Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(освітній рівень)

на тему: *" Розробка та налаштування захищеного корпоративного. поштового сервера "*

Виконав: студент (ка)
Спеціальності:
125 «Кібербезпека»
(шифр і назва напряму підготовки, спеціальності)
Луговський П.В.
підпис (прізвище та ініціали)

Керівник *Александер Марек Богуслав*
підпис (прізвище та ініціали)

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

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

Рецензент
підпис (прізвище та ініціали)

м. Тернопіль – 2023

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

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра кібербезпеки
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Загородна Н.В.
(підпис) (прізвище та ініціали)
«__» _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 125 Кібербезпека
(шифр і назва спеціальності)

Студенту Луговському Павлу Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка та налаштування захищеного корпоративного поштового сервера.

Керівник роботи Александер Марек Богуслав Антонович, д.т.н., професор кафедри КБ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «03» 04 2023 року № 4/7-349

2. Термін подання студентом завершеної роботи 12.06.2023

3. Вихідні дані до роботи Вимоги до корпоративного поштового сервера

4. Зміст роботи (перелік питань, які потрібно розробити)
Огляд загальних принципів розробки та налаштування поштових серверів.
Розглянути загрози безпеці та методи захисту електронних поштових сервісів
Розглянути можливі архітектури та вимоги до поштового сервера
Встановити та налаштувати відповідні компоненти поштового сервера
Здійснити тестування функціональності та провести оцінку безпеки поштового сервера
Безпека життєдіяльності, основи охорони праці
Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
Тема, мета, задачі. Загрози безпеці та методи захисту електронних поштових сервісів. Архітектура корпоративного поштового сервера. Postfix. Dovecot, Dovecot-pigeonhole (Sieve). ClamAV та clamsmtpd. ClamAV. Cyrus-SASL та Cyrus-SASL-saslauthd. Налаштування поштового сервера. Налаштування DNS. Встановлення та налаштування Dovecot та dovecot-pigeonhole. Встановлення та налаштування ClamAV та Clamsmtpd для поштового сервера. Тестування функціональності. Записи в файлі журналу /var/log/maillog. Оцінка безпеки поштового сервера. Перевірка SPF запису для домену cs.networkacad.net. Перевірка за допомогою ***checktls.com.*** Результати сканування на вразливості сканером nessus . Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи хорони праці	Пилипець М.І., проф. кафедри МТ Гурик О.Я., доц. кафедри МТ		

7. Дата видачі завдання 22.01.2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	22.01 – 25.01	Виконано
2.	Підбір джерел аналізу роботи і налаштування поштових серверів	26.01 – 30.01	Виконано
3.	Опрацювання джерел в галузі дослідження	01.02 – 14.02	Виконано
4.	Налаштування операційної системи та корпоративного поштового сервера	05.03 – 20.03	Виконано
5.	Тестування роботи та безпеки корпоративного поштового сервера	20.03-08.04	Виконано
6.	Оформлення розділу «Теоретичні основи електронних поштових серверів»	14.02 – 17.03	Виконано
7.	Оформлення розділу «Розробка та налаштування поштового сервера»	20.03 – 29.04	Виконано
8.	Оформлення розділу «Тестування та оцінка безпеки»	28.04 – 10.05	Виконано
9.	Виконання завдання до підрозділу «Безпека життєдіяльності, основи охорони праці»	11.05 – 25.05	Виконано
10.	Оформлення кваліфікаційної роботи	25.05 – 05.06	Виконано
11.	Нормоконтроль	07.06 – 14.06	Виконано
12.	Перевірка на плагіат	11.06 – 12.06	Виконано
13.	Попередній захист кваліфікаційної роботи	14.06 – 15.06	Виконано
14.	Захист кваліфікаційної роботи	20.06.2023	

Студент _____ *Луговський П.В.*
(підпис) (прізвище та ініціали)

Керівник роботи _____ *Александер Марек Богуслав*
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Розробка та налаштування захищеного корпоративного поштового сервера // Кваліфікаційна робота ОР «Бакалавр» // Луговський Павло Володимирович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБ-41 // Тернопіль, 2023 // С. _74_ , рис. – 28__, табл. – _-_ , кресл. 21 , додат. – _1__.

Ключові слова: FREEBSD, POSTFIX, DOVECOT, CLAMAV, CLAMSMTPD, CYRUS-SASL, SIEVE, OPENSSL.

Ця кваліфікаційна робота присвячена розробці та налаштуванню захищеного корпоративного поштового сервера з використанням вільної операційної системи FreeBSD та набору програмних засобів, таких як Postfix, Dovecot, ClamAV, Clamsmtpd, Cyrus-SASL, Cyrus-SASL-saslauthd та OpenSSL.

Основною метою роботи є створення безпечної та надійної інфраструктури електронної пошти для корпоративного використання. В роботі досліджуються основні аспекти налаштування кожного компонента поштового сервера та їх взаємодії для забезпечення надійності, безпеки, ефективності обробки та доставки повідомлень.

У роботі проводиться аналіз особливостей використання операційної системи FreeBSD у якості базової платформи для поштового сервера. Детально вивчаються можливості та налаштування компонентів Postfix, Dovecot, Clamsmtpd, Cyrus-SASL-saslauthd для забезпечення безпеки, аутентифікації, обробки спаму та вірусів, шифрування та інших функціональних можливостей.

В рамках дипломної роботи розробляється практичний поштовий сервер з використанням зазначених компонентів і проводяться тестування його функціоналу та ефективності.

Результати дослідження та розробки поштового сервера можуть бути використані для практичної реалізації безпечних корпоративних поштових сервісів які забезпечують надійну комунікацію та обмін інформацією в організації.

ANNOTATION

Development and configuration of a secure corporate mail server // Thesis of educational level "Bachelor" // Pavlo Lugovskyi // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Cybersecurity, СБ-41 group // Ternopil, 2023 // P. _74__ , fig. -_28___, table. - -_ , chair. - _21_ , added. -__1_.

Keywords: FREEBSD, POSTFIX, DOVECOT, CLAMAV, CLAMSMTPD, CYRUS-SASL, SIEVE, OPENSSL.

This qualification work is dedicated to the development and configuration of a secure corporate mail server using the FreeBSD operating system and a set of software tools such as Postfix, Dovecot, ClamAV, Clamsmtpd, Cyrus-SASL, Cyrus-SASL-saslauthd, and OpenSSL. The main goal of the work is to create a secure and reliable email infrastructure for corporate use. The study explores the key aspects of configuring each component of the mail server and their interactions to ensure message reliability, security, processing efficiency, and delivery.

The work analyzes the features of using the FreeBSD operating system as the underlying platform for the mail server. The capabilities and configurations of Postfix, Dovecot, Clamsmtpd, Cyrus-SASL-saslauthd components are thoroughly examined to ensure security, authentication, spam and virus processing, encryption, and other functional capabilities.

Within the scope of the thesis, a practical mail server is developed using the mentioned components, and testing of its functionality and efficiency is conducted. The research and development results of the mail server can be used for the practical implementation of secure corporate mail services that provide reliable communication and information exchange within organizations.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

RBL – Real-time Blackhole List

SPF – Sender Policy Framework

TLS – Transport Layer Security

MTA – Mail Transfer Agent

SMTP – Simple Mail Transfer Protocol

MDA – Mail Delivery Agent

SSL – Secure Sockets Layer

IMAP – Internet Message Access Protocol

POP3 – Post Office Protocol

ZFS – Zettabyte File System

FQDN – Fully Qualified Domain Name

SASL – Simple Authentication and Security Layer

TCP/IP – Transmission Control Protocol/Internet Protocol

DNS – Domain Name System

MX – Mail Exchanger

LDA – Local Delivery Agent

ClamAV – Clam Anti-Virus

ВСТУП

Сучасне корпоративне середовище вимагає ефективних та безпечних засобів обміну електронними повідомленнями. Електронна пошта є одним з найпоширеніших та найважливіших інструментів комунікації в бізнесі, і забезпечення безпеки цього засобу є критичним завданням для організацій.

Захист важливої корпоративної інформації від несанкціонованого доступу, спаму, вірусів та інших загроз є пріоритетом для будь-якої компанії. Досягнення цієї мети вимагає комплексного підходу та використання сучасних технологій безпеки.

Ця дипломна робота присвячена розробці та налаштуванню захищеного корпоративного поштового сервера з використанням програмних засобів FreeBSD, Postfix, Dovecot, Dovecot-pigeonhole, ClamAV, Clamsmtpd, Cyrus-SASL, Cyrus-SASL-saslauthd та OpenSSL. Головною метою цього дослідження є створення безпечного та надійного середовища для обміну електронними повідомленнями у корпоративному секторі.

Основні завдання цієї дипломної роботи включають розробку архітектури корпоративного поштового сервера, налаштування компонентів сервера для забезпечення безпеки, аутентифікації, обробки спаму та вірусів, а також шифрування комунікації. Після розробки та налаштування сервера проводиться його тестування та оцінка результатів згідно визначених критеріїв.

Ця дипломна робота важлива з точки зору практичної реалізації безпечних корпоративних поштових сервісів, що сприятимуть надійній комунікації та обміну інформацією в організації. Результати цього дослідження можуть бути використані як основа для подальшого розвитку та вдосконалення захищених поштових серверів.

У наступних розділах роботи будуть розглянуті основні аспекти розробки та налаштування захищеного корпоративного поштового сервера з використанням зазначених компонентів. Результати дослідження та розробки допоможуть покращити безпеку і ефективність корпоративного обміну електронними повідомленнями.

1 ТЕОРЕТИЧНІ ОСНОВИ ЕЛЕКТРОННИХ ПОШТОВИХ СЕРВІСІВ.

Електронні поштові сервіси (electronic mail services) забезпечують можливість обміну електронними листами між користувачами через Інтернет. Вони включають в себе всі необхідні компоненти для відправки, отримання, зберігання та управління електронними листами.

1.1 Загрози безпеці та методи захисту електронних поштових сервісів.

Електронні поштові сервери піддаються різноманітним загрозам, які можуть поставити під загрозу безпеку та надійність обміну електронними повідомленнями. До основних загроз належать:

- Спам: небажані повідомлення, які розсилаються автоматично до великої кількості користувачів. Спам може спричиняти перевантаження поштового сервера та завдати шкоди ефективності та продуктивності.
- Віруси та шкідливі програми: електронні повідомлення можуть містити вкладені файли, які містять віруси, троянські програми або інші шкідливі коди. Це може призвести до порушення безпеки користувачів та інфраструктури.
- Перехоплення конфіденційної інформації: незахищена комунікація між поштовими серверами може бути перехопленою, що призведе до ризику розголошення конфіденційних даних.

Методи захисту від спаму та вірусів.

Для захисту поштового сервера від спаму та вірусів використовуються різні методи:

- Фільтрація спаму: застосування різних алгоритмів та правил для виявлення та блокування спамових повідомлень. Це може включати аналіз заголовків та вмісту повідомлень, перевірку IP-адреси відправника, використання списків блокування спаму (RBL) та багато іншого.

- Антивірусний сканер: використання спеціального програмного забезпечення для виявлення та блокування вірусів та інших шкідливих програм у вхідних та вихідних повідомленнях. Антивірусний сканер може перевіряти вкладені файли, посилання та інші компоненти повідомлення на наявність загроз.
- SPF: механізм, який дозволяє перевіряти автентичність відправника електронного повідомлення. SPF допомагає виявляти фальшиві електронні адреси та запобігає підробці поштового сервера.

Принципи шифрування комунікації.

Забезпечення шифрування комунікації між поштовими серверами та між поштовим клієнтом і сервером допомагає запобігти перехопленню та незаконному доступу до електронних повідомлень.

Основні принципи шифрування включають:

- Використання протоколу TLS: Шифрування комунікації між поштовими серверами за допомогою протоколу TLS. Це забезпечує конфіденційність та цілісність даних під час їх передачі.
- Цифрові сертифікати: використання цифрових сертифікатів для перевірки автентичності поштових серверів перед встановленням захищеної з'єднання.
- Криптографічні алгоритми: використання сильних криптографічних алгоритмів для шифрування та розшифрування повідомлень. Це забезпечує надійний рівень захисту даних.

TLS - це криптографічний протокол, який забезпечує безпечну комунікацію та захист даних під час передачі через мережу, таку як Інтернет. Він використовується для шифрування та аутентифікації даних, що передаються між клієнтом і сервером. TLS використовує криптографічні алгоритми для шифрування даних. Це дозволяє захистити конфіденційність інформації, що передається між двома точками, шляхом перетворення її у зашифрований вигляд, який може бути розшифрований тільки з використанням правильного ключа.

Даний протокол забезпечує аутентифікацію сторін, що здійснюють з'єднання, забезпечуючи перевірку їх ідентичності. Це досягається за допомогою

цифрових сертифікатів, які містять публічні ключі, що використовуються для перевірки підпису та підтвердження автентичності сторін.

Шифрування TLS в поштовому сервері postfix забезпечують безпечну комунікацію між поштовими серверами шляхом застосування шифрування та аутентифікації.

