Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

(повна назва факультету)

(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавра
(назва освітнього ступеня)

на тему: Дослідження криптографічних протоколів захисту інформації в мережі Інтернет

Виконала: студентка 4 курсу, групи СБ-41
спеціальності 125 - кібербезпека
(шифр і назва спеціальності)

(підпис) Костюк К.О. (прізвище та ініціали)

Керівник (підпис) Загородна Н.В. (прізвище та ініціали)

Нормоконтроль (підпис) Лобур Т.Б. (прізвище та ініціали)

Завідувач кафедри (підпис) Загородна Н.В. (прізвище та ініціали)

Рецензент (підпис) (прізвище та ініціали)

Тернопіль
2023

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет _____
(повна назва факультету)

Кафедра _____
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____ _____
(підпис) (прізвище та ініціали)
« » 20___ р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня _____
(назва освітнього ступеня)

за спеціальністю _____
(шифр і назва спеціальності)

студентці _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____

Керівник роботи _____
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «___» _____ 20___ року № _____

2. Термін подання студентом завершеної роботи _____

3. Вихідні дані до роботи _____

4. Зміст роботи (перелік питань, які потрібно розробити)

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка

Студент _____ _____
(підпис) (прізвище та ініціали)

Керівник роботи _____ _____
(підпис) (прізвище та ініціали)

АННОТАЦІЯ

Дослідження криптографічних протоколів захисту інформації в мережі Інтернет // Кваліфікаційна робота ОР «Бакалавр» // Костюк Катерина Олегівна // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБ-41 // Тернопіль, 2023 // с. 63, рис. – 20, табл. – 16, лістинги – 1, бібліогр. – 24.

Ключові слова: КРИПТОГРАФІЯ, КРИПТОГРАФІЧНІ ПРОТОКОЛИ, ШИФРУВАННЯ, ВРАЗЛИВОСТІ, МЕРЕЖА ІНТЕРНЕТ, АУТЕНТИФІКАЦІЯ, ЦІЛІСНІСТЬ, КОНФІДЕНЦІЙНІСТЬ, ВЕБСЕРВЕР.

Метою даної роботи є детальний аналіз, порівняння та оцінка криптографічних протоколів захисту інформації в мережі Інтернет, що дасть можливість виявити їх переваги та недоліки, а також потенційні вразливості.

Об'єкт дослідження – криптографічні протоколи захисту інформації в мережі Інтернет.

Предмет дослідження – алгоритми захисту інформації в криптографічних Інтернет-протоколах, виявлення та усунення вразливостей протоколів.

В кваліфікаційній роботі проведено порівняння криптографічних примітивів, аналіз криптографічних алгоритмів, порівняння криптографічних протоколів захисту в мережі Інтернет та їх версій, оцінка стану безпеки вебсервера, аналіз та усунення виявлених вразливостей вебсервера, що пов'язані з підтримкою небезпечних версій криптографічних протоколів SSL/TLS.

Результатом роботи є виявлення та усунення вразливостей вебсервера, пов'язаних підтримкою небезпечних версій криптографічних Інтернет-протоколів.

Для реалізації даної роботи були використані такі програмні продукти: VMware Workstation Pro, Tenable Nessus Vulnerability Scanner, Draw.io.

ABSTRACT

Research on cryptographic information protection protocols in the Internet // Qualification work for a Bachelor's degree // Kostiuk Kateryna Olegivna // Ternopil Ivan Puluj National Technical University, Faculty of Computer and Information Systems and Software Engineering, Department of Cybersecurity, SB-41 group // Ternopil, 2023 // p. 64, fig. - 20, tab. - 16, listings - 1, bibl. - 24.

Keywords: CRYPTOGRAPHY, CRYPTOGRAPHIC PROTOCOLS, ENCRYPTION, VULNERABILITIES, INTERNET, AUTHENTICATION, INTEGRITY, CONFIDENTIALITY, WEB SERVER.

The purpose of this work is a detailed analysis, comparison, and evaluation of cryptographic information protection protocols in the Internet, which will allow identifying their advantages, disadvantages, and potential vulnerabilities.

The object of research is cryptographic information protection protocols in the Internet.

The subject of research is information protection algorithms in cryptographic Internet protocols, detection and elimination of protocol vulnerabilities.

In the qualification work, a comparison of cryptographic primitives is conducted, cryptographic algorithms are analyzed, comparison of cryptographic protection protocols in the Internet and their versions is performed, the security state of a web server is evaluated, and identified vulnerabilities related to the support of insecure versions of SSL/TLS cryptographic protocols in the web server are analyzed and eliminated.

The result of the work is the detection and elimination of vulnerabilities in the web server related to the support of insecure versions of cryptographic Internet protocols.

The following software products were used for the implementation of this work: VMware Workstation Pro, Tenable Nessus Vulnerability Scanner, Draw.io.

ЗМІСТ

ВСТУП

У сучасному цифровому суспільстві, інформаційна безпека у глобальній мережі Інтернет є особливо важливим аспектом. Масштабний обмін даними між персональними комп'ютерами та серверами часто містить конфіденційну інформацію. Сьогодні, онлайн-операції є максимально різноманітними, включаючи банківські транзакції, корпоративні комунікації та багато іншого, що реалізується через мережу Інтернет, роблячи цю інформацію потенційною ціллю для кіберзлочинців.

Ці проблеми спонукали до створення та розвитку криптографічних протоколів для забезпечення інформаційної безпеки в мережі Інтернет. Такі протоколи формують необхідні умови для безпечного передавання даних, використовуючи різноманітні методики шифрування, які забезпечують недоступність інформації для зловмисників. Завдання інформаційного захисту в Інтернеті стає все більш складним, вимагаючи неперервних наукових досліджень та розробки новітніх технологій.

Актуальність данної дипломної роботи визначається необхідністю детального вивчення та оцінки різних криптографічних протоколів, як-то SSL та TLS, для оптимізації захисту вебсерверів. Цільовим завданням роботи є аналіз, порівняння та оцінювання криптографічних протоколів інформаційного захисту в мережі Інтернет, що сприятиме виявленню їх сильних та слабких сторін, а також потенційних вразливостей. Об'єктом дослідження є криптографічні протоколи інформаційного захисту в мережі Інтернет, а предметом – алгоритми цих протоколів, виявлення та усунення їх вразливостей.

Попри наявність великої кількості криптографічних протоколів, наукові дослідження в цій області продовжуються, найчастіше в контексті появи передових технологій, таких як квантові обчислення, що можуть кардинально змінити існуючі моделі захисту. Сучасні досягнення у галузі криптографії включають розробку нових алгоритмів та протоколів, які можуть враховувати потенційно можливі квантові атаки. Для вдосконалення цих систем потрібно розуміти актуальний стан протоколів захисту, їх сильних та слабких сторін.

Отже, представлена робота відіграє важливу роль в науковому та практичному контекстах, вносить вагомий вклад у розуміння інформаційного захисту в мережі Інтернет. Робота спрямована на детальне дослідження ключових криптографічних протоколів, що використовуються для захисту даних під час передачі по мережі. Вивчення механізмів цих протоколів та їх впливу на інформаційну безпеку підсилює розуміння оптимізації процесу передачі даних.

Особлива увага в роботі приділяється перевірці рівня безпеки, що надається цими протоколами. Отримані дані підтверджують, що правильно інтегровані та налаштовані протоколи можуть суттєво покращити безпеку даних, що передаються в мережі Інтернет, роблячи цей процес більш надійним та безпечним.

РОЗДІЛ 1 ЗАГАЛЬНІ ПОНЯТТЯ КРИПТОГРАФІЇ

1.1 Загальні криптографічні поняття

Криптографія – це наука та практика забезпечення конфіденційності, автентичності та цілісності даних шляхом застосування математичних алгоритмів та методів. Основна мета криптографії полягає у захисті інформації від несанкціонованого доступу, змін чи некоректного використання.

Криптографія відіграє важливу роль у багатьох галузях, включаючи інформаційну безпеку, електронну комерцію, банківську справу, мобільні комунікації та багато іншого. Вона забезпечує захист даних, які передаються через мережу, зберігаються на носіях чи пристроях або передаються між учасниками.

Основні принципи криптографії включають:

– Конфіденційність – інформація ніколи не розкривається не авторизованим користувачам.

– Ідентифікацію та автентифікацію – перед обміном даними відправник та одержувач ідентифікуються, а потім проходять автентифікацію.

– Цілісність – інформація не змінюється і не переміщається.

– Запобігання відмови від відповідальності – не можна відмовитися від створення або передачі повідомлення, що забезпечує цифрову легітимність та відстеження транзакцій.

Дамо визначення основним термінам, що використовуються у криптографії:

– шифрування – процес перетворення вихідного повідомлення (відкритого тексту) на зашифроване з використанням певного алгоритму та ключа;

– розшифрування – процес зворотного перетворення зашифрованого повідомлення у вихідне повідомлення (відкритий текст) використовуючи відповідний ключ та алгоритм;

– дешифрування – процес зворотного перетворення зашифрованого повідомлення у вихідне повідомлення (відкритий текст) не використовуючи відповідний ключ;

– ключ – параметр, який використовується у процесі шифрування та розшифрування. Ключ може бути симетричним (секретним) (у разі симетричного шифрування) – коли для шифрування і розшифрування використовується один ключ, або особистим (секретним) чи відкритим (у разі асиметричного шифрування), коли різні ключі використовуються для різних операцій;

– алгоритм шифрування – математична процедура, що визначає спосіб перетворення вихідного повідомлення на зашифроване;

– автентифікація – перевірка справжності ідентичності чи підтвердження, що певна особа чи сутність є тим, ким себе видає;

– хешування – процес перетворення довільних даних у хеш-значення фіксованої довжини з використанням хеш-функції.

Для побудови криптостійких систем використовують з багаторазовим повторенням криптографічні примітиви. Криптографічні примітиви використовують прості перетворення: підстановки, перестановки, циклічні зсуви та шифри XOR.

Наведемо основні криптографічні примітиви:

– симетричне шифрування;

– асиметричне шифрування;

– цифровий підпис;

– хеш-функції.

Нижче наведено схему, яка ілюструє основні криптографічні примітиви та їх зв'язок один з одним (див. рисунок 1.1).

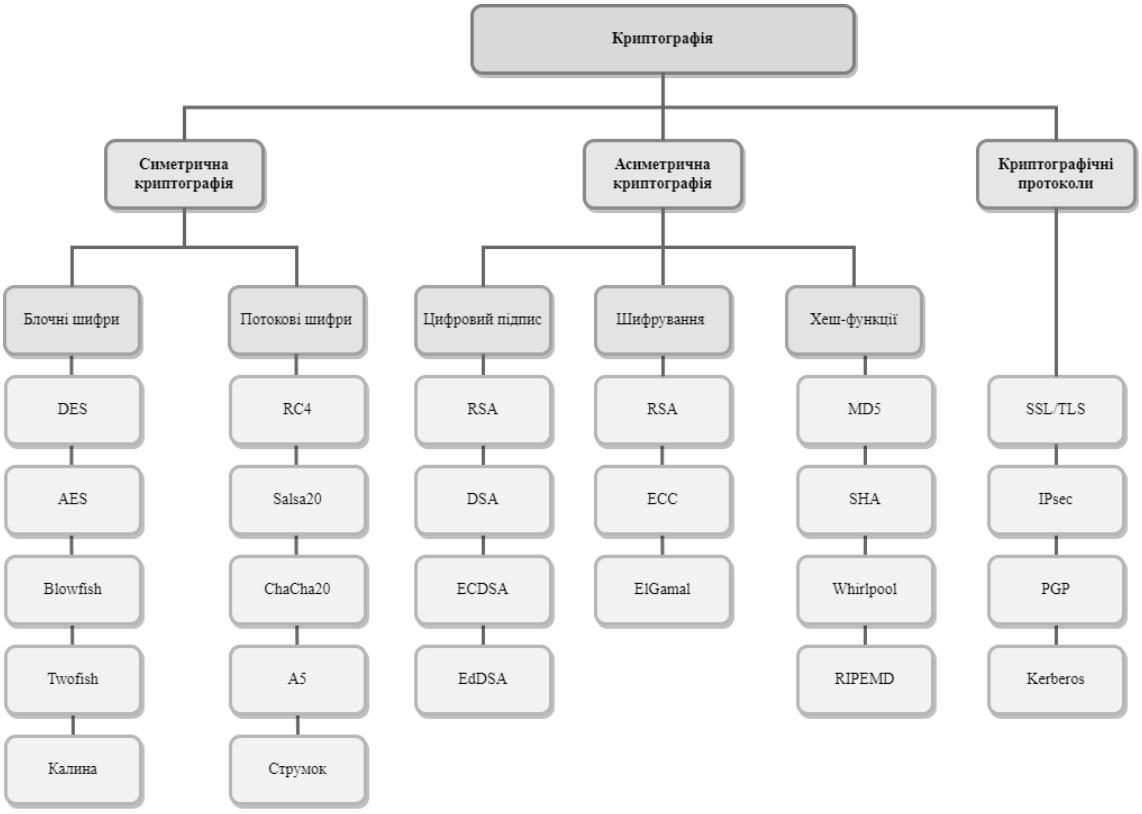

Рисунок 1.1 – Основні криптографічні примітиви

Симетричне шифрування – це примітив, у якому секретний (симетричний) ключ використовується для шифрування і розшифрування даних (див. рисунок 1.2). Приклади алгоритмів симетричного шифрування включають AES та DES. Цей метод ефективний та швидкий, але вимагає безпечного обміну ключами між комунікуючими сторонами.

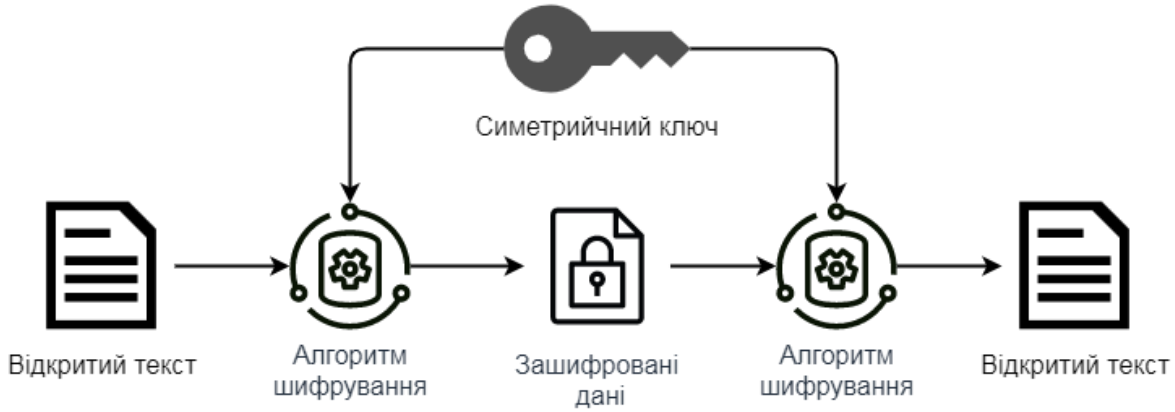

Рисунок 1.2 – Схема симетричного шифрування

Асиметричне шифрування – це примітив, у якому використовується два різні ключі: відкритий та особистий (див. рисунок 1.3). Для шифрування даних

використовується відкритий ключ, а особистий – для їх розшифрування. RSA та ECC є найпоширенішими прикладами асиметричних шифрів. Цей метод забезпечує безпеку передачі ключів, але може бути менш ефективним при обробці великих обсягів даних.

