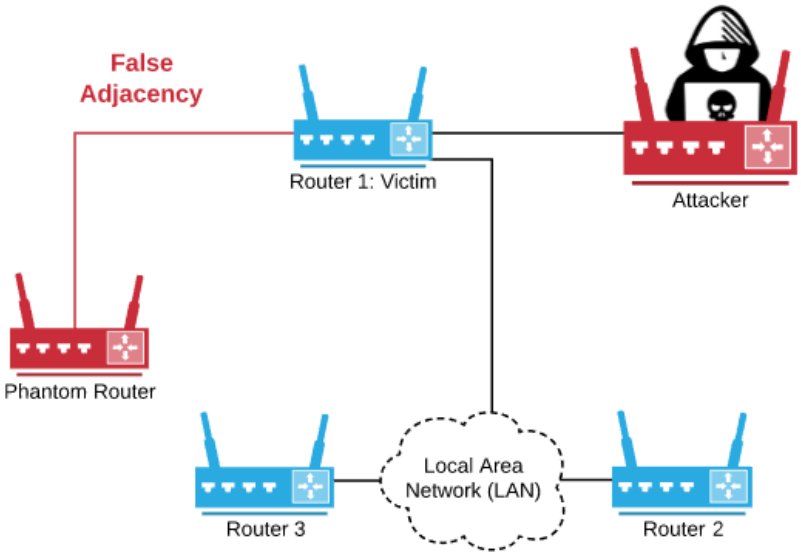

Рисунок 3.5 – Приклад можливої топологічної реалізації атаки Remote False Adjacency

Однак, все одно, ці два типи є схожими за принципом отримання привілеїв в системі маршрутизації, хоч і Adjacency Spoofing, на мою думку, є більш універсальною та потужною. Атака даного типу, а саме її послідовність, може бути використана не тільки при експлуатації вразливості протоколу маршрутизації, а й інших комп’ютерних технологій.

3.5 Атаки з фальсифікацією порядкового номера повідомлення

Атаки, що пов’язані з фальсифікацією порядкового номеру оголошення про стан каналу мають на меті створити труднощі в роботі системи маршрутизації, тобто, можливе переповнення баз даних, перевантаження апаратного забезпечення.

На даний момент відомо про дві атаки, що експлуатують недоліки технічної документації протоколу – Seq++ та MaxSeq [39].

З назви першої атаки, можна зрозуміти, що основним завданням для зловмисника стає перехоплення оголошення про стан каналу та інкрементація його порядкового номеру. Він також має можливість перерахувати контрольні суми, однак, більшого ефекту це не додасть. Після виконання вищезазначених дій, злодію достатньо запустити сфальсифіковане оголошення про стан каналу назад в систему маршрутизації. Після отримання редагованого пакету даних, маршрутизатор, що згенерував це оголошення спочатку, запускає процес роботи механізму «fight-back» [39]. Потенційно можливими наслідками є сповільнення процесу маршрутизації, у випадку масованого надсилання пакетів такого виду для різних маршрутизаторів автономної системи.

Атака MaxSeq також пов’язана зі зміною порядкового номеру оголошення про стан каналу [40]. Однак, на відміну від вищезгаданої, порядковий номер змінюється на максимальний і надсилається назад в систему маршрутизації. Маршрутизатор, що створив оригінальне оголошення, звісно його відкине, а при спрацюванні механізму «fight-back» створиться нове оголошення з мінімально можливим порядковим номером.

Хоч, ці атаки різняться за принципом модифікації одного й того ж поля, але ефект лишається однаковим. Проте, слід зазначити, що атаки даного типу досить легкі в імплементації та можуть завдати великої шкоди системі маршрутизації автономної системи, тому їх популярність ніколи не зменшиться.

3.6 Реалізація атаки на автентифікацію повідомлень

RFC 5709, що був написаний в 2009 році, описує криптографічну автентифікацію повідомлень з використанням алгоритму шифрування HMAC-SHA. До цього часу у використанні була лише хеш-функція MD5, що є вразливою до атак.

Для експлуатації даної вразливості використана така ж топологія емульованої системи, які і при впровадженні механізмів захисту (див. рисунок 4.1). Для перевірки захищеності пакетів даних в такій системі маршрутизації була проведена спроба атаки на автономну систему. Налаштування автентифікації наведені на рисунку 3.6.

```
R3(config-router)#area 0 authentication message-digest
R3(config-router)#exit
R3(config)#int se0/0
R3(config-if)#ip ospf message-digest-key 1 md Master_DIPLOMA
R3(config-if)#exit
R3(config)#
```

Рисунок 3.6 – Налаштування криптографічної автентифікації на маршрутизаторі

Встановлений пароль для автентифікації між маршрутизаторами R2 та R3 – «Master_DIPLOMA».

Так як автономна система, у нашому випадку, є емульована, було зроблено припущення, що зловмисник має доступ до аналізу трафіку системи маршрутизації і нічого не знає про налаштування безпеки.

Наступним кроком, видаючи себе за порушника цілісності системи маршрутизації, необхідно отримати пакети даних, що передаються між відповідними пристроями маршрутизації. Для цього була використана утиліта Wireshark, що дає змогу перехоплювати на аналізувати пакети [41]. Результат виконання даного кроку можна побачити на рисунках 3.7 та 3.8.

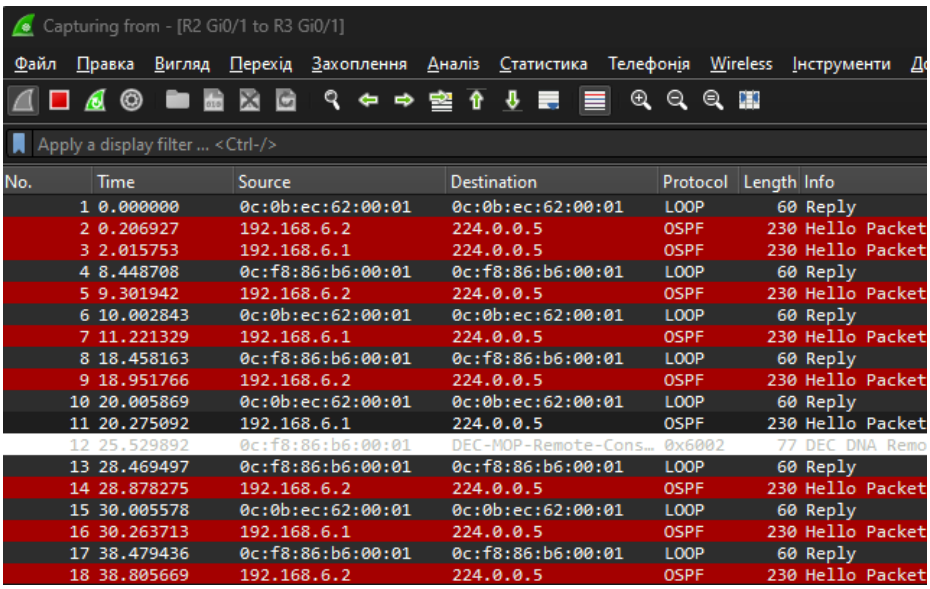

Рисунок 3.7 – Перехоплення пакетів з використанням Wireshark

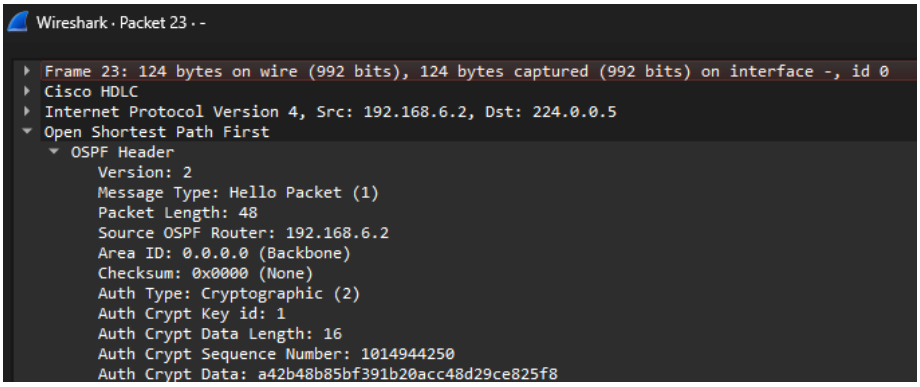

Рисунок 3.8 – Заголовок перехопленого OSPF Hello-пакету

Підсумком даної цілеспрямованої діяльності став файл формату .pcapng з переліком пакетів типу OSPF-Hello.

Згідно з рисунком 3.6.3, поле «Auth Crypt Data Length» має значення 16, тобто 128 біт. Дана інформація дає зрозуміти, що між відповідними пристроями маршрутизації встановлена криптографічна автентифікація з використанням алгоритму хешування MD5. Хоч певна інформація міститься в заголовку пакету перехопленого за допомогою Wireshark, використати її неможливо, адже вона неповна. Було проведено кілька спроб розшифрування даних, взятих одразу з заголовку пакету, однак успішного завершення вони не мали.

На цьому робота з Wireshark закінчена. Для вивільнення та читання хешів з отриманого дампу трафіку було використано утиліту «ettercap», що встановлена на віртуальну машину під керуванням операційної системи Kali Linux [42]. Результати можна побачити на рисунку 3.9.

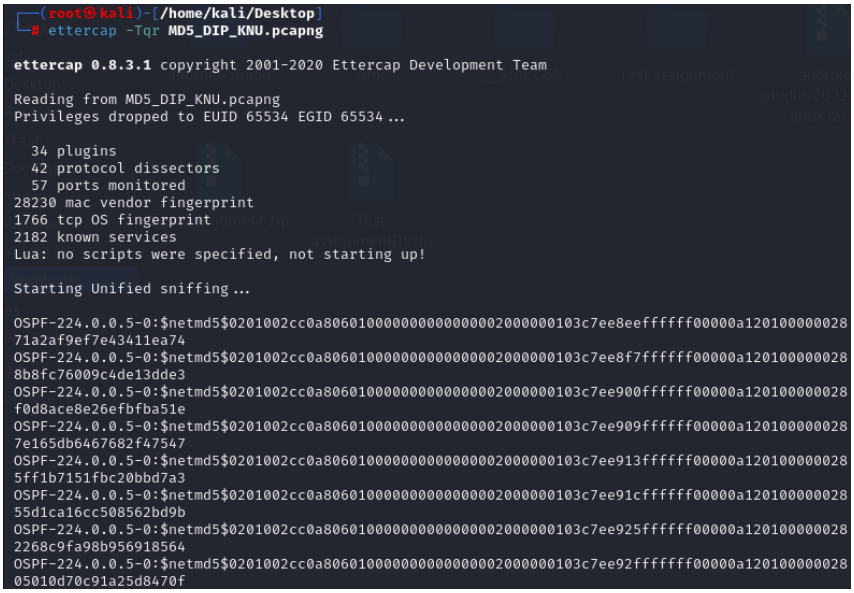

Рисунок 3.9 – Вивільнення хешів з дампу

Утиліта спрацювала бездоганно. Вивільнені хеші потребують змін, адже вони містять початкові символи, які не мають ніякого відношення до криптографічної автентифікації, а просто вказують на використаний протокол та адресу шлюзу. Результат видозмін можна побачити на рисунку 3.10.

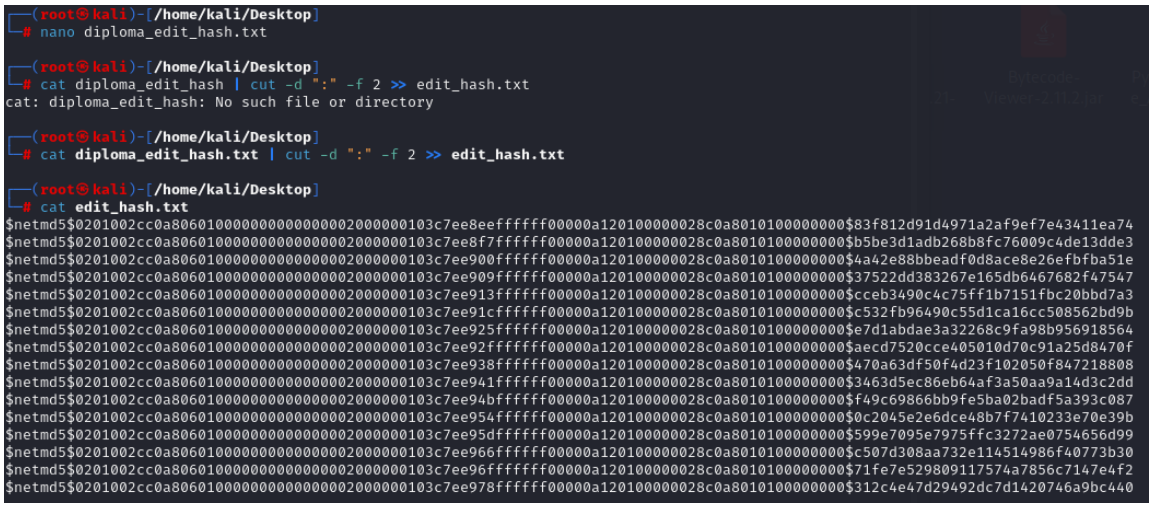

Рисунок 3.10 – Модифікація хешів

Останньою дією є спроба зламу відформатованих хешів. Для цього була використана утиліта John The Ripper, яка підтримує безліч типів хешів та шифрів [43]. Результат виконання процедури взлому можна побачити на рисунках 3.11 та 3.12.

```
┌──(root㉿kali)-[/home/kali/Desktop]
└─# john --format=net-md5 edit_hash.txt
Using default input encoding: UTF-8
Loaded 16 password hashes with 16 different salts (net-md5, "Keyed MD5" RIPv2, OSPF, BGP, SNMPv2 [MD5 32/64 or dynamic_39])
No password hashes left to crack (see FAQ)

┌──(root㉿kali)-[/home/kali/Desktop]
└─# john --show --format=net-md5 edit_hash.txt >> output.txt
```

Рисунок 3.11 – Робота утиліти John The Ripper

```
┌──(root㉿kali)-[/home/kali/Desktop]
└─# cat output.txt
?:Master_DIPLOMA
?:Master_DIPLOMA
?:Master_DIPLOMA
?:Master_DIPLOMA
?:Master_DIPLOMA
?:Master_DIPLOMA
?:Master_DIPLOMA
?:Master_DIPLOMA
?:Master_DIPLOMA
?:Master_DIPLOMA
?:Master_DIPLOMA
?:Master_DIPLOMA
?:Master_DIPLOMA
?:Master_DIPLOMA

16 password hashes cracked, 0 left
```

Рисунок 3.12 – Зламаний ключ автентифікації

Спроба атаки виявилась успішною, а слово-пароль було розшифроване з використанням потрібних утиліт.