Нижче наведені основні моменти шифрування TLS в Postfix:

1. Встановлення зв'язку. Коли поштовий сервер Postfix встановлює з'єднання з іншим поштовим сервером, він ініціює процес TLS Handshake. Під час цього процесу сервери обмінюються сертифікатами для аутентифікації та встановлюють загальний секретний ключ для шифрування даних.
2. Аутентифікація. Кожен поштовий сервер, який бере участь у з'єднанні, повинен мати валідний цифровий сертифікат. Під час TLS Handshake сервери перевіряють сертифікати один одного, щоб забезпечити правильність аутентифікації.
3. Шифрування. Після успішного TLS Handshake поштові сервери використовують загальний секретний ключ для шифрування даних, що передаються між ними. Це забезпечує конфіденційність інформації під час передачі через мережу.
4. Валідація сертифікатів. Поштовий сервер перевіряє валідність сертифіката, отриманого від іншого поштового сервера, використовуючи ланцюжок довірених сертифікатів, або встановлює що цей сертифікат є самопідписним.
5. Підтримка протоколів TLS. Postfix підтримує різні версії протоколу TLS, такі як TLSv1.2 та TLSv1.3, які є надійними та безпечними. Конфігурація postfix може бути налаштована для підтримки потрібних версій протоколу TLS та відповідних шифрів.
6. Інспекція сертифікатів. Postfix надає можливість перевірити сертифікати, використовуючи зовнішні інструменти, такі як OpenSSL. Це дозволяє здійснити детальний аналіз сертифікатів, перевірити їх дійсність, термін дії, власника та інші атрибути.

7. Конфігурація параметрів TLS:. Postfix дозволяє налаштувати різні параметри TLS, такі як рівень шифрування, криптографічні алгоритми, методи аутентифікації та інші. Це дозволяє адміністраторам точно налаштувати безпеку комунікації згідно з вимогами організації.

Застосування TLS в поштовому сервері Postfix допомагає забезпечити конфіденційність, цілісність та аутентичність комунікації між поштовими серверами. Це важливий аспект безпеки в корпоративному поштовому середовищі, який допомагає захистити важливі дані та запобігти несанкціонованому доступу до поштових повідомлень.

1.2 Архітектура корпоративного поштового сервера.

Корпоративні поштові сервери можуть бути побудовані на різних операційних системах, в залежності від вимог, вподобань та доступності. Ось деякі з популярних операційних систем, які використовуються для побудови корпоративних поштових серверів:

Microsoft Windows Server: Microsoft Windows Server є однією з поширених операційних систем для побудови корпоративних поштових серверів. Він підтримує такі поштові сервери, як Microsoft Exchange Server, який є одним з провідних рішень для корпоративної електронної пошти [1].

Linux: Linux є популярною операційною системою для побудови корпоративних поштових серверів. Багато розповсюджених поштових служб, таких як Postfix, Exim та Sendmail, можуть бути встановлені та налаштовані на базі Linux. Популярними дистрибутивами Linux для поштових серверів є CentOS, Ubuntu, Redhat та Debian.

FreeBSD: FreeBSD є Unix операційною системою, яка використовується для побудови корпоративних поштових серверів. Вона має високу стабільність, надійність та безпеку, і підтримує такі поштові сервери, як Postfix, Sendmail та інші.

В даній кваліфікаційній роботі для побудови корпоративного поштового сервера буде використано операційну систему FreeBSD [2].

Огляд компонентів поштового сервера.

Корпоративний поштовий сервер на основі FreeBSD буде розроблено та налаштовано з використанням таких компонентів, як Postfix, Dovecot, Dovecot-pigeonhole (Sieve), ClamAV, Clamsmtpd, Cyrus-SASL, Cyrus-SASL-saslauthd та openssl. Огляд цих компонентів надає загальне уявлення про структуру та функціональні можливості корпоративного поштового сервера.

Postfix - це поштовий агент передачі (MTA), який відповідає за отримання та пересилання електронних повідомлень. Postfix є одним з найпоширеніших MTA і відзначається своєю надійністю та безпекою. Він використовується для прийому повідомлень від зовнішніх поштових серверів та їх подальшої доставки до локальних скриньок або інших внутрішніх серверів [3].

Основні компоненти Postfix включають:

SMTP-сервер: відповідає за приймання та доставку електронних повідомлень між поштовими серверами. SMTP-сервер postfix отримує, перевіряє та передає повідомлення відправників до відповідних поштових скриньок отримувачів.

MTA-агент: відповідає за приймання та доставку повідомлень в межах внутрішньої мережі. MTA-агент Postfix обробляє повідомлення внутрішніх користувачів, здійснює пересилання повідомлень між вузлами мережі та виконує різні фільтраційні операції.

MDA-агент: відповідає за збереження та доставку повідомлень до поштових скриньок користувачів. Агент доставки Postfix зберігає повідомлення у відповідних директоріях, забезпечує контроль доступу та керування повідомленнями.

База даних транспорту (Transport Database): використовується для налаштування маршрутизації повідомлень до відповідних поштових серверів. База даних транспорту Postfix містить інформацію про адреси поштових серверів та правила маршрутизації.

Dovecot - це сервер IMAP та POP3, який забезпечує доступ користувачів до їхніх поштових скриньок [4]. Dovecot підтримує безпечний доступ до пошти за допомогою TLS і забезпечує розширені можливості управління повідомленнями,

включаючи фільтрацію та сортування. Використання протоколу TLS в Dovecot забезпечує шифрування даних, аутентифікацію сторін та захист від перехоплення або модифікації інформації під час передачі. Для використання TLS необхідно мати дійсний сертифікат, який підтверджує ідентичність сервера. Це може бути самопідписаний сертифікат або сертифікат, виданий довіреним центром сертифікації. Сертифікат повинен бути налаштований у Dovecot для використання під час TLS з'єднань. Dovecot підтримує різні версії протоколу TLS, включаючи TLSv1.2 та TLSv1.3.

Dovecot-pigeonhole - це розширення для Dovecot, яке додає підтримку мови Sieve. Sieve є мовою скриптів, яка дозволяє користувачам налаштовувати правила автоматичної обробки повідомлень, включаючи фільтрацію, пересилання, сортування та інші дії [5].

ClamAV ma clamsmtpd. ClamAV є антивірусним пакетом, який використовується для виявлення та блокування вірусів та шкідливих програм у електронних повідомленнях [6]. Clamsmtpd є проксі-сервером для ClamAV, який перехоплює вхідні та вихідні повідомлення та перевіряє їх на наявність загроз.

Cyrus-SASL ma Cyrus-SASL-saslauthd - це компоненти, які відповідають за механізми аутентифікації та авторизації користувачів [7]. Вони забезпечують безпеку при обміні даними між поштовим сервером та клієнтськими програмами, такими як поштові клієнти.

OpenSSL є бібліотекою криптографічних функцій та інструментом, що надає широкі можливості для шифрування, дешифрування, створення цифрових підписів, верифікації, генерації ключів та інших криптографічних операцій. Вона використовується для захищеної комунікації та забезпечення безпеки даних[8].

OpenSSL підтримує широкий спектр криптографічних алгоритмів, включаючи асиметричне шифрування (наприклад, RSA, DSA, ECC), симетричне шифрування (наприклад, AES, 3DES), хеш-функції (наприклад, SHA-1, SHA-256, MD5) та багато інших. Openssl надає реалізацію SSL та його наступника TLS. Вона дозволяє зашифровану комунікацію між клієнтом та сервером, забезпечуючи конфіденційність, цілісність та аутентичність даних. OpenSSL підтримує різні версії TLS (наприклад, TLS 1.2, TLS 1.3) та різні криптографічні

алгоритми для шифрування та аутентифікації. OpenSSL надає інструменти для генерації криптографічних ключів різних типів (наприклад, RSA, ECC) та їх керування. Вона підтримує створення самопідписаних сертифікатів та роботу з сертифікатами, що видані сертифікаційними органами. OpenSSL дозволяє створювати цифрові підписи для даних за допомогою приватного ключа та верифікувати їх за допомогою відповідного публічного ключа. Це використовується для перевірки цілісності даних та аутентичності відправника. OpenSSL надає набір інструментів командного рядка, що дозволяють виконувати різні криптографічні операції, такі як шифрування, розшифрування, генерація ключів, підписи та верифікація. Ці інструменти є потужними для тестування та налагодження криптографічних процесів.

OpenSSL широко використовується в сферах інформаційної безпеки, веб-розробки, серверної безпеки та багатьох інших областях. У контексті розробки та налаштування захищеного корпоративного поштового сервера, OpenSSL буде використовуватись для шифрування комунікації між поштовими клієнтами та сервером та між поштовими серверами, генерації та керування TLS сертифікатами що забезпечить безпеку та конфіденційність електронних повідомлень.

Взаємодія компонентів.

Архітектура корпоративного поштового сервера базується на взаємодії різних компонентів для забезпечення повнофункціонального та безпечного сервісу електронної пошти. Взаємодія відбувається за допомогою протоколів та механізмів, таких як:

SMTP - це протокол передачі електронної пошти, який використовується для взаємодії між поштовими серверами під час передачі повідомлень.

IMAP та POP3 - ці протоколи використовуються для доступу користувачів до їхніх поштових скриньок. Вони дозволяють зчитувати та керувати повідомленнями, які зберігаються на сервері.

Архітектура корпоративного поштового сервера передбачає, що поштовий сервер Postfix використовується для прийому та пересилання повідомлень,

Dovecot - для доступу користувачів до поштових скриньок, Dovecot-pigeonhole - для обробки правил автоматичної обробки повідомлень, ClamAV та Clamsmtpd - для виявлення шкідливих програм у повідомленнях, а Cyrus-SASL та Cyrus-SASL-saslauthd - для аутентифікації та авторизації користувачів.

Така архітектура забезпечує надійну та безпечну роботу поштового сервера, здатного виконувати всі необхідні функції електронної пошти в корпоративному середовищі.

Схема обробки поштових повідомлень.

Послідовність обробки пошти виглядає наступним чином:

1. Прийом пошти. На цьому етапі поштовий сервер Postfix приймає вхідні повідомлення від зовнішніх поштових серверів або користувачів. Postfix використовує протокол SMTP для прийому повідомлень і виконує перевірку на достовірність та валідацію вхідних повідомлень. Для забезпечення конфіденційності та цілісності даних, використовується протокол TLS з використанням компонента OpenSSL. Це дозволяє зашифровувати комунікацію між поштовим сервером та клієнтськими програмами та між поштовими серверами.
2. Фільтрація та обробка. Після прийому повідомлення, воно проходить крізь фільтри та обробку. На цьому етапі використовуються різні компоненти, такі як ClamAV, для виявлення та блокування вірусів та шкідливих програм у повідомленнях. Також використовується компонент Dovecot-pigeonhole для застосування правил автоматичної обробки повідомлень.
3. Зберігання повідомлень. Після фільтрації та обробки, повідомлення передається до поштового сервера Dovecot, де вони зберігаються у поштових скриньках користувачів. Dovecot підтримує протоколи IMAP та POP3, що дозволяє користувачам отримувати доступ до своїх повідомлень з різних пристроїв та поштових клієнтів.

4. Аутентифікація та авторизація. Для забезпечення безпеки та контролю доступу до поштових скриньок використовуються компоненти Cyrus-SASL та Cyrus-SASL-saslauthd. Вони відповідають за аутентифікацію користувачів та перевірку їх прав доступу до поштових скриньок.

2 РОЗРОБКА ТА НАЛАШТУВАННЯ ПОШТОВОГО СЕРВЕРА

2.1 Вимоги до поштового сервера.

Перед розробкою та налаштуванням поштового сервера необхідно визначити вимоги до системи.

Загальні вимоги включають:

Надійність: поштовий сервер повинен бути стабільним та надійним, здатним опрацьовувати велику кількість повідомлень та запитів без перебоїв у роботі.

Безпека: система повинна мати високий рівень безпеки, включаючи захист від вірусів, шкідливих програм та несанкціонованого доступу до поштових скриньок.

Швидкодія: поштовий сервер повинен працювати ефективно та забезпечувати швидку доставку повідомлень користувачам.

Масштабованість: система повинна бути здатною масштабуватися для врахування зростаючих потреб користувачів та обсягів поштових повідомлень.

2.2 Розробка поштового сервера.

2.2.1 Встановлення та налаштування FreeBSD.

FreeBSD - це безкоштовна та відкрита операційна система, базована на системі UNIX. Вона розроблена та підтримується FreeBSD Project, який забезпечує вільний доступ до вихідного коду та можливість модифікувати систему.

FreeBSD має репутацію надійної та стабільної операційної системи, яка знаходить широке застосування у різних сферах, таких як серверні платформи, мережеві пристрої, хмарні обчислення та інші. Вона підтримує різні архітектури, включаючи x86, AMD64, ARM та інші, що дає можливість використовувати FreeBSD на різноманітних платформах [2].

FreeBSD має потужний інфраструктурний стек, який включає широкий набір інструментів та сервісів, що полегшують розробку, управління та підтримку системи. Крім того, FreeBSD має добре розвинену документацію та активну спільноту, що надає підтримку та допомогу користувачам.

Операційна система FreeBSD буде використана для розгортання та налаштування захищеного корпоративного поштового сервера. FreeBSD надає потужні інструменти та можливості для реалізації такого сервера, а також забезпечує високу стабільність та безпеку системи.

Першим кроком у розробці поштового сервера є встановлення та налаштування операційної системи FreeBSD.

Встановлення та налаштування операційної системи FreeBSD включає кілька кроків. Ось загальний опис процесу:

1. Завантажуємо останню версію FreeBSD з офіційного веб-сайту проекту https://www.freebsd.org/where/

Обираємо підходящу архітектуру та формат образу.

2. Після завантаження з обраного носія установки розпочинаємо процес встановлення. Вибираємо опції, такі як розміщення, розділи диска, мережеві налаштування та інші параметри. При встановленні вибираємо файлову систему ZFS.

Файлова система ZFS є однією з ключових особливостей операційної системи FreeBSD. ZFS розроблена компанією Sun Microsystems (тепер Oracle) і відрізняється своєю надійністю та гнучкістю [9]. Ось деякі важливі аспекти використання файлової системи ZFS у FreeBSD [10]:

Надійність даних. ZFS забезпечує високий рівень надійності та захисту даних. Вона використовує методи контролю цілісності, копіювання та резервного копіювання, які дозволяють виявляти та виправляти пошкодження даних. Завдяки цьому, ZFS може забезпечити високий рівень безпеки для корпоративної поштової системи.