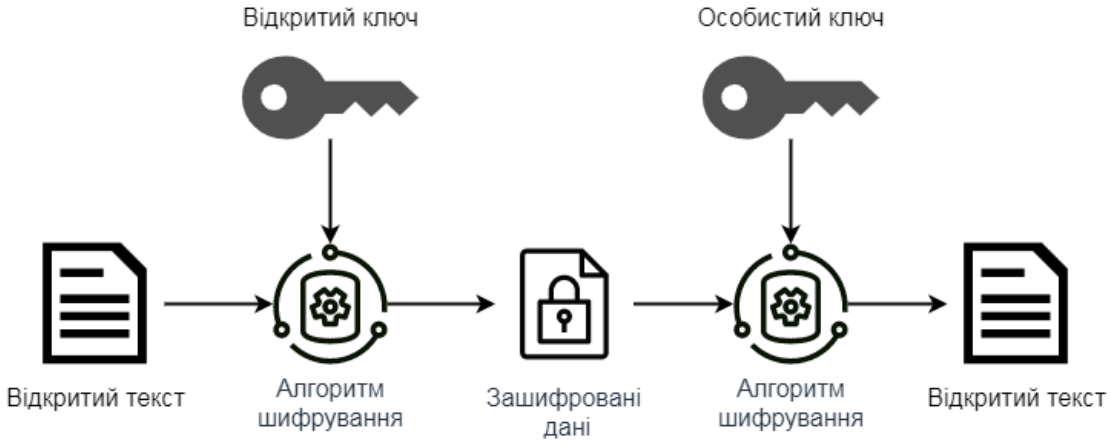

Рисунок 1.3 – Схема асиметричного шифрування

Симетричне та асиметричне шифрування є двома основними методами криптографії, які використовуються для захисту інформації в цифровому світі. Кожен з цих методів має свої переваги і обмеження, а їх вибір залежить від конкретних вимог безпеки та зручності використання. В таблиці 1.1 наведено порівняння симетричного та асиметричного шифрування за ключовими аспектами, такими як принципи шифрування, обробка даних, обмін ключами, застосування, а також наводяться приклади алгоритмів, які були розроблені на базі кожного з цих типів шифрування. Це дозволяє краще зрозуміти особливості та унікальні характеристики кожного з типів шифрування, а також дати об'єктивне уявлення про їх можливості та обмеження.

Таблиця 1.1 – Порівняння симетричного та асиметричного шифрування за ключовими аспектами

Аспект	Симетричне шифрування	Асиметричне шифрування
Принцип шифрування	Використовується один і той же ключ для шифрування та розшифрування даних	Використовуються два ключі: приватний і публічний ключі
Обробка даних	Швидка та ефективна	Може бути повільною залежно від розміру даних
Обмін ключами	Вимагає безпечного обміну однаковим ключем між відправником і отримувачем	Використовується публічний ключ для шифрування та приватний ключ для розшифрування
Застосування	Захист великих обсягів даних, шифрування файлів та дисків	Електронна підписка, аутентифікація, обмін ключами
Приклади алгоритмів	AES, DES, 3DES	RSA, ECC, ElGamal

Наступна таблиця 1.2 надає огляд деяких популярних алгоритмів шифрування, разом з їх коротким описом. Ці алгоритми розроблені для різних цілей і мають різні характеристики щодо безпеки, швидкодії та масштабованості.

Таблиця 1.2 – Популярні алгоритми шифрування

Алгоритм шифрування	Опис
AES (Advanced Encryption Standard)	Симетричний блочний алгоритм шифрування, що використовується для захисту конфіденційності даних
RSA (Rivest-Shamir-Adleman)	Асиметричний алгоритм шифрування, який використовується для шифрування та цифрових підписів
ECC (Elliptic Curve Cryptography)	Асиметричний алгоритм шифрування, оснований на математичних властивостях еліптичних кривих
Blowfish	Симетричний блочний алгоритм шифрування з високою швидкодією
3DES (Triple Data Encryption Standard)	Симетричний блочний алгоритм шифрування, що застосовується у трьох послідовних раундах

Цифровий підпис – електронна аналогія звичайного підпису, що

використовується для підтвердження справжності та цілісності даних. Цифровий підпис створюється з використанням особистого ключа та може бути перевірений з використанням відповідного відкритого ключа. Цифрові підписи забезпечують доказ авторства та неможливість відмови від авторства. Приклади алгоритмів цифрового підпису – RSA, DSA, ECDSA, EdDSA. Короткий опис алгоритмів цифрового підпису знаходиться в таблиці 1.3.

Таблиця 1.3 – Популярні алгоритми цифрового підпису

Тип цифрового підпису	**Опис**
RSA (Rivest-Shamir-Adleman)	Асиметричний алгоритм, який використовується для цифрових підписів та шифрування даних
DSA (Digital Signature Algorithm)	Стандартний асиметричний алгоритм для цифрових підписів
ECDSA (Elliptic Curve Digital Signature Algorithm)	Асиметричний алгоритм, оснований на еліптичних кривих, використовується для цифрових підписів
EdDSA (Edwards-curve Digital Signature Algorithm)	Сучасний асиметричний алгоритм для цифрових підписів, який базується на кривих Едвардса

Хеш-функції – використовуються для перетворення даних довільної довжини у хеш-значення фіксованої довжини. Хеш-функції повинні мати властивості односторонності (неможливість відновлення вихідних даних з хеш-значення) і стійкості до колізій (складності знаходження двох різних повідомлень з однаковим хешем). Прикладами широко використовуваних хеш-функцій є SHA (Secure Hash Algorithm) та MD5 (Message Digest Algorithm 5). Обидва ці алгоритми представляють собою стандартні хеш-функції, які забезпечують високий рівень безпеки. Короткий опис кількох типів хеш-функцій, що включають інформацію про їх характеристики, міститься в таблиці 1.4.

Таблиця 1.4 – Опис найбільш відомих хеш-функцій

Тип хеш-функції	Опис
MD5 (Message Digest Algorithm 5)	Широко використовуваний криптографічний хеш-алгоритм, який створює 128-бітний хеш-значення. Зараз вважається небезпечним через вразливості до атак.
SHA-1 (Secure Hash Algorithm 1)	Криптографічний хеш-алгоритм, який створює 160-бітний хеш-значення. Вважається небезпечним через здатність до колізій.
SHA-2 (Secure Hash Algorithm 2)	Сімейство хеш-алгоритмів з різними довжинами хеш-значення (224, 256, 384, 512, 512/224, 512/256 біт). Використовується в багатьох застосунках безпеки.
SHA-3 (Secure Hash Algorithm 3)	Останнє покоління Secure Hash Algorithm. Переможець конкурсу NIST на криптографічний хеш-алгоритм.

Після огляду типів хеш-функцій порівняємо взагальному хеш-функції з цифровим підписом для кращого розуміння їх застосування в криптографії. Наступна порівняльна таблиця 1.5 надасть огляд їх основних характеристик та ефективності використання в конкретних цілях.

Таблиця 1.5 – Порівняння хеш-функцій і цифрового підпису

Аспект	Хеш-функція	Цифровий підпис
Використання ключів	Не потребує використання ключів	Приватний ключ для підписування та публічний ключ для перевірки
Конфіденційність	Ні	Ні
Цілісність	Так	Так
Застосування	Верифікація цілісності даних, порівняння даних	Автентифікація, цифровий підпис, безвідмовність даних
Алгоритми	MD5, SHA-1, SHA-256	RSA, DSA, ECDSA

Криптографічні протоколи – набори правил, які визначають, як деталі

криптографії мають бути використані для забезпечення безпеки в певному контексті. Вони включають різні методи для автентифікації, шифрування, цифрового підпису та інших операцій.

В таблиці 1.6 перераховані різні аспекти криптографічних протоколів, такі як захист даних, безпека мереж, захист електронної пошти та захист з'єднання. Кожен аспект включає популярний криптографічний протокол, пов'язаний з цим аспектом, а також основні характеристики цього протоколу.

Таблиця 1.6 –Ключові аспекти криптографічних протоколів

Аспект	**Криптографічний протокол**	**Основні характеристики**
Захист даних	SSL/TLS	Шифрування, автентифікація, цифровий підпис
Безпека мереж	IPsec	Шифрування на рівні мережі
Захист електронної пошти	PGP, S/MIME	Шифрування, автентифікація, цифровий підпис
Захист з'єднання	SSH	Шифрування, автентифікація

При застосуванні криптографічного захисту для інформації важливо враховувати сучасні стандарти та рекомендації щодо криптографічної безпеки. Додатково, необхідно використовувати надійні алгоритми шифрування та забезпечити безпеку ключів, так як вони є основою криптографічного захисту.
Вибір відповідних методів криптографічного захисту залежить від конкретних вимог, обмежень та ризиків, пов'язаних з системою або застосуванням. Важливо використовувати сучасні та надійні алгоритми шифрування та враховувати рекомендації щодо криптографічної безпеки.

1.2 Порівняння криптографічних примітивів

У криптографії використовуються різні примітиви, які забезпечують

безпеку і конфіденційність інформації. Розуміння різниці між цими примітивами є ключовим для вибору відповідних методів криптографічного захисту для системи.

Симетричне шифрування є одним з основних примітивів, при якому використовується один і той самий секретний ключ для шифрування та розшифрування даних. Воно відоме своєю швидкодією і ефективністю, але вимагає безпечного обміну секретним ключем між комунікуючими сторонами.

Асиметричне шифрування використовує пару ключів: публічний і приватний. Публічний ключ використовується для шифрування повідомлень, а приватний ключ - для розшифрування. Цей примітив надає велику перевагу у безпеці, оскільки немає необхідності обмінюватися секретним ключем, але він є обчислювально більш витратним.

Цифровий підпис використовується для підтвердження автентичності повідомлення та забезпечення цілісності даних. Він використовується з хеш-функціями та приватним ключем для створення унікального підпису, який може бути перевірений за допомогою публічного ключа. Цей примітив є важливим для забезпечення недеформованості та незаперечності повідомлень.

Хеш-функції, з свого боку, перетворюють будь-які дані фіксованої довжини в хеш-код. Вони використовуються для перевірки цілісності повідомлень та даних, а також для генерації унікальних ідентифікаторів. Хеш-функції необоротні, що означає, що неможливо відновити початкові дані з хеш-значення.

Таблиця 1.7 надає загальний огляд різних примітивів криптографії та їх характеристик. Ці примітиви – це базові інструменти, на основі яких будуються більш складні криптографічні системи. Варто зазначити, що вибір відповідних методів криптографічного захисту залежить від конкретних потреб, вимог і обмежень системи. Ретельний аналіз цих примітивів та їх властивостей допоможе прийняти обґрунтовані рішення щодо використання криптографічних протоколів, що найкраще відповідають потребам у безпеці та захисті інформації. Завдяки цьому можна розробити і реалізувати стратегії захисту, які найкращим чином відповідатимуть конкретним потребам в захисті та безпеці інформації.

Таблиця 1.7 – Порівняння криптографічних примітивів

Аспект	Симетричне шифрування	Асиметричне шифрування	Цифровий підпис	Хеш-функція
Основний принцип	Використовується один і той же ключ для шифрування та розшифрування даних	Використовується публічний та приватний ключі для шифрування та розшифрування даних	Використовується приватний ключ для підписування даних та публічний ключ для перевірки підпису	Генерація унікального хешу для вхідних даних, який використовується для верифікації цілісності
Ключі	Вимагає обмін та зберігання одного і того ж ключа між відправником і отримувачем	Використовує два ключі: приватний (зберігається в таємниці) та публічний (розповсюджується)	Використовується приватний ключ для підписування та перевірки підпису	Ключі не використовуються
Підтримка конфіденційності	Так	Так	Ні	Ні
Підтримка цілісності	Ні	Так	Так	Так
Витрати обчислювальних ресурсів	Низькі	Високі	Високі	Залежить від протоколу та алгоритму шифрування
Використання у реальному часі	Так	Так	Так	Так
Основне застосування	Шифрування даних на місці та під час передачі	Шифрування, аутентифікація, цифрові підписи	Підтвердження авторства, цілісності та безвідмовності	Верифікація цілісності, створення відбитків даних
Приклади алгоритмів	AES, DES, 3DES	RSA, ECC	RSA, ECDSA	MD5, SHA-1, SHA-2, SHA-3

Ця таблиця надає загальну інформацію про різні аспекти та властивості симетричного шифрування, асиметричного шифрування, цифрового підпису та хеш-функцій. Важливо враховувати конкретні потреби та вимоги системи для вибору найбільш підходящих методів криптографічного захисту, що залежить від конкретних вимог, обмежень та ризиків, пов'язаних з системою або застосуванням.

РОЗДІЛ 2 КРИПТОГРАФІЧНІ ПРОТОКОЛИ ЗАХИСТУ ІНФОРМАЦІЇ В МЕРЕЖІ ІНТЕРНЕТ

2.1 Криптографічні протоколи

2.1.1 Огляд криптографічних протоколів

Криптографічний протокол – це набір правил та процедур, які визначають спосіб взаємодії між різними учасниками чи системами для забезпечення безпеки даних та комунікацій. Протоколи криптографії використовують різні криптографічні примітиви та алгоритми для захисту інформації від несанкціонованого доступу, зміни чи підробки.

Криптографічні протоколи визначають послідовність кроків та обмін повідомленнями між учасниками, включаючи ініціалізацію, автентифікацію, ключовий обмін, шифрування, розшифрування, перевірку цілісності та інші операції. Вони забезпечують конфіденційність, цілісність, автентифікацію та інші властивості безпеки даних у межах комунікації.

Під учасником протоколу розуміють як людей, так і застосунки, програми, групи людей чи організації. Формально учасниками вважають лише тих, хто виконує активну роль у рамках протоколу.

Протокол складається з циклів (проходів). Цикл – часовий інтервал активності лише одного учасника. За винятком першого циклу, зазвичай протокол починається з прийому повідомлення, а закінчується – відправкою.

Цикл (прохід) складається з кроків – конкретних закінчених дій, які виконує учасник протоколу (див. рисунок 2.1). Наприклад:

- генерація нового (випадкового) значення;
- обчислення значень функції;
- перевірка ключів, сертифікатів, підписів тощо;
- прийом, відправка повідомлень.

Рисунок 2.1 – Схематичне зображення циклу дій учасника протоколу

Приклади криптографічних протоколів включають SSL/TLS для безпечного з'єднання веб-сайтів, протоколи автентифікації (OAuth, OIDC), протоколи обміну ключами (наприклад, DH), та інших, що застосовуються в різних галузях, включаючи безпеку електронної пошти, мобільних застосунків, мережевих протоколів тощо.

Мета криптографічних протоколів – забезпечити захист даних у відкритих чи ненадійних мережевих середовищах, де є вірогідність атак чи спроб неправомірного доступу до інформації. Правильне проектування, реалізація та використання криптографічних протоколів відіграють важливу роль у забезпеченні безпеки інформаційних систем, захисті конфіденційності, доступності та цілісності даних.

2.1.2 Класифікація криптографічних протоколів

Не існує загальновизнаної класифікація криптографічних протоколів, проте можна класифікувати їх за різними параметрами, наприклад, за функціональним призначенням, за кількістю учасників, за характером взаємодії між учасниками та іншими параметрами.

Однак за наборами ознак та характеристик (див. рисунок 2.2) можна розділити протоколи на класи.

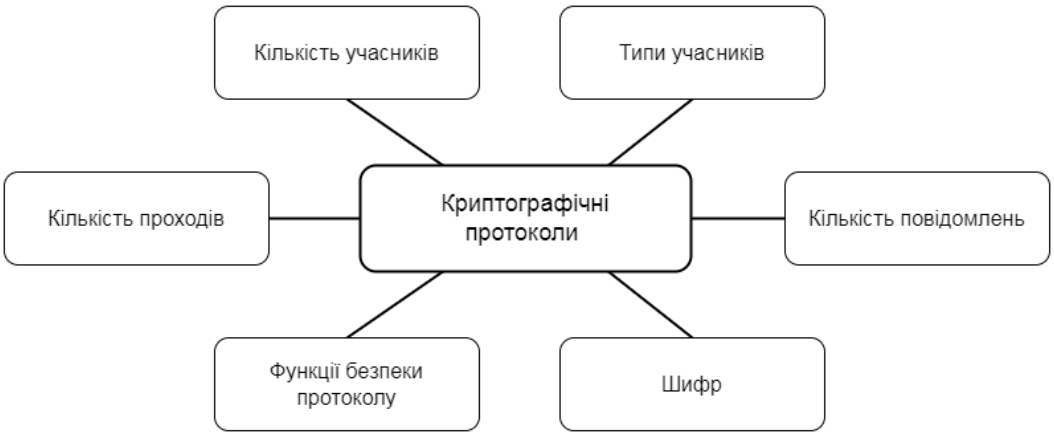

Рисунок 2.2 – Схематичне представлення переліку розповсюджених критеріїв класифікацій криптографічних протоколів

За кількістю учасників протоколи можуть поділятись на:

- двосторонні;
- тристоронні;
- багатосторонні.