Даний факт ставить під сумнів безпечність використання алгоритму хешування MD5 при налаштуванні криптографічної автентифікації повідомлень, адже у випадку отримання такого паролю не дослідницьких цілях, а в зловмисних – буде отриманий доступ до вмісту пакетів даних, що порушує конфіденційність автономної системи, а також підвищує ймовірність розсилання фальсифікованих пакетів даних.

3.7 Реалізація атаки Adjacency Spoofing

Детальний аналіз атаки Adjacency Spoofing, наведений у підрозділі 3.4, дає зрозуміти, що основною дією, яку треба виконати зловмиснику для успішної експлуатації вразливості – це надсилання сфальсифікованого пакету OSPF-Hello до шлюзу автономної системи. Саме це запустить процес встановлення відносин сусідства з зловмисним пристроєм.

Для емуляції процесу атаки зі сторони зловмисника була створена окрема автономна система з топологією, зображеною на рисунку 3.13. Окремо була налаштована маршрутизація з використанням протоколу OSPF.

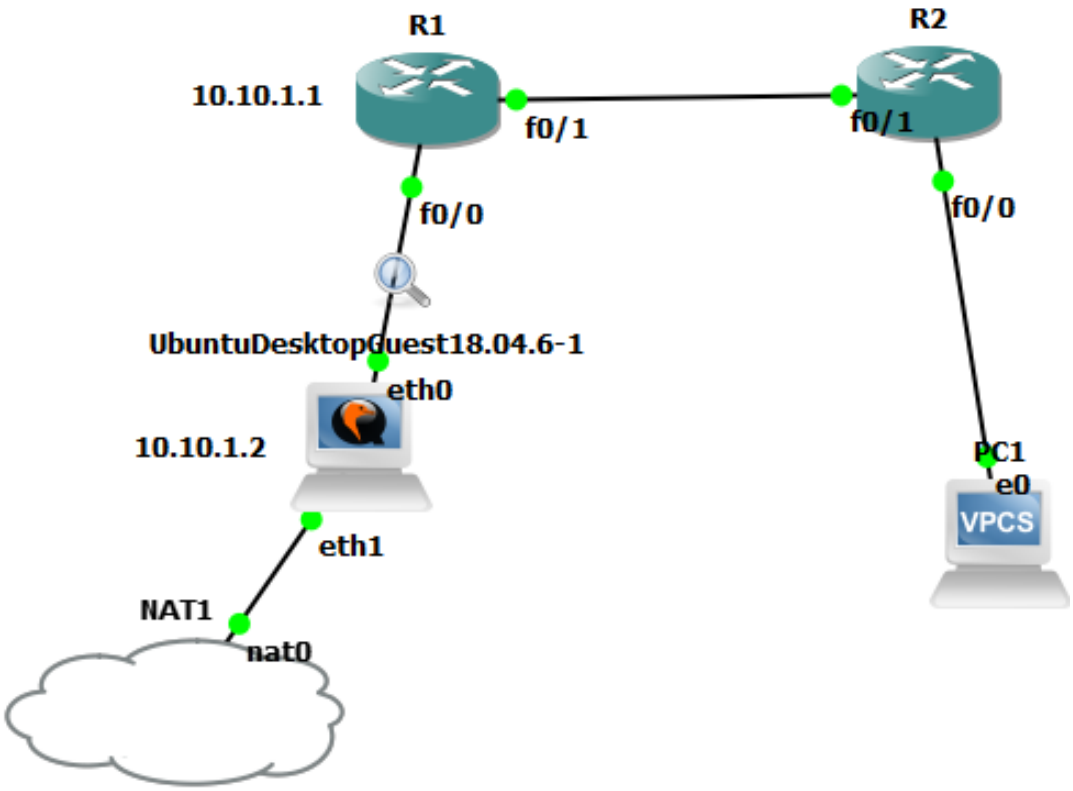

Рисунок 3.13 – Топологія, що буде використана для атаки

Зловмисник отримав повний доступ до персонального комп’ютера «UbuntuDesktop» шляхом надсилання на пошту користувачу пристрою фішингового повідомлення з вбудованим скриптом (або будь-яким іншим способом, адже це впливає на перебіг досліджуваної атаки).

Маршрутизатором-жертвою обирається пристрій R1. Для подальшої впевненості було переглянуто список маршрутизаторів-сусідів. Результати можна побачити на рисунку 3.14.

```
R1#show ip ospf neigh

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.10.3.1         1   FULL/DR         00:00:31    10.10.2.2       FastEthernet0/
1
```

Рисунок 3.14 – Список маршрутизаторів-сусідів R1 до атаки

Активні відносини сусідства встановлені з пристроєм маршрутизації R2.

Першочерговим кроком є отримання перехопленого пакету OSPF-Hello, який надсилається маршрутизатором R1 на шлюз. Він містить всю необхідну інформацію для експлуатації вразливості. Для виконання даної дії був використаний scapy. Scapy – це комплекс інструментів обробки пакетів даних комп'ютерних мереж, написаний на мові програмування Python [44]. Перехопимо пакет даних за наступним критерієм: «кінцева точка призначення – IP-адреса 224.0.0.5». Результат можна побачити на рисунку 3.15.

```
>>> D = sniff(filter="ip dst 224.0.0.5",count=6)
>>> D.summary()
Ether / 10.10.1.1 > 224.0.0.5 ospf / OSPF_Hdr / OSPF_Hello / OSPF_LLS_Hdr
Ether / 10.10.1.1 > 224.0.0.5 ospf / OSPF_Hdr / OSPF_Hello / OSPF_LLS_Hdr
Ether / 10.10.1.1 > 224.0.0.5 ospf / OSPF_Hdr / OSPF_Hello / OSPF_LLS_Hdr
Ether / 10.10.1.1 > 224.0.0.5 ospf / OSPF_Hdr / OSPF_Hello / OSPF_LLS_Hdr
Ether / 10.10.1.1 > 224.0.0.5 ospf / OSPF_Hdr / OSPF_Hello / OSPF_LLS_Hdr
Ether / 10.10.1.1 > 224.0.0.5 ospf / OSPF_Hdr / OSPF_Hello / OSPF_LLS_Hdr
```

Рисунок 3.15 – Перехоплені пакети OSPF_Hello

Із використанням вищеописаного програмного забезпечення були отримані потрібні пакети даних. Така кількість для перехоплення була обрана для запевнення того, що все спрацювало вірно і помилкових пакетів не показано. За коротким описом можна зрозуміти, що породжені вони були пристроєм маршрутизації R1. Для розуміння побудови сфальсифікованого

пакету, було переглянуто вміст одного з перехоплених. З результатами можна ознайомитись на рисунку 3.16.

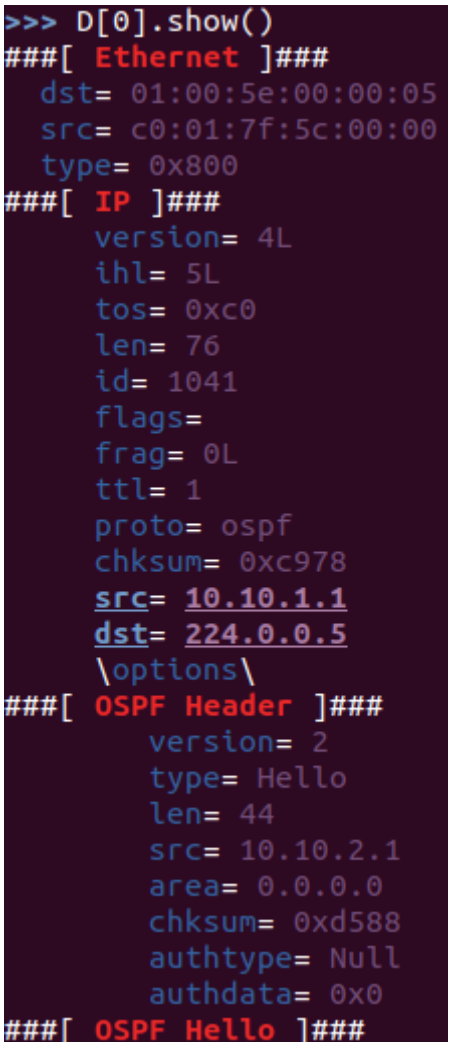

Рисунок 3.16 – Вміст перехопленого пакету OSPF-Hello

Після останньої виконаної дії, можна починати будувати сфальсифікований пакет даних. З інформації в заголовку перехопленого пакету, зрозуміло, що автентифікація повідомлень не налаштована, отже шлях до успішної експлуатації вразливості істотно спрощується.

Створюватиметься та надсилатиметься пакет даних також з використанням комплексу scapy. Для початку було проведено перевірку правильності роботи комплексу та чи надсилатимуться взагалі будь-які пакети-даних через активний інтерфейс. Було створено довільний IP-пакет та надіслано через інтерфейс кінцевого пристрою, через який буде відбуватись

подальша експлуатація вразливості [45]. Результати можна побачити на рисунках 3.17 та 3.18.

```
>>> sendp(Ether()/IP(dst="1.2.3.4",ttl=(1,4)), iface="ens3")
....
Sent 4 packets.
```

Рисунок 3.17 – Надсилання пакету даних

223	727.745755	192.168.133.141	1.2.3.4	IPv4	34
224	727.746360	192.168.133.141	1.2.3.4	IPv4	34
225	727.746683	192.168.133.141	1.2.3.4	IPv4	34
226	727.746957	192.168.133.141	1.2.3.4	IPv4	34

Рисунок 3.18 – Перехоплені новостворені пакети

Всі компоненти працюють вірно, отже можна починати будувати потрібний OSPF-Hello пакет. Для цього покроково конфігуруватимемо пакет даних з потрібної інформації. Найголовніше – встановлення IP-адреси скомпроментованого кінцевого пристрою, як джерела походження пакету даних, а також встановлення IP-адреси маршрутизатора R1 у поле «neighbors». Інші поля заголовку підроблюються відповідно до перехоплених. Результати можна побачити на рисунках 3.19 та 3.20.

```
>>> a = sniff(filter="proto ospf", count=1)
>>> a.show()
0000 Ether / 10.10.1.1 > 224.0.0.5 ospf / Raw
>>> a.summary()
Ether / 10.10.1.1 > 224.0.0.5 ospf / Raw
>>> sendp(Ether()/IP(dst="1.2.3.4",ttl=(1,4)), iface="ens3")
....
Sent 4 packets.
>>> packet = Ether(src='00:06:28:b9:85:31',dst='01:00:5e:00:00:05')
>>> packet.show()
###[ Ethernet ]###
  dst= 01:00:5e:00:00:05
  src= 00:06:28:b9:85:31
  type= 0x9000

>>> packet = packet/Dot1Q(vlan=33)
>>> packet = packet/IP(src='10.10.1.2',dst='224.0.0.5')
>>> packet= packet/OSPF_Hdr(src=host)
Traceback (most recent call last):
  File "<console>", line 1, in <module>
NameError: name 'OSPF_Hdr' is not defined
>>> load_contrib('ospf')
>>> packet= packet/OSPF_Hdr(src=host)
Traceback (most recent call last):
  File "<console>", line 1, in <module>
NameError: name 'host' is not defined
>>> packet= packet/OSPF_Hdr(src='10.10.1.2')
>>> sendp(packet,iface='ens3')
.
Sent 1 packets.
```

Рисунок 3.19 – Створення та надсилання сфальсифікованого пакету OSPF-Hello

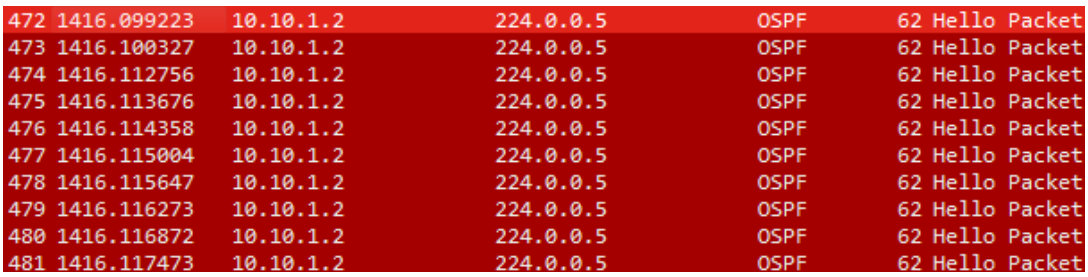

Рисунок 3.20 – Надіслані сфальсифіковані пакети OSPF-Hello

Пакети були надіслані до шлюзу. Для перевірки успішності реалізації вразливості необхідно переглянути налаштування відносин сусідства на маршрутизаторі R1. Із результатами можна ознайомитись на рисунку 3.21.