Гнучкість та масштабованість. ZFS пропонує ряд потужних функцій, які полегшують управління даними та забезпечують гнучкість. Наприклад, ви можете створювати та управляти "пулами" даних, які об'єднують кілька

фізичних пристроїв у єдиний логічний простір. Ви також можете використовувати функції снапшотів та клонування для швидкого створення копій даних.

Керування сховищами. ZFS надає потужні засоби керування сховищами даних. Ви можете змінювати розмір та параметри пулів, додавати або видаляти фізичні пристрої, проводити резервне копіювання та відновлення даних. Засоби керування ZFS спрощують адміністрування та забезпечують високу гнучкість для корпоративного поштового сервера.

Компресія. ZFS має вбудовану підтримку компресії даних, що дозволяє ефективно використовувати дисковий простір. Ви можете вибрати різні алгоритми компресії відповідно до сценарію використання.

Інтеграція з іншими функціями FreeBSD. ZFS щільно інтегрована з іншими компонентами та функціями FreeBSD. Наприклад, ви можете використовувати ZFS разом зі засобами реплікації, які дозволяють створювати резервні копії та відновлювати дані на інших серверах. Вона також добре працює з мережевими пристроями та дозволяє створювати засоби зберігання для мережевих облікових записів.

3. Налаштування мережі. Після завершення установки налаштовуємо мережеві параметри. Це включає призначення IP-адреси, налаштування DNS-серверів та інших мережевих параметрів. Сервер буде налаштований для отриманні пошти домену ***cs.networkacad.net*** з іменем хоста ***mail.cs.networkacad.net***. На рисунку 2.1 показано вміст конфігураційного файлу ***/etc/rc.conf*** на початковому етапі налаштування поштового сервера.

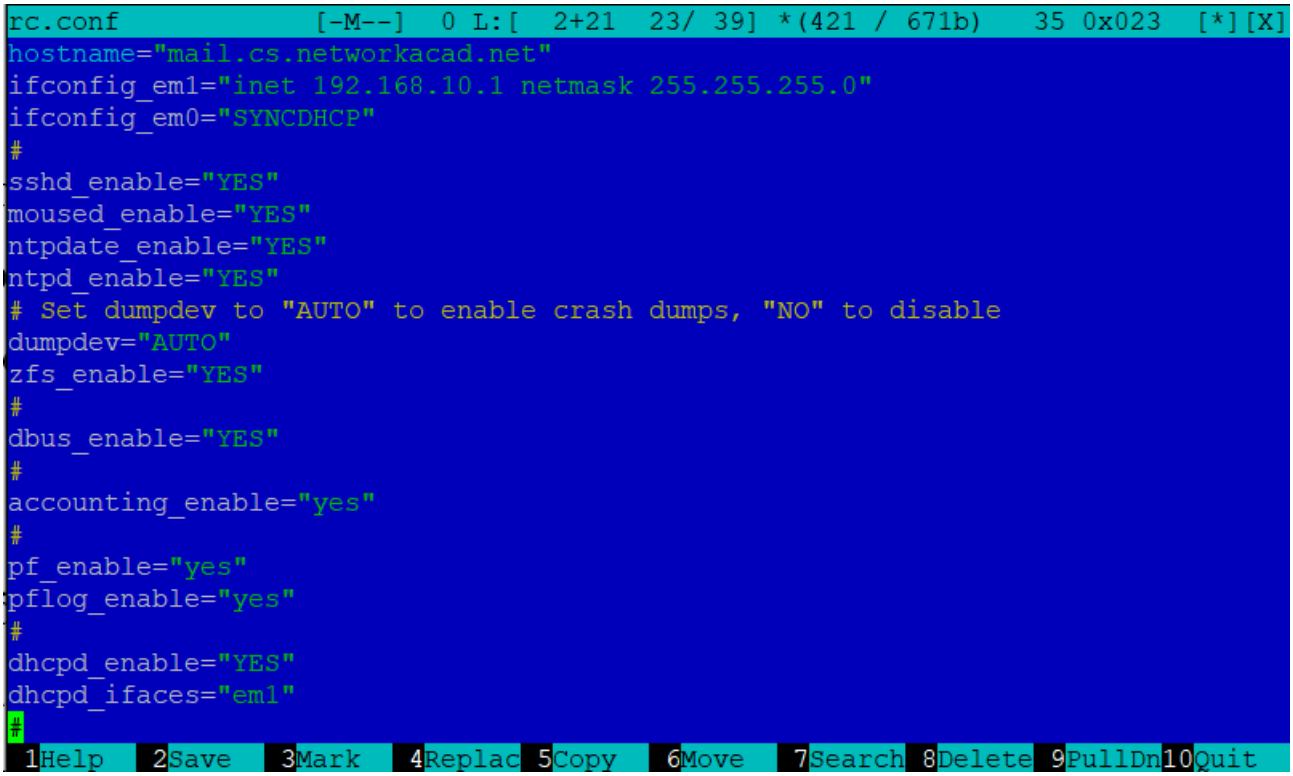


Рисунок 2.1 – Вміст конфігураційного файлу /etc/rc.conf на початковому етапі налаштування

2.2.2 Встановлення та налаштування компонентів поштового сервера.

Після успішного встановлення FreeBSD переходимо до встановлення та налаштування компонентів поштового сервера. Для встановлення необхідних пакетів використаємо пакетний менеджер pkg в FreeBSD.

Встановлення та налаштування Postfix.

Для встановлення та налаштування Postfix як основного MTA для прийому та пересилання повідомлень для домену ***cs.networkacad.net*** слід виконати наступні кроки [11]:

1.Встановлення Postfix:

В терміналі виконуємо наступну команду для встановлення Postfix:

```
#pkg install postfix
```

2.Налаштування основних параметрів Postfix [12]:

Відредагуємо файл конфігурації Postfix ***main.cf***, використовуючи текстовий редактор mcedit. Файл ***main.cf*** містить основні параметри конфігурації для

Postfix. Тут визначаються налаштування, що стосуються мережевих інтерфейсів, доменних імен, обмежень безпеки, параметрів TLS, аутентифікації, маршрутизації повідомлень та інші важливі параметри. Цей файл зазвичай містить багато параметрів, які визначаються у форматі параметр = значення.

Налаштування проведені нижче відносяться до базової конфігурації поштового сервера Postfix і використовуються для налаштування параметрів інформації про хост, мережі та маршрутизацію електронної пошти.

Встановлюємо наступні параметри та значення:

`myhostname = mail.cs.networkacad.net`: Цей параметр вказує ім'я хоста, яке використовується для ідентифікації поштового сервера.

`mydomain = cs.networkacad.net`: Цей параметр вказує доменне ім'я, пов'язане з поштовим сервером.

`myorigin = $myhostname`: Цей параметр вказує вихідне ім'я домену для неканонічних адрес електронної пошти. Значення "$myhostname" вказує, що вихідне ім'я буде відповідати значенню параметра "myhostname" (тобто "mail.cs.networkacad.net" у цьому випадку).

`inet_interfaces = all`: Цей параметр вказує, які мережеві інтерфейси слухають поштовий сервер Postfix. Значення "all" означає, що сервер буде слухати всі доступні мережеві інтерфейси.

`mydestination = $myhostname, localhost.$mydomain, localhost, cs.networkacad.net`: Цей параметр вказує список локальних доставок поштового сервера. Він визначає, які адреси вважатимуться "локальними" для сервера Postfix. У цьому випадку, локальними адресами є "mail.cs.networkacad.net", "localhost.$mydomain", "localhost", "mail.cs.networkacad.net" та "cs.networkacad.net".

`unknown_local_recipient_reject_code = 550`: Цей параметр встановлює код помилки, який сервер Postfix повертає, коли отримує поштове повідомлення для невідомого локального отримувача. Значення "550" означає, що сервер поверне код помилки "550" (Permission denied) для невідомих локальних отримувачів.

`mynetworks_style = host`: Цей параметр вказує тип мережі, з якої приймаються з'єднання. Значення "host" вказує, що приймаються з'єднання лише з локального хоста, а не зовнішніх мереж.

`relay_domains = $mydestination`: Цей параметр вказує список доменів, для яких сервер Postfix буде виконувати релейну маршрутизацію. Значення "$mydestination" вказує, що список релейних доменів буде збігатися зі значенням параметра "mydestination".

Налаштування нижче визначають різні обмеження і поведінку при обробці поштових повідомлень.

Встановлюємо наступні параметри та значення:

`message_size_limit = 30000000`: Цей параметр встановлює максимальний розмір поштового повідомлення, який дозволяється для прийняття. У цьому випадку, максимальний розмір повідомлення обмежений приблизно 30 мегабайтами.

`mailbox_size_limit = 1000000000`: Цей параметр встановлює максимальний розмір поштової скриньки користувача. У цьому випадку, максимальний розмір поштової скриньки обмежений приблизно 1 гігабайтом.

`smtpd_recipient_limit = 50`: Цей параметр встановлює максимальну кількість отримувачів, які дозволено вказувати в команді RCPT TO під час відправки поштового повідомлення через SMTP. У цьому випадку, максимальна кількість отримувачів обмежена 50.

`disable_vrfy_command = yes`: Цей параметр вказує, чи потрібно вимкнути команду VRFY в SMTP. Команда VRFY використовується для перевірки наявності поштової скриньки або перевірки правильності адреси. Значення "yes" вказує, що команда VRFY вимкнена.

`strict_rfc821_envelopes = yes`: Цей параметр вказує, чи потрібно дотримуватись строгих стандартів RFC 821 щодо формату зворотної адреси (MAIL FROM) і адреси отримувача (RCPT TO) в протоколі SMTP. Значення "yes" вказує, що повинні дотримуватись строгих стандартів RFC 821.

`smtpd_helo_required = yes`: Цей параметр вказує, чи потрібно вимагати від клієнта вказувати команду HELO або EHLO під час встановлення з'єднання з

сервером SMTP. Значення "yes" вказує, що клієнт повинен надіслати команду HELO або EHLO.

`smtpd_data_restrictions`: Цей параметр визначає обмеження для обробки даних поштового повідомлення, які надходять під час транзакції DATA в протоколі SMTP. У цьому випадку, вказані наступні обмеження:

- `reject_unauth_pipelining` – відхилити запит, коли клієнт надсилає команди SMTP завчасно, де це заборонено, або коли клієнт надсилає команди SMTP завчасно, не знаючи, що Postfix насправді підтримує конвеєрну передачу команд ESMTP. Це зупиняє пошту від програмного забезпечення масової розсилки, яке неналежним чином використовує конвеєрну передачу команд ESMTP для прискорення доставки..
- `reject_multi_recipient_bounce` – відхилити запит, якщо відправником листа є нульова адреса, а повідомлення має кілька одержувачів листа.
- `permit` – дозволяти все решту.

`smtpd_delay_reject = no`: Цей параметр вказує, чи потрібно відкладати відхилення поштових повідомлень. Значення "no" вказує, що відхилення повідомлень не буде відкладатись, а буде відхилятись негайно.

Ці налаштування встановлюють обмеження та правила для перевірки відправника (sender) поштових повідомлень.

Встановлюємо наступні параметри та значення для `smtpd_sender_restrictions`:

`permit_mynetworks`: Ця директива дозволяє відправникам з локальної мережі (зазначеної в параметрі mynetworks) надсилати поштові повідомлення без обмежень.

`permit_sasl_authenticated`: Ця директива дозволяє аутентифікованим користувачам через SASL надсилати поштові повідомлення без обмежень.

`reject_invalid_hostname`: Ця директива відхиляє повідомлення від відправників, чия хост-система має неправильно встановлене ім'я хоста.

`reject_non_fqdn_hostname`: Ця директива відхиляє повідомлення від відправників, чия хост-система має неправильне доменне ім'я (не є повністю кваліфікованим доменним ім'ям - FQDN).

`reject_non_fqdn_sender`: Ця директива відхиляє повідомлення від відправників, чия адреса електронної пошти не є повністю кваліфікованою (не є FQDN).

`reject_unknown_client`: Ця директива відхиляє повідомлення від клієнтів (відправників), чия ідентифікація клієнта невідома або не пройшла перевірку.

`reject_unknown_sender_domain`: Ця директива відхиляє повідомлення від відправників, чия доменна частина адреси електронної пошти (sender domain) не існує або не вдається її перевірити.

`reject_unknown_hostname`: Ця директива відхиляє повідомлення від відправників, чия хост-система має невідоме або неправильне ім'я хоста.

`reject_rbl_client bl.spamcop.net`: Ця директива відхиляє повідомлення від відправників, чиї IP-адреси знаходяться в списку блокованих IP-адрес RBL на сервері bl.spamcop.net. Цей список використовується для виявлення спаму.

`reject_rbl_client zen.spamhaus.org`: Ця директива відхиляє повідомлення від відправників, чиї IP-адреси знаходяться в списку блокованих IP-адрес на сервері zen.spamhaus.org. Цей список також використовується для виявлення спаму.

3. Налаштування автоматично запуску Postfix:

Щоб налаштувати автоматичний запуск Postfix при завантаженні системи, виконаємо наступні команди:

```
#sysrc postfix_enable="YES"
#sysrc sendmail_enable="NONE"
```

Дані команди виконуються в командному рядку FreeBSD і мають наступні ефекти:

`sysrc postfix_enable="YES"`: Ця команда встановлює значення YES для параметра postfix_enable в системному файлі /etc/rc.conf. Це дозволяє автоматично запускати сервіс Postfix при запуску системи.