За кількістю повідомлень, що передаються, протоколи можуть класифікуватись наступним чином:

- інтерактивний (взаємний обмін повідомленнями);
- неінтерактивний (одноразова передача повідомлень між учасниками).

У криптографічних протоколах можуть використовуватись:

- симетричні шифри;
- асиметричні шифри.

Захищена система та відповідно захищений протокол можуть виконувати різні функції безпеки. Багато цих функцій (цілей) можна сформулювати як стійкість до певного класу атак. Найбільш повним і актуальним вважається перерахування та тлумачення цих цілей у документі проекту AVISPA, що підсумовує описи з різних документів IETF. Прийнято вважати, що, для окремих протоколів, можна формально довести або спростувати досягнення цих цілей.

За типами учасників протоколи можуть поділятись на:

- однорангові – всі учасники можуть виконувати будь-які ролі в рамках протоколу;
- з довіреним посередником – у протоколі завжди бере участь третя

довірена сторона;

– з довіреним арбітром – у протоколі може брати участь третя довірена сторона, якщо решта учасників не дійшли згоди.

2.2 Криптографічні протоколи захисту інформації у мережі Інтернет

Інтернет це розподілена мережа, де дані передаються через велику кількість вузлів і маршрутизаторів. Це означає, що повинно бути забезпечено безпеку даних в умовах непередбачуваного та ненадійного оточення. У тому числі повинно передбачити захист даних від перехоплення, заміни та втручання у процесі їх передачі. В Інтернеті передається велика кількість даних, включаючи веб-сторінки, електронну пошту, файли, потокове відео і аудіо. Таким чином засоби забезпечення захисту даних повинні бути ефективними та працювати у штатному режимі в умовах високого завантаження мережі та великого обсягу інформації, що передається.

У мережі Інтернет існує велика кількість різних протоколів, пристроїв, операційних систем та програмних платформ, які взаємодіють один з одним. Засоби захисту повинні бути стандартизованими та забезпечувати сумісність між різними середовищами, пристроями та платформами, щоб забезпечити безпеку та захист даних незалежно від технологій, що використовуються.

Криптографія стала невід'ємною частиною мережі Інтернет завдяки комерціалізації мережі, а криптографічні протоколи - невід'ємною частиною протоколів Інтернету, оскільки вони дозволяють забезпечити безпеку інформації, що передається.

Вперше криптографія була застосована у мережі Інтернет для забезпечення безпеки електронної пошти та забезпечення безпечної передачі.

Для електронної пошти використовувались протоколи безпеки S/MIME та PGP. Вони забезпечує шифрування та цифрові підписи для захисту конфіденційності та підтвердження справжності електронних повідомлень.

Для забезпечення шифрованого з'єднання між веб-сервером та веб-браузером було розроблено протокол SSL. Цей протокол було інтегровано у веб-

браузери і він дозволяв шифрувати конфіденційну інформацію, що надсилається через веб-сайти.

SSL та його наступник TLS – це протоколи, які забезпечують безпечне з'єднання між клієнтом та сервером. Вони використовуються для захисту даних, включаючи веб-сторінки, електронну пошту, онлайн-платежі та інші онлайн-сервіси. SSL/TLS забезпечує конфіденційність, цілісність та автентифікацію даних шляхом шифрування та перевірки сертифікатів.

Нові криптографічні протоколи були розроблені для захисту пакетів IP, включаючи протоколи PPTP, L2TP та IPsec. Ці протоколи часто використовуються для створення віртуальних приватних мереж (VPN) та забезпечують захист більшої частини трафіку. IPsec – це протокол безпеки на мережевому рівні, який забезпечує захист IP-пакетів під час їх передачі через мережу. Він використовується для створення VPN та забезпечує шифрування, автентифікацію та цілісність даних на рівні IP.

Також існують інші протоколи, такі як SSH для шифрування та автентифікації віддалених з'єднань. SSH – це протокол для безпечного віддаленого доступу та виконання команд на віддалених комп'ютерах. Він забезпечує шифрування комунікацій та автентифікацію користувачів, захищаючи від несанкціонованого доступу та підміни даних.

У сучасному стані криптографічні протоколи в мережі Інтернет продовжують відігравати критичну роль у забезпеченні безпеки передачі даних та захисті конфіденційності користувачів. Однак, з розвитком технологій та появою нових загроз, сучасна криптографія стикається з новими викликами та потребує постійного оновлення та вдосконалення.

Однією з основних тенденцій у сучасній криптографії є перехід до використання сильніших алгоритмів шифрування. Традиційні алгоритми, такі як RSA та DES, поступово замінюються безпечнішими алгоритмами, наприклад, алгоритмами на основі криптографії з відкритим ключем (наприклад, ECC) та алгоритмами блокового шифрування (наприклад, AES). Ці алгоритми забезпечують вищий рівень захисту та криптостійкості.

Ще однією важливою тенденцією є розширення застосування

криптографічних протоколів на різних рівнях Інтернету. Разом з традиційними протоколами, такими як SSL/TLS для захисту веб-з'єднань, існують нові протоколи, такі як DNSSEC (для забезпечення автентичності та цілісності DNS), HTTPS (захищена версія HTTP), IPsec (для захисту передачі даних на рівні IP), OAuth та OIDC для автентифікації. Це дозволяє забезпечити безпеку на різних рівнях інфраструктури мережі.

Протокол DNSSEC є набором розширень до протоколу DNS. Мета протоколу DNSSEC полягає у запобіганні атакам на систему DNS, таких як підміна даних та підроблені відповіді від DNS-серверів. Він забезпечує цілісність даних DNS та автентифікацію їх джерел, що дозволяє користувачам впевнено довіряти отриманим DNS-відповідям.

OAuth – це протокол автентифікації та авторизації, який використовується для безпечного надання доступу до захищених ресурсів від імені користувача без передачі пароля. Він широко застосовується в авторизації через соціальні мережі та API-інтерфейси.

OIDC – це протокол автентифікації, що базується на протоколі OAuth. Він надає безпечний механізм, який дозволяє застосункам зв'язатися зі службою ідентифікації, щоб отримати необхідні дані про користувача та повернути їх назад у програму, забезпечивши повний захист даних користувачів в Інтернеті.

В цілому, сучасний стан використання криптографічних протоколів у мережі Інтернет відображає постійний розвиток та адаптацію технологій до нових проблем та вимог безпеки. Використання сильних алгоритмів, розширення сфери застосування протоколів та урахування балансу між безпекою та зручністю використання є ключовими аспектами у забезпеченні безпеки в Інтернеті.

2.3 Технологія VPN як приклад використання криптографічних протоколів в мережі Інтернет

VPN - це технологія, що створює безпечне з'єднання через небезпечну мережу, таку як Інтернет. Вона дозволяє користувачам отримувати та відправляти

дані через публічні чи загальні мережі, наче їх пристрої підключені до приватної мережі (див. рисунок 2.3).

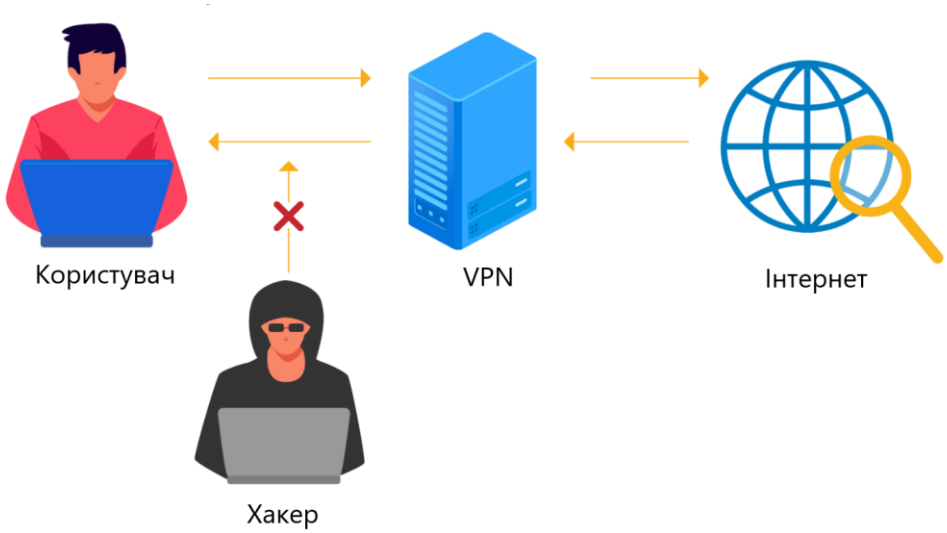

Рисунок 2.3 – Схематичне зображення використання технології VPN

Сценарії використання VPN можуть бути різними, найпопулярніші з них:

– Побудова захищеного каналу між двома або більше віддаленими сегментами мережі.

– Безпечне підключення працівників з віддалених локацій до корпоративної мережі.

– Віртуальна зміна розташування або обхід геоблокування за допомогою послуг VPN Providers.

Для реалізації цих сценаріїв існують різні види протоколів VPN: для зв'язку, для шифрування трафіку та інші, і вже на підставі відповідного протоколу можна приймати рішення. Два найвідоміші протоколи, що широко використовуються – це OpenVPN і IPSec, а порівняно недавно з'явився ще WireGuard. Є й інші альтернативи, які вже застарілі, але цілком здатні вирішувати певні завдання.

Перевага того чи іншого протоколу VPN залежить від низки факторів та умов використання. Фактори та умови, які можуть впливати на використання VPN, включають якість Інтернет-з'єднання, тип шифрування, що використовується VPN-провайдером, а також законодавство країни, в якій перебуває користувач. Наприклад, деякі країни забороняють або обмежують

використання VPN, тому користувачам важливо знати чи є використання VPN законним в цій країні. Проте можна проаналізувати загальні переваги та недоліки VPN, що й робимо в таблиці 2.1.

Таблиця 2.1 – Переваги та недоліки VPN

Переваги VPN	**Недоліки VPN**
Захищена передача даних	Потребує налаштування та обслуговування серверів VPN
Конфіденційність інформації	Можливість обмеження швидкості під час використання VPN
Забезпечує анонімність в мережі	Додаткові витрати на підтримку інфраструктури VPN
Доступ до внутрішньої мережі ззовні	Можливість блокування VPN провайдером або країною
Захищений доступ до віддалених ресурсів	Можливість витоку DNS запитів через VPN

Технологія VPN є одним з найвідоміших прикладів використання криптографічних протоколів в Інтернеті. Криптографія використовується в VPN для забезпечення конфіденційності, шифруючи дані, так що лише призначений отримувач може їх розшифрувати, цілісності, перевіряючи, що дані не були змінені під час передачі без виявлення за допомогою та аутентифікації даних за допомогою цифрових сертифікатів та ключів, що використовуються для підтвердження ідентифікації перед встановленням з'єднання.

VPN може використовувати різні криптографічні протоколи, такі як IPSec, PPTP, SSTP та інші, щоб забезпечити безпечне з'єднання. Отож перейдемо до огляду цих протоколів в контексті використання технологією VPN.

Point-to-Point Tunneling Protocol (PPTP) - один із найстаріших VPN протоколів, що використовуються досі та був розроблений компанією Microsoft (див. рисунок 2.4).

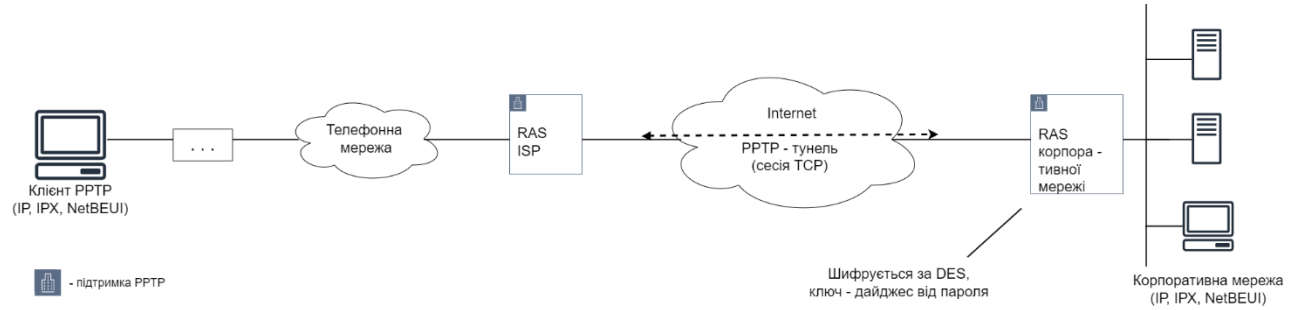

Рисунок 2.4 – Схематичне зображення захищеного каналу PPTP

PPTP використовує два з'єднання - одне для управління, інше для інкапсуляції даних. Перше працює з використанням TCP, у якому порт сервера 1723. Друге працює з допомогою протоколу GRE, який є транспортним протоколом (тобто заміною TCP/UDP). Цей факт заважає клієнтам, що перебувають за NAT, встановити підключення з сервером, так як для них встановлення підключення точка-точка неможливо за умовчанням. Однак, оскільки в протоколі GRE, що використовує PPTP (а саме enhanced GRE), є заголовок Call ID, маршрутизатори можуть ідентифікувати та зіставити GRE трафік, що йде від клієнта локальної мережі до зовнішнього сервера і навпаки. Це дозволяє клієнтам за NAT встановити підключення point-to-point і користуватися протоколом GRE. Ця технологія називається VPN PassTrough. Вона підтримується великою кількістю сучасного клієнтського мережного обладнання.

PPTP підтримується на всіх версіях Windows і більшості інших операційних систем. Незважаючи на відносно високу швидкість, PPTP не надто надійний: після обриву з'єднання він не відновлюється так само швидко, як, наприклад, OpenVPN.

На даний час PPTP є достатньо застарілим і Microsoft радить користуватися іншими рішеннями VPN. Звичайно, якщо VPN використовується виключно для розблокування контенту, PPTP підійде, однак, є більш безпечні варіанти, на які варто звернути увагу.

Secure Socket Tunneling Protocol (SSTP) – продукт від Microsoft. Як і PPTP, SSTP не дуже широко використовується в індустрії VPN, але, на відміну від PPTP, він не діагностує серйозні проблеми з безпекою. SSTP відправляє трафік SSL

через TCP-порт 443. SSTP також доступний і на Linux, RouterOS і SEIL, але переважно використовується Windows-системами.

SSTP влаштований досить просто завдяки тому, що він використовує функціонал інших криптографічних протоколів. Власне, єдина криптографічна функція, що реалізується самим SSTP – це «cryptographic binding». Все шифрування даних здійснюється протоколом SSL. Усі пакети протоколів SSTP, PPP та інших передаються лише у зашифрованому вигляді.

Авторизація проходить відразу за трьома протоколами: SSL, PPP і, власне, сам SSTP. При встановленні SSL з'єднання проходить авторизація сервера клієнтом за сертифікатом SSL. Аутентифікація клієнта сервером допускається, однак не підтримується жодна з серверних Windows.

На рівні PPP відбувається авторизація клієнта сервером і додатково може відбуватися автентифікація сервера. Windows Server підтримує автентифікацію клієнта на рівні PPP за допомогою MS-CHAPv2, EAP-TLS, PEAP-MSCHAPv2, PEAP-TLS. Також підтримуються Password Authentication Protocol (PAP – не шифрований пароль) та CHAP, проте їх використання не рекомендується, тому що вони не передбачають обміну ключовою інформацією, яка потрібна для «cryptographic binding». Власне, тут ті самі методи аутентифікації, що і в PPTP. Відмінність від PPTP полягає в тому, що обмін відбувається всередині вже створеного шифрованого каналу SSL.

Internet Protocol Security (IPsec) – це набір протоколів для забезпечення захисту даних, що передаються IP-мережею (див. рисунок 2.5). На відміну від SSL, який працює на прикладному рівні, IPsec працює на мережевому рівні і може використовуватися з багатьма операційними системами, що дозволяє використовувати його без сторонніх програм (на відміну від OpenVPN).