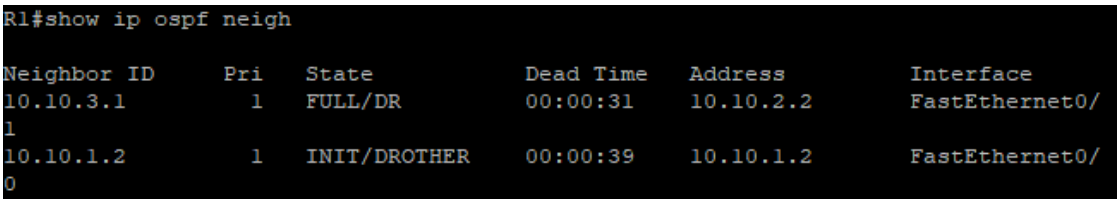

Рисунок 3.21 – Встановлення відносин сусідства з скомпроментованим кінцевим пристроєм

Потрібного результату було досягнуто і вразливість була проексплуатована, проте повних відносин сусідства встановлено не було по причині необхідності автоматизації процесу обміну DBD-повідомлень між жертвою та зловмисником, на що потребується великий обсяг обчислювальних ресурсів і лише високовартісне програмне забезпечення.

Тим не менш, виконані дії доводять фактичне існування вразливості. У випадку встановлення повних відносин сусідства із зловмисником, отримується повний доступ до системи маршрутизації, проте навіть велика кількість таких скомпрометованих пакетів, надісланих у шлюз, може спричинити атаку типу DoS, яка також може стояти на меті у зловмисника.

3.8 Класифікація кібератак за рівнем їх впливу на автономну систему маршрутизації

Класифікації кібератак дають систематичні та ґрунтовні знання про вразливості тієї чи іншої технології, програмного або апаратного забезпечення.

У випадку кібератак на протокол динамічної маршрутизації OSPF, класифікація кібератак за певними критеріями може допомогти навіть спеціалістам для швидшого реагування на інцидент, що трапився, адже перегляд офіційної технічної документації протоколу займає дуже багато часу, який можна витратити більш ефективніше.

Класифікувати будь-які атаки можна за багатьма критеріями [46]:

- тип;
- метод;
- мета;
- вразливості, що експлуатуються;
- сценарій;
- застосування;

Проте, кібератак на протокол OSPF не дуже багато, а більшість з них використовують більше недосконалості структури протоколу, а не його вразливості. Недоліки програмного забезпечення пристроїв маршрутизації також можливі, однак, у постійно змінному ландшафті кібербезпеки вони швидко усуваються оновленнями. Саме тому немає майже ніякого сенсу аналізувати старі документації пов’язані з вразливостями програмного забезпечення маршрутизаторів при роботі з протоколом, адже нові їх версії, зазвичай, все виправляють.

Говорячи про класифікації кібератак на мережеві протоколи, вважаю, що найдоцільніше їх розробляти та впроваджувати, аналізуючи прямий вплив на уражені вузли. Зрозуміло, що атака типу DDoS на автономну систему тягне за собою не тільки збої в процесі маршрутизації, а й незадоволення

користувачів або репутаційні збитки. Однак, розробляти класифікацію атак на будь-який мережевий протокол за рівнем незадоволеності користувачів не має ніякого сенсу, адже вона не дасть спеціалістам потрібного розуміння та знань.

Ґрунтуючись на всіх вищезазначених фактах, було прийняте рішення про розробку класифікації кібератак за рівнем їх впливу на автономну систему, адже вона допоможе систематизувати та упорядкувати знання про експлуатації вразливостей, а також зрозуміти які компоненти автономної системи та системи маршрутизації страждають найбільше від тієї чи іншої атаки.

Таблиця 3.1

Класифікація кібератак на протокол динамічної маршрутизації OSPF за рівнем їх впливу на автономну систему

Вплив	Disguised LSA	Adjacency Spoofing	Single Path Injection	Seq++	MaxSeq	LSA flood	MD5-attack
Постійність	+	+	+	+	+	+	-
Задіяння «fight-back»	-	+	-	+	+	+	-
Кількість потрібних пакетів	2	3	1	≥1	≥1	≥1	1
Отримання доступу до маршрутизатора-жерти	+	+	+	-	-	-	-
Зараження всієї автономної системи	+	+	+	-	-	-	-
Перевантаження системи	-	+	-	+	+	-	-

Як можна побачити, таблиця має перелік атак, який може доповнюватись, та відповідний вплив на автономну систему, де використовується досліджуваний протокол маршрутизації. Згідно з даною класифікацією, атака типу Adjacency Spoofing має найбільший вплив, але й

вона найскладніша в реалізації. Атаки, що пов’язані з експлуатацією вразливості порядкового номеру пакету даних найлегші в здійсненні, проте і вплив від них відповідний.

Слід також зазначити, що комбінування кібератак також можливе. Тобто навмисне заповнення системи маршрутизації в поєднанні з спробою встановлення неправдивих відносин сусідства також може бути здійснене зловмисником. На відміну від прямої атаки, така комбінація може лишитись непоміченою адміністраторами мережі, через високу активність фіктивних пакетів в системі маршрутизації [47].

Висновки до розділу 3

Третій розділ роботи містить аналіз вразливостей технічної конструкції протоколу; виокремлення потенційно вразливих вузлів протоколу OSPF; практичну експлуатацію вразливостей на емульованій автономній системі; класифікацію атак на протокол OSPF за рівнем їх впливу на автономну систему.

За результатом аналізу, найбільш вразливим вузлом протоколу є оголошення про стан каналу, адже вони містять максимальну кількість інформації про систему маршрутизації. Фальсифікація такого оголошення може бути як повна, так і часткова. Відповідно, рівень впливу на автоматизовану систему також буде різний.

Варто зазначити, що значна кількість задокументованих вразливостей у відкритій базі даних вразливостей є застарілими і стосувались певних версій програмного забезпечення маршрутизаторів. Версія програмного забезпечення використана в процесі дослідження майже не має таких проблем. Проте, від атак, що використовують вразливості конструкції протоколу наявна конфігурація не захищає.

Окремо було проекслуатовано вразливість криптографічної автентифікації повідомлень з використанням алгоритму шифрування MD5 на

емульованій автономній системі. Виконання правильного порядку дій дало змогу отримати і розшифрувати ключ, який був попередньо налаштований у системі маршрутизації.

Розробка класифікації кібератак за рівнем їх впливу на автономну систему відбувалась шляхом виокремлення вузлів автономної системи, які страждали у процесі кібератаки того чи іншого типу.

Відповідно до даної класифікації, найбільш руйнівною є атака Adjacency Spoofing, яка дає доступ зловмиснику до маршрутизатора-жертви, а також заражає усю автономну систему.

Атаки з фальсифікацією порядкових номерів оголошень мають найменший вплив на автономну систему, адже єдині можливі наслідки від них це перевантаження системи маршрутизації. Однак, у випадку правильних налаштувань механізмів роботи протоколу, навіть такого впливу досягти буде дуже складно. Тому атаки такого типу більш прийнятні у вигляді допоміжних атак, що передуватимуть більш комплексній.

Наслідки від атаки на криптографічну автентифікацію повідомлень досить контраверсійні. По факту, як результат від атаки, зловмисник отримує розшифрований ключ, що використовується для автентифікації повідомлень. Окремий ключ не дає доступу ні до маршрутизатора, ні до всієї системи. Проте, у випадку продовження зловмисником атак на нібито захищену автономну систему він зможе зробити будь-які дії, адже криптографічна автентифікація більше не стане йому на заваді.

Останній розділ містить практичну побудову удосконаленої конфігурації OSPF-маршрутизатора шляхом налаштування обраних механізмів безпеки, а також оцінку ефективності новоствореної конфігурації.

РОЗДІЛ 4
РОЗРОБКА МЕТОДУ ПРОТИДІЇ КІБЕРАТАКАМ ТА ОЦІНКА ЙОГО ЕФЕКТИВНОСТІ

Під час аналізу джерел інформації було знайдено безліч вразливих вузлів, як в офіційній технічній специфікації протоколу, так і в різних версіях програмного забезпечення пристроїв маршрутизації. Розробка та впровадження методів протидії на різні кібератаки створить унікальну конфігурацію пристрою маршрутизації з захистом від різних типів атак. Більше того, з постійним розвитком інформаційних технологій, така конфігурація може захистити як пристрій, так і всю автономну систему від нових типів атак, або вже існуючих, але зі зміненим методом експлуатації вразливості.

Задля кращого розуміння структури побудови захисту для маршрутизаторів було проведено експлуатацію декількох вразливостей у підрозділі 3.6, що додало більшого розуміння практичної роботи протоколу.

Реалізація практичної складової роботи відбуватиметься в мережевому емуляторі GNS3, який дає можливість побудувати автономні системи довільної топології з використанням пристроїв різних видів. Для зведення потрібної автономної системи були завантажені образи віртуальних пристроїв маршрутизації від компанії Cisco зі встановленим програмним забезпеченням Cisco vIOS 15.6.2 [48]. Причиною вибору цього програмного забезпечення є вже виправлені недоліки попередніх версій програмного забезпечення, а також наявність більшого вибору алгоритмів шифрування при налаштуванні криптографічної автентифікації повідомлень.

Емульована система включає до свого складу 3 вищеописані маршрутизатори, 3 комутатори та 3 персональні комп’ютери. Більш детально ознайомитись з топологією мережі можна ознайомитись на рисунку 4.1.

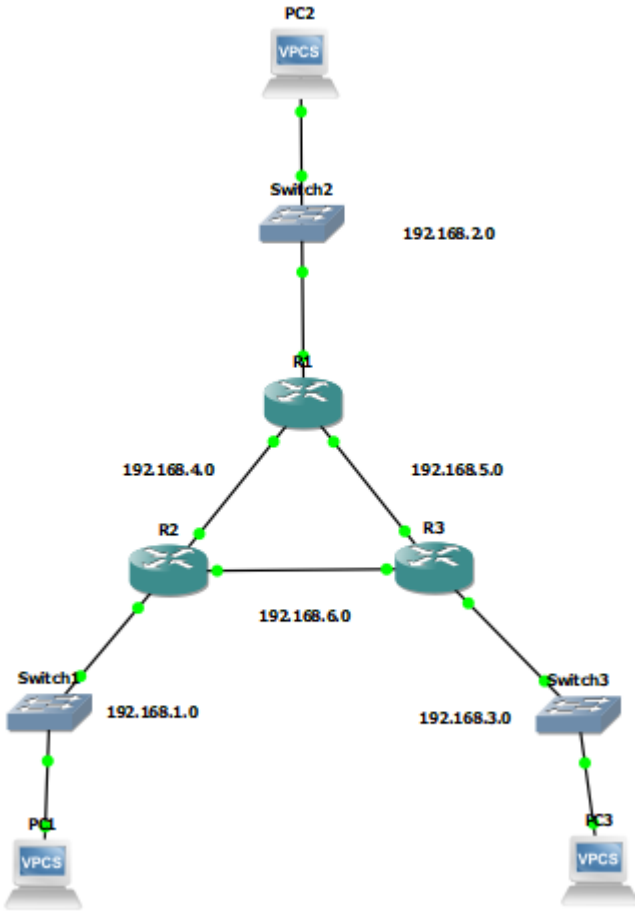

Рисунок 4.1 – Топологія емульованої системи

Попередньо було проведено налаштування протоколу динамічної маршрутизації OSPF. Конфігурація маршрутизаторів R1, R2 та R3 наразі виглядає чином:

```
Router#sh ip ospf data

            OSPF Router with ID (192.168.5.1) (Process ID 1)

                Router Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Link count
192.168.5.1     192.168.5.1     199         0x80000005 0x009174 3
192.168.6.1     192.168.6.1     208         0x80000005 0x005AA9 3
192.168.6.2     192.168.6.2     198         0x80000005 0x00E418 3

                Net Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum
192.168.4.2     192.168.5.1     413         0x80000001 0x00C6B4
192.168.5.1     192.168.5.1     199         0x80000001 0x00D3A6
192.168.6.1     192.168.6.1     208         0x80000001 0x00CAAC
```

Рисунок 4.2 – Конфігурація OSPF-маршрутизатора R1

```
Router#sh ip ospf data

            OSPF Router with ID (192.168.6.1) (Process ID 1)

                Router Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Link count
192.168.5.1     192.168.5.1     226         0x80000005 0x009174 3
192.168.6.1     192.168.6.1     232         0x80000005 0x005AA9 3
192.168.6.2     192.168.6.2     224         0x80000005 0x00E418 3

                Net Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum
192.168.4.2     192.168.5.1     439         0x80000001 0x00C6B4
192.168.5.1     192.168.5.1     226         0x80000001 0x00D3A6
192.168.6.1     192.168.6.1     232         0x80000001 0x00CAAC
```

Рисунок 4.3 – Конфігурація OSPF-маршрутизатора R2

```
Router#sh ip ospf data

            OSPF Router with ID (192.168.6.2) (Process ID 1)

                Router Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Link count
192.168.5.1     192.168.5.1     294         0x80000005 0x009174 3
192.168.6.1     192.168.6.1     302         0x80000005 0x005AA9 3
192.168.6.2     192.168.6.2     291         0x80000005 0x00E418 3

                Net Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum
192.168.4.2     192.168.5.1     508         0x80000001 0x00C6B4
192.168.5.1     192.168.5.1     294         0x80000001 0x00D3A6
192.168.6.1     192.168.6.1     302         0x80000001 0x00CAAC
```

Рисунок 4.4 – Конфігурація OSPF-маршрутизатора R3

Відповідно до рисунків 4.2, 4.3, 4.4, базова система маршрутизації в автономній системі налаштована, а пристрої встановили відносини сусідства між собою. Для перевірки вищезазначеного факту було проведено успішну спробу запиту на досяжність з персонального комп’ютеру PC1 до інших кінцевих пристроїв автономної системи. Ознайомитись з результатами можна на рисунку 4.5.