`sysrc sendmail_enable="NONE"`: Ця команда встановлює значення NONE для параметра sendmail_enable в системному файлі /etc/rc.conf. Це вимикає автоматичний запуск служби Sendmail, яка є стандартним MTA в FreeBSD. Встановлення значення NONE забезпечує, що Sendmail не буде конфліктувати з Postfix, оскільки ви встановлюєте Postfix як основний MTA.

Окрім конфігураційного файлу ***main.cf*** Postfix має ще один конфігураційний файл ***master.cf***.

Цей файл визначає конфігурацію сервісів Postfix, які слухають на різних мережевих інтерфейсах та портах. Кожен рядок в ***master.cf*** описує окремий сервіс або процес, який виконується в Postfix, такий як SMTP-сервер, LMTP-сервер, або сервіс аутентифікації. Цей файл дозволяє налаштувати різні аспекти обробки повідомлень, включаючи маршрутизацію, фільтрацію, аутентифікацію, захист та інші операції. У ***master.cf*** визначаються параметри, такі як тип сервісу, спосіб з'єднання, команда або програма, яка виконується для обробки повідомлень, та інші налаштування.

Взаємодія між ***main.cf*** і ***master.cf*** дозволяє налаштувати різні аспекти роботи Postfix, зокрема параметри конфігурації, мережеві сервіси, маршрутизацію та обробку повідомлень. Ці два файли співпрацюють для створення повнофункціонального поштового сервера з належними налаштуваннями.

Встановлення та налаштування компонентів Cyrus-SASL та Cyrus-SASL-saslauthd.

Сервери SMTP повинні вирішувати, чи має клієнт SMTP право надсилати пошту до віддалених адресатів чи лише до адресатів, за які відповідає сам сервер. Зазвичай сервери SMTP приймають пошту та пересилають до віддалених адресатів, коли IP-адреса клієнта прописана як дозволена в конфігураційному файлі ***main.cf***.

Клієнтам SMTP за межами дозволених мереж потрібен інший спосіб отримання привілеїв надсилати пошту. Для задоволення цієї потреби Postfix підтримує автентифікацію SASL (RFC 4954). За допомогою цього віддалений

клієнт SMTP може автентифікуватися на сервері SMTP Postfix. Після автентифікації клієнта сервер дозволяє відправку поштових повідомлень.

Postfix сам по собі не реалізує SASL, а використовує існуючі реалізації як будівельні блоки.

Cyrus SASL є бібліотекою та набором інструментів для реалізації механізмів аутентифікації та захисту в різних додатках, таких як поштові сервери, сервери LDAP, FTP-сервери та інші [7]. Вона надає фреймворк для безпечної обробки аутентифікаційних протоколів і шифрування даних.

Основні компоненти Cyrus SASL:

cyrus-sasl - це основний пакет Cyrus SASL, який містить бібліотеки та утиліти, необхідні для роботи з механізмами аутентифікації та захисту SASL.

cyrus-sasl-saslauthd - це додатковий пакет, який містить сервер аутентифікації SASL (saslauthd). Saslauthd - це окремий процес, який використовується для аутентифікації клієнтів на основі SASL. Він надає інтерфейс до різних методів аутентифікації, що дозволяє використовувати різні схеми аутентифікації.

Для встановлення та налаштування компоненти Cyrus SASL та Cyrus SASL Saslauthd на операційній системі FreeBSD, виконаємо наступні кроки:

1.Встановимо Cyrus SASL та Cyrus SASL Saslauthd за допомогою команди:

```
#pkg install cyrus-sasl cyrus-sasl-saslauthd
```

2.Конфігурація Cyrus SASL:

Додамо в конфігураційному файлі ***/usr/local/lib/sasl2/smtpd.conf*** наступний парметр:

```
pwcheck_method: saslauthd
```

3. Налаштування запуску служби Saslauthd:

Щоб налаштувати автоматичний запуск Saslauthd при завантаженні системи, виконаємо наступні команди:

```
#sysrc saslauthd_enable="YES"
#sysrc saslauthd_flags="-a getpwent"
```

Тепер компонента Cyrus SASL та Saslauthd успішно встановлена та налаштована. Їх можна використовувати для аутентифікації.

4.Інтеграція з Postfix.

Щоб інтегрувати компоненти Cyrus SASL та Saslauthd з поштовим сервером Postfix додамо наступні параметри в файлі ***main.cf***:

`smtpd_sasl_auth_enable = yes`: Цей параметр вказує, що SMTP-сервер Postfix дозволяє аутентифікацію SASL для вхідних з'єднань. SASL дозволяє клієнтам (наприклад, поштовим клієнтам) аутентифікуватись на сервері за допомогою різних механізмів аутентифікації, таких як паролі, криптографічні сертифікати тощо.

`smtpd_sasl_security_options = noanonymous`: Цей параметр встановлює обмеження на аутентифікацію SASL. Значення "noanonymous" означає, що анонімна аутентифікація вимкнена, тобто клієнтам потрібно надати валідні облікові дані для аутентифікації.

`smtpd_sasl_local_domain = $mydomain`: Цей параметр встановлює локальний домен для SASL. Змінна $mydomain зазвичай містить доменне ім'я поштового сервера, і вона використовується для локального розрізнення доменів при аутентифікації.

`broken_sasl_auth_clients = yes`***:*** Цей параметр дозволяє обробку "поломаних" клієнтів SASL. Деякі поштові клієнти можуть некоректно реалізувати аутентифікацію SASL. Якщо цей параметр встановлено в "yes", сервер Postfix намагатиметься обробити таких клієнтів, навіть якщо вони надсилають некоректні запити аутентифікації.

`smtpd_sasl_path = /var/spool/postfix/private/auth`: Цей параметр вказує шлях до сокета аутентифікації SASL. Коли клієнт намагається аутентифікуватись, сервер Postfix взаємодіє з SASL через цей сокет для перевірки облікових даних.

Ці налаштування дозволяють використовувати аутентифікацію SASL на сервері Postfix, забезпечуючи безпечний спосіб перевірки облікових даних клієнтів перед надсиланням або отриманням поштових повідомлень.

Встановлення OpenSSL та налаштування підтримки TLS-шифрування трафіку

TLS-шифрування трафіку використовується для захисту комунікації між клієнтами, що перебувають за межами надійних мереж, та нашим поштовим сервером, а також для захисту з'єднань між нашим поштовим сервером та іншими поштовими серверами [8]. Використаємо функції OpenSSL для забезпечення TLS-шифрування трафіку та створення самопідписного довіреного сертифікат X.509.

Щоб встановити OpenSSL та додати підтримку TLS-шифрування трафіку в Postfix та можливості використання STARTTLS, виконаємо наступні кроки:

1.Встановимо OpenSSL за допомогою команди:

```
#pkg install openssl
```

2.Створимо сертифікат та ключ.

Для створення сертифіката та ключа, будучи в каталозі */usr/local/etc/postfix*, потрібно виконати наступну команду:

```
#openssl req -new -nodes -x509 -out smtpd.pem -keyout smtpd.pem -days 3650
```

Ось детальний опис кожного параметра команди:

`req`: Цей параметр вказує на те, що ми виконуємо операцію заявки на сертифікат (Certificate Signing Request, CSR).

`-new`: Цей параметр створює новий запит на сертифікат.

`-nodes`: Цей параметр дозволяє залишити закритий ключ незашифрованим. Це зручно для автоматизації процесу, але потрібно бути обережним з безпекою, оскільки незашифрований закритий ключ може бути потенційно небезпечним.

`-x509`: Цей параметр генерує самопідписний сертифікат X.509 замість CSR.

`-out smtpd.pem`: Цей параметр вказує ім'я вихідного файлу, в якому буде збережений сертифікат. У даному випадку, файл буде називатися "smtpd.pem".

`-keyout smtpd.pem`: Цей параметр вказує ім'я вихідного файлу, в якому буде збережений закритий ключ. В даному випадку, файл також буде називатися "smtpd.pem".

`-days 3650`: Цей параметр встановлює термін дії сертифіката в днях. У даному випадку, термін дії становитиме 3650 днів (10 років).

Ця команда створить самопідписний довірений сертифікат X.509 з використанням OpenSSL. В результаті буде створено файл з назвою "smtpd.pem", який міститиме сертифікат та відповідний приватний ключ.

Використання самопідписних сертифікатів може спричинити попередження про безпеку в деяких програмах або браузерах, оскільки вони не довіряють самопідписаним сертифікатам за замовчуванням. Для надійного захисту комунікації рекомендується отримати довірений сертифікат від відповідного видавця сертифікатів.

3.Включимо підтримку TLS в Postfix [14].

Відредагуємо файл конфігурації Postfix ***main.cf.*** Встановлюємо наступні параметри та значення:

`smtp_use_tls = yes`: Цей параметр вказує, що Postfix має використовувати TLS для вихідних з'єднань з поштовими серверами. TLS забезпечує шифрування з'єднання, що дозволяє безпечно передавати поштові повідомлення між серверами.

`smtpd_use_tls = yes`: Цей параметр вказує, що Postfix має використовувати TLS для вхідних з'єднань з поштовими клієнтами. Це забезпечує шифрування з'єднання при отриманні поштових повідомлень від клієнтів.

`smtpd_tls_auth_only = no`: Цей параметр вказує, що сервер Postfix дозволяє аутентифікацію SASL через TLS, але не обмежує аутентифікацію лише на основі TLS-сертифікатів. Іншими словами, клієнти можуть використовувати ім'я користувача та пароль для аутентифікації під час використання захищеного з'єднання TLS.

`smtp_tls_note_starttls_offer = yes`: Цей параметр вказує, що сервер Postfix буде включати прапорець STARTTLS у відповідь на з'єднання SMTP, щоб повідомити клієнтам про підтримку TLS. Це спонукає клієнтів до встановлення захищеного з'єднання TLS з сервером.

`smtpd_tls_key_file = /usr/local/etc/postfix/smtpd.pem`: Цей параметр вказує шлях до приватного ключа TLS, який використовується сервером Postfix під час встановлення з'єднання TLS з поштовими клієнтами.

`smtpd_tls_cert_file = /usr/local/etc/postfix/smtpd.pem`: Цей параметр вказує шлях до сертифіката TLS, який використовується сервером Postfix при встановленні з'єднання TLS з поштовими клієнтами.

`smtpd_tls_CAfile = /usr/local/etc/postfix/smtpd.pem`: Цей параметр вказує шлях до файлу сертифікату центру сертифікації (CA), який містить публічні ключі випущених сертифікатів. Він використовується для перевірки дійсності сертифікату поштового клієнта під час аутентифікації.

`smtpd_tls_loglevel = 1`: Цей параметр встановлює рівень деталізації ведення журналу для подій TLS на сервері Postfix. Значення "1" вказує базовий рівень журналювання.

`smtpd_tls_received_header = yes`***:*** Цей параметр вказує, що сервер Postfix має додавати заголовок "Received" до отриманих поштових повідомлень, що містить інформацію про захищене з'єднання TLS, через яке повідомлення було отримано.

`smtpd_tls_session_cache_timeout = 3600s`: Цей параметр встановлює таймаут для кешування сесій TLS на сервері Postfix. Значення "3600s" вказує, що сесії будуть кешуватись протягом 3600 секунд (1 години) для покращення продуктивності.

`tls_random_source = dev:/dev/urandom`: Цей параметр вказує, звідки брати випадкові дані для генерації криптографічних ключів TLS. У цьому випадку, випадкові дані беруться з пристрою /dev/urandom в операційній системі.

Відредагуємо файл конфігурації Postfix ***master.cf.*** Встановлюємо наступні параметри та значення:

```
submission inet n       -       n       -       -       smtpd
  -o syslog_name=postfix/submission
  -o smtpd_tls_security_level=encrypt
  -o smtpd_sasl_auth_enable=yes
```

Ці налаштування відносяться до служби submission у поштовому сервері Postfix і визначають параметри для обробки поштових повідомлень, що надходять через цю службу.

Розберемо кожен параметр докладніше:

```
submission inet n       -       n       -       -       smtpd
```

Цей рядок визначає конфігурацію служби `submission. inet` вказує на використання TCP/IP протоколу для мережевого з'єднання.

Решта параметрів `- n - - smtpd` вказують на тип служби і специфічні параметри, пов'язані з цією службою.

`-o syslog_name=postfix/submission`

Цей параметр встановлює ім'я для syslog-журналу (system log) для подій, пов'язаних зі службою submission. В даному випадку, ім'я журналу встановлено як "postfix/submission".

`-o smtpd_tls_security_level=encrypt`

Цей параметр встановлює рівень безпеки для TLS з'єднання у службі submission. Значення "encrypt" означає, що з'єднання повинно бути шифрованим за допомогою TLS.

`-o smtpd_sasl_auth_enable=yes`

Цей параметр вказує, що аутентифікація SASL повинна бути увімкненою для служби submission. SASL дозволяє відправникам аутентифікуватися перед відправленням поштових повідомлень.

Служба submission використовує порт 587 (TCP) для мережевого з'єднання. Цей порт призначений для прийому поштових повідомлень від клієнтів, які надсилають повідомлення через захищене TLS-з'єднання з аутентифікацією.

За замовчуванням, протокол SMTP на порту 25 (TCP) не надає шифрування і передає дані в незахищеному вигляді. Протокол SMTP на порту 25 є стандартним для передачі поштових повідомлень між поштовими серверами.

Однак, в нашому поштовому сервері застосовується шифрування для протоколу SMTP на порту 25 (TCP), якщо обидва поштові сервери, відправник і отримувач, підтримують розширення протоколу STARTTLS. Розширення STARTTLS дозволяє встановити захищене з'єднання TLS під час з'єднання між поштовими серверами.