- Фаза I для вузла A, автентифікація
- Тунель
- Фаза II для вузлів A і B, обмін ключами
- Встановлення тунелю
- Контроль стану тунелю, мінімум кожні 10 с

Рисунок 2.5 - Загальна процедура IPsec

Одна з важливих особливостей IPsec полягає в тому, що він шифрує весь IP-пакет, що передається по мережі. Це означає, що не лише дані внутрішнього повідомлення шифруються, але й заголовки IP-пакету, що містять метадані про передачу, також захищені. IPsec використовує два основні протоколи для забезпечення захисту даних: Authentication Header (AH) який ставить цифровий підпис на кожному пакеті і Encapsulating Security Protocol (ESP), який забезпечує конфіденційність, цілісність та аутентифікацію пакету під час передачі.

IPsec часто використовується разом з протоколами IKEv2 (Internet Key Exchange version 2) та L2TP (Layer 2 Tunneling Protocol) для створення віртуальних приватних мереж (VPN) забезпеченням захищеного каналу передачі даних. Ці комбінації протоколів дозволяють забезпечити конфіденційність, цілісність та аутентифікацію даних, переданих через мережу. Тож розглянемо, як IPsec використовується в парі з L2TP та IKEv2.

Протокол тунелювання другого рівня (L2TP) був вперше представлений в 1999 році як удосконалення протоколів L2F від Cisco і PPTP від Microsoft. Оскільки L2TP не надає шифрування та автентифікації, він часто використовується разом з IPsec. Дана комбінація протоколів, відома як L2TP/IPsec, підтримується багатьма операційними системами і стандартизована RFC 3193.

L2TP/IPsec вважається безпечним і не має серйозних виявлених проблем, що робить його набагато безпечнішим, ніж PPTP. Цей протокол може використовувати шифрування 3DES або AES, але 3DES, який вже вважається слабким, застосовується рідко. L2TP/IPsec може забезпечити високий рівень безпеки передачі даних, простоту в налаштуванні і є сумісним з усіма сучасними операційними системами. Втім, через подвійну інкапсуляцію переданих даних, L2TP/IPsec може бути менш ефективним та повільнішим в порівнянні з іншими VPN-протоколами. До того ж, з протоколом L2TP іноді виникають проблеми, оскільки він стандартно використовує UDP-порт 500, який може бути заблокований деякими файрволами.

Internet Key Exchange version 2 (IKEv2) є протоколом IPsec, що використовується для виконання взаємної аутентифікації, створення та обслуговування Security Associations (SA), стандартизований в RFC 7296. Також захищений IPsec, як і L2TP, що може говорити про їхній однаковий рівень безпеки. Хоча IKEv2 був розроблений Microsoft спільно з Cisco, існують реалізації протоколу з відкритим вихідним кодом (наприклад, OpenIKEv2, Openswan та strongSwan).

Завдяки підтримці Mobility and Multi-homing Protocol (MOBIKE) IKEv2 дуже стійкий до зміни мереж. Це робить IKEv2 відмінним вибором для користувачів смартфонів, які регулярно перемикаються між домашнім Wi-Fi та мобільним з'єднанням або переміщуються між точками доступу.

IKEv2/IPsec може використовувати низку різних криптографічних алгоритмів, включаючи AES, Blowfish та Camellia, у тому числі з 256-бітними ключами.

IKEv2 підтримує Perfect Forward Secrecy. У багатьох випадках IKEv2 швидше за OpenVPN, оскільки він менш ресурсоємний. З точки зору продуктивності IKEv2 може бути найкращим варіантом для мобільних користувачів, оскільки він добре встановлює з'єднання. IKEv2 підтримується на Windows 7+, Mac OS 10.11+, iOS, а також на деяких Android-пристроях.

OpenVPN – це універсальний протокол VPN з відкритим вихідним кодом, розроблений компанією OpenVPN Technologies. На сьогоднішній день це, мабуть,

найпопулярніший протокол VPN. Будучи відкритим стандартом, він пройшов не одну незалежну безпекову експертизу (див. рисунок 2.6).

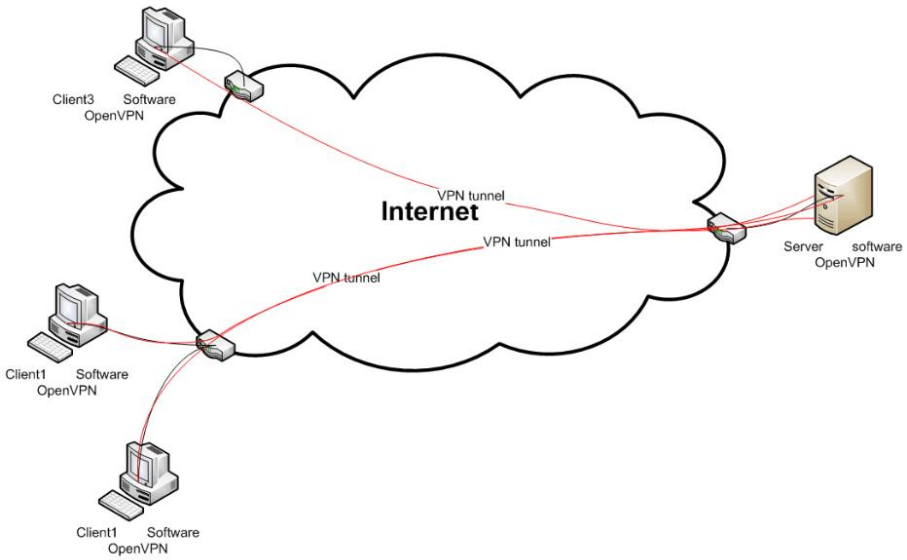

Рисунок 2.6 – Схематичне зображення застосування OpenVPN

У більшості ситуацій, коли потрібне підключення через VPN, швидше за все підійде OpenVPN. Він стабільний та пропонує хорошу швидкість передачі даних. OpenVPN використовує TCP та UDP, які є стандартними протоколами, що й дозволяє OpenVPN стати альтернативою IPsec, коли деякі протоколи VPN блокує провайдер.

Для можливості роботи з OpenVPN необхідно мати спеціальне клієнтське програмне забезпечення. Більшість VPN-сервісів створюють свої програми для роботи з OpenVPN, які можна використовувати в різних операційних системах та пристроях. Протокол може працювати на будь-якому з портів TCP та UPD і може використовуватись на всіх основних платформах через сторонні клієнти: Windows, Mac OS, Linux, Apple iOS, Android.

Найновіший і незвіданий протокол VPN - WireGuard. Позиціонується розробниками як заміна IPsec і OpenVPN для більшості випадків їх використання, будучи більш безпечним, продуктивним і простим у використанні.

Всі IP-пакети, що приходять на інтерфейс WireGuard, інкапсулюються в UDP і безпечно доставляються. WireGuard використовує сучасну криптографію:

– ChaCha20 для шифрування;

- Curve25519 для обміну ключами;
- SipHash для ключів хеш-таблиці;
- Poly1305 для аутентифікації даних;
- BLAKE2 для хешування.

Код WireGuard виглядає набагато простіше, ніж код OpenVPN, внаслідок чого його простіше дослідити на вразливості (4 тисяч рядків коду проти кількох сотень тисяч), також його легше розгорнути та налаштувати.

Результати тестів продуктивності можна побачити на офіційному сайті. Варто відзначити, що найкращі результати WireGuard покаже на Linux-системах, так як там він реалізований у вигляді модуля ядра. Включений до складу ядра Linux код пройшов додатковий аудит безпеки, виконаний незалежною фірмою, який не виявив жодних проблем. Для багатьох це чудові новини, але чи зможе WireGuard стати гідною заміною IPsec та OpenVPN покаже час та незалежні дослідження безпеки.

Вище було охарактеризовано найбільш популярні VPN протоколи, в якості заключення наводжу порівняльну таблицю 2.2, в якій відзначаю важливі технічні характеристики протоколів.

Таблиця 2.2 – Важливі технічні характеристики VPN протоколів

Назва	Розробник	Ліцензія	Розгортання	Шифрування	Порти	Недоліки безпеки
PPTP	Microsoft	Proprietary	Windows, macOS, IOS, якийсь час GNU/Linux. Працює "з коробки", не вимагаючи встановлення додаткового ПЗ	Використовує Microsoft Point-to-Point Encryption (MPPE), що реалізує RSA RC4 з максимум 128-бітовими сеансовими ключами	TCP-порт 1723	Має серйозні вразливості. MSCHAP-v2 вразливий для атаки за словником, алгоритм RC4 піддається атаці Bit-flipping.
SSTP	Microsoft	Proprietary	Windows. Працює "з коробки", не вимагаючи встановлення додаткового ПЗ	SSL (шифруються всі частини, крім TCP- та SSL-заголовків)	TCP-порт 443	Серйозних недоліків безпеки не було виявлено.
L2TP/IPsec	12TP - спільна розробка Cisco та Microsoft	Proprietary	Windows, Mac OS X, Linux, IOS. Android. Багато ОС (включно з Windows 2000/XP + Mac OS 10.3+) мають вбудовану підтримку, немає необхідності в додатковому ПЗ	3DES або AES	UDP-порт 500 для початкового обміну ключами та UDP-порт 1701 для початкової конфігурації L2TP, UDP-порт 5500 для обходу NAT	Не вдалося знайти інформації про наявні недоліки безпеки, крім інциденту з витоком доповідей АНБ щодо IPsec.
OpenVPN	OpenVPN Technologies	GNU GPL	Windows, macOS, GNU/Linux Apple IOS, Android та маршрутизатори. Необхідна установка спеціалізованого ПЗ, що підтримує роботу з цим протоколом	Використовує бібліотеку OpenSS (реалізує більшість популярних криптографічних стандартів)	Будь-який UDP- або TCP-порт	Серйозних недоліків безпеки не було виявлено.
WireGuard	Jason A. Donenfeld	GNU GPL	Windows, macOS, GNU/Linux Apple IOS, Android. Встановити сам WireGuard, потім налаштувати згідно інструкцій	Обмін ключами по 1-RTT, Curve25519 для ECDH RFC7539 для ChaCha20 і Poly1305 для автентифікаційного шифрування, і BLAKE2s для хешування	Будь-який UDP-порт	Серйозних недоліків безпеки не було виявлено.

2.4 Порівняльний аналіз криптографічних протоколів захисту інформації в мережі Інтернет та їх версій

2.4.1 Порівняльний аналіз криптографічних протоколів захисту інформації в мережі Інтернет

Кожен протокол має свої унікальні характеристики та властивості, які дозволяють забезпечити безпеку та конфіденційність даних під час їх передачі по мережі Інтернет. Ключовою метою криптографічних протоколів є забезпечення конфіденційності, цілісності та аутентичності даних, а також забезпечення безпечних каналів комунікації та аутентифікації користувачів. Кожен протокол має свою власну комбінацію криптографічних алгоритмів, ключів, протоколів обміну ключами та механізмів шифрування, які впливають на його ефективність, стійкість, ресурсомісткість та зручність використання.

У цьому розділі ми розглянемо декілька ключових криптографічних протоколів, таких як SSL/TLS, IPsec, SSH, OpenPGP, S/MIME, Kerberos, IKEv2 та PGP. Кожен з цих протоколів має власні особливості та застосування, які варто врахувати при виборі протоколу для конкретного сценарію.

Проаналізувавши порівняльну таблицю 2.3, побачимо такі речі:

– SSL/TLS є широко використовуваним протоколом, який має високу ефективність шифрування та стійкість. Він забезпечує зручність використання та управління, а також підтримує різні типи даних та пристроїв. Однак, ступінь анонімності в SSL/TLS є низьким.

– IPsec, з своєю високою стійкістю та ресурсомісткістю, є протоколом, який широко використовується для захисту IP-трафіку. Він забезпечує високий рівень підтримки перевірки цілісності даних, але використання та конфігурація IPsec можуть бути складними.

– SSH є протоколом, спеціально розробленим для безпечного з'єднання та передачі даних по мережі. Він має високу ефективність шифрування та стійкість, але середню зручність використання. SSH підтримує різні алгоритми шифрування та аутентифікації.

– OpenPGP та S/MIME є протоколами для захисту електронної пошти.

Вони обидва мають високу ефективність шифрування та стійкість, але середню зручність використання. Вони підтримують різні розміри ключів та типи даних.

– Kerberos є протоколом для аутентифікації та авторизації в розподілених системах. Він має середню ефективність шифрування, але високу стійкість та ресурсомісткість. Протокол Kerberos забезпечує високий рівень ступеня анонімності, але його використання може бути складним через обмежену підтримку.

– IKEv2 є протоколом для безпечної обміну ключами та встановлення віртуальних приватних мереж (VPN). Він має високу ефективність шифрування, стійкість та ресурсомісткість. IKEv2 підтримує різні алгоритми шифрування та аутентифікації.

Таблиця 2.3 - Порівняння властивостей криптографічних протоколів в мережі Інтернет

Протокол	**Ефективність шифрування**	**Стійкість**	**Ресурсомісткість**	**Зручність використання та управління**	**Підтримка перевірки цілісності даних**	**Ступінь анонімності**	**Протоколи в основі**
SSL/TLS	Висока	Висока	Середня	Зручна	Так	Низький	RSA, AES, ECC
IPsec	Висока	Висока	Висока	Складна	Так	Високий	DES, 3DES, AES, SHA
SSH	Висока	Висока	Середня	Зручна	Так	Середній	RSA, DSA, AES
OpenPGP	Висока	Висока	Середня	Зручна	Так	Середній	RSA, DSA, AES
S/MIME	Висока	Висока	Середня	Зручна	Так	Середній	RSA, AES
Kerberos	Середня	Висока	Висока	Складна	Так	Високий	DES, RC4, AES
IKEv2	Висока	Висока	Висока	Складна	Так	Високий	AES, SHA
PGP	Висока	Висока	Середня	Зручна	Так	Середній	RSA, DSA, AES

Нижче наведено таблицю 2.4, яка допомагає порівняти основні властивості криптографічних протоколів, що використовуються в мережі Інтернет, окрім того, вона вказує, які протоколи забезпечують конфіденційність, цілісність та аутентичність даних, підтримують безпечні канали комунікації та аутентифікацію користувачів.

Таблиця 2.4 – Порівняння ключових властивостей криптографічних протоколів в мережі Інтернет

Протокол	**Забезпечення конфіденційності**	**Забезпечення цілісності**	**Забезпечення аутентичності**	**Забезпечення безпечних каналів**	**Аутентифікація користувачів**
SSL/TLS	+	+	+	+	+
IPsec	+	+	+	+	+
SSH	+	+	+	+	+
OpenPGP	+	+	+	-	-
S/MIME	+	+	+	-	-
Kerberos	-	-	+	-	+
IKEv2	+	+	+	+	+
PGP	+	+	+	-	-

З огляду таблиці можна зробити такі висновки:

– Протокол SSL/TLS надає високий рівень конфіденційності, цілісності та аутентичності даних. Він забезпечує шифрування передачі даних, перевірку цілісності повідомлень та аутентифікацію сервера і клієнта. Також протокол SSL/TLS підтримує безпечні канали комунікації і має механізми аутентифікації користувачів.

– IPsec є набором протоколів, які забезпечують конфіденційність, цілісність та аутентичність даних на рівні мережевого протоколу. Він

використовує шифрування та аутентифікацію на рівні IP-пакетів для захисту передачі даних. IPsec також може забезпечувати безпечні канали комунікації та аутентифікацію користувачів.

– SSH (Secure Shell) є протоколом для захищеного віддаленого доступу до системи. Він забезпечує конфіденційність, цілісність та аутентичність даних, шифрування комунікації та механізми аутентифікації користувачів. SSH також може створювати безпечні тунелі для захищеної передачі даних.

– Протокол OpenPGP використовується для шифрування та підпису електронної пошти і файлів. Він надає конфіденційність, цілісність та аутентичність даних за допомогою симетричного та асиметричного шифрування. Однак, OpenPGP не підтримує безпечні канали комунікації і аутентифікацію користувачів.