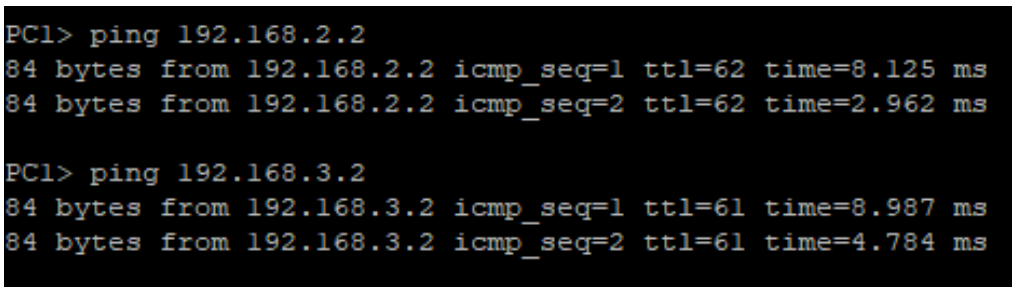

Рисунок 4.5 – Запит на досяжність з PC1

Як можна помітити, для емуляції була побудована топологія невеликого розміру по причині того, що протокол погано реагує на збільшення пристроїв. При роботі з даною автономною системою, швидкодія системи маршрутизації буде максимальна.

4.1 Розробка безпечної конфігурації OSPF-маршрутизатора

Як вже було описано, в автономній системі використовуються маршрутизатори під керуванням програмного забезпечення Cisco vIOS 15.6.2 [48]. Однак базового захисту від описаних атак та вразливостей воно не має, лише виправлені недоліки попередніх версій. Тому конфігурація даного маршрутизатора буде сукупністю можливих методів протидії атакам на протокол OSPF. Вона може бути використана, як допоміжний засіб при налаштуванні як особистих, так і корпоративних систем маршрутизації, а за певних обставин може бути просто імпортована.

4.1.1 Інсталювання автентифікації пакетів даних з використанням криптографічних алгоритмів шифрування

У підрозділі 3.6 було проведено атаку, що підриває довіру до алгоритму шифрування MD5. Пароль, що був встановлений в системі маршрутизації, був скомпрометований.

Відповідно до вищезазначеного факту, було прийняте рішення налаштовувати автентифікацію повідомлень з використанням алгоритму шифрування HMAC-SHA зі створенням окремих ланцюжків ключів, що містять рядки з відповідними до них паролями.

RFC 5709 описує використання алгоритму HMAC-SHA з різною довжиною ключа: від 1 до 512 біт. Більша довжина ключа збільшує стійкість алгоритму шифрування, тобто збільшує кількість та складність математичних операцій, що використовуються [49]. Це в свою чергу впливає на час, який витратять зловмисники на розшифрування перехоплених даних.

Для налаштування був обраний ключ з довжиною 512 біт, адже практичних методів для його ефективного обчислення не існує. Спроба розшифрування з використанням методу «brute-force» неможлива [50].

Кожен ключ має певний термін життя. По завершенні цього часу, створюється новий ключ, що розсилається по маршрутизаторам всієї автономної системи. Після імплементації нового ключа, старі – видаляються.

Технічні специфікації криптографічних алгоритмів MD5 та HMAC-SHA-512 мають певні відмінності з якими можна ознайомитись у таблиці 4.1 [50].

Таблиця 4.1

Порівняння алгоритмів MD5 та HMAC-SHA-512

Особливість	MD5	HMAC-SHA-512
Безпечність використання	Вразливий до атак	Безпечний
Автентифікація на основі ключа	Нативно не підтримується	Підтримується
Довжина ключа	16 байт (128 біт)	64 байт (512 біт)
Налаштування	Базова конфігурація зі спільним секретним ключем	Конфігурація з використанням одного або кількох ланцюжків ключів
Використання	Використовувався в більш старих версіях протоколу OSPF	Рекомендований для використання в усіх причетних автономних системах

продовження табл 4.1

Цілісність повідомлення	Забезпечується базова цілісність повідомлень	Забезпечується більша цілісність за рахунок використання криптографічного шифрування
Швидкодія	Швидкий	Повільний, за рахунок використання складніших шифрування
Сумісність	Сумісний з великою кількістю мережевого обладнання	Сумісний з певними версіями програмного забезпечення пристроїв маршрутизації

Інсталювання криптографічної автентифікації повідомлень з використанням алгоритму HMAC-SHA-512 починається з створення окремого ланцюга ключів та встановлення паролю. Наступним кроком, на потрібному інтерфейсі встановлюється автентифікація з використанням відповідного ланцюга. Більш детально з процесом налаштування на маршрутизаторі R1 можна ознайомитись на рисунку 4.6.

```
Router(config)#key chain R1_R2
Router(config-keychain)#key 1
Router(config-keychain-key)#cry
Router(config-keychain-key)#cryptographic-algorithm hmac-sha-512
Router(config-keychain-key)#key-string R1_R2_SeCrEt
Router(config-keychain-key)#exit
Router(config-keychain)#exit
Router(config)#key chain R1_R3
Router(config-keychain)#key 1
Router(config-keychain-key)#cry
Router(config-keychain-key)#cryptographic-algorithm hmac-sha-512
Router(config-keychain-key)#key-string R1_R3_SeCrEt
Router(config-keychain-key)#exit
Router(config-keychain)#exit
Router(config)#int gi0/1
Router(config-if)#ip ospf authentication key-chain R1_R2
Router(config-if)#exit
Router(config)#int gi0/2
Router(config-if)#ip ospf authentication key-chain R1_R3
Router(config-if)#exit
Router(config)#
*May  1 09:27:00.585: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.6.1 on GigabitEther
net0/1 from FULL to DOWN, Neighbor Down: Dead timer expired
Router(config)#
*May  1 09:27:20.460: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.6.2 on GigabitEther
net0/2 from FULL to DOWN, Neighbor Down: Dead timer expired
```

Рисунок 4.6 – Інсталювання криптографічної автентифікації на маршрутизаторі R1

Були створені два ланцюжки ключів R1_R2 та R1_R3 для відповідних підключень по інтерфейсах. Після увімкнення криптографічної автентифікації на одному пристрої маршрутизації, його відносини сусідства розірвались. Причиною є те, що він вже очікує зашифровані пакети даних, а інші маршрутизатори надсилають невідповідні пакети.

Відповідні пари ланцюжків були створені і на інших пристроях. Після налаштування криптографічної автентифікації на усіх відповідних інтерфейсах маршрутизаторів, відносини сусідства між ними відновились, а в специфікаціях інтерфейсу з’явилось відповідне оголошення. Переконатись в цьому можна на рисунках 4.7 та 4.8.

```
*May  1 09:34:14.071: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.5.1 on GigabitEther
net0/2 from LOADING to FULL, Loading Done
Router(config)#
*May  1 09:35:02.055: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.6.1 on GigabitEther
net0/1 from LOADING to FULL, Loading Done
```

Рисунок 4.7 – Поновлення відносин сусідства між маршрутизаторами

```
Router#sh ip ospf int gi0/1 | begin auth
  Cryptographic authentication enabled
    Sending SA: Key 1, Algorithm HMAC-SHA-512 - key chain R1_R2
```

Рисунок 4.8 – Використаний алгоритм шифрування для автентифікації

Для повної впевненості було проведено додаткову спробу запиту на досяжність з PC1 до інших пристроїв автономної системи. Результати зображені на рисунку 4.9.

```
PC1> ping 192.168.2.2
192.168.2.2 icmp_seq=1 timeout
192.168.2.2 icmp_seq=2 timeout
84 bytes from 192.168.2.2 icmp_seq=3 ttl=62 time=2.523 ms
84 bytes from 192.168.2.2 icmp_seq=4 ttl=62 time=3.153 ms

PC1> ping 192.168.3.2
192.168.3.2 icmp_seq=1 timeout
192.168.3.2 icmp_seq=2 timeout
84 bytes from 192.168.3.2 icmp_seq=3 ttl=62 time=2.592 ms
84 bytes from 192.168.3.2 icmp_seq=4 ttl=62 time=2.987 ms
84 bytes from 192.168.3.2 icmp_seq=5 ttl=62 time=3.372 ms
```

Рисунок 4.9 – Запит на досяжність з PC1

4.1.2 Інсталювання захисного механізму від переповнення трафіку

За своєю конструкцією, протокол OSPF вимагає оновлення оголошення про стан каналу кожні 30 хвилин. По закінченню строку їх дії, тобто досягнення значення MaxAge, що встановлене у 3600 секунд, вони видаляються з баз даних про стан каналів. У певних реалізаціях, в автономних системах встановлювалось значення оновлень від 30 до 50 хвилин, але зміни у системі маршрутизації при цьому майже не спостерігались.

Водночас, існує механізм OSPF Flooding Reduction, що працює за рахунок зменшення непотрібних оновлень и розсилання вже відомої незмінної інформації. Це досягається шляхом встановлення додаткового біта DoNotAge до оголошення про стан каналу, що вимикає потребу в його оновленні у встановлений проміжок часу [51].

За рахунок використання даного механізму, система маршрутизації розвантажується та працює значно швидше.

Налаштування даного механізму не займає багато часу і здійснюється за рахунок виконання відповідної команди в налаштуваннях потрібного інтерфейсу маршрутизатора. Більш детально з результатами можна ознайомитись на рисунку 4.10.

```
Router(config)#int gi 0/1
Router(config-if)#ip ospf flood-reduction
Router(config-if)#
*Apr 30 21:49:19.497: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.6.1 on GigabitEther
net0/1 from FULL to DOWN, Neighbor Down: Interface down or detached
*Apr 30 21:49:19.515: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.6.1 on GigabitEther
net0/1 from LOADING to FULL, Loading Done
Router(config-if)#exit
Router(config)#int gi0/2
Router(config-if)#ip ospf flood-reduction
Router(config-if)#
*Apr 30 21:49:40.955: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.6.2 on GigabitEther
net0/2 from FULL to DOWN, Neighbor Down: Interface down or detached
*Apr 30 21:49:40.973: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.6.2 on GigabitEther
net0/2 from LOADING to FULL, Loading Done
Router(config-if)#exit
```

Рисунок 4.10 – Налаштування механізму Flood Reduction

Окремо, можна налаштувати фільтрацію бази даних для блокування масового розсилання оголошення про стан каналу за межі автономної системи.

Однак, у випадку використовуваної під час розробки топології, налаштування даного механізму не матиме ніякого практичного сенсу, адже підключень поза межі автономної системи не має жоден пристрій.

4.1.3 Інсталювання захисного механізму перевірки часу життя пакетів даних

Досліджуваний протокол має функцію перевірки часу життя пакетів даних. За умови увімкнення даного механізму, OSPF надсилатиме вихідні новостворені пакети зі значенням часу життя IP-заголовку рівним 255. За такої ситуації, вхідні пакети, що матимуть менший показник часу життя будуть відкинуті.

Механізм роботи даної функції є досить логічним та простим. Кожний пристрій, що пересилає IP-пакет зменшує час його життя на одиницю при проходженні одного вузла системи маршрутизації [52]. Тому, пакети даних, що були передані прямим підключенням матимуть значення 255, а ті що потребуватимуть пересилання через один проміжний пристрій маршрутизації матимуть 254.

Так як OSPF-маршрутизатори встановлюють відносини сусідства, пакети даних між ними передаватимуться лише за умови прямого підключення, тому робота даного механізму захищає від різного типу атак, навіть віддаленого типу.