Коли поштовий сервер, що надсилає повідомлення, виявляє, що сервер-отримувач підтримує STARTTLS, він може ініціювати процес шифрування, запитуючи сервер-отримувач про встановлення захищеного з'єднання. Якщо обидва сервери успішно встановлять захищене з'єднання, подальша передача повідомлення відбуватиметься в захищеному режимі.

Для клієнтської аутентифікації та надсилання повідомлень використовується окремий порт 587 (submission). Використання окремого порту дозволяє розрізняти трафік передачі повідомлень.

```
smtps     inet  n       -       n       -       -       smtpd
  -o syslog_name=postfix/smtps
  -o smtpd_tls_wrappermode=yes
  -o smtpd_sasl_auth_enable=yes
```

Ці налаштування відносяться до служби smtps у поштовому сервері Postfix і визначають параметри для обробки поштових повідомлень, що надходять через цю службу.

Розберемо кожен параметр докладніше:

```
smtps     inet  n       -       n       -       -       smtpd
```

Цей рядок визначає конфігурацію служби `smtps`. `inet` вказує на використання TCP/IP протоколу для мережевого з'єднання.

Решта параметрів `- n - - smtpd` вказують на тип служби і специфічні параметри, пов'язані з цією службою.

```
-o syslog_name=postfix/smtps
```

Цей параметр встановлює ім'я для syslog-журналу (system log) для подій, пов'язаних зі службою smtps. В даному випадку, ім'я журналу встановлено як "postfix/smtps".

```
-o smtpd_tls_wrappermode=yes
```

Цей параметр вказує, що служба smtps має працювати у режимі TLS wrapper. TLS wrapper дозволяє запускати TLS-з'єднання на стандартному порті SMTPS без необхідності використання окремого порту для TLS.

```
-o smtpd_sasl_auth_enable=yes
```

Цей параметр вказує, що аутентифікація SASL повинна бути увімкненою для служби smtps. SASL дозволяє відправникам аутентифікуватися перед відправленням поштових повідомлень.

Служба smtps використовує порт 465 (TCP) для мережевого з'єднання. Цей порт призначений для забезпечення захищеного TLS-з'єднання для передачі поштових повідомлень між поштовими серверами та клієнтами.

Це забезпечує конфіденційність та безпеку передачі поштових повідомлень, оскільки дані, що передаються між клієнтом і сервером, шифруються.

При використанні порту 465 для smtps, клієнти надсилають свої поштові повідомлення до сервера SMTP через захищене TLS-з'єднання на цьому порті. Це забезпечує надійну і захищену передачу пошти з аутентифікацією та шифруванням даних.

Тепер Postfix на FreeBSD має підтримку TLS-шифрування трафіку.

Налаштування DNS.

Налаштування DNS включає різні записи, які дозволяють встановлювати зв'язок між доменним ім'ям та відповідними IP-адресами або конфігурацією електронної пошти [15].

Здійснимо наступні налаштування для домену "cs.networkacad.net":

```
mail.cs.networkacad.net. IN A 77.121.15.237
```

Цей запис вказує на те, що хост "mail.cs.networkacad.net" має IP-адресу 77.121.15.237. На рисунку 2.2 показано вивід команди nslookup для хоста mail.cs.networkacad.net.

```
C:\Users\admin>nslookup mail.cs.networkacad.net 8.8.8.8
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
Name:    mail.cs.networkacad.net
Address:  77.121.15.237


C:\Users\admin>
```

Рисунок 2.2 – Вивід команди nslookup mail.cs.networkacad.net

```
cs.networkacad.net. IN MX 10 mail.cs.networkacad.net.
```

Цей запис вказує на налаштування поштового сервера MX для домену "cs.networkacad.net". Він говорить про те, що всі поштові повідомлення, адресовані до отримувачів електронної пошти в домені "cs.networkacad.net", повинні бути направлені на сервер з ім'ям "mail.cs.networkacad.net" (Рис.2.3.).

Число 10 вказує на пріоритет поштового сервера, де менше значення означає вищий пріоритет.

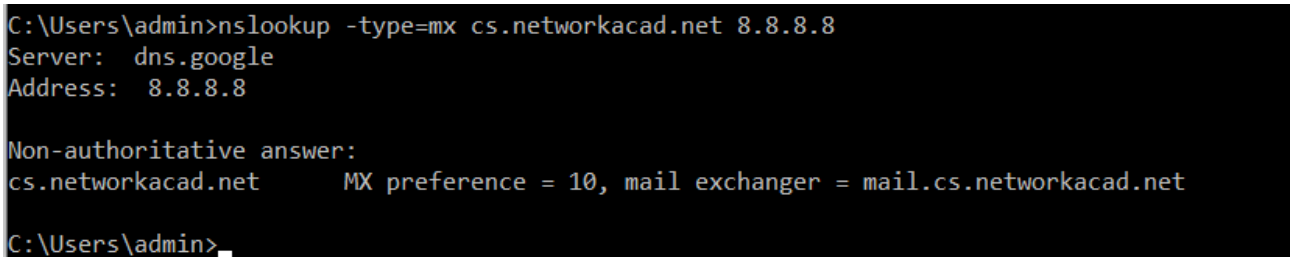


Рисунок 2.3 – Вивід команди nslookup -type=mx cs.networkacad.net

```
cs.networkacad.net. IN TXT "v=spf1 +a +mx -all"
```

Цей запис вказує на налаштування SPF для домену "cs.networkacad.net". SPF є механізмом автентифікації електронної пошти, який допомагає запобігти спаму та підробці [16]. Значення "v=spf1" вказує на версію SPF і початок правила. "+a" дозволяє використовувати IP-адреси, що відповідають основному домену "cs.networkacad.net". "+mx" дозволяє використовувати IP-адреси, які вказані в MX-записах для цього домену. "-all" вказує на жорстку перевірку SPF. Це означає, що всі інші IP-адреси, які не вказані в SPF-записах (вказаних в записах A та MX.), повинні вважатись недійсними. Листи, які надсилаються з домену "cs.networkacad.net". з таких IP-адрес, будуть відхилятися поштовими серверами.

Ці налаштування дозволяють коректно налаштувати електронну пошту для домену "cs.networkacad.net". IP-адреса, поштовий сервер та SPF-правила забезпечують коректну адресацію та доставку електронних листів, пов'язаних з цим доменом.

Встановлення та налаштування Dovecot та dovecot-pigeonhole.

Dovecot є сервером IMAP і POP3, призначеним для обробки електронної пошти [4]. Він є одним з найпопулярніших серверів пошти у світі і використовується для забезпечення доступу користувачів до своїх поштових скриньок через протоколи IMAP та POP3. Dovecot відмінно працює зі стеком поштових програм, таких як Postfix або Exim. Він підтримує широкий набір

функцій, включаючи аутентифікацію, шифрування, фільтрацію та керування поштовими скриньками.

Dovecot-pigeonhole є розширенням для Dovecot, яке надає підтримку для Sieve-фільтрів [5]. Sieve є мовою скриптів для фільтрації пошти на сервері. Dovecot-pigeonhole дозволяє використовувати ці скрипти для автоматичної обробки вхідної пошти.

Щоб встановити та налаштувати Dovecot та Dovecot-pigeonhole на FreeBSD, слід виконати наступні кроки [17]:

1.Встановлення Dovecot та Dovecot-pigeonhole:

В терміналі виконуємо наступну команду:

```
#pkg install dovecot dovecot-pigeonhole
```

2.Налаштування postfix:

Внесемо наступні зміни в конфігураційній файл ***main.cf***.

```
mailbox_command = /usr/local/libexec/dovecot/dovecot-lda -f "$SENDER" -a "$RECIPIENT"
```

Це налаштування відноситься до поштового сервера postfix і визначає команду, яка використовується для доставки поштових повідомлень до поштових скриньок [18].

Цей параметр вказує шлях до виконуваної команди, яка забезпечує доставку поштових повідомлень до поштових скриньок. У цьому випадку використовується команда /usr/local/libexec/dovecot/dovecot-lda.

Опція -f "$SENDER" вказує команді dovecot-lda передачу параметру -f зі значенням $SENDER. $SENDER є змінною, яка містить адресу електронної пошти відправника.

Опція -a "$RECIPIENT" вказує команді dovecot-lda передачу параметру -a зі значенням $RECIPIENT. $RECIPIENT є змінною, яка містить адресу електронної пошти отримувача.

Таким чином, коли Postfix отримує поштове повідомлення для доставки, він виконує команду dovecot-lda і передає їй відправника та отримувача. Команда dovecot-lda відповідає за доставку повідомлення до відповідної поштової скриньки в системі, використовуючи Dovecot.

3.Налаштування конфігураційних файлів Dovecot:

Внесемо наступні зміни в конфігураційній файл ***dovecot.conf***.

`protocols = imap sieve`: Це налаштування визначає протоколи, які підтримуються dovecot. У даному випадку, dovecot підтримує протоколи IMAP і Sieve.

`listen = *`: Це налаштування визначає, на якому інтерфейсі сервер має прослуховувати запити. Зі значенням "*" Dovecot прослуховує всі доступні мережеві інтерфейси. Також можна вказати конкретну IP-адресу, якщо потрібно обмежитись прослуховуванням лише цієї адреси.

`base_dir = /var/run/dovecot/`: Це шлях до базової директорії Dovecot. У цій директорії зберігаються різні файли, необхідні для роботи Dovecot, такі як PID-файл тощо.

`shutdown_clients = yes`: Це налаштування вказує Dovecot закривати з'єднання з клієнтами, якщо він самостійно завершує роботу. Якщо встановити значення "yes", то Dovecot спробує відправити сповіщення про закриття з'єднань клієнтам перед завершенням роботи.

Внесемо наступні зміни в конфігураційній файл ***10-master.conf***.

```
service imap-login {
  inet_listener imap {
    port = 143
  }
  inet_listener imaps {
    port = 993
    ssl = yes
  }
}
```

`inet_listener imap`: Вказує, що IMAP має використовувати порт 143. Це стандартний порт IMAP для незашифрованого з'єднання.

`inet_listener imaps`: Вказує, що IMAPS має використовувати порт 993 і включає шифрування (`ssl = yes`). Порт 993 є стандартним портом IMAP для зашифрованого TLS з'єднання.

```
service auth {
  unix_listener /var/spool/postfix/private/auth {
   mode = 0660
```

```
}
}
```

`unix_listener /var/spool/postfix/private/auth`: Вказує шлях до файлу сокету Unix для комунікації зі службою аутентифікації з поштовим сервером postfix. У цьому прикладі вказано шлях /var/spool/postfix/private/auth, а також задано режим доступу mode = 0660.

Внесемо наступні зміни в конфігураційній файл ***10-ssl.conf***.

`ssl = yes`: Це налаштування вказує, що TLS має бути включений для з'єднань Dovecot. TLS - це протоколи шифрування, які забезпечують безпечну комунікацію між сервером Dovecot і клієнтами.

`ssl_cert = </etc/ssl/dovecot/cert.pem`: Це налаштування вказує шлях до сертифіката TLS, який використовується для шифрування з'єднань Dovecot. У даному випадку, шлях до сертифіката вказаний як /etc/ssl/dovecot/cert.pem. Сертифікат TLS використовується для ідентифікації сервера і шифрування комунікації.

`ssl_key = </etc/ssl/dovecot/key.pem`: Це налаштування вказує шлях до приватного ключа, який використовується для розшифрування з'єднань Dovecot. У даному випадку, шлях до приватного ключа вказаний як /etc/ssl/dovecot/key.pem. Приватний ключ використовується для розшифрування даних, що отримуються від клієнтів.

`ssl_min_protocol = TLSv1.2`: Це налаштування визначає мінімальний рівень протоколу TLS, який приймається для з'єднань Dovecot. У даному випадку, вказано TLSv1.2, що означає, що Dovecot буде приймати тільки з'єднання, які використовують TLS версії 1.2 або вище. TLSv1.2 є одним з найбільш безпечних протоколів шифрування і рекомендується використовувати його для забезпечення безпеки з'єднань.

Внесемо наступні зміни в конфігураційній файл ***15-lda.conf***.

```
protocol lda {
mail_plugins = $mail_plugins sieve
}
```

`mail_plugins = $mail_plugins sieve`: Ця директива вказує, які плагіни повинні бути завантажені для агента LDA. В даному випадку, вказано Sieve, що означає, що плагін Sieve повинен бути завантажений.

Плагін Sieve - це мова скриптів, яка використовується для фільтрації та обробки вхідної пошти на сервері. Він дозволяє налаштовувати правила фільтрації, пересилання, розподілу та інші дії над вхідними повідомленнями.

За допомогою плагіна Sieve, можна налаштовувати правила фільтрації пошти для кожного користувача окремо, використовуючи файл скрипту Sieve. Ці скрипти можуть виконуватися автоматично для обробки вхідних повідомлень.

Внесемо наступні зміни в конфігураційній файл ***90-plugin.conf***

```
plugin {
  sieve = file:~/sieve;active=~/.dovecot.sieve
}
```

`sieve`: Ця директива вказує, як використовувати плагін Sieve і вказує шляхи до файлів скриптів Sieve.

`file:~/sieve`: Це вказує, що скрипти Sieve знаходяться у каталозі sieve в домашньому каталозі користувача. Тут ~ позначає домашній каталог користувача.

`active=~/.dovecot.sieve`: Це вказує, що активний скрипт Sieve знаходиться у файлі ~/.dovecot.sieve в домашньому каталозі користувача. Файл .dovecot.sieve - це файл, в якому вказуються правила фільтрації, які будуть застосовуватися до вхідних повідомлень.

3. Налаштування автоматично запуску Dovecot:

Щоб налаштувати автоматичний запуск Dovecot при завантаженні системи, виконаємо наступну команду:

```
#sysrc dovecot_enable="YES"
```

4. Створення сертифіката TLS та приватного ключа.