– S/MIME (Secure/Multipurpose Internet Mail Extensions) є протоколом для захищеного обміну електронною поштою. Він надає конфіденційність, цілісність та аутентичність даних через шифрування та підписи. Але, подібно до OpenPGP, S/MIME також не підтримує безпечні канали комунікації та аутентифікацію користувачів.

– Протокол Kerberos забезпечує механізми аутентифікації користувачів в розподілених обчислювальних середовищах. Він підтримує аутентичність даних, але не забезпечує конфіденційність або цілісність даних. Kerberos використовується для централізованого керування аутентифікацією та авторизацією в мережевих середовищах.

– IKEv2 (Internet Key Exchange version 2) є протоколом для встановлення безпечних з'єднань VPN. Він забезпечує конфіденційність, цілісність та аутентичність даних, а також безпечні канали комунікації і механізми аутентифікації користувачів.

– Pretty Good Privacy (PGP) є протоколом для шифрування, підпису та захисту електронної пошти та файлів. PGP забезпечує конфіденційність, цілісність та аутентичність даних, але не підтримує безпечні канали комунікації і аутентифікацію користувачів.

2.4.2 Порівняльний аналіз версій криптографічних протоколів захисту інформації в мережі Інтернет

При аналізі криптографічних протоколів, які широко використовуються в Інтернеті, важливо звертати увагу на їх ключові властивості, що включають здатність забезпечити конфіденційність, цілісність, автентичність та доступність даних. Протоколи, як-то SSL/TLS, SSH, IPsec, забезпечують різні рівні безпеки в залежності від використовуваних алгоритмів шифрування, методів автентифікації та механізмів обміну ключами. Проте, переходячи до детального розгляду конкретного протоколу, важливо проаналізувати його історію розвитку, особливості різних версій та виявлені вразливості. Наводжу таблиці 2.5 та 2.6, в яких для прикладу розглянемо протокол TLS, який є одним із найбільш поширених криптографічних протоколів в Інтернеті, і зосередимося на його версіях: TLS 1.0, 1.1, 1.2 і 1.3, а також на версіях його попередника SSL 1.0, 2.0, 3.0.

Таблиця 2.5 – Порівняння версій SSL 1.0, 2.0 та 3.0

Версія	Рік стандартизації	Оновлення / поліпшення	Виявлені проблеми безпеки та вразливості	Інші характеристики
SSL 1.0	1995	Перша версія протоколу, яка була розроблена компанією Netscape. Ніколи не була опублікована через виявлені в ній серйозні вразливості	Містить багато вразливостей, через що ніколи не була офіційно випущена	Ця версія ніколи не була загальнодоступна
SSL 2.0	1995	Усунення деяких вразливостей, виявлених у SSL 1.0. Підтримка механізмів аутентифікації та захисту передачі даних	Містить вразливості до атак на повторення сесій та проблеми з автентифікацією сервера	Зараз вважається застарілим і не безпечним
SSL 3.0	1996	Більш сучасні алгоритми шифрування, виправлення вразливостей SSL 2.0.	Став жертвою атаки POODLE (Padding Oracle On Downgraded Legacy Encryption), яка дозволила зловмисникам розшифровувати зашифровану інформацію	Зараз вважається застарілим і не безпечним, замінений на TLS 1.0

Як бачимо, SSL 1.0 не був ніколи офіційно випущений через серйозні вразливості, SSL 2.0 і SSL 3.0 з часом стали менш безпечними через відкриття нових вразливостей, а також через появу більш ефективних методів атак. SSL 3.0 став жертвою атаки POODLE, яка показала серйозні недоліки в цій версії протоколу. З цієї причини ці версії зараз вважаються застарілими і не безпечними.

Таблиця 2.6 – Порівняння версій TLS 1.0, 1.1, 1.2 та 1.3

Версія	Рік стандартизації	Оновлення / поліпшення	Виявлені проблеми безпеки та вразливості	Інші характеристики
TLS 1.0	1999	Покращений варіант SSLv3 з підтримкою більш сучасних алгоритмів шифрування та механізмом шифрування	Вразливості до атак типу man-in-the-middle (POODLE), а також проблеми з блоковим шифруванням (BEAST)	Часто вважається застарілим та не безпечним
TLS 1.1	2006	Захист від атаки CBC replay, яка використовувала IV; підтримка шифрування, заснованого на AEAD	Багато вразливостей, що могли дозволити атаки посередника (man-in-the-middle)	Вже вважається застарілим та не безпечним
TLS 1.2	2008	Підтримка більш сучасних алгоритмів шифрування; покращена гнучкість при виборі алгоритмів	Чутливість до атак на Padding Oracle (POODLE) та атаки на виробничий процес рандомізації (Lucky 13)	Надає більш безпечний та гнучкий вибір шифрування
TLS 1.3	2018	Скорочення часу рукостискання; підтримка Forward Secrecy; усунення старих і небезпечних алгоритмів шифрування	Немає широко відомих вразливостей на момент написання	Набагато швидше рукостискання, ніж у попередніх версіях; зміцнена безпека

Важливим моментом є те, що перехід від SSL до TLS не був випадковим, а був обумовлений необхідністю підвищити безпеку передачі даних в мережі. TLS 1.0, TLS 1.1 і TLS 1.2 з часом теж стали менш безпечними через відкриття нових вразливостей, а також через появу більш ефективних методів атак. TLS 1.3

вважається значно більш безпечним за своїх попередників, та є останньою версією протоколу, допоки не вийшов реліз версії TLS 1.4.

Оскільки ми зосередилися на важливості оновлення протоколів для забезпечення безпеки, ми можемо застосувати цей же принцип до іншого важливого протоколу - IPsec. Як і в TLS, зміни і оновлення в різних версіях IPsec відображають розвиток стандартів безпеки та відповідь на виявлені вразливості. В таблиці 2.7 розглянемо та порівняємо різні версії ключового компонента IPsec - протоколу обміну ключами Internet (IKE), включаючи IKEv1 та IKEv2. IKE (Internet Key Exchange) використовується у IPsec для обміну ключами та управління безпекою.

Таблиця 2.7 – Порівняння версій компонента IPsec - IKE

Версія	**Рік стандартизації**	**Оновлення / поліпшення**	**Виявлені проблеми безпеки та вразливості**	**Інші характеристики**
IKEv1	1998	Перший стандарт для IPsec key management	Aggressive Mode, numerous RFC complaints	Використовується в багатьох старіших системах, не рекомендується для нових розробок через вразливості
IKEv2	2005	Більш стабільний і безпечний, зменшена складність	Fewer known vulnerabilities	Використовується в сучасних системах заради його кращої безпеки і ефективності, більші можливості розширення

Кожна нова версія протоколу зазвичай вносить поліпшення у сферу безпеки, виправляє виявлені вразливості та адаптується до сучасних умов комунікації в Інтернеті. Неправильний вибір версії протоколу може призвести до серйозних наслідків, зокрема до витоку конфіденційної інформації, до злому системи, або навіть до повної втрати контролю над мережевими ресурсами. Тому, для забезпечення максимального рівня безпеки, важливо завжди використовувати найновішу та найбільш надійну версію протоколу.

РОЗДІЛ 3 ДОСЛІДЖЕННЯ БЕЗПЕКИ ВЕБСЕРВЕРА ДЛЯ РІЗНИХ ВЕРСІЙ ПРОТОКОЛІВ SSL ТА TLS

3.1 Оцінка стану безпеки вебсервера

3.1.1 Опис об’єкту дослідження

Об'єктом нашого дослідження є вебсервер, що працює на базі операційної системи FreeBSD.

FreeBSD — це розповсюджена відкрита операційна система, відома своєю відмінною підтримкою мережевих сервісів, надійністю та стабільністю, що базується на Unix-подібній системі, що вийшла з проекту Berkeley Software Distribution (BSD).

FreeBSD відзначається високою продуктивністю та надійністю, що робить цю операційну систему доречною для масштабних серверних рішень, надійних у розподілених мережах. Особливість цієї системи полягає в гнучкості при налаштуванні. Адміністратори можуть налаштовувати майже кожний аспект системи, щоб відповідати конкретним вимогам і потребам їхнього середовища. Це включає налаштування мережі, управління системними ресурсами, налаштування служб та застосунків, налаштування безпеки і багато іншого. Також FreeBSD має розширений набір інструментів безпеки. Вони включають вбудовані механізми для ізоляції процесів, мережевого фільтрування, аудиту та шифрування. З цими інструментами адміністратори можуть створити міцні бар'єри для зловмисників, мінімізуючи ризик вторгнень та атак.

FreeBSD також відрізняється своєю здатністю до масштабування. Система може бути налаштована для підтримки великого навантаження, використовуючи різноманітні механізми для розподілення навантаження та балансування навантаження. Це означає, що вона може ефективно обробляти високу кількість запитів від клієнтів, забезпечуючи стабільність та надійність обслуговування.

Окрім цього, FreeBSD має розширену підтримку вебсерверних технологій. Вона підтримує велику кількість вебсерверів, включаючи Apache, Nginx, Lighttpd та інші. Також, система дозволяє встановлювати різноманітні модулі та додаткові

компоненти, що розширюють можливості вебсервера і забезпечують більш гнучкі та потужні функціональні можливості. У комбінації з активною спільнотою розробників і регулярними оновленнями безпеки, FreeBSD забезпечує високий рівень стабільності і безпеки. Розробники активно працюють над виявленням і виправленням потенційних вразливостей, а оновлення безпеки регулярно випускаються для забезпечення захисту від нових загроз. Такий підхід гарантує, що FreeBSD залишається надійною та безпечною операційною системою для вебсерверів.

Вебсервер, над яким ведеться дослідження, працює на програмному забезпеченні Apache. Apache є одним з найпопулярніших і широко використовуваних вебсерверів у світі. Він славиться своєю надійністю, гнучкістю та розширюваністю, що робить його відмінним вибором для багатьох веб-проектів. Apache заснований на відкритому програмному забезпеченні, що означає, що весь його вихідний код відкритий для громадськості і доступний для змін. Це сприяє активному розвитку та покращенню сервера завдяки внескам різних розробників зі всього світу.

Одна з головних переваг Apache полягає у його підтримці великої кількості модулів, які можна використовувати для різних цілей, включаючи модулі для шифрування даних. Завдяки цим модулям, Apache може забезпечити ефективний захист інформації, що передається між вебсервером та клієнтами. Наприклад, модуль для шифрування даних дозволяє використовувати протокол SSL/TLS для захищеної передачі даних через Інтернет. Це особливо важливо для захисту конфіденційної інформації, такої як паролі, особисті дані користувачів або фінансова інформація, від несанкціонованого доступу.

Apache сервер, який працює на операційній системі FreeBSD, використовує стандартні порти для з'єднання. Він використовує порт 80 для протоколу HTTP і порт 443 для протоколу HTTPS. HTTP є основним протоколом передачі даних в Інтернеті, тоді як HTTPS є захищеним варіантом HTTP з використанням SSL/TLS для шифрування.

Використання стандартних портів дозволяє забезпечити надійний захист інформації, яка передається через мережу Інтернет. Наприклад, шифрування

SSL/TLS, що використовується в протоколі HTTPS на порті 443, дозволяє захистити дані від прослуховування та незаконного доступу. Це важливо для підтримки конфіденційності та цілісності інформації, переданої між вебсервером та клієнтами, і забезпечує безпечну взаємодію з веб-додатками.

3.1.2 Оцінка поточного стану безпеки вебсервера та аналіз виявлених вразливостей

В рамках аналізу поточного стану безпеки вебсервера було використано сканер вразливостей Nessus з метою виявлення можливих проблем та оцінки ризиків безпеки. На основі цього інструменту було проведено сканування сервера з метою виявлення потенційних вразливостей, включаючи проблеми з програмним забезпеченням, слабкі паролі, недостатні налаштування безпеки та інші аспекти, які можуть створювати загрози для безпеки системи. Nessus є потужним інструментом, спеціально розробленим для виявлення таких проблем, і його використання дало змогу отримати детальну інформацію про потенційні вразливості сервера та визначити рівень ризику.

Після налаштування параметрів сканування, запускаємо Nessus для аналізу вебсервера. Інструмент сканував різні порти та сервіси, що працюють на сервері, та перевіри їх на наявність відомих вразливостей. Також Nessus провів сканування мережевих протоколів.

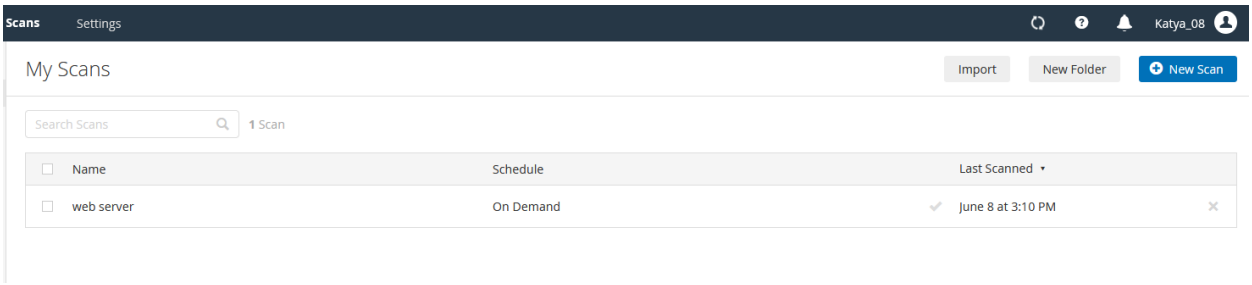

Рисунок 3.1 – Вікно My Scans

Після завершення сканування Nessus генерується звіт, який містить результати аналізу (див. рисунок 3.2). Цей звіт включає інформацію про виявлені вразливості, їх рівень серйозності та рекомендації щодо виправлення проблем.

Звіт також містить додаткову інформацію про кожну вразливість, включаючи опис, можливі наслідки та рекомендовані заходи безпеки.

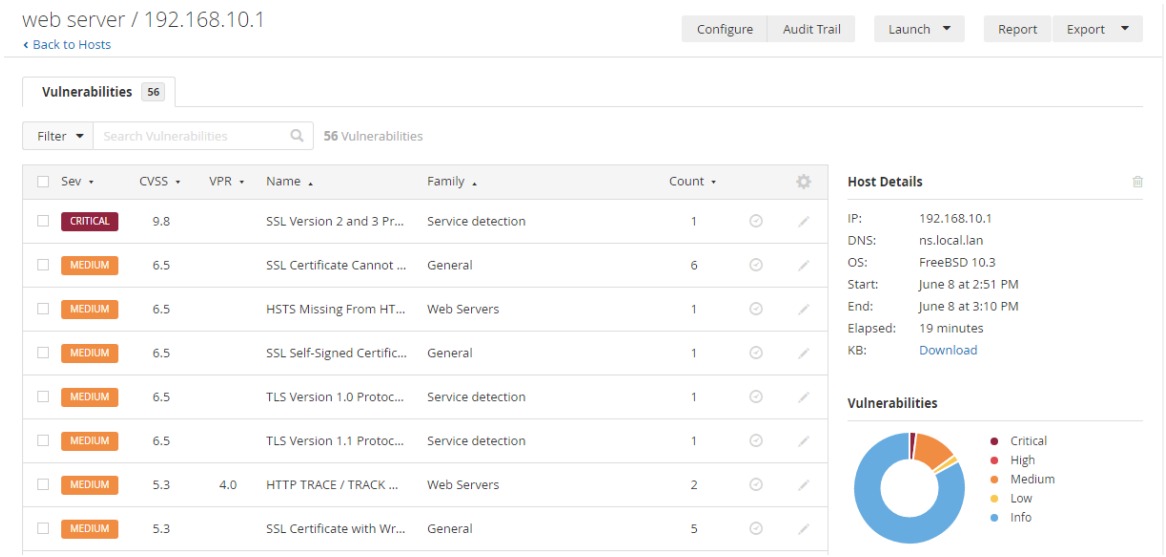

Рисунок 3.2 – Результат Nessus сканування

З огляду результатів сканування стає зрозуміло, що сканер виявив критичну вразливість безпеки, пов'язану з використанням небезпечних протоколів на вебсервері, зокрема ssl2 і ssl3, а саме «SSL version 2 and 3 Protocol Detection». Ці протоколи мають відомі проблеми безпеки, які можуть бути використані зловмисниками для несанкціонованого доступу до системи або виконання шкідливих дій. Крім того, виявлено вразливості середнього рівня небезпеки, пов'язані з використанням протоколів TLS 1.0 та TLS 1.1, які також відомі своїми проблемами безпеки, через що несуть ризик для захищеності нашої системи.