Налаштування даного функціоналу можливе як для всіх, так і для окремих інтерфейсів маршрутизатора на яких налаштована робота протоколу OSPF. Результат налаштування можна побачити на рисунку 4.11.

```
Router(config-router)#ttl-security all-interfaces
Router(config-router)#
Router(config-router)#
Router(config-router)#
*May  1 09:34:56.513: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.6.2 on GigabitEther
net0/2 from FULL to DOWN, Neighbor Down: Dead timer expired
*May  1 09:34:56.913: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.6.1 on GigabitEther
net0/1 from FULL to DOWN, Neighbor Down: Dead timer expired
```

Рисунок 4.11 – Налаштування захисного механізму перевірки часу життя пакетів

Як можна побачити, після налаштування на одному з маршрутизаторів, відносини сусідства розірвались. Причиною є те, що маршрутизатор з налаштованим механізмом очікує пакети з встановленим значенням часу життя 255. Проте, інші пристрої надсилають їх з часом життя, встановленим за замовчуванням – 1. Але після увімкнення цієї функції на всіх маршрутизаторах автономної системи, відносини сусідства відновились.

Для перевірки, було перехоплено один пакет даних за допомогою утиліти Wireshark. Ознайомитись с з ним можна на рисунку 4.12.

```
▸ Frame 1: 230 bytes on wire (1840 bits), 230 bytes captured (1840 bits) on interface -, id 0
▸ Ethernet II, Src: 0c:f8:86:b6:00:01 (0c:f8:86:b6:00:01), Dst: IPv4mcast_05 (01:00:5e:00:00:05)
▾ Internet Protocol Version 4, Src: 192.168.6.1, Dst: 224.0.0.5
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
    Total Length: 216
    Identification: 0x178b (6027)
  ▸ 000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
  ▸ Time to Live: 255
    Protocol: OSPF IGP (89)
    Header Checksum: 0xfbd2 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.6.1
    Destination Address: 224.0.0.5
▾ Open Shortest Path First
  ▾ OSPF Header
      Version: 2
      Message Type: Hello Packet (1)
```

Рисунок 4.12 – Перехоплений пакет за допомогою Wireshark

4.1.4 Інсталювання системи логування змін відносин сусідства

Пристрої не обмінюються інформацією про маршрути, якщо вони втрачають відносини сусідства. Кожного разу, коли такий зв’язок зникає, відповідні маршрути видаляються з бази даних маршрутизатора. Кожен інший пристрій автономної системи, у такому випадку, повинен також оновити свої бази даних з урахуванням нової топології [53].

Запис таких подій в журнал може значно полегшити процес усунення несправностей, адже системний адміністратор матиме більше інформації про час виникнення проблеми та її суть.

Також, певним чином, ця функція додає захищеності автономній системі від атак типу Adjacency Spoofing або Remote False Adjacency, де зловмисник намагається встановити фальшиві відносини сусідства. Хоч, сам процес логування не захистить маршрутизатор, але він дасть інформацію спеціалісту про встановлення нових відносин сусідства. Людина матиме змогу відновити правильну роботу атакованого пристрою.

Налаштування логування виконується шляхом виконання відповідної команди в консолі маршрутизатора. Детальніше на рисунку 4.13.

```
Router(config-router)#log-adjacency-changes
```

Рисунок 4.13 – Налаштування логування змін відносин сусідства

4.1.5 Інсталювання ліміту кількості оголошень про стан каналу

Функція налаштування ліміту кількості оголошень про стан каналу визначає максимальну кількість таких повідомлень в базі даних маршрутизатора, послідовність його дій у випадку наближення їх кількості до граничної. У випадку перевантаження бази даних можливі наступні варіанти розвитку подій [54]:

- маршрутизатор записує повідомлення типу MaxLSAWarning до журналу подій, у випадку співпадіння кількості наявних повідомлень з максимальною;
- якщо кількість оголошень в базі даних маршрутизатора перевищила встановлений ліміт, протокол маршрутизації відключається і не приймає нових оголошень про стан каналу. Після 5-хвилинного відключення маршрутизатор перезапускає роботу протоколу;
- маршрутизатор повністю вимикає протокол OSPF, у випадку п’яти повторюваних періодичних відключення, описаних вище. Перезавантаження відбувається мануально в консолі маршрутизатора;

Ліміт можливо зняти, шляхом встановлення його значення нульовим. Однак, це може призвести до перевантаження маршрутизатора, адже він виділятиме більше ресурсів на обробку нових та збереження існуючих оголошень про стан каналу.

Налаштування ліміту кількості оголошень про стан каналу в базі даних маршрутизатора відбувається шляхом виконання команди в консолі маршрутизатора. Більш детально можна ознайомитись на рисунку 4.14.

```
Router(config)#router ospf 1
Router(config-router)#max-lsa 1000 40 ignore-time 10 ignore-count 4 reset-time$
```

Рисунок 4.14 – Інсталювання ліміту максимальної кількості оголошень про стан каналу

Дана команда встановлює ліміт у 1000 оголошень, а також у випадку перевантажень провокує наступні дії маршрутизатора:

- маршрутизатор записує повідомлення MaxLSAWarning, якщо в базі даних повідомлень про стан каналу збережено 400 оголошень (40% від ліміту);

- маршрутизатор тимчасово вимикає протокол маршрутизації на 10 хвилин при досягненні встановленого ліміту;
- маршрутизатор повністю вимикає протокол маршрутизації після 4 тимчасових ітерацій;
- інкрементор тимчасових ітерацій скидається, у випадку наявності в базі даних меншої за ліміт кількості оголошень протягом 20 хвилин.

У залежності від автономної системи та топології ліміт може бути іншим.

4.1.6 Інсталювання списків контролю доступу

Списки контролю доступу для маршрутизаторів не настільки надійні як міжмережеві екрани з відстеженням стану, проте певний їх функціонал все ж переймають.

Такі списки зазвичай стосуються зовнішніх маршрутизаторів для фільтрації небажаного трафіку. Однак, навіть у випадку закритої автономної системи вони також застосовні. Наприклад, заборона доступу кінцевих пристроїв до головного серверу підвищує захищеність, адже у випадку компрометації одного з таких пристроїв швидкого доступу до сервера зловмисник не отримає [55].

Для досліджуваної автономної системи було прийняте рішення обмежити доступ пакетів ICMP з PC3 до PC1, адже зовнішніх маршрутів у автономній системі немає.

Для цього було створено новий розширений список контролю доступу та додано відповідне правило. Детальніше ознайомитись можна на рисунку 4.15.

```
Router(config)#access-list 100 deny icmp host 192.168.3.2 host 192.168.1.2
```

Рисунок 4.15 – Правило обмеження доступу пакетів ICMP між двома кінцевими пристроями

Наступним кроком список контролю доступу був введений в роботу та була проведена спроба запиту на досяжність з PC3 до PC1. Результати зображені на рисунку 4.16.

```
PC3> ping 192.168.1.2
*192.168.6.1 icmp_seq=1 ttl=254 time=3.610 ms (ICMP type:3, code:13, Communicati
on administratively prohibited)
*192.168.6.1 icmp_seq=2 ttl=254 time=3.236 ms (ICMP type:3, code:13, Communicati
on administratively prohibited)
*192.168.6.1 icmp_seq=3 ttl=254 time=3.050 ms (ICMP type:3, code:13, Communicati
on administratively prohibited)
*192.168.6.1 icmp_seq=4 ttl=254 time=1.955 ms (ICMP type:3, code:13, Communicati
on administratively prohibited)
*192.168.6.1 icmp_seq=5 ttl=254 time=2.442 ms (ICMP type:3, code:13, Communicati
on administratively prohibited)
```

Рисунок 4.16 – Спроба запиту на досяжність з PC3 до PC1

Як можна побачити, трафік був відкинутий та були повернені пакету третього типу ICMP з кодом 13, що означає адміністративну заборону в комунікації між двома пристроями.

Іншим, та більш правильним, вектором застосування розширених списків доступу в досліджуваній мережі є створення правил доступу для трафіку між маршрутизаторами.

У випадку такої топології першочергово було прийнято рішення про обмеження трафіку на маршрутизаторі R2, а саме його інтерфейсі Gi0/1. Так як даний інтерфейс є підключеним до пристрою R3. Для виконання поставленої задачі було написано правило доступу для пакетів даних лише з відповідної підмережі 192.168.6.0/24 на підключеному інтерфейсі. Із результатом можна ознайомитись на рисунку 4.17.

```
Router(config)#access-list 1 permit 192.168.6.0 0.0.0.255
Router(config)#exit
Router#
*May  1 19:51:45.060: %SYS-5-CONFIG_I: Configured from console by console
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int gi0/1
Router(config-if)#ip access-group 1 in
```

Рисунок 4.17 – Правило обмеження трафіку на інтерфейсі

Результатом даного правила в розширеному списку доступу є заборона доступу будь-якого іншого трафіку до потрібного інтерфейсу, окрім того, що був вказаний. Таким чином інтерфейс маршрутизатора стає більш захищеним від атак з фальсифікацією відносин сусідства.

Відповідне правило було введене для всіх інтерфейсів маршрутизаторів автономної системи з активним протоколом OSPF.

4.1.7 Налаштування часових інтервалів

Одним із підходів до підвищення захищеності автономної системи є зміна налаштувань з базових, що встановлені за замовчуванням, на модифіковані. Таким чином, зловмисник, який захоче скомпрометувати систему муситиме витратити більше часу на дослідження нововстановлених параметрів.

Одним з таких параметрів, які точно варто змінити, у протоколі динамічної маршрутизації OSPF є часові інтервали. Використовується два основних типи часових інтервалів.

Hello-таймер контролює частоту надсилання повідомлень до своїх сусідів для ідентифікації себе як працюючого [56].

Dead-таймер визначає час, який вираховує пристрій маршрутизації, у випадку неотримання пакету типу OSPF-Hello від свого сусіда. Після закінчення визначеного часу, маршрутизатор видаляє сусідній пристрій з таблиці маршрутизації, так як вважає, що він більше не є доступним [56].

За замовчуванням ці часові інтервали рівні 10 та 40 секундам відповідно. Звичайним емпіричним правилом для OSPF є збереження різниці між двома інтервалами у чотири рази. Однак, технічна конструкція протоколу дозволяє змінити значення на довільні.

Було прийняте рішення змінити два часових інтервали на значення 15 та 60 секунд відповідно. У такому випадку, проміжок кардинально не збільшується. З процесом налаштування можна ознайомитись на рисунку 4.18.

```
Router(config)#int gi0/1
Router(config-if)#ip ospf hell
Router(config-if)#ip ospf hello-interval 15
Router(config-if)#ip ospf dead-interval 60
Router(config-if)#exit
Router(config)#int gi0/2
Router(config-if)#ip ospf
Router(config-if)#ip ospf he
Router(config-if)#ip ospf hello-interval 15
Router(config-if)#ip ospf dead-interval 60
```

Рисунок 4.18 – Налаштування інтервалів на інтерфейсах маршрутизатора

4.1.8 Налаштування привілеїв на маршрутизаторах

Маршрутизатори Cisco підтримують налаштування рівнів привілеїв для користувачів [57]. Встановлюються вони у проміжку від 0 до 15, де вищий рівень надає доступ до більшої кількості команд.

Даний механізм захищає систему маршрутизації від віддаленого захоплення пристрою маршрутизації зловмисниками, а також можливого несанкціонованого виконання команд людиною, яка не мала б отримати до них доступ.

Для конфігурації було створено два користувачі Admin та John з рівнями привілеїв 15 та 5 відповідно. Процес створення зображений на рисунку 4.19.

```
Router (config)#username admin privilege 15 secret Admin_DIPL
Router (config)#username john privilege 5 secret Johny_DIP
Router (config)#privelege exec level 5 show running-config
                ^
% Invalid input detected at '^' marker.

Router (config)#privilege exec level 5 show running-config
Router (config)#privilege exec level 5 ping
```

Рисунок 4.19 – Створення користувачів та налаштування їх привілеїв

Користувач Admin отримав максимальні привілеї, тобто матиме доступ до всього функціоналу маршрутизатора та його консолі.

Користувач John, у свою чергу, отримав рівень привілеїв 5 та доступ лише до вищезазначених команд.

Наступним кроком було проведено спробу перевірити працездатність встановлених привілеїв. Для цього було використане підключення до маршрутизатора через telnet. Варто зазначити, що підключення до консолі маршрутизатора можливе через будь-який кінцевий пристрій, однак для пришвидшення процесу перевірки було обрано створювати підключення з іншого пристрою маршрутизації автономної системи.

Першочергово було проведено спробу авторизації під користувачем Admin і перевірено рівень його привілеїв. Результати зображені на рисунку 4.20.

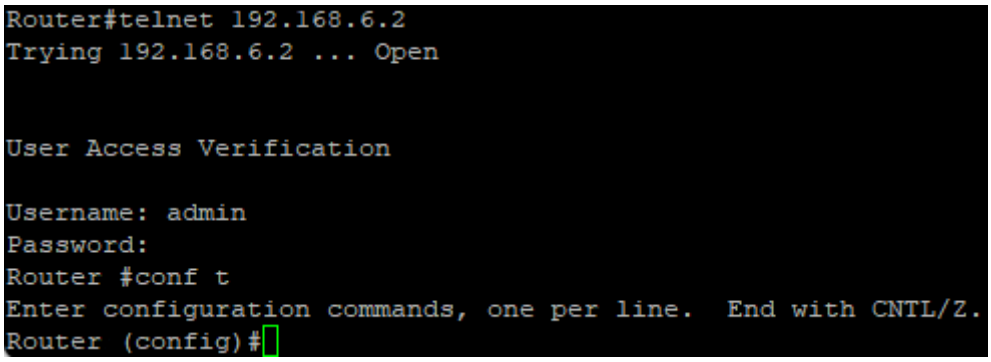

Рисунок 4.20 – Авторизація під користувачем Admin

За результатами перевірки, користувач має доступ до конфігураційного рядку маршрутизатора, тобто встановлений рівень привілеїв працює.