Файл ***dovecot-openssl.cnf*** та скрипт ***mkcert.sh*** використовуються для створення та налаштування TLS сертифікатів для сервера Dovecot.

dovecot-openssl.cnf: Цей файл є конфігураційним файлом сумісним з OpenSSL, який використовується для налаштування параметрів генерації сертифікатів. У цьому файлі визначаються різні налаштування, такі як довжина

ключа, параметри шифрування, розширення сертифікату та інші. Конфігураційний файл dovecot-openssl.cnf може бути використаний для налаштування генерації сертифікату TLS для Dovecot за допомогою утиліти OpenSSL.

mkcert.sh: Це скрипт, який використовує конфігураційний файл dovecot-openssl.cnf та утиліту OpenSSL для генерації TLS сертифікатів для сервера Dovecot. Скрипт mkcert.sh автоматизує процес створення сертифікатів, виконуючи необхідні команди OpenSSL з використанням вказаного конфігураційного файлу. В результаті виконання скрипту mkcert.sh будуть створені сертифікати та приватні ключі, які можна використовувати для налаштування TLS з'єднань для Dovecot.

Скрипт ***mkcert.sh*** разом з конфігураційним файлом ***dovecot-openssl.cnf*** використовується, щоб згенерувати необхідний сертифікат ***cert.pem*** та приватний ключ ***key.pem*** для сервера Dovecot, які використовуються для налаштування безпечного з'єднання TLS.

Встановлення та налаштування ClamAV та Clamsmtpd.

ClamAV - це відкрите програмне забезпечення, яке надає антивірусний захист. Він призначений для сканування файлів на віруси та інші види шкідливого програмного забезпечення. ClamAV використовується як самостійний антивірусний сканер, а також може бути інтегрований у різноманітне програмне забезпечення, таке як поштові сервери, файлові сервери тощо [6].

Clamsmtpd - це простий проксі-сервер для електронної пошти, який інтегрується з ClamAV для сканування електронних листів на віруси та інше шкідливе програмне забезпечення перед їх доставкою.

Інтеграція ClamAV та Clamsmtpd з Postfix дозволить автоматично сканувати електронну пошту, що проходить через поштовий сервер Postfix.

Щоб встановити та налаштувати ClamAV та Clamsmtpd на FreeBSD, слід виконати наступні кроки:

1.Встановлення ClamAV та Clamsmtpd:

В терміналі виконуємо наступну команду:

```
#pkg install clamav clamsmtp
```

2.Налаштування Postfix для підтримки Clamsmtpd та ClamAV:

Внесемо наступні зміни в конфігураційній файл ***main.cf***.

`content_filter = scan:127.0.0.1:10025`: Цей параметр встановлює фільтр вмісту для поштових повідомлень. У даному випадку, вказано, що поштові повідомлення будуть направлятись на адресу 127.0.0.1:10025 для подальшої обробки фільтром з назвою "scan". Це дозволяє передавати повідомлення до зовнішнього програмного забезпечення або фільтра, яке буде проводити аналіз та сканування вмісту повідомлення з метою виявлення вірусів.

`receive_override_options = no_address_mappings`: Цей параметр вказує не виконувати перетворення адрес доставки під час обробки вхідних повідомлень. Значення "no_address_mappings" означає, що адреса в заголовку повідомлення буде використовуватись без змін.

Внесемо наступні зміни в конфігураційній файл ***master.cf***.

```
scan      unix  -       -       n       -       16      smtp
        -o smtp_send_xforward_command=yes
```

`scan unix - - n - 16 smtp`: Це означає, що використовується Unix-доменний сокет для отримання вхідних поштових повідомлень від сканера антивірусного програмного забезпечення. З'єднання з сканером буде встановлено через локальний Unix-доменний сокет.

`-o smtp_send_xforward_command=yes` вказує, що поштовий сервер Postfix буде надсилати команду XFORWARD під час відправки повідомлень іншим серверам. XFORWARD є розширенням протоколу SMTP, яке дозволяє передавати додаткові інформаційні поля про клієнта і з'єднання між поштовими серверами. Ця інформація може бути корисною для обліку, журналювання або додаткової обробки повідомлень.

```
127.0.0.1:10026 inet  n -       n       -       16      smtpd
        -o content_filter=
        -o              receive_override_options              =
no_unknown_recipient_checks, no_header_body_checks
```

```
            -o smtpd_helo_restrictions=
            -o smtpd_client_restrictions=
            -o smtpd_sender_restrictions=
            -o      smtpd_recipient_restrictions=permit_mynetworks,
reject
            -o mynetworks_style=host
            -o smtpd_authorized_xforward_hosts=127.0.0.0/8
```

`127.0.0.1:10026 inet n - n - 16 smtpd`: Це налаштування стосується прийому просканованих ClamAV поштових повідомлень, які надсилаються на локальний інтерфейс (IP-адреса 127.0.0.1) на порт 10026. Далі наведено детальний опис кожного параметра:

`-o content_filter=`: Цей параметр не вказує жодного фільтра контенту. Повідомлення будуть оброблятися безпосередньо сервером SMTP.

`-o receive_override_options = no_unknown_recipient_checks, no_header_body_checks`: Цей параметр вказує, що жодна перевірка невідомих отримувачів або перевірка заголовків в тілі повідомлення не буде виконуватися. Повідомлення будуть прийматися безпосередньо, без додаткових перевірок.

`-o smtpd_helo_restrictions=`: Цей параметр не має визначених обмежень привітання (HELO/EHLO). Клієнти можуть надсилати привітання без будь-яких обмежень.

`-o smtpd_client_restrictions=`: Цей параметр не має визначених обмежень для клієнтів. Клієнти можуть надсилати повідомлення без будь-яких обмежень.

`-o smtpd_sender_restrictions=`: Цей параметр не має визначених обмежень для відправників. Відправники можуть надсилати повідомлення без будь-яких обмежень.

`-o smtpd_recipient_restrictions=permit_mynetworks,reject`: Цей параметр встановлює обмеження для отримувачів повідомлень. Він дозволяє приймати повідомлення, якщо вони надходять зі змінної mynetworks і відхиляє повідомлення в інших випадках.

`-o mynetworks_style=host`: Цей параметр вказує, що mynetworks використовується для визначення локальної мережі на основі імен хостів (IP-адреси) замість підмереж.

`-o smtpd_authorized_xforward_hosts=127.0.0.0/8`: Цей параметр вказує список дозволених хостів для використання XFORWARD. В даному випадку дозволені хости з підмережі 127.0.0.0/8 (локальний інтерфейс).

3.Налаштування конфігураційних файлів ClamAV ат Clamsmtpd:

Внесемо наступні зміни в конфігураційній файл ***clamd.conf***.

```
LogFile /var/log/clamav/clamd.log
```

Цей параметр визначає шлях до файлу журналу ClamAV, куди будуть записуватися повідомлення про роботу антивірусного сканера.

```
PidFile /var/run/clamav/clamd.pid
```

Цей параметр вказує шлях до файлу, в якому зберігається ідентифікатор процесу (PID) антивірусного сканера ClamAV. Цей файл використовується для контролю за процесом та управління ним.

```
DatabaseDirectory /var/db/clamav
```

Цей параметр визначає шлях до каталогу, де знаходиться база даних ClamAV. У цьому каталозі зберігаються вірусні бази даних та інші файли, необхідні для роботи ClamAV.

```
LocalSocket /var/run/clamav/clamd.sock
```

Цей параметр встановлює шлях до локального сокету (Unix-сокету), через який здійснюється зв'язок між ClamAV та іншими програмами чи службами.

```
FixStaleSocket yes
```

Цей параметр вказує, що при запуску антивірусного сканера ClamAV буде автоматично виправлено можливі проблеми з локальним сокетом, якщо такі виникнуть.

```
User clamav
```

Цей параметр вказує, під яким користувачем запускати процес антивірусного сканера ClamAV. У даному випадку, вказано, що процес повинен запускатися з правами користувача "clamav".

```
ScanMail yes
```

Цей параметр вказує, чи слід сканувати електронну пошту, яка проходить через антивірусний сканер ClamAV.

Внесемо наступні зміни в конфігураційній файл ***freshclam.conf***.

```
DatabaseDirectory /var/db/clamav
```

Цей параметр визначає шлях до каталогу, де знаходиться база даних ClamAV. Це місце, де freshclam (клієнт оновлення баз даних ClamAV) зберігає оновлені вірусні бази даних та інші файли.

```
UpdateLogFile /var/log/clamav/freshclam.log
```

Цей параметр вказує шлях до файлу журналу оновлення ClamAV. У цьому файлі будуть зберігатися повідомлення про стан та результати процесу оновлення баз даних.

```
PidFile /var/run/clamav/freshclam.pid
```

Цей параметр визначає шлях до файлу, в якому зберігається ідентифікатор процесу (PID) freshclam. Цей файл використовується для контролю за процесом та управління ним.

```
DatabaseOwner clamav
```

Цей параметр вказує користувача, власника баз даних ClamAV. У даному випадку, бази даних будуть належати користувачеві "clamav".

```
DatabaseMirror database.clamav.net
```

Цей параметр вказує URL-адресу сервера-дзеркала, з якого будуть завантажуватися оновлені вірусні бази даних ClamAV.

```
NotifyClamd /usr/local/etc/clamd.conf
```

Цей параметр вказує шлях до файлу конфігурації clamd.conf, який використовується для сповіщення clamd про зміни у базах даних ClamAV. Це дозволяє clamd автоматично завантажувати та оновлювати бази даних без перезавантаження.

Внесемо наступні зміни в конфігураційній файл ***clamsmtpd.conf***.

```
OutAddress: 10026
```

Вказує порт, на який clamsmtpd буде відправляти електронну пошту після сканування. Таким чином, після того, як clamsmtpd сканує електронну пошту за допомогою ClamAV, він передає її на цей порт для подальшої обробки або доставки. Тобто повертає в Postfix.

```
Listen: 127.0.0.1:10025
```

Цей параметр вказує IP-адресу та порт, на якому Clamsmtpd буде приймати з'єднання від Postfix

```
ClamAddress: /var/run/clamav/clamd.sock
```

Цей параметр вказує шлях до локального сокету (Unix-сокету), через який Clamsmtpd здійснює зв'язок з антивірусним сканером ClamAV (clamd).

```
TempDirectory: /tmp
```

Цей параметр вказує шлях до тимчасового каталогу, який використовується Clamsmtpd для зберігання тимчасових файлів, пов'язаних з обробкою електронної пошти.

```
User: clamav
```

Цей параметр вказує, під яким користувачем запускати Clamsmtpd.

4. Налаштування автоматично запуску Clamd, Freshclam та Clamsmtpd:

Щоб налаштувати автоматичний запуск Clamd, Freshclam та Clamsmtpd при завантаженні системи, виконаємо наступні команди:

```
#sysrc  clamsmtpd_enable="YES"
#sysrc clamav_freshclam_enable="YES"
#sysrc  clamav_clamd_enable="YES"
```

Після успішної інтеграції ClamAV та Clamsmtpd з Postfix поштовий сервер буде автоматично сканувати електронну пошту на наявність шкідливого програмного забезпечення перед її доставкою.

2.3. Запуск служб поштового сервера.

Запуск служб та перевірка коректного старту є важливими кроками під час розробки та налаштування поштового сервера.

Перевіримо чи встановлено атоматичний запуск відповідних служб в конфігураційному файлі /etc/rc.conf (Рис. 2.4.).

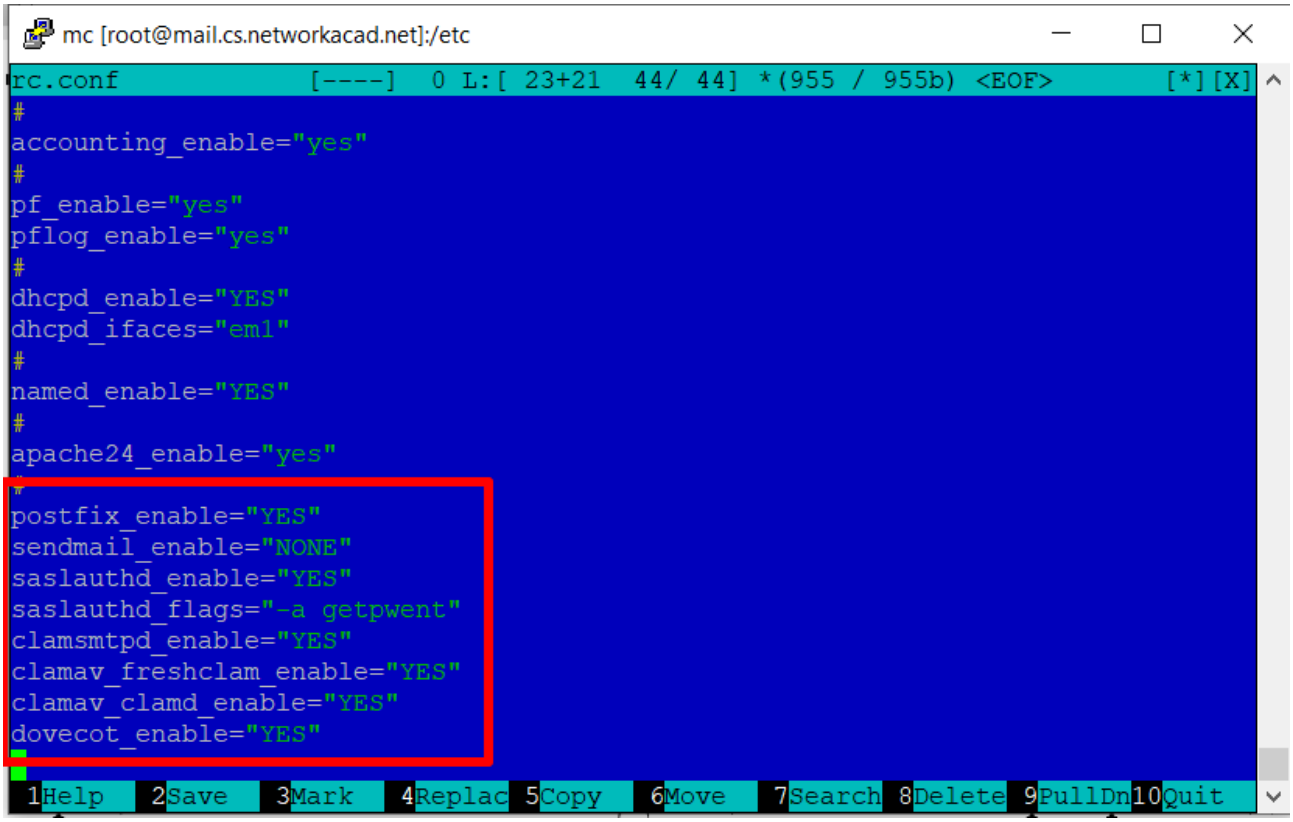


Рисунок 2.4 – Вміст конфігураційного файлу /etc/rc.conf після встановлення та налаштування відповідних служб

Проведемо поетапний запуск відповідних служб. Перевіримо чи служби успішно стартують. Для цього проаналізуємо відповідні лог-файли, де зберігається інформація про стан та події служб.