Враховуючи серйозність цих вразливостей, необхідно прийняти негайні заходи для усунення проблем та підвищення рівня безпеки, а саме відключити використання протоколів TLS 1.0, TLS 1.1 та всіх версій протоколу SSL, підключивши більш безпечні версій, такі як TLS 1.2 або TLS 1.3, залежно від можливостей підтримки сервером. Це допоможе запобігти можливим атакам та забезпечить високий рівень безпеки системи.

Проте, перед тим як приступати до виправлення вразливостей важливо розуміти, що знайдені та оцінені вразливості не обов'язково потрібно виправляти одразу після виявлення. Необхідно провести пріоритезацію, за допомогою якої

можна ефективно розподілити ресурси та час на виправлення вразливостей згідно їх важливості та впливу на систему.

Отже, після виявлення та оцінки вразливостей, наступним кроком є встановлення пріоритетів щодо їх виправлення. Оцінка критичності вразливості, яку надає сканер, може бути чудовим стартовим пунктом для цього процесу. Виходячи з цього, можна визначити, які вразливості потребують негайного виправлення.

В нашому випадку є одна вразливість, яка оцінена як критична - SSL version 2 and 3 Protocol Detection. Це означає, що вона має високий рівень ризику і може мати серйозний вплив на систему, якщо буде використана зловмисниками. Тому таку вразливість слід виправити негайно.

Щодо вразливостей середнього рівня, а в нас це SSL Certificate Cannot Be Trusted, SSL Self-Signed Certificate, TLS Version 1.0 Protocol Detection, TLS Version 1.1 Protocol Detection та інші, то їх також необхідно виправити, але вони мають менший пріоритет порівняно з критичною вразливістю і не є невідкладними. Тому вони будуть виправлені після виправлення критичної вразливості, особливо якщо ресурси обмежені.

Процес пріоритизації вимагає ретельного планування і балансування між рівнем ризику, який створює кожна вразливість, і ресурсами, які доступні для їх виправлення. Пріоритизація допомагає зосередитись на найбільш важливих задачах, забезпечуючи, що найбільш серйозні ризики будуть усунені в першу чергу.

3.2 Аналіз безпеки вебсервера для різних версій протоколів SSL та TLS

3.2.1 Структура конфігураційного файлу

Після оцінки поточного стану захищеності системи шляхом сканування системи, було виявлено загрози, які ми проаналізувати та пріоритезували за їх важливість і потенційним впливом на систему. Наступним кроком буде розробка стратегій і планів для виправлення вразливостей.

Стратегія полягає у зміні конфігурації файлу системи httpd-ssl.conf. У операційній системі FreeBSD, конфігураційний файл httpd-ssl.conf використовується для налаштування параметрів безпеки, пов'язаних з SSL і TLS, для вебсервера Apache HTTP. Файл знаходиться за шляхом /usr/local/etc/apache24/extra/httpd-ssl.conf, але розташування може відрізнятись в залежності від версії FreeBSD або Apache.

Цей конфігураційний файл містить налаштування для модуля SSL/TLS на Apache HTTP сервері. В лістингу 1.1 подано фрагмент вмісту конфігураційного файлу httpd-ssl.conf, ака містить ключові налаштування директив пов'язаних з SSL і TLS.

Лістинг 1.1 - Фрагмент вмісту конфігураційного файлу httpd-ssl.conf з ключовими налаштуваннями

```
Listen 443
SSLCipherSuite HIGH:MEDIUM:!MD5:!RC4:!3DES
SSLProxyCipherSuite HIGH:MEDIUM:!MD5:!RC4:!3DES
SSLHonorCipherOrder on
SSLProtocol all -SSLv3 -SSLv2 +TLSv1 +TLSv1.1 +TLSv1.2 +TLSv1.3
SSLProxyProtocol all -SSlv3 -SSLv2 -TLSv1.2 -TLSv1.3
SSLPassPhraseDialog builtin
SSLSessionCache "shmcb:/var/run/ssl_scache(512000)"
SSLSessionCacheTimeout 300
<VirtualHost _default_:443>
DocumentRoot "/usr/local/www/apache24/data"
ServerName www.local.lan:443
ServerAdmin admin@local.lan
ErrorLog "/var/log/httpd-error.log"
TransferLog "/var/log/httpd-access.log"
SSLEngine on
SSLCertificateFile "/usr/local/etc/apache24/server.crt"
SSLCertificateKeyFile "/usr/local/etc/apache24/server.key"
<FilesMatch "\.(cgi|shtml|php)$">
SSLOptions +StdEnvVars
</FilesMatch>
<Directory "/usr/local/etc/apache24/cgi-bin">
    SSLOptions +StdEnvVars
</Directory>
BrowserMatch "MSIE [2-5]"\
    nokeepalive ssl-unclean-shutdown\
    downgrade-1.0 force-response-1.0
CustomLog "/var/log/httpd-ssl_request.log"\
    "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"
</VirtualHost>
```

Цей конфігураційний файл містить налаштування для модуля SSL/TLS на нашому Apache HTTP сервері. Ось детальний опис ключових директив вашого конфігураційного файлу:

– Listen 443: Apache вебсервер за замовчуванням слухає порт 80 для HTTP з'єднань. Ця директива вказує серверу слухати порт 443 для HTTPS з'єднань.

– SSLCipherSuite і SSLProxyCipherSuite: Вказують на шифри, які будуть використовуватися під час домовленостей SSL/TLS. В даному випадку, сервер використовує лише шифри високого та середнього рівня, виключаючи MD5, RC4 та 3DES шифри.

– SSLHonorCipherOrder on: Ця директива вказує, що сервер повинен використовувати шифри в порядку перелічених у SSLCipherSuite, починаючи з найбільш сильного.

– SSLProtocol і SSLProxyProtocol: Визначають протоколи, які сервер повинен приймати під час домовленостей SSL/TLS. В даному випадку, усі протоколи крім SSLv3, SSLv2 доступні, і TLSv1.2, TLSv1.3 виключені для проксі-протоколу.

– SSLPassPhraseDialog builtin: Ця директива контролює, як Apache буде запитувати парольну фразу для зашифрованих приватних ключів SSL/TLS.

– SSLSessionCache "shmcb:/var/run/ssl_scache(512000)" і SSLSessionCacheTimeout 300: Ці директиви контролюють, як сеанси SSL будуть кешуватися, що може поліпшити продуктивність SSL.

В середині <VirtualHost _default_:443> блоку, ви знайдете директиви, специфічні для віртуального хоста:

– DocumentRoot "/usr/local/www/apache24/data": Це директива вказує на каталог, в якому знаходяться файли сайту.

– ServerName www.local.lan:443: Ця директива встановлює ім'я сервера, яке використовується для визначення цього віртуального хоста на мережі.

– SSLEngine on: Ця директива включає SSL для цього віртуального хоста.

– SSLCertificateFile та SSLCertificateKeyFile: Ці директиви вказують на розташування файлів сертифікатів, які використовуються для автентифікації сервера.

– BrowserMatch "MSIE [2-5]" nokeepalive ssl-unclean-shutdown downgrade-1.0 force-response-1.0: Ця директива містить налаштування для старих версій браузера Internet Explorer (версії 2-5), що забезпечують зворотну сумісність.

– CustomLog "/var/log/httpd-ssl_request.log" "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x "%r" %b": Ця директива встановлює журнал, який зберігає детальну інформацію про кожен SSL запит, отриманий сервером.

Ці та інші директиви, що містяться в цьому файлі, дозволяють нам детально налаштовувати параметри SSL/TLS, щоб максимально використовувати можливості безпеки нашого сервера. В наступному пункті розділу ми проведемо оптимізацію нашого вебсервера, вносячи зміни до директив SSL/TLS, які було розглянуто в цьому пункті розділу.

3.2.2 Усунення виявлених критичних вразливостей вебсервера

Оскільки було проведено всі необхідні підготовчі етати, що передують виправленню вразливостей: оцінку поточного стану безпеки, аналіз виявлених вразливостей системи, визначено пріритетність та стратегію для їх виправлення, можемо безпосередньо переходити до виправлення вразливостей, дотримуючись визначеного пріоритету. Отож розпочнемо з виправлення вразливості SSL version 2 and 3 Protocol Detection. Для цього потрібно внести зміни в конфігураційний файл системи, що містить налаштування для модуля SSL/TLS на Apache HTTP сервері, а саме httpd-ssl.conf. Для цього через вікно терміналу, в режимі root користувача, за допомогою команди mc переходимо консольний файловий менеджер Midnight Commander (див. рисунок 3.3).

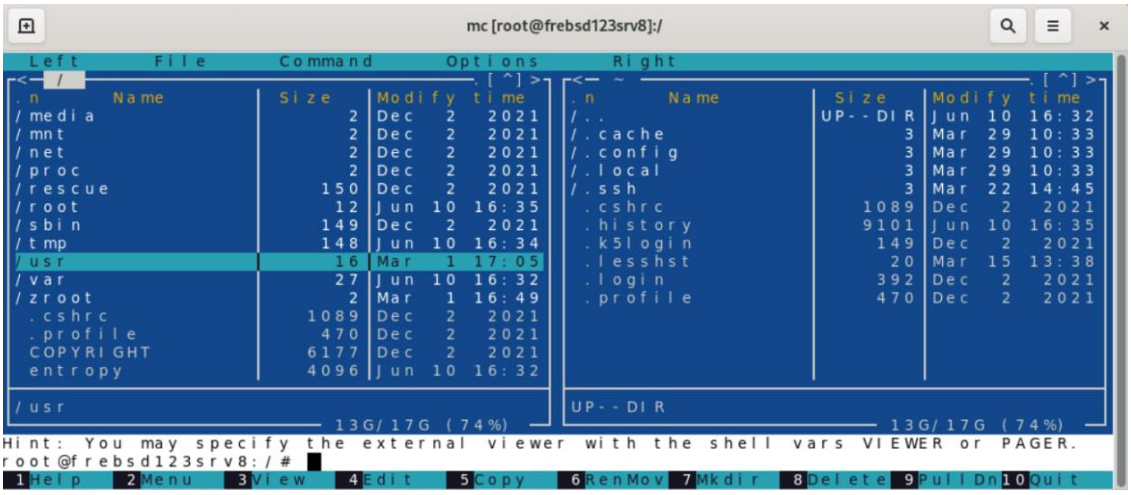

Рисунок 3.3 – Кореневий каталог файлового менеджера Midnight Commander

Знаходимо та відкриваємо в режимі редагування файл httpd-ssl.conf, що знаходиться за шляхом /usr/local/apache24/extra/httpd-ssl.conf. На рисунку 3.3. частково відображено вміст файлу httpd-ssl.conf. В рядку «SSLProtocol all +SSLv3 +SSlv2 +TLSv1 +TLSv1.1» бачимо, що в нашій системі підключені такі криптографічні протоколи як SSL 3.0, SSL 2.0, TLS 1.0 та TLS 1.1 (дивю рисунок 3.4), а це саме ті вразливі протоколи, що були виявлені під час сканування системи.

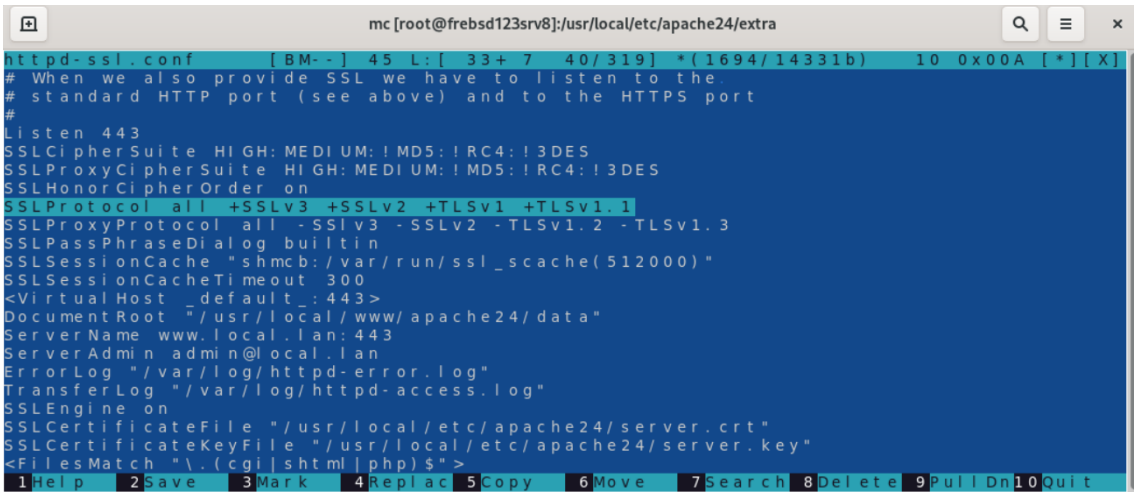

Рисунок 3.4 – Початкові налаштування сертифікатів системи в httpd-ssl.conf

Згідно пріоритизації, спочатку відключаємо протоколи SSL 3.0 та SSL 2.0, адже вони несуть критичну небезпеку для нашою системи. Щоб зробити це, заміняємо рядок конфігураційно файлу з «SSLProtocol all +SSLv3 +SSlv2 +TLSv1

+TLSv1.1» на «SSLProtocol all -SSLv3 -SSlv2 +TLSv1 +TLSv1.1» та зберігаємо зміни (див. рисунок 3.5).

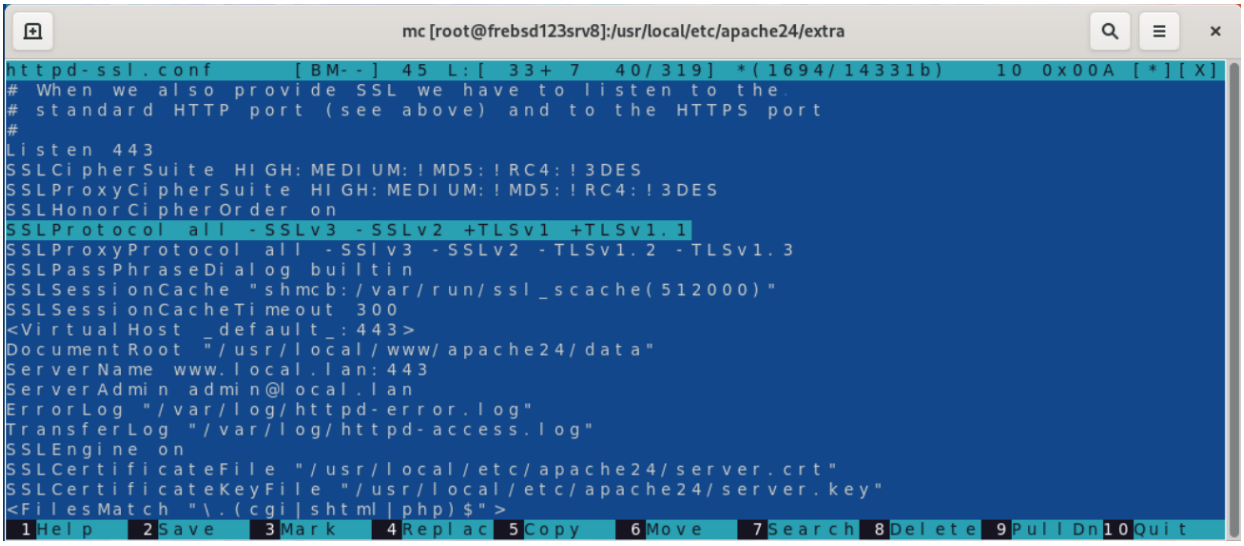

Рисунок 3.5 – Вміст конфігураційного файлу після внесення змін

Після цього в терміналі необхідно виконати команду #service apache24 restart під правами root-користувача для перезапуску вебсервера Apache версії 2.4. Команда є корисною, якщо були внесені зміни в конфігураційні файли або модулі Apache і потрібно, щоб щоб ці зміни набрали чинності без повного зупинення та запуску сервера. З рисунку 3.6 бачимо результат виконання команди, який показує що було провелено перевірку конфігураційного файлу Apache, зупинено процеси Apache, після зупинки процесів проведено перевірку синтаксису конфігурації, визначено, що файл містить правильний синтаксис, а потім запущено Apache з оновленою конфігурацією.

```
root@frebsd123srv8:/usr/local/etc/apache24/extra# service apache24 restart
Performing sanity check on apache24 configuration:
Syntax OK
Stopping apache24.
Waiting for PIDS: 1002.
Performing sanity check on apache24 configuration:
Syntax OK
Starting apache24.
root@frebsd123srv8:/usr/local/etc/apache24/extra#
```

Рисунок 3.6 – Перезапуск сервера Apache

Після перезапуску вебсервера Apache з оновленою конфігурацію, проведемо повторне сканування нашої системи, для підтвердження того, що вразливість було виправлено.