Наступним кроком стала авторизація і перевірка привілеїв під користувачем John. Детальніше з результатами можна ознайомитись на рисунку 4.21.

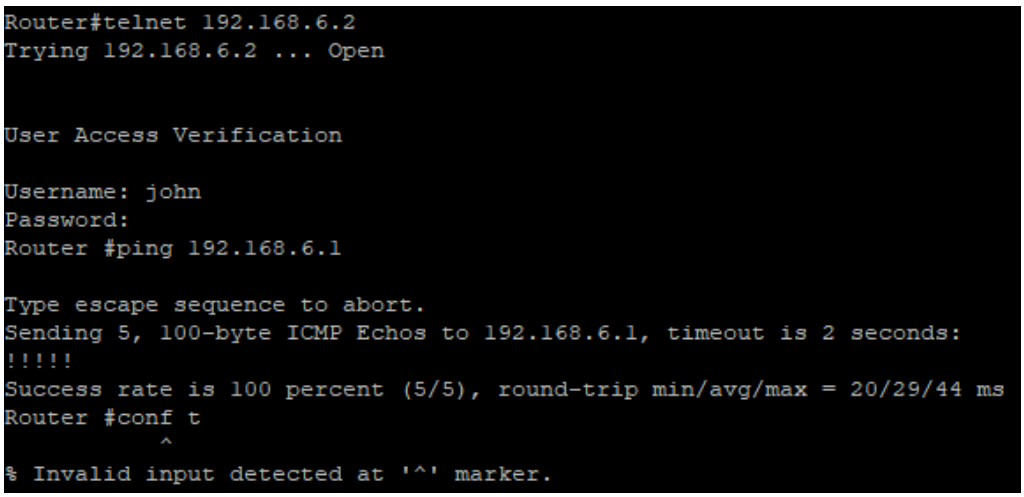

Рисунок 4.21 – Авторизація під користувачем John

Перевірка доступу до команди «ping» виявилась успішною, отже встановлені команди дозволені до виконання. Однак, доступу до конфігураційого рядку пристрою маршрутизації користувач отримати не зміг. Консоль повернула помилкове значення.

4.2 Оцінка ефективності удосконалено конфігурації

Як вже було обумовлено, базова конфігурація маршрутизатора Cisco під керуванням програмного забезпечення Cisco vIOS 15.6.2 має меншу кількість програмних недоліків у порівнянні з попередніми версіями.

Розроблена удосконалена конфігурація (Додаток Б) забезпечує захист від всіх згаданих в цій роботі атак та певної кількості вразливостей, а саме:

- Disguised LSA:
- Adjacency Spoofing;
- LSA-flood;
- Seq++/MaxSeq;
- MD5 Cryptographic Authentication Attack;
- Single Path Injection;
- MaxAge Attack;
- CVE-20**-****.

Найкращим показником для оцінки ефективності в даному випадку є продуктивність того чи іншого механізму в протидії певному типу атаки.

Відповідно до вищезазначеного факту, найбільшу продуктивність має криптографічна автентифікація повідомлень з використанням алгоритму шифрування HMAC-SHA-512. Цей механізм буквально захищає систему маршрутизації від усіх згаданих атак, а також великої низки задокументованих вразливостей програмного забезпечення, таких як CVE-2024-20313, CVE-2017-6770. Причиною є неможливість або навіть безперспективність зламу хешу. Як вже відомо, довжина ключа при використанні даного алгоритму шифрування – 512 біт, тобто для пошуку колізії знадобиться $O(2^{512})$ часу, адже жодних атак на SHA-2 досі не існує. Імовірність знаходження колізії після хешування певного набору символів буде рівна [58]:

$$P \approx \frac{1}{2} \times 2^{\frac{-n}{2}} \tag{4.1}$$

де P – шукана ймовірність, а n – кількість бітів в хеші.

Відповідно до даної формули ймовірність знаходження колізії для 512-бітового хешу рівна:

$$P \approx \frac{1}{2} \times 2^{\frac{-512}{2}} \approx \frac{1}{2} \times 2^{-256} \tag{4.2}$$

Отримана величина є неймовірно малою, що логічно, адже криптографічні хеш-функції розроблялись з урахуванням максимально низької ймовірності колізій для гарантування безпеки.

Розглядаючи інші механізми не можна не зазначити логування змін відносин сусідства маршрутизаторів. Дана технологія не надає прямого захисту від будь-яких типів атак, однак дає можливість швидше зреагувати на інцидент, що стався в автономній системі. Це не обов’язково має бути певна атака, адже механізми безпеки відповідають також і за захист доступності в

мережі Переповнення бази даних оголошень про стан каналу може спричинити розірвання відносин сусідства з маршрутизаторами, що спричинить порушення доступності в автономній системі. Проте, із налаштованим механізмом логування, адміністратор мережі помітить зміни набагато швидше і зреагує на нього потрібним чином.

Налаштування часу життя пакетів даних захищає автономну систему від атак типу LSA-flood, атак, пов'язаних з фальсифікацією порядкового номеру оголошення про стан каналу, а також атак з-поза меж автономної системи. Під час виконання атак такого виду, зловмисники можуть не звернути увагу на налаштування часу життя пакету. У такому випадку, вони можуть бути одразу відкинуті або просто не мати подальшого маршруту.

Була проведена спроба повторити атаку Adjacency Spoofing шляхом надсилання пакету OSPF-Hello, проте завдяки налаштованим механізмам захисту нічого не вийшло. Пакет даних був ідентифікований, як сформований зловмисно. Докази можна побачити на рисунку 4.22.

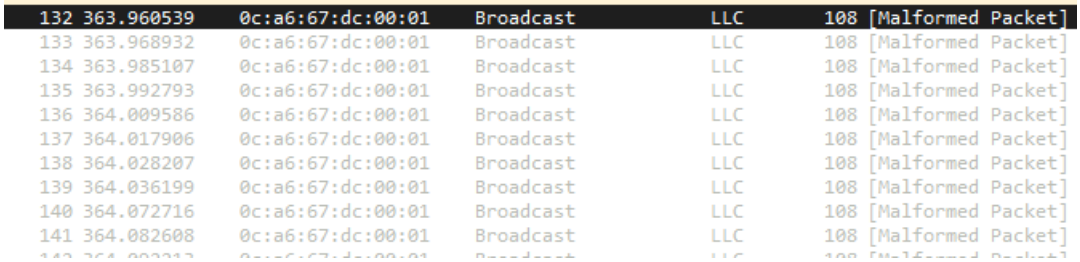

Рисунок 4.22 – Спроба реалізації атаки Adjacency Spoofing на удосконаленій версії конфігурації

Ця спроба візуалізує активну роботу криптографічної автентифікації повідомлень, перевірки часу життя пакетів даних та перевірки часових інтервалів.

Для оцінювання удосконаленої конфігурації було підсумовано загальну кількість вразливостей і атак, захист від яких було впроваджено. Також, окремо, було звісно підраховано загальну кількість взагалі існуючих незахищеностей. Варто зазначити, що у процесі калькуляції до уваги брались

лише вразливості, що стосуються недоліків програмного забезпечення Cisco, адже удосконалена конфігурація створена для маршрутизаторів Cisco.

У процесі також було стверджено, що базова конфігурація маршрутизатора з налаштованою маршрутизацією протоколом OSPF захисту від вразливостей та атак не має.

Загальна кількість CVE для протоколу динамічної маршрутизації OSPFv2 – 10. Загальна кількість задокументованих атак, що стосуються саме конструкції протоколу – 11.

Удосконалена конфігурація OSPF-маршрутизатора захищена від 6 вразливостей, що стосуються недоліків програмного забезпечення, а також від 9 атак, що використовують вразливості конструкції протоколу.

Оцінка ефективності рахувалась за наступною формулою:

$$E = \frac{d}{n} \times 100\% \quad (4.3)$$

де E – ефективність удосконаленої конфігурації;

d – кількість вразливостей і атак протоколу, захист від яких був налаштований;

n – загальна кількість вразливостей і атак на протокол OSPF.

Необхідні калькуляції для використання формули 4.3 вже виконані, отже можна вирахувати ефективність удосконаленої конфігурації:

$$E = \frac{15}{21} \times 100\% = 71\% \quad (4.4)$$

За результатами математичних обрахунків, було встановлено, що розроблена удосконалена версія OSPF-маршрутизатора на базі програмного забезпечення Cisco vIOS 15.6.2 на 71% ефективніша за базову.

Висновки до розділу 4

Останній розділ містить практичну побудову удосконаленої конфігурації OSPF-маршрутизатора шляхом інсталювання обраних ефективних механізмів захисту. Після завершення було проведено оцінку її ефективності.

Для удосконаленої конфігурації OSPF-маршрутизатора було налаштовано криптографічну автентифікацію повідомлень з використанням алгоритму шифрування HMAC-SHA-512, механізм OSPF Flood Reduction, механізм перевірки часу життя пакету даних, механізм логування змін відносин сусідства, механізм обмеження максимальної кількості оголошень про стан каналу в базі даних маршрутизатора, а також впроваджено розширені списки контролю доступу і привілейованих користувачів на маршрутизаторі.

Усі механізми, що були налаштовані та додані до конфігурації OSPF-маршрутизатора певним чином захищають автономну систему від зловмисників та їх атак. У процесі оцінки ефективності порівнювалась продуктивність того чи іншого механізму безпеки в протидії атакам різних типів.

Найбільш продуктивним механізмом виявилась криптографічна автентифікація з використанням ключа довжини 512 біт. Ймовірність розкриття такого ключа автентифікації дуже мала.

Інші методи захисту, хоч і менш продуктивні, проте в кооперації надають захист від атак з фальсифікацією порядкового номеру оголошення, а також певних атак з фальсифікацією оголошень про стан каналу.

За результатами математичних обрахунків, було встановлено, що розроблена удосконалена версія OSPF-маршрутизатора на базі програмного забезпечення Cisco vIOS 15.6.2 на 71% ефективніша за базову.

ВИСНОВКИ

Метод динамічної маршрутизації надає можливість автоматично змінювати маршрути для доставлення пакетів даних у випадку відмови одного з вузлів системи. Такий підхід до маршрутизації наразі використовується у великій кількості комп'ютерних мереж через його відмовостійкість, легкість у налаштуванні та здатності до масштабування.

Протокол Open Shortest Path First – це протокол динамічної маршрутизації, що відноситься до сімейства протоколів внутрішньої маршрутизації, та ґрунтується на алгоритмі відстеження стану каналу. Оскільки, він належить до категорії протоколів внутрішньої маршрутизації, область його дії, зазвичай, обмежується однією автономною системою.

Основною структурною одиницею протоколу є оголошення про стан каналу, які поділяються на 7 різних типів. Такі пакети даних розсилаються маршрутизаторами поміж пристроїв з якими налагоджені відносини сусідства.

Протокол має три версії, дві з яких активно використовуються зараз. Цей факт, на фоні зростаючої кількості кібератак, робить його потенційно можливою точкою для атаки зловмисника.

Базова конфігурація протоколу маршрутизації та вбудованих механізмів захисту, що описані в офіційних технічних документаціях, повною мірою не забезпечують ефективного відбиття атак.

У дипломній роботі розв'язано поставлене актуальне наукове завдання щодо розробки удосконаленої конфігурації OSPF-маршрутизатора і класифікації кібератак за рівнем їх впливу на автономну систему. У процесі вирішення поставлених на початку роботи завдань були одержані такі наукові та практичні результати:

1. Здійснено детальний аналіз офіційних технічних документацій з метою розкриття більшої кількості специфічних аспектів. Проведено

порівняння різних протоколів динамічної маршрутизації, а також різних версій протоколу OSPF.

2. Визначено найбільш вразливі до атак вузли протоколу. Проексплуатовано вразливість криптографічної автентифікації повідомлень. З огляду на це, було зроблено висновок про необхідність створення комплексного ефективного механізму для захисту від атак.
3. На основі аналізу та експлуатації кібератак різних видів розроблено їх класифікацію за рівнем впливу на автономну систему.
4. Запропоновано набір механізмів захисту, який мінімізує вплив атак на систему маршрутизації.
5. Розроблено удосконалену конфігурацію OSPF-маршрутизатора на базі програмного забезпечення Cisco vIOS 15.6.2, у якій реалізовано усі запропоновані механізми захисту.
6. Проведено теоретичну оцінку ефективності створеної конфігурації, базуючись на сукупній ефективності протидії атакам різних типів.

Тематика захисту систем маршрутизації, зокрема на базі протоколу динамічної маршрутизації OSPF не є повністю дослідженою, тож залишатиметься актуальною ще довгий проміжок часу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Скляр Б., Лаптєв О. Methods of counteracting attacks on the OSPF dynamic routing protocol. Проблеми кібербезпеки інформаційно-телекомунікаційних систем: зб. тез та доп. міжнар. науково-практ. конф., 26 квіт. 2024 р. Київ, 2024. С. 60-62.