1.Запуск Postfix:

```
#service postfix start
```

В файлі журналу /var/log/maillog можна побачити що дана служба запустилась коректно (Рис.2.5).

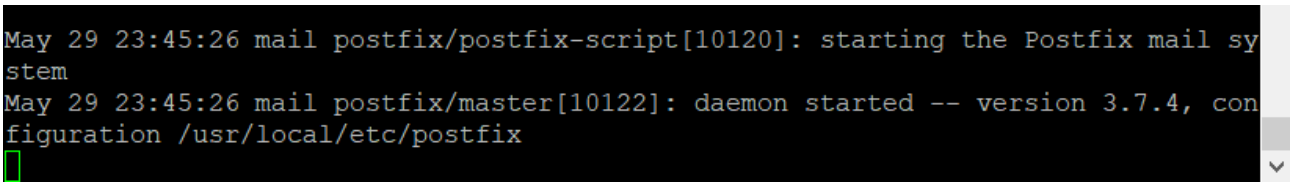


Рисунок 2.5 – Підтвердження коректного запуску postfix

2.Запуск Dovecot:

```
#service dovecot start
```

В файлі журналу /var/log/dovecot.log можна побачити що дана служба запустилась коректно з підтримкою протоколу imap та sieve (Рис.2.6).

Рисунок 2.6 – Підтвердження коректного запуску dovecot

3.Запуск Saslauthd:

```
# service saslauthd start
```

В файлі журналу /var/log/auth.log можна побачити що дана служба запустилась коректно (Рис.2.7).

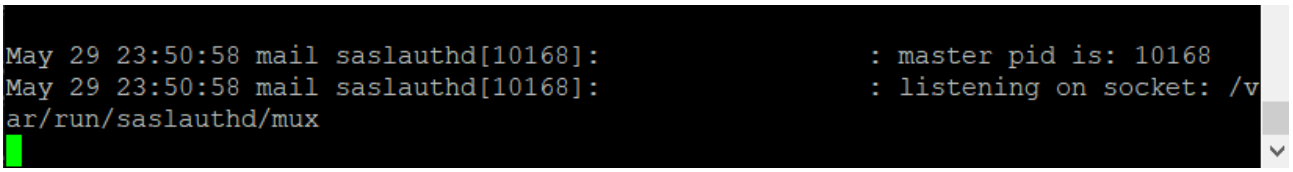


Рисунок 2.7 – Підтвердження коректного запуску saslauthd

4.Запуск Clamsmtpd та Clamd:

```
# service clamsmtpd start
# service clamav_clamd start
```

В файлі журналу /var/log/clamav/clamav.log можна побачити що дані служби запустились коректно (Рис.2.8).

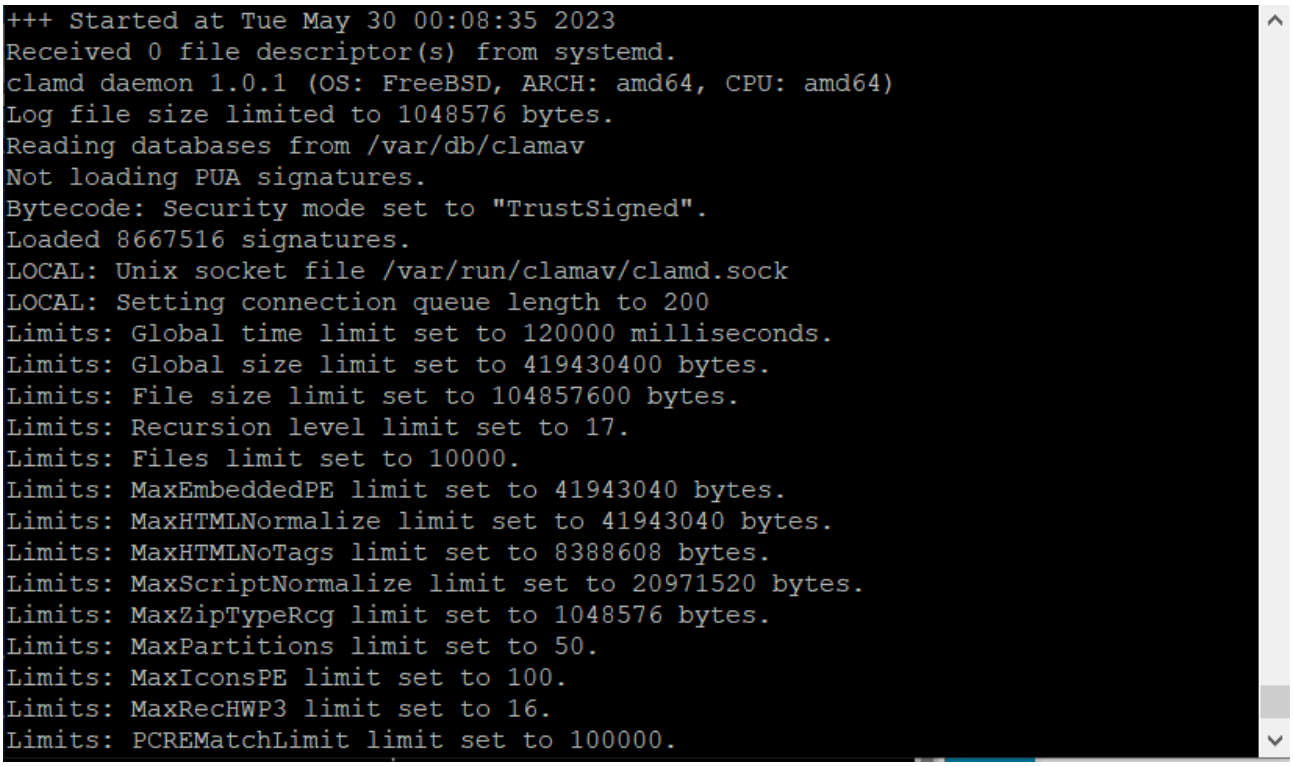


Рисунок 2.8 – Підтвердження коректного запуску clamsmtpd та clamd

5.Запуск Freshclam (оновлення вірусних баз ClamAV):

```
# service clamav_freshclam start
```

В файлі журналу /var/log/clamav/freshclam.log можна побачити що дана служба запустилась коректно (Рис.2.9).

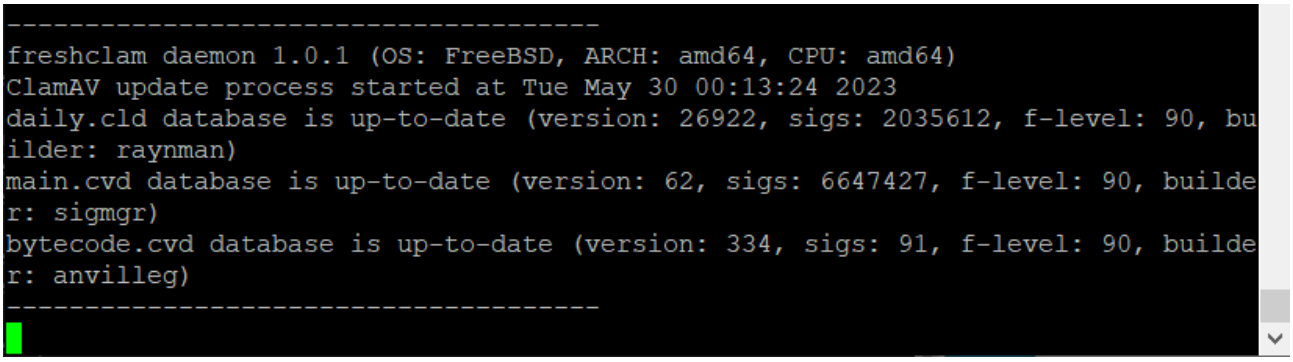


Рисунок 2.9 – Підтвердження коректного запуску freshclam

Перевіримо за допомогою команди ***netstst –an*** чи служби слухають відповідні їм порти. Як можна побачити з виводу даної команди служби працюють та очікують з’єднання по відповідних портах (Рис.2.10).

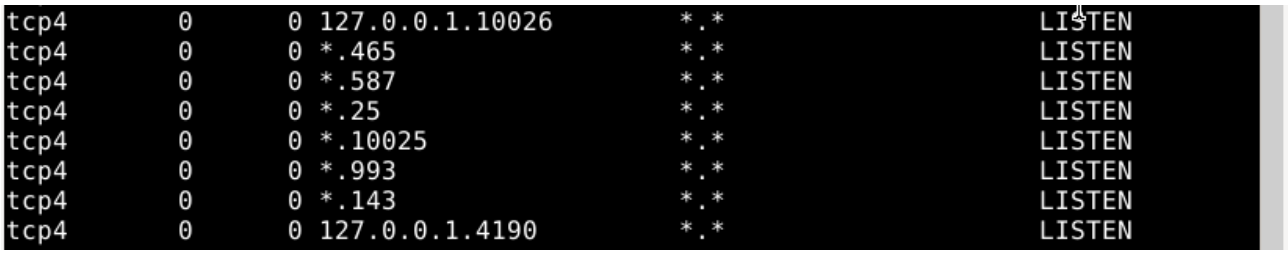


Рисунок 2.10 – Вивід команди netstst –an

3. ТЕСТУВАННЯ ТА ОЦІНКА БЕЗПЕКИ.

Після успішного налаштування поштового сервера необхідно провести тестування щоб переконатись у його працездатності та ефективності. Також потрібно провести оцінку безпеки поштового сервера. Оцінка безпеки покаже чи сервер належним чином захищений від потенційних загроз та вразливостей.

3.1 Тестування функціональності.

Перед введенням корпоративного поштового сервера в експлуатацію необхідно провести тестування його функціональності. Тестування допоможе переконатися, що поштовий сервер відповідає вимогам і працездатний у різних сценаріях використання.

3.1.1 Тестування локальної доставки пошти.

Щоб перевірити, чи можливо успішно відправити та отримати повідомлення в межах поштового сервера та доставити їх до відповідних скриньок, давайте розглянемо процес відправки та отримання листів між двома поштовими скриньками pavlo@cs.networkacad.net і mailadmin@cs.networkacad.net, використовуючи Mozilla Thunderbird версії 102.11.2 на операційній системі Windows 10 для відправки та Microsoft Outlook 2016 на операційній системі Windows 11 для отримання.

1.Налаштування поштового клієнта Mozilla Thunderbird.