По завершенню повторного сканування, на рисунку 3.7, бачимо, що вразливість SSL version 2 and 3 Protocol Detection виявлено не було, отже ми успішно її виправили. Успішне виправлення вразливості "SSL version 2 and 3 Protocol Detection" демонструє, що ми можемо ефективно управляти цими вразливостями і забезпечити надійну роботу нашого сервера.

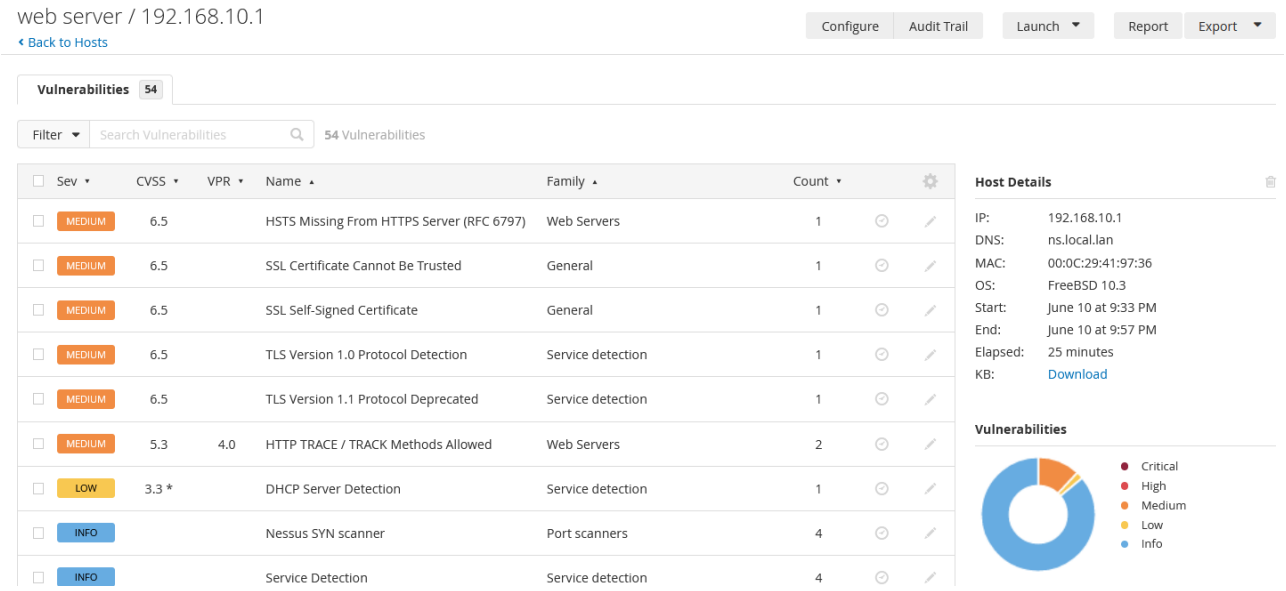

Рисунок 3.7 – Результат повторного сканування

Проте роботу ще не завершено. Незважаючи на те, що ми вже виправили найбільш критичні вразливості, в нашій системі все ще існують середні вразливості. Хоча вони можуть не представляти такої серйозної загрози, як критичні вразливості, вони все ще можуть використовуватись зловмисниками для отримання несанкціонованого доступу або завдання шкоди нашій системі.

3.2.3 Усунення виявлених середніх вразливостей вебсервера

Після успішного виправлення критичних вразливостей у нашій системі, настав час приділити увагу вразливостям середнього рівня. Хоча ці вразливості можуть бути менш серйозними ніж критичні, їх не потрібно залишати без уваги. Середні вразливості можуть містити широкий спектр потенційних проблем, від слабких місць в налаштуваннях та конфігурації до проблем з сумісністю та застарілими версіями програмного забезпечення.

Для виправлення вразливостей TLS Version 1.0 Protocol Detection та TLS Version 1.1 Protocol Deprecated використовуємо ту ж стратегію, що й для виправлення вразливості SSL version 2 and 3 Protocol Detection. Тому вносимо відповідні зміни до того ж конфігураційного файлу /httpd-ssl.conf, а саме замінимо «SSLProtocol all -SSLv3 -SSlv2 +TLSv1 +TLSv1.1» на «SSLProtocol all -SSLv3 -SSlv2 -TLSv1 -TLSv1.1 +TLSv1.2 +TLSv1.3» та зберігаємо зміни (див. рисунок 3.8). Ці зміни мають вплинути на SSL/TLS таким чином:

– "-TLSv1 -TLSv1.1" відключає протоколи TLS 1.0 та TLS 1.1.

– "+TLSv1.2 +TLSv1.3" додає протоколи TLS 1.2 та TLS 1.3.

Після здійснених змін, вебсервер Apache буде підтримувати тільки протоколи TLS версій 1.2 та 1.3, а протоколи TLS 1.0 та 1.1 будуть відключені, це має виправити такі вразливості, як TLS version 1.0 Protocol Detection та TLS version 1.0 Protocol Deprecated.

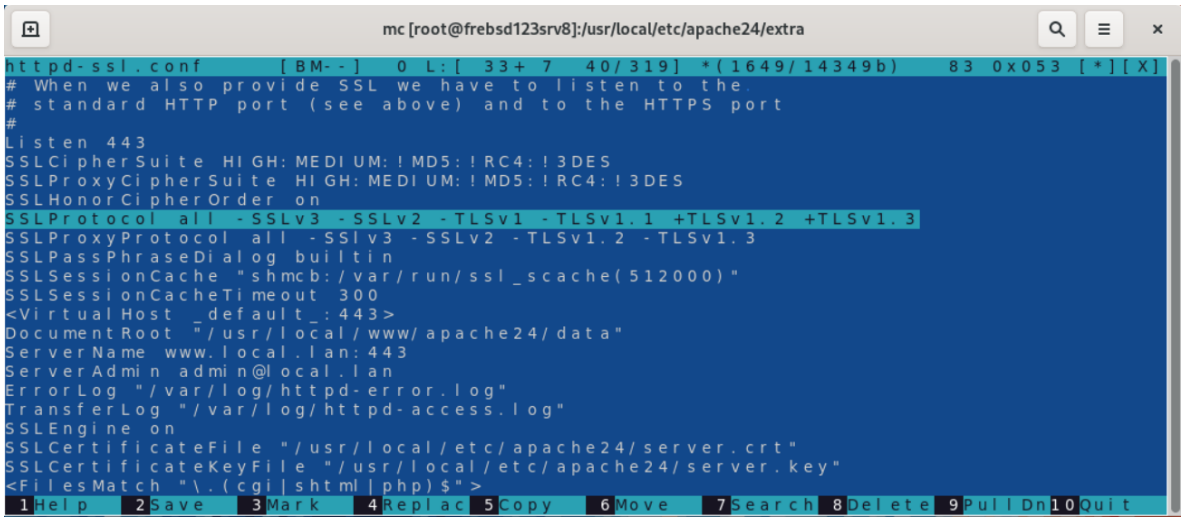

Рисунок 3.8 – Безпечні налаштування версій протоколів вебсерверу Apache

Після внесення змін до конфігураційного файлу, ідентично попередньому випадку виконуємо команду #service apache24 restart для перезапуску вебсервера Apache. Як бачимо з рисунку 3.9 у нас знову все вдалось – перевірка синтаксису конфігурації пройшла успішно і зміни набули чинності.

```
root@freebsd123srv8:/usr/local/etc/apache24/extra# service apache24 restart
Performing sanity check on apache24 configuration:
Syntax OK
Stopping apache24.
Waiting for PIDS: 1041.
Performing sanity check on apache24 configuration:
Syntax OK
Starting apache24.
root@freebsd123srv8:/usr/local/etc/apache24/extra#
```

Рисунок 3.9 – Повторний перезапуск серверу Apache

Тепер, після того як повторно внесено зміни в конфігураційний файл для виправлення, цього разу, середніх вразливостей, знову проведемо Nessus сканування, що мати підтвердження, того, що вразливості дійсно виправлені.

На рисунку 3.10 бачимо, що вразливості TLS version 1.0 Protocol Detection та TLS version 1.0 Protocol Deprecated під час сканування не виявлено, отже завдяки внесеним змінам до конфігураційного файлу Apache, а саме httpd-ssl.conf, було поліпшено безпеку нашого сервера.

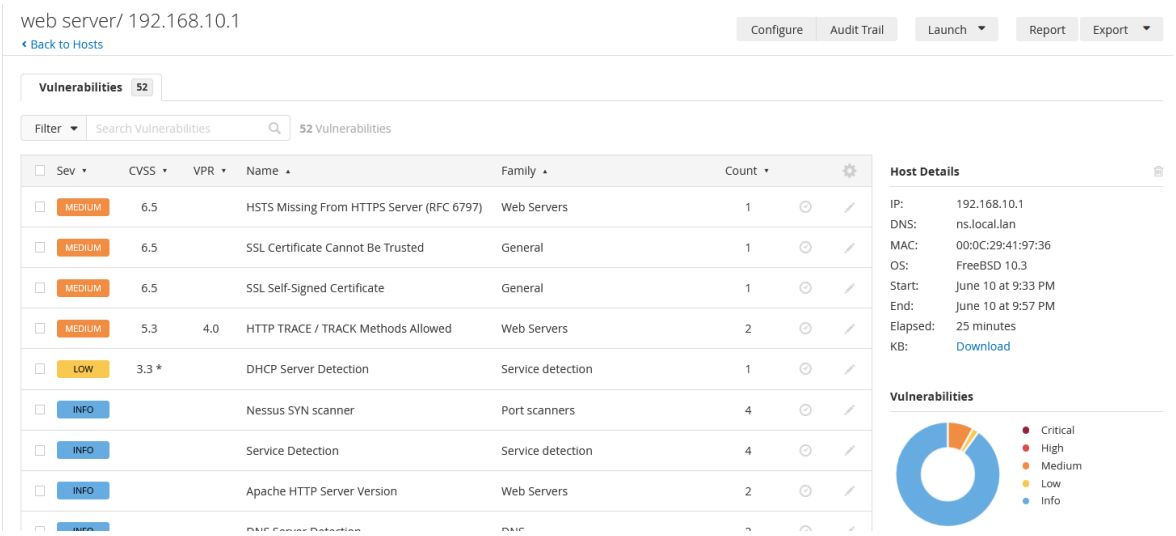

Рисунок 3.10 – Результат сканування після виправлення вразливостей

Як зображено на рисунку 3.10, сервер все-ще має вразливості такі як SSL Certificate Cannot Be Trusted та SSL Self-Signet Certificate:

– Вразливість SSL Certificate Cannot Be Trusted виникає, коли веб-сайт використовує SSL-сертифікат, який не може бути перевірений довірений браузерами або системами безпеки. Це може статися, якщо сертифікат був виданий ненадійним центром сертифікації або якщо його термін дії закінчився, але веб-сайт продовжує використовувати його.

– Вразливість SSL Self-Signet Certificate виникає коли використовується

самопідписаний SSL-сертифікат, замість сертифікату, що підписаний визнаним центром сертифікації. Браузери та системи безпеки не довіряють самопідписаним сертифікатам, оскільки вони не можуть перевірити їх аутентичність.

Для виправлення вразливостей SSL Certificate Cannot Be Trusted і SSL Self-Signed Certificate потрібно отримати та використовувати довірений SSL-сертифікат, підписаний визнаним центром сертифікації. В нашому випадку, головний пріоритет - це проведення тестувань і досліджень на сервері, а не вирішення питань, пов'язаних з сертифікатами. Оскільки самопідписані сертифікати не мають прямого впливу на нашу роботу та безпеку на сервері під час досліджень, ми можемо прийняти рішення не вносити зміни щодо виправлення цих сертифікатів. Але важливо зазначити, що це рішення може бути прийняте тільки у контексті проведення досліджень, і для публічного сервера або виробничого середовища рекомендується використовувати валідні сертифікати, щоб забезпечити безпеку та довіру користувачів.

В процесі аналізу нашої системи було виявлено ще одну вразливість - HSTS Missing from HTTPS Server (RFC 6797). Однак, попри це, ми не зможемо виправити цю конкретну вразливість, адже вона є false positive. Виявлення false positive вказує на те, що вразливість була ідентифікована сканером безпеки, але насправді вона не існує в реальному середовищі. Це може статися через ряд причин, включаючи недосконалості алгоритмів детекції, неправильні налаштування сканера, або неправильне інтерпретування результатів. У випадку HSTS Missing from HTTPS Server (RFC 6797) сканер безпеки неправильно інтерпретує відсутність заголовка HSTS як вразливість, хоча в деяких випадках це може бути нормальною конфігурацією. Тому, незважаючи на те, що ця вразливість з'являється у звіті сканування, ми можемо її ігнорувати, оскільки вона не відображає реальну проблему в нашій системі.

В цілому, за допомогою досліджень та виправлень, описаних в цьому розділі, вдалось значно підвищити рівень безпеки вебсервера.

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Вимоги ергономіки до організації робочого місця оператора ПК

Ергономічна організація робочого місця користувача ЕОМ враховує як специфіку діяльності, що виконується, так і забезпечує комфортні умови перебування людини. Тому основними ергономічними завданнями щодо організації робочого місця є наступні:

– забезпечення просторових параметрів робочого місця, які відповідають антропометричним характеристикам користувача;

– раціональне розташування елементів робочого місця відносно користувача на підставі поглибленого кількісного та якісного аналізу діяльності, яка виконується;

– оптимізацію умов робочого середовища.

На рисунку 4.1 наведено робоче місце користувача ЕОМ та позначено основні ергономічні та просторові параметри його складових.

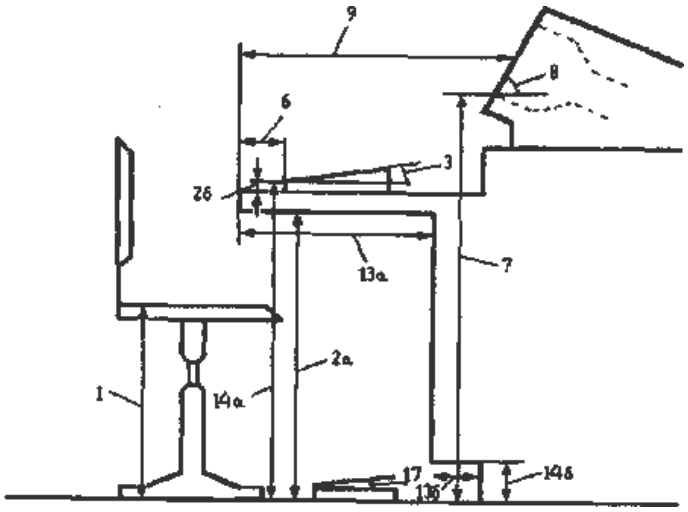

Рисунок 4.1 – Робоче місце користувача ЕОМ

Основні просторові параметри робочого місця користувача ЕОМ приведені в таблиці 4.1 [11].