2. What is routing? [Електронний ресурс] – Режим доступу до ресурсу: https://www.cloudflare.com/learning/network-layer/what-is-routing/.

3. Balchunas A. Static vs. Dynamic Routing [Електронний ресурс]. / Aaron Balchunas – Режим доступу до ресурсу: https://www.routeralley.com/guides/static_dynamic_routing.pdf.

4. Dynamic Routing Protocols [Електронний ресурс] – Режим доступу до ресурсу: https://www.linkedin.com/pulse/dynamic-routing-protocols-olabode-stephen-opeyemi-wixvf/.

5. Difference between Static and Dynamic Routing [Електронний ресурс] – Режим доступу до ресурсу: https://www.geeksforgeeks.org/difference-between-static-and-dynamic-routing/.

6. T.Moy J. OSPF: Anatomy of an Internet Routing Protocol [Електронний ресурс]. / John T.Moy. – 1998. – Режим доступу до ресурсу: https://bit.ly/4agO4RF.

7. OSPF RFC Documents [Електронний ресурс] – Режим доступу до ресурсу: https://www.rfc-editor.org/search/rfc_search_detail.php?title=ospf&pubstatus%5B%5D=Any&pub_date_type=any.

8. RFC 5340 [Електронний ресурс] – Режим доступу до ресурсу: https://www.rfc-editor.org/info/rfc5340.

9. OSPF CVE [Електронний ресурс] – Режим доступу до ресурсу: https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=OSPF.

10. Open Shortest Path First (OSPF) protocol [Електронний ресурс] – Режим доступу до ресурсу: https://www.geeksforgeeks.org/open-shortest-path-first-ospf-protocol-states/.

11. What is Dijkstra’s Algorithm? [Електронний ресурс] – Режим доступу до ресурсу: https://www.geeksforgeeks.org/introduction-to-dijkstras-shortest-path-algorithm/.

12. How OSPF protocol implements Dijkstra Algorithm [Електронний ресурс] – Режим доступу до ресурсу: https://medium.com/@kp-the-great/how-ospf-protocol-implements-dijkstra-algorithm-53c390199ee8.

13. OSPF Design [Електронний ресурс] – Режим доступу до ресурсу: https://community.cisco.com/t5/routing/ospf-design/td-p/2452999.

14. RFC 2328 [Електронний ресурс] – Режим доступу до ресурсу: https://www.rfc-editor.org/rfc/pdfrfc/rfc2328.txt.pdf.

15. OSPF Cost - Everything You Need to Know [Електронний ресурс] – Режим доступу до ресурсу: https://orhanergun.net/ospf-cost.

16. OSPF - Part 5: Analysis Of OSPF Link State Update (LSU) - Link State Advertisement (LSA) Packet Structure [Електронний ресурс] – Режим доступу до ресурсу: https://www.firewall.cx/networking/routing-protocols/ospf-lsu-lsa-packet-structure-lsa-types-overview.html.

17. OSPF LSA Types Explained [Електронний ресурс]. – Режим доступу до ресурсу: https://networklessons.com/ospf/ospf-lsa-types-explained.

18. LSA types [Електронний ресурс] – Режим доступу до ресурсу: https://techhub.hpe.com/eginfolib/networking/docs/switches/5710/5200-4992_l3-ip-rtng_cg/content/517702239.htm.

19. Link State ID [Електронний ресурс] – Режим доступу до ресурсу: https://www.freesoft.org/CIE/RFC/1583/63.htm.

20. Link-state advertisement [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Link-state_advertisement.

21. Understanding OSPF Authentication [Електронний ресурс] – Режим доступу до ресурсу:

https://support.huawei.com/enterprise/en/doc/EDOC1100290924/ab450cef/understanding-ospf-authentication.

22. The Security Analysis and Attacks Detection of OSPF Routing Protocol [Електронний ресурс] – Режим доступу до ресурсу: https://sci-hub.se/10.1109/ICICTA.2014.200.

23. OSPF Packet Types [Електронний ресурс] – Режим доступу до ресурсу: https://sites.google.com/site/amitsciscozone/ospf/ospf-packet-types.

24. OSPF MD5 Authentication [Електронний ресурс] – Режим доступу до ресурсу: https://itskillbuilding.com/networking/network/ospf/ospf-md5-authentication/.

25. OSPF HMAC-SHA Cryptographic Authentication [Електронний ресурс] / R. White, M. Barnes, V. Manral та ін. – Режим доступу до ресурсу: https://www.ietf.org/proceedings/70/IDs/draft-ietf-ospf-hmac-sha-00.txt.

26. Hierarchical Structure of OSPF [Електронний ресурс] – Режим доступу до ресурсу: https://stucknactive.com/2019/03/29/6-4-hierarchical-structure-of-ospf/.

27. Monitoring OSPF Routing Protocols [Електронний ресурс] – Режим доступу до ресурсу: https://logrhythm.com/blog/monitoring-ospf-routing-protocols/.

28. LSA & SPF Throttling [Електронний ресурс] – Режим доступу до ресурсу: http://www.bscottrandall.com/3.6.9.3.

29. Comparison between OSPFv2 vs OSPFv3 [Електронний ресурс] – Режим доступу до ресурсу: https://www.geeksforgeeks.org/comparison-between-ospfv2-vs-ospfv3/.

30. OSPF Security Vulnerabilities Analysis [Електронний ресурс] – Режим доступу до ресурсу: https://datatracker.ietf.org/doc/html/draft-ietf-rpsec-ospf-vuln-01#ref-3.

31. OSPF LSA Manipulation Vulnerability in Multiple Cisco Products [Електронний ресурс] – Режим доступу до ресурсу:

https://www.cisco.com/c/dam/en/us/support/docs/csa/cisco-sa-20130801-lsaospf.html.

32. CVE-2013-0149 [Електронний ресурс] – Режим доступу до ресурсу: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-0149.\

33. CVE-2024-20313 [Електронний ресурс] – Режим доступу до ресурсу: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2024-20313.

34. Interconnections: Bridges, Routers, Switches, and Internetworking [Електронний ресурс] – Режим доступу до ресурсу: https://dokumen.pub/interconnections-bridges-routers-switches-and-internetworking-protocols-2nbsped-0201634481-9780201634488.html.

35. Persistent OSPF Attacks [Електронний ресурс] – Режим доступу до ресурсу: https://theory.stanford.edu/~dabo/papers/ospf.pdf.

36. OSPF Vulnerability to Persistent Poisoning Attacks: A Systematic Analysis [Електронний ресурс] – Режим доступу до ресурсу: https://csaws.cs.technion.ac.il/~gnakibly/papers/ACSAC14.pdf.

37. Novel Attacks in OSPF Networks to Poison Routing Table [Електронний ресурс] – Режим доступу до ресурсу: https://www4.comp.polyu.edu.hk/~shanggao/publications/Novel_Attacks_in_OSPF_Networks_to_Poison_Routing_Table.pdf.

38. Remote false adjacency [Електронний ресурс] – Режим доступу до ресурсу: https://www.oreilly.com/library/view/advanced-infrastructure-penetration/9781788624480/abe071e1-a754-4115-bc54-72bc45071278.xhtml.

39. OSPF Security: Attacks and Defenses [Електронний ресурс] – Режим доступу до ресурсу: https://docplayer.net/61426194-Ospf-security-attacks-and-defenses.html.

40. Real-Time Protocol Analysis for Detecting Link-State Routing Protocol Attacks [Електронний ресурс] – Режим доступу до ресурсу: https://www.researchgate.net/publication/220593668_Real-Time_Protocol_Analysis_for_Detecting_Link-State_Routing_Protocol_Attacks.

41. Wireshark [Електронний ресурс] – Режим доступу до ресурсу: https://www.wireshark.org/.

42. Ettercap [Електронний ресурс] – Режим доступу до ресурсу: https://www.ettercap-project.org/.

43. John the Ripper password cracker [Електронний ресурс] – Режим доступу до ресурсу: https://www.openwall.com/john/.

44. Scapy [Електронний ресурс] – Режим доступу до ресурсу: https://scapy.net/.

45. Create Packets from Scratch with Scapy [Електронний ресурс] – Режим доступу до ресурсу: https://null-byte.wonderhowto.com/how-to/create-packets-from-scratch-with-scapy-for-scanning-dosing-0159231/.

46. Cyber-security for Mobile Service Robots [Електронний ресурс] – Режим доступу до ресурсу: https://www.researchgate.net/publication/334097376_Cyber-security_for_Mobile_Service_Robots_-_Challenges_for_Cyber-physical_System_Safety.

47. 10 MOST COMMON TYPES OF CYBER ATTACKS [Електронний ресурс] – Режим доступу до ресурсу: https://www.crowdstrike.com/cybersecurity-101/cyberattacks/most-common-types-of-cyberattacks/.

48. 1120 Connected Grid Router [Електронний ресурс] – Режим доступу до ресурсу: https://software.cisco.com/download/home/284174271/type/280805680/release/15.6.2T.

49. SHA-512 Algorithm [Електронний ресурс] – Режим доступу до ресурсу: https://bit.ly/4bwJShF.

50. OSPFv2 HMAC-SHA Cryptographic Authentication [Електронний ресурс] – Режим доступу до ресурсу: https://datatracker.ietf.org/doc/html/rfc5709.

51. OSPF Flooding Reduction [Електронний ресурс] – Режим доступу до ресурсу: https://www.cisco.com/en/US/docs/ios/12_1t/12_1t2/feature/guide/dt_ospff.html.

52. Configuring OSPF TTL Security Check and OSPF Graceful Shutdown [Електронний ресурс] – Режим доступу до ресурсу: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_ospf/configuration/xe-3e/iro-xe-3e-book/iro-ttl.pdf.

53. Logging OSPF Adjacency Changes [Електронний ресурс] – Режим доступу до ресурсу: https://www.oreilly.com/library/view/cisco-ios-cookbook/0596527225/ch08s17.html.

54. Open Shortest Path First – Version 2 [Електронний ресурс] – Режим доступу до ресурсу: https://www.arista.com/en/um-eos/eos-open-shortest-path-first-version-2.

55. Configure and Filter IP Access Lists [Електронний ресурс] – Режим доступу до ресурсу: https://www.cisco.com/c/en/us/support/docs/security/ios-firewall/23602-confaccesslists.html.

56. OSPF Hello and Dead Interval [Електронний ресурс] – Режим доступу до ресурсу: https://networklessons.com/ospf/ospf-hello-and-dead-interval.

57. Cisco IOS - Privilege Levels [Електронний ресурс] – Режим доступу до ресурсу: https://learningnetwork.cisco.com/s/blogs/a0D3i000002eeWTEAY/cisco-ios-privilege-levels.

58. Hash Collision [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Hash_collision.

ДОДАТОК А

Алгоритм Дейкстри

Алгоритм Дейкстри – алгоритм пошуку найкоротшого шляху у від однієї вершини граф до всіх інших, порівнюючи вагу кожного ребра. Він був винайдений у 1959 нідерландським вченим Е.Дейкстрой.

Даний алгоритм вважається найпройстішим для вирішення проблеми найкоротшого шляху.

Для прикладу розглянемо наступний граф зображений на рисунку А.1 для пошуку найкоротшого шляху з вершини 0 до всіх інших вершин.

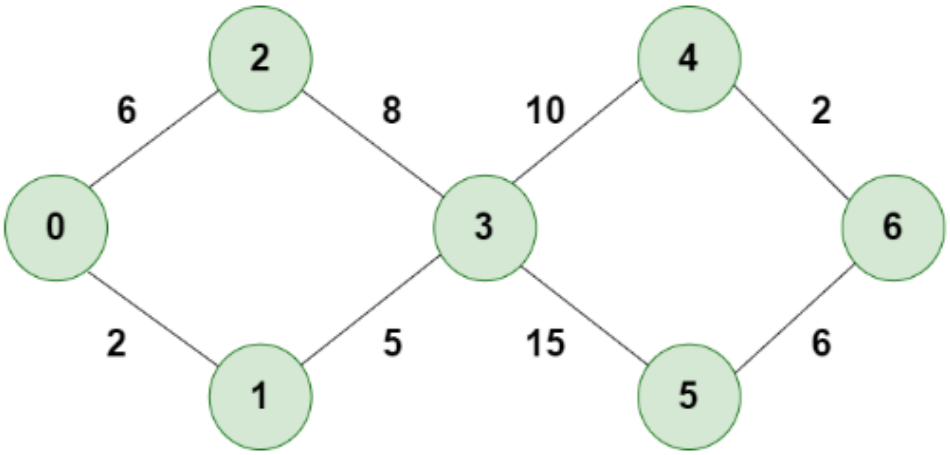

Рисунок А.1 – Граф для прикладу роботи алгоритму Дейкстри

У випадку даного графу, вага кожного ребра буде рівна відстанню між двома вершинами. Тобто, потрібно знайти найкоротшу відстань від:

- від вузла 0 до вузла 1;
- від вузла 0 до вузла 2;
- від вузла 0 до вузла 3;
- від вузла 0 до вузла 4;
- від вузла 0 до вузла 6.

Початковий набір ресурсів виглядає наступним чином:

- відстань з вихідного вузла до самого себе рівна 0;
- відстань до всіх інших вузлів поки невідома, тому поки рахується як нескінченність.