Параметри налаштування поштового облікового запису pavlo@cs.networkacad.net показано на рисунку 3.1-3.3

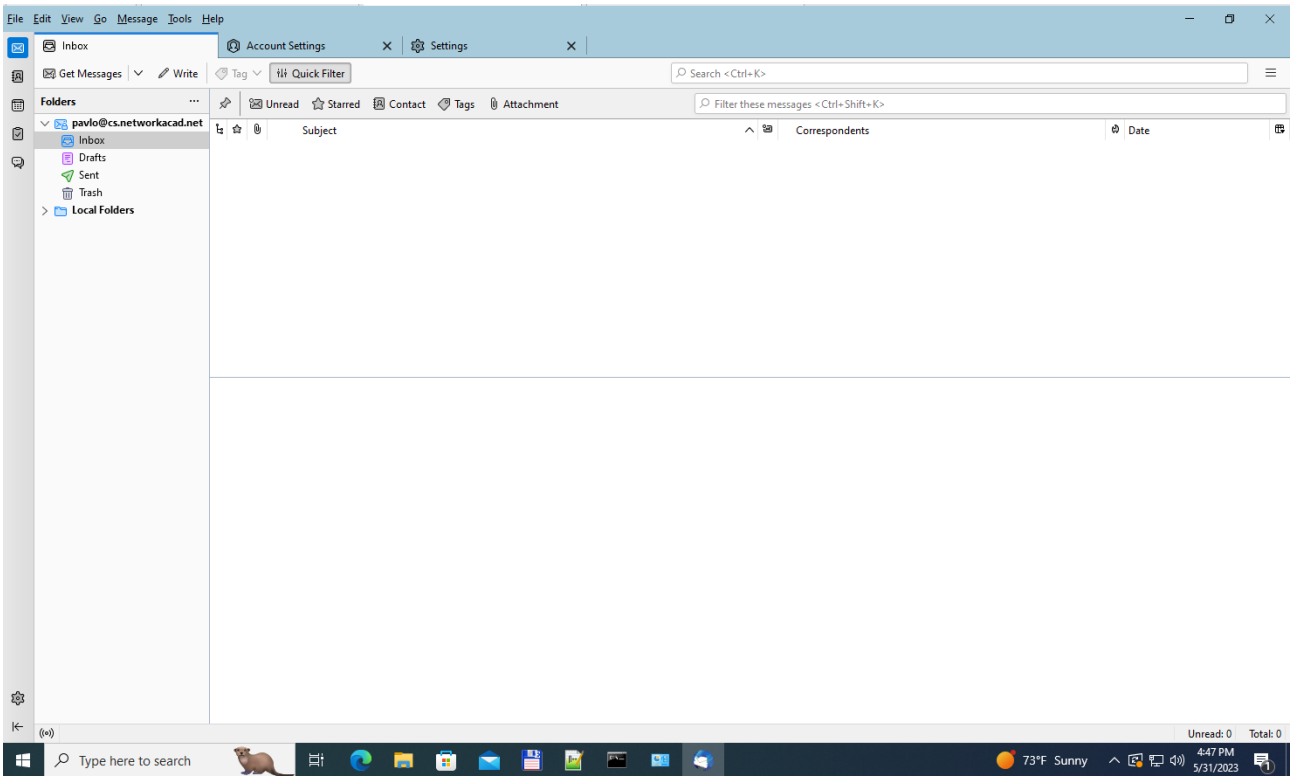


Рисунок 3.1 – Налаштований поштовий обліковий запис pavlo@cs.networkacad.net

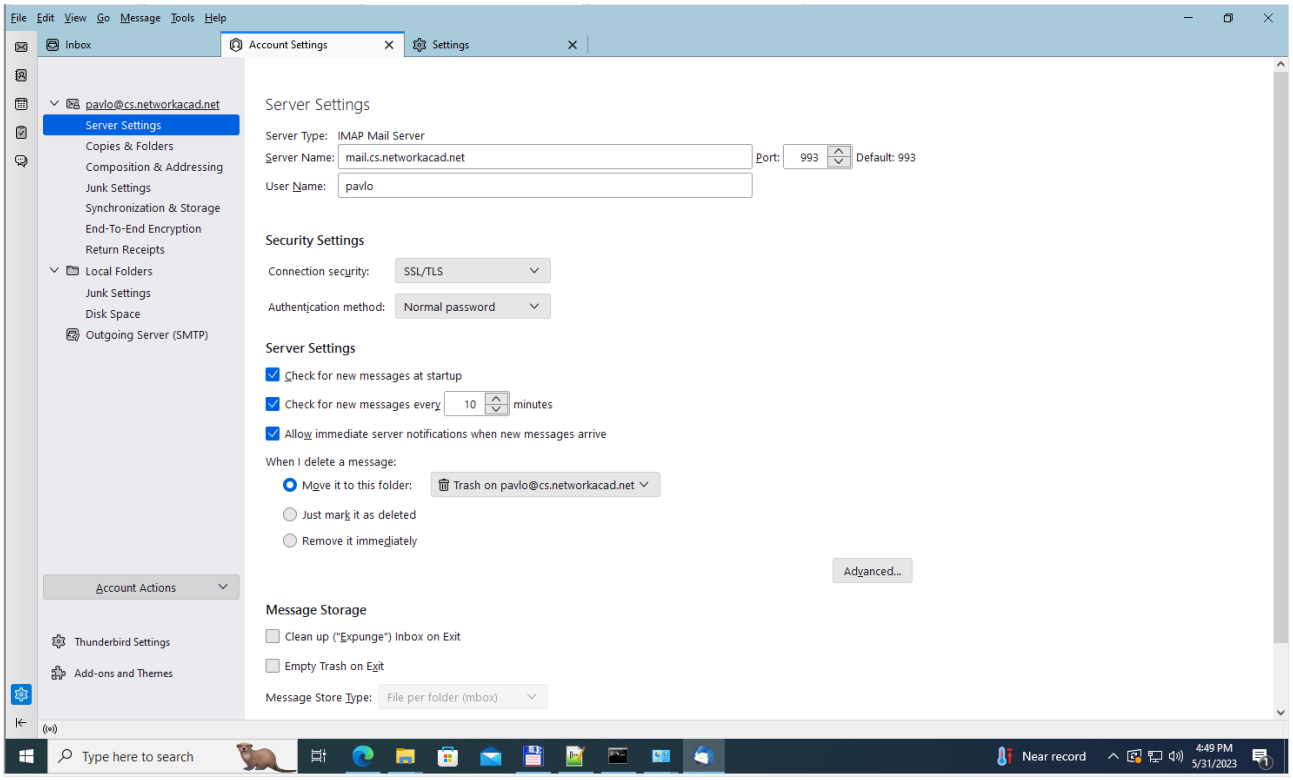


Рисунок 3.2 – Параметри налаштування сервера вхідної пошти для pavlo@cs.networkacad.net

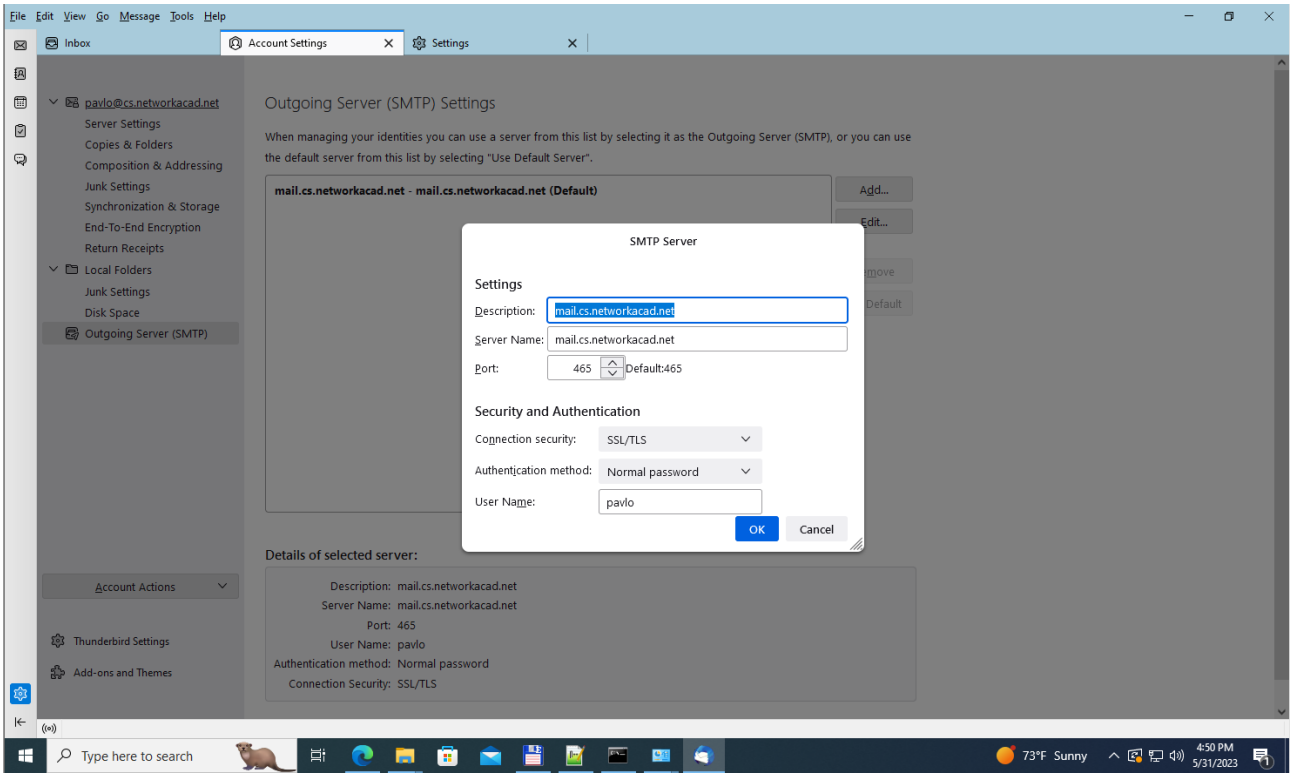


Рисунок 3.3 – Параметри налаштування сервера вихідної пошти для pavlo@cs.networkacad.net

2. Поштовий клієнт при підключення до поштового сервера завантажив відповідні сертифікати для отримання (Рис. 3.4) та надсилання (Рис. 3.5) поштових повідомлень по шифрованому каналу.

Certificate

mail.cs.networkacad.net

Subject Name

Country	UA
State/Province	Ternopil
Locality	Ternopil
Organization	TNTU
Organizational Unit	CS
Common Name	mail.cs.networkacad.net
Email Address	admin@cs.networkacad.net

Issuer Name

Country	UA
State/Province	Ternopil
Locality	Ternopil
Organization	TNTU
Organizational Unit	CS
Common Name	mail.cs.networkacad.net
Email Address	admin@cs.networkacad.net

Validity

Not Before	Sat, 27 May 2023 10:20:46 GMT
Not After	Tue, 24 May 2033 10:20:46 GMT

Public Key Info

Algorithm	RSA
Key Size	2048
Exponent	65537
Modulus	D6:B3:03:49:63:4D:F8:63:27:14:3F:0A:65:90:8E:77:CC:EF:C2:8A:9B:08:C8:F8:0B:08:F...

Miscellaneous

Serial Number	02:47:E0:61:BA:94:60:CA:75:41:6E:8B:CE:F9:22:AC:78:A0:90:77
Signature Algorithm	SHA-256 with RSA Encryption
Version	3
Download	PEM (cert) PEM (chain)

Fingerprints

SHA-256	D9:D7:B2:F8:CC:4B:C3:CD:70:B7:A1:91:FC:5E:E7:F5:6E:96:25:92:31:D4:A9:1C:5B:80...
SHA-1	CB:F9:4D:4A:B2:3B:42:28:BE:AA:1D:04:44:85:62:16:02:76:84:25

Рисунок 3.4 – Сертифікат, який використовується поштовим клієнтом при встановленні з'єднання TLS з сервером IMAP для отримання пошти.

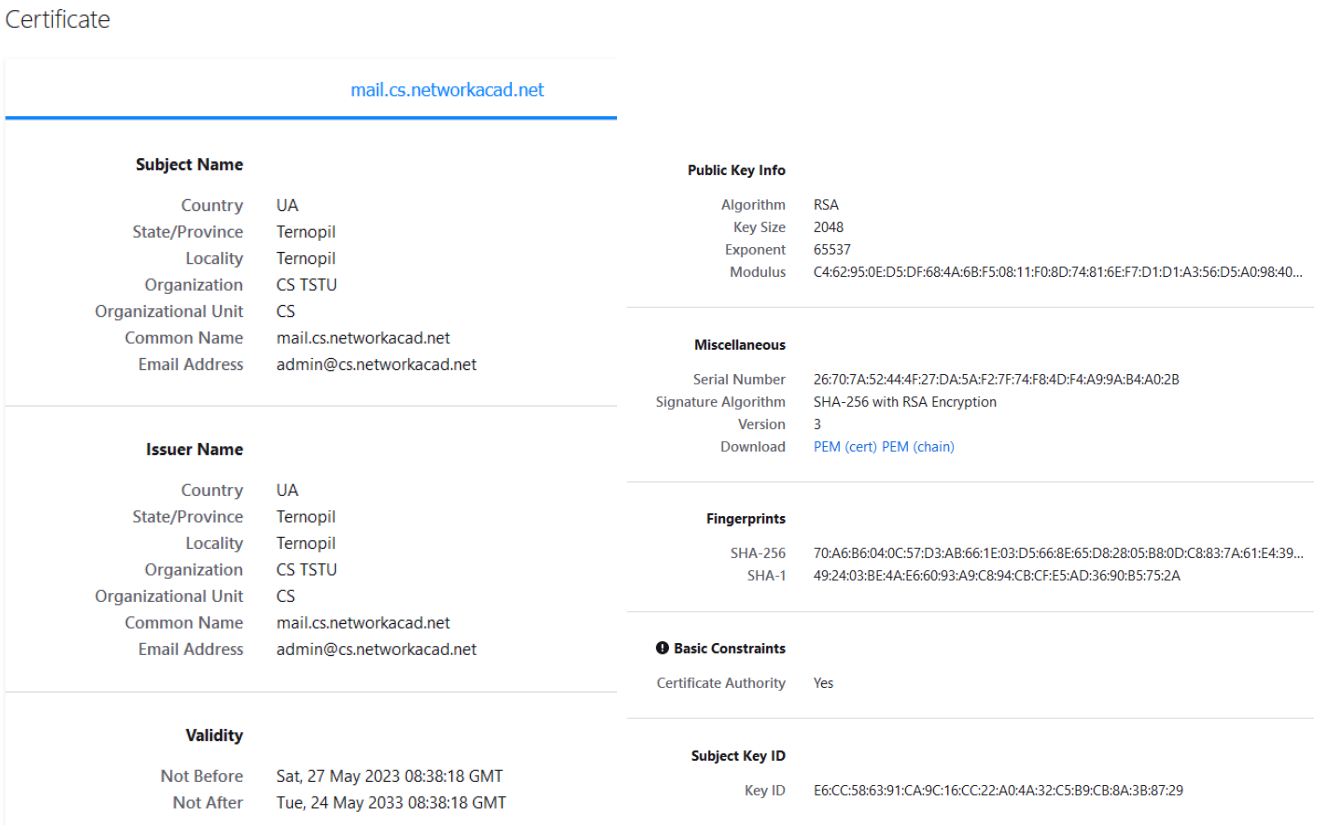


Рисунок 3.5 – Сертифікат, який використовується поштовим