Таблиця 4.1 – Просторові параметри робочого місця

Умовні позначення	Параметри	Спосіб вимірювання параметра	Значення параметра
1	Висота сидіння	Від підлоги до верхньої площини сидіння	400-500 мм
2	Висота клавіатури (від рівня підлоги)	Від підлоги до нижнього ряду клавіатури	600-700 мм
2а	Висота клавіатури (від рівня стола)	Від базової поверхні до нижнього ряду клавіатури	20 мм
3	Кут нахилу клавіатури	Від горизонтальної площини	7-15°
4	Ширина основної клавіатури	Визначається оптимальною зоною моторного поля	До 400 мм
5	Глибина основної клавіатури	Визначається оптимальною зоною моторного поля	До 200 мм
6	Відстань від клавіатури до краю стола	Від переднього краю стола до клавіатури	Понад 80 - 100 мм
7	Висота екрана	Від підлоги до нижнього краю екрана	950-1050 мм
8	Кут нахилу екрана	Від вертикальної площини	15°
9	Відстань від екрана до краю стола	Від переднього краю стола до екрана	500-700 мм
10	Висота поверхні для запису	Від підлоги	870-860 мм
11	Площа поверхні для запису	Визначається оптимальною зоною моторного поля	600 х 400 мм 900 х 600 мм
12	Кут нахилу поверхні для запису	Від горизонтальної площини	0 – 10°
13	Глибина простору для ніг на рівні колін	Від переднього краю стола	Понад 400 мм
13а	Глибина простору для ніг на рівні колін	Від підлоги	Понад 600 мм
14	Висота простору для ніг на рівні колін	Від переднього краю стола	Понад 600 мм
14а	Висота простору для ніг на рівні ступень	Від підлоги	Понад 100 мм
15	Ширина простору для ніг на рівні колін		Понад 500 мм
15а	Ширина простору для ніг на рівні		Понад 250 мм
16	Висота підставки для ніг	Від підлоги до передньої частини підставки	50-130 мм
17	Кут нахилу підставки для ніг	Від горизонтальної площини	0-25°
18	Глибина підставки для ніг	Від переднього краю підставки до її заднього краю	400 мм
19	Ширина підставки для ніг		300 мм
20	Пюпітр-підставка для документів	Від горизонтальної площини	15 - 20°

В ході організації робочих місць на кожну ЕОМ виділяється площа, яка складає не менш, ніж 6 м^2, та об'єм, який становить не менш, ніж 20

$м^3$. Причому, зона, де розташовується робочий стіл, сервер або робоча станція, принтер, екран для графопроектора, займає відповідно 6-8 $м^2$. Висота приміщення не менша, ніж 4 м.

Робоче місце користувача ПК облаштоване одномісним столом та напівм'яким стільцем, висоту сидіння яких можна змінювати. Довжина стола користувача не менше 700 мм, ширина – забезпечує місце перед клавіатурою для розташування зошита або іншого приладдя. Поверхня стола має кут нахилу у межах 12-15^o, лише іноді припустимою є її розташування у горизонтальній площині [11].

На робочому місці користувача ПК забезпечена відповідність висоти краю стола і стільця до росту та антропометричних особливостей організму користувачів. Як нормативні визначають показники, що приведені у таблиці 4.2.

Таблиця 4.2 – Нормативні показники

Ріст, мм	Висота над підлогою, мм		
	стіл	простір для ніг	стілець
1450-1600	640	530	380
1610 -1750	700	590	420
Понад 1750	760	650	460

Глибина простору для ніг під столом не менше 450 мм, а у випадку застосування високого стола та низького стільця і, отже, відсутності відповідності росту користувача конструктивним елементам робочого місця, використовується підставка для ніг, ширина якої становить – 350 мм, довжина – 400 мм, кут нахилу опорної поверхні – 15^o.

Столи з ЕОМ розміщено без розривів між ними, але при незначній кількості робочих столів з відеотерміналами перевагу варто віддавати розташуванню їх біля внутрішньої стіни.

Робота з комп'ютерною технікою вимагає обов'язкового дотримування правильної посадки. Користувач ЕОМ повинен сидіти прямо, з невеликим нахилом (до 5° – 7°) голови вперед, не сутулитися, спираючись нижніми краями

лопаток на спинку стільця. Передпліччя повинні спиратися на поверхню стола, забезпечуючи зниження статичного напруження м'язів плечового поясу і рук, кути, що утворюються передпліччям і плечем, а також гомілкою і стегном, – складати не менш, ніж 90°.

Рівень очей припадає на центр екрана або на точку, яка розташована між верхньою та середньою третинами екрану, причому, лінія погляду є перпендикулярною до площини екрана, а її відхилення у вертикальній площині – знаходитися у межах ±5–10º. Оптимальний огляд у горизонтальній площині від центральної осі екрана у межах ±15–30°. Лише під час спостереження за інформацією, яка розміщена у найвіддаленіших ділянках екрану, кут огляду становить 40–45º.

Кут розглядання цифр та букв на екрані монітора не менше 20 кутових хвилин, а його величину розраховують за формулою 4.1 [11]:

$$tg\ \alpha\ /\ 2 = \frac{S}{2L}, \tag{4.1}$$

де S – висота букви або цифри, мм;

L – відстань від очей до об'єкта інформації на екрані, мм;

α – кут розглядання, кутові хвилини.

Оптимальна відстань від очей до площини екрана монітора складає 600 – 700 мм, допустима – не менше 500 мм. Розглядати інформацію на екрані з відстані менш, ніж 500 мм не рекомендується.

4.2 Організація служби охорони праці на підприємстві

Роботодавець зобов'язаний згідно Закону України «Про охорону праці» стаття 13 «Управління охороною праці та обов’язки роботодавця» створити на робочому місці в кожному структурному підрозділі умови праці відповідно до нормативно-правових актів, а також забезпечити додержання вимог законодавства щодо прав працівників у галузі охорони праці.

Із цією метою роботодавець забезпечує функціонування системи управління охороною праці, а саме:

– створює відповідні служби і призначає посадових осіб, які забезпечують вирішення конкретних питань охорони праці, затверджує інструкції про їхні обов'язки, права та відповідальність за виконання покладених на них функцій, а також контролює їх додержання;

– розробляє за участю сторін колективного договору і реалізує комплексні заходи для досягнення встановлених нормативів та підвищення існуючого рівня охорони праці;

– забезпечує виконання необхідних профілактичних заходів відповідно до обставин, що змінюються;

– впроваджує прогресивні технології, досягнення науки і техніки, засоби механізації та автоматизації виробництва, вимоги ергономіки, позитивний досвід з охорони праці тощо;

– забезпечує належне утримання будівель та споруд, виробничого обладнання та устаткування, моніторинг за їх технічним станом;

– забезпечує усунення причин, що призводять до нещасних випадків, професійних захворювань, та здійснення профілактичних заходів, визначених комісіями за підсумками розслідування цих причин;

– організовує проведення аудиту охорони праці, лабораторних досліджень умов праці, оцінку технічного стану виробничого обладнання та устаткування, атестацій робочих місць на відповідність нормативно-правовим актам з охорони праці в порядку і строки, що визначаються законодавством, та за їх підсумками вживає заходів з усунення небезпечних і шкідливих для здоров'я виробничих факторів;

– розробляє і затверджує положення, інструкції, інші акти з охорони праці, що діють у межах підприємства та встановлюють правила виконання робіт і поведінки працівників на території підприємства, у виробничих приміщеннях, на будівельних майданчиках, робочих місцях відповідно до нормативно-правових актів з охорони праці, забезпечує безоплатно працівників нормативно-правовими актами підприємства з охорони праці;

– здійснює контроль за додержанням працівником технологічних процесів, правил поведінки з машинами, механізмами, устаткуванням та іншими засобами виробництва, використанням засобів колективного та індивідуального захисту, виконанням робіт відповідно до вимог з охорони праці.

Спеціалісти служби охорони праці у разі виявлення порушень охорони праці мають право:

– видавати керівникам структурних підрозділів підприємства обов'язкові для виконання приписи щодо усунення наявних недоліків, одержувати від них необхідні відомості, документацію і пояснення з питань охорони праці;

– вимагати відсторонення від роботи осіб, які не пройшли передбачених законодавством медичного огляду, навчання, інструктажу, перевірки знань і не мають допуску до відповідних робіт або не виконують вимог нормативно-правових актів з охорони праці;

– зупиняти роботу виробництва, дільниці, машин, механізмів, устаткування та інших засобів виробництва у разі порушень, які створюють загрозу життю або здоров'ю працівників;

– надсилати роботодавцю подання про притягнення до відповідальності працівників, які порушують вимоги щодо охорони праці.

Ліквідація служби охорони праці допускається тільки у разі ліквідації підприємства чи припинення використання найманої праці фізичною особою.

Законодавство про охорону праці передбачає і обов'язки працівників. Зокрема вони зобов'язані:

– дбати про особисту безпеку і здоров'я, а також про безпеку і здоров'я оточуючих людей у процесі виконання будь-яких робіт під час перебування на території підприємства;

– знати і виконувати вимоги нормативно-правових актів з охорони праці, правила поведінки з машинами, механізмами, устаткуванням та іншими засобами виробництва, користуватися засобами колективного та індивідуального захисту;

– проходити у встановленому законодавством порядку попередні та періодичні медичні огляди.

Працівник несе безпосередню відповідальність за порушення зазначених вимог. Дотримання правил безпеки і виробничої санітарії залежить не тільки від виконання роботодавцем своїх обов'язків, а й від того, наскільки кожен працівник знає і виконує правила під час роботи. Тому всі працівники при прийомі на роботу і в процесі роботи проходять на підприємстві інструктаж з охорони праці, надання першої медичної допомоги потерпілим від нещасних випадків, правил поведінки при виникненні аварій.

Навчання й інструктаж працівників з охорони праці є складовою частиною системи управління охороною праці і проводиться з усіма працівниками в процесі їхньої трудової діяльності. Інструктаж працівників залежно від характеру та часу його проведення буває вступний (при прийомі на роботу); первинний (на робочому місці з усіма працівниками: на роботах із підвищеною небезпекою - один раз на квартал, на інших роботах — один раз на півроку; проводиться або індивідуально, або з групою працівників, що виконують однотипні роботи, за програмою первинного інструктажу); позаплановий (при зміні правил з охорони праці, заміні устаткування чи за інших змін факторів, що впливають на безпеку праці); цільовий (при виконанні разових робіт, не пов'язаних із прямими обов'язками за фахом).

ВИСНОВКИ

У результаті виконання даної дипломної роботи було здійснено детальний аналіз і порівняння основних криптографічних протоколів захисту інформації в мережі Інтернет, зокрема протоколів SSL і TLS. Це дослідження виявилося важливим кроком у розумінні способів підвищення безпеки інформаційних обмінів в мережі Інтернет. Було виявлено, що незважаючи на високий рівень захисту, який надають ці протоколи, існують потенційні вразливості, які можуть бути використані зловмисниками. До них належать слабкі місця у конфігурації вебсервера, неоновлене програмне забезпечення та використання застарілих версій протоколів.

Результати цієї роботи мають велике значення для галузі захисту інформації. Вони допомагають у розумінні, як правильно конфігурувати вебсервери для надання максимального захисту від кібератак. Крім того, рекомендації з покращення безпеки, викладені в цій роботі, можуть бути використані практиками в області кібербезпеки.

Результати даного дослідження можуть бути використані для покращення існуючих криптографічних протоколів, зокрема SSL і TLS. Детальний аналіз виявив потенційні вразливості цих протоколів та можливі шляхи їх усунення, що може сприяти підвищенню рівня безпеки при передачі інформації через Інтернет. Окрім цього, аналіз та порівняльна оцінка цих протоколів можуть бути використані розробниками та інженерами кібербезпеки при модернізації існуючих протоколів.

З урахуванням складності і постійної зміни технологічного середовища, рекомендується продовжити дослідження в цій області. В перспективі особливої уваги вимагає дослідження впливу квантових технологій на криптографічні протоколи. Адже розвиток квантового обчислення може категорично змінити існуючі моделі захисту і, як наслідок, потребуватиме від нас нових стратегій захисту від кібератак.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Automated Validation of Internet Security Protocols and Applications (AVISPA) [Електронний ресурс] / AVISPA. – 2003. – Режим доступу: http://www.avispa-project.org/delivs/6.1/d6-1/node3.html – Дата звернення: 01.05.2023.

2. Katz J. Introduction to Modern Cryptography: Third Edition (Chapman & Hall/CRC Cryptography and Network Security Series) / J. Katz, Y. Lindell. – Chapman and Hall/CRC, 2020. – 648 с.

3. Schneier B. Applied Cryptography / Bruce Schneier. – John Wiley & Sons, 1996. – 784 с.

4. Stallings W. Cryptography and Network Security: Principles and Practice (6th Edition) / William Stallings. – Pearson, 2013. – 752 с.

5. Sunshine C. A. Computer Network Architectures and Protocols / Carl A. Sunshine. – Springer Science & Business Media, 2013. – 542 с.

6. Tanenbaum A. S. Computer Networks (6th Edition) / A. S. Tanenbaum, N. Feamster, D. J. Wetherall. – Pearson, 2020. – 960 с.

7. Вербіцький О.В. Вступ до криптології / О. В. Вербіцький. – Львів: Вид-во НТЛ, 2008. – 248 с.

8. Лагун А. Е. Криптографічні системи та протоколи: навч. посіб. / А. Е. Лагун. – Львів: Вид-во Львів. політехніки, 2013. – 96 с.

9. Diffie W., Hellman M. New directions in cryptography // IEEE Transactions on Information Theory. – 1976. – Vol. 22, No. 6. – P. 644–654.

10. Rivest R., Shamir A., Adleman L. A method for obtaining digital signatures and public-key cryptosystems // Communications of the ACM. – 1978. – Vol. 21, No. 2. – P. 120–126.

11. Головченко С.І. Охорона праці. Основи ергономіки: Підручник / С.І. Головченко, І.А. Косарєв, М.А. Словак. – К.: Вища школа, 2016. – 384 с.

12. НПАОП 0.00-1.28-10 Правила охорони праці при експлуатації електронно-обчислювальних машин. – К., 2010. – 16 с.

13. Singh S. The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography / Simon Singh. – Anchor Books, 2000. – 432 с.

14. Ferguson N., Schneier B., Kohno T. Cryptography Engineering: Design Principles and Practical Applications / Niels Ferguson, Bruce Schneier, Tadayoshi Kohno. – John Wiley & Sons, 2010. – 384 с.

15. Rivest R. L., Shamir A., Adleman L. A method for obtaining digital signatures and public-key cryptosystems / R. L. Rivest, A. Shamir, L. Adleman. – Communications of the ACM, 1978. – 21 (2). – С. 120–126.

16. Stinson D. R. Cryptography: Theory and Practice (3rd Edition) / Douglas R. Stinson. – Chapman and Hall/CRC, 2005. – 616 с.

17. Lucas M. Absolute FreeBSD: The Complete Guide to FreeBSD / M. Lucas. – No Starch Press, 2018. – 720 с.

18. Nessus [Електронний ресурс] / Tenable, Inc. – 2023. – Режим доступу: https://www.tenable.com/products/nessus - Дата звернення: 11.05.2023.

19. Nessus Documentation [Електронний ресурс] / Tenable, Inc. – 2023. – Режим доступу: https://docs.tenable.com/nessus/Content/Welcome.htm - Дата звернення: 10.05.2023.

20. Apache HTTP Server Documentation Version 2.4 [Електронний ресурс] / The Apache Software Foundation. – 2023. – Режим доступу: https://httpd.apache.org/docs/2.4/ - Дата звернення: 17.05.2023.

21. Bowen A., Coar K. Apache Cookbook: Solutions and Examples for Apache Administration / A. Bowen, K. Coar. – O'Reilly Media, 2007. – 252 с.

22. Rescorla E. SSL and TLS: Designing and Building Secure Systems / E. Rescorla. – Addison-Wesley Professional, 2000. – 528 с.

23. Dierks T., Rescorla E. The Transport Layer Security (TLS) Protocol Version 1.2 [Електронний ресурс] / T. Dierks, E. Rescorla. – The Internet Engineering Task Force (IETF). – 2008. – Режим доступу: https://tools.ietf.org/html/rfc5246 - Дата звернення: 07.05.2023.

24. FreeBSD Ports Collection: Apache24 [Електронний ресурс]. – The FreeBSD Project. – 2023. – Режим доступу: https://www.freshports.org/www/apache24/ - Дата звернення: 01.06.2023