Крок 1

Починається робота алгоритму з вершини 0 та саме вона позначається як відвідана. Відвідані вершини відмічаємо червоним кольором.

Вершина 0 → Вершина 0 = 0+0 = 0.

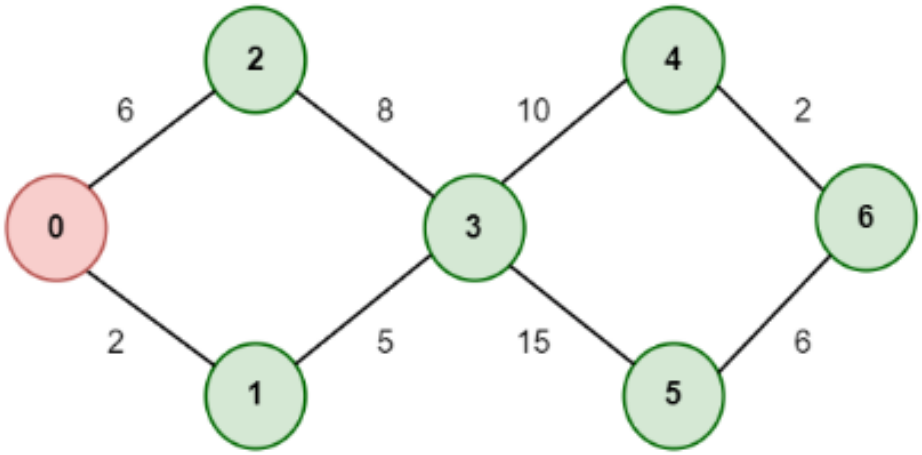

Рисунок А.2 – Перший крок алгоритму Дейкстри

Крок 2

Наступним кроком перевіряється наявність суміжних вершин. Треба обрати вершину з найменшою вагою. На цьому кроці відстань до вершини 1 є мінімальною, тож позначимо її як відвідану та занотуємо шлях до цієї вершини, як 2.

Вершина 0 → Вершина 1 = 0 + 2 = 2.

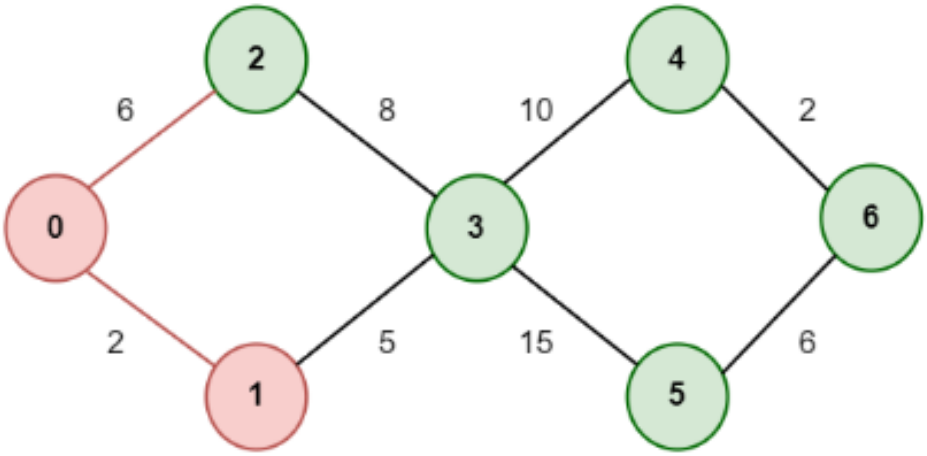

Рисунок А.3 – Другий крок алгоритму Дейкстри

Крок 3

Переміщаємось далі і відвідуємо вершину 2 починаючи знову з вершини 0, адже шлях до неї буде коротшим ніж інші. Позначимо її як відвідану та занотуємо шлях до цієї вершини, як 6.

Вершина 0 → Вершина 2 = 0 + 6 = 6.

Для пришвидшення, у цьому ж кроці знайдемо найкоротший шлях до вершини 3 через вершину 1. Позначимо її як відвідану та занотуємо шлях до цієї вершини, як 7.

Вершина 0 → Вершина 1 → Вершина 3 = 0 + 2 + 5 = 7.

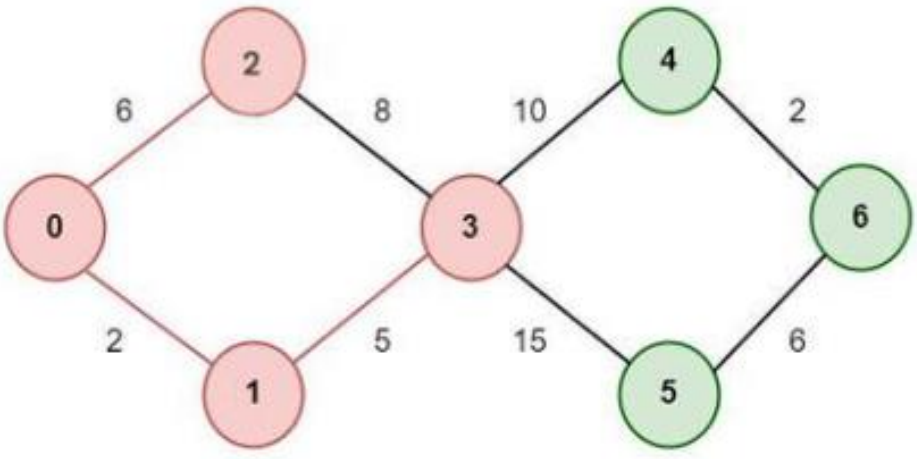

Рисунок А.4 – Третій крок алгоритму Дейкстри

Крок 4

Знову існує два варіанти для суміжних вершин. На цьому кроці вершина 4 має мінімальну відстань до сусідньої вершини 3. Позначимо її як відвідану та занотуємо шлях до цієї вершини, як 17.

Вершина 0 → Вершина 1 → Вершина 3 → Вершина 4 = 0 + 2 + 5 + 10 = 17.

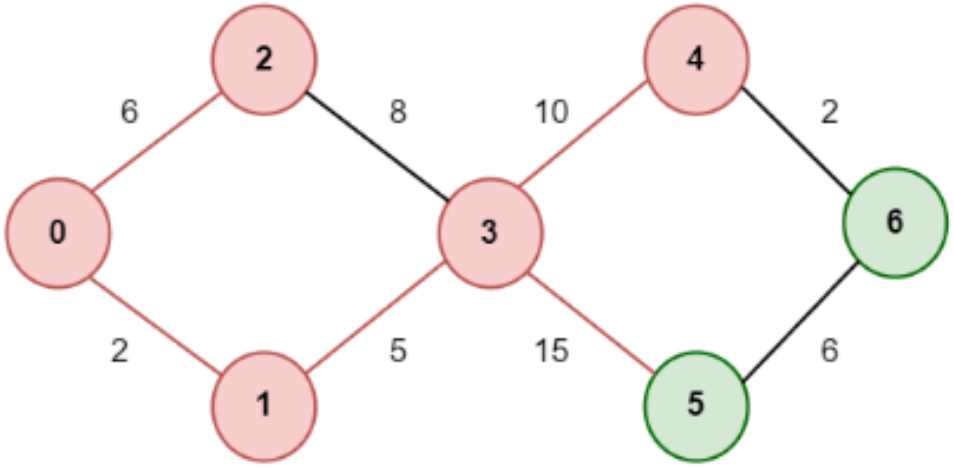

Рисунок А.5 – Четвертий крок алгоритму Дейкстри

Крок 5

Наступною вершиною для відвідування буде вершина 6. Вона має меншу відстань від початкової вершини 0, ніж вершина 5, що залишилась. Позначимо вершину 6 як відвідану та занотуємо шлях до цієї вершини, як 19.

Вершина 0 → Вершина 1 → Вершина 3 → Вершина 4 → Вершина 6 = 0 + 2 + 5 + 10 + 2 = 19.

У цьому ж кроці знайдемо шлях до останньої вершини 5. Позначимо її як відвідану та занотуємо шлях до цієї вершини, як 22.

Вершина 0 → Вершина 1 → Вершина 3 → Вершина 5 = 0 + 2 + 5 + 15 = 22.

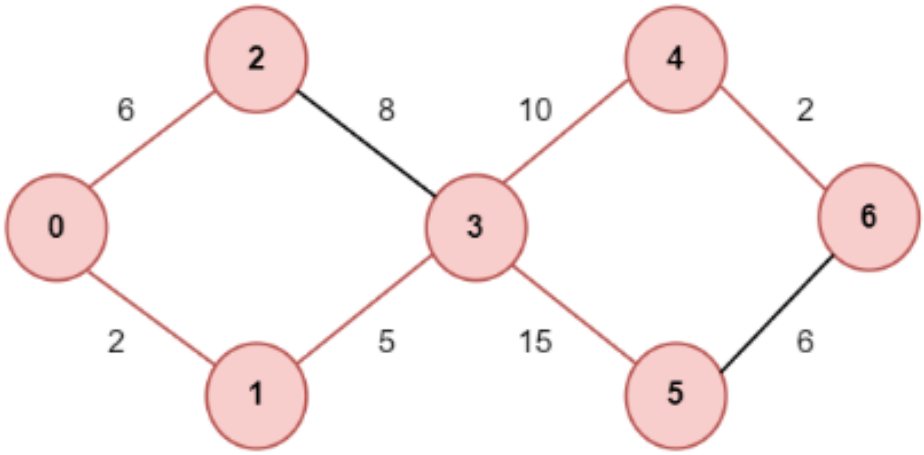

Рисунок А.6 – Останній крок алгоритму Дейкстри

ДОДАТОК Б

```
show start
Using 3906 out of 262144 bytes
!
! Last configuration change at 22:31:36 UTC Thu May 2 2024
!
version 15.6
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
!
!
no aaa new-model
ethernet lmi ce
!
!
!
no process cpu autoprofile hog
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
```

```
!
!
!
!
!
no ip icmp rate-limit unreachable
!
!
!
!
!
!
no ip domain lookup
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
!
!
key chain R1_R2
 key 1
  key-string R1_R2_SeCrEt
  cryptographic-algorithm hmac-sha-512
key chain R1_R3
 key 1
  key-string R1_R3_SeCrEt
  cryptographic-algorithm hmac-sha-512
!
!
```

```
!
redundancy
!
no cdp log mismatch duplex
no cdp run
!
ip tcp synwait-time 5
!
!
!
!
!
!
!
!
!
!
!
!
!
!
interface GigabitEthernet0/0
 ip address 192.168.2.1 255.255.255.0
 duplex auto
 speed auto
 media-type rj45
 no cdp enable
!
interface GigabitEthernet0/1
 ip address 192.168.4.2 255.255.255.0
 ip ospf authentication key-chain R1_R2
```

```
 ip ospf dead-interval 60
 ip ospf hello-interval 15
 ip ospf flood-reduction
 duplex auto
 speed auto
 media-type rj45
 no cdp enable
!
interface GigabitEthernet0/2
 ip address 192.168.5.1 255.255.255.0
 ip ospf authentication key-chain R1_R3
 ip ospf dead-interval 60
 ip ospf hello-interval 15
 ip ospf flood-reduction
 duplex auto
 speed auto
 media-type rj45
 no cdp enable
!
interface GigabitEthernet0/3
 no ip address
 shutdown
 duplex auto
 speed auto
 media-type rj45
 no cdp enable
!
router ospf 1
 max-lsa 1000 40 ignore-time 10 ignore-count 4
 ttl-security all-interfaces
```

```
 network 192.168.2.0 0.0.0.255 area 0
 network 192.168.4.0 0.0.0.255 area 0
 network 192.168.5.0 0.0.0.255 area 0
!
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
!
!
!
!
control-plane
!
banner exec ^C
**********************************************************************
**********
* IOSv is strictly limited to use for evaluation, demonstration and IOS  *
* education. IOSv is provided as-is and is not supported by Cisco's      *
* Technical Advisory Center. Any use or disclosure, in whole or in part, *
* of the IOSv Software or Documentation to any third party for any       *
* purposes is expressly prohibited except as otherwise authorized by     *
* Cisco in writing.                                                      *
**********************************************************************
**********^C
banner incoming ^C
**********************************************************************
**********
* IOSv is strictly limited to use for evaluation, demonstration and IOS  *
```

```
        * education. IOSv is provided as-is and is not supported by Cisco's      *
        * Technical Advisory Center. Any use or disclosure, in whole or in part, *
        * of the IOSv Software or Documentation to any third party for any       *
        * purposes is expressly prohibited except as otherwise authorized by     *
        * Cisco in writing.                                                      *
        **************************************************************************
**************^C
        banner login ^C
        **************************************************************************
**************
        * IOSv is strictly limited to use for evaluation, demonstration and IOS  *
        * education. IOSv is provided as-is and is not supported by Cisco's      *
        * Technical Advisory Center. Any use or disclosure, in whole or in part, *
        * of the IOSv Software or Documentation to any third party for any       *
        * purposes is expressly prohibited except as otherwise authorized by     *
        * Cisco in writing.                                                      *
        **************************************************************************
**************^C
        !
        line con 0
         exec-timeout 0 0
         privilege level 15
         logging synchronous
        line aux 0
         exec-timeout 0 0
         privilege level 15
         logging synchronous
        line vty 0 4
         login
         transport input none
```

```
!
no scheduler allocate
!
end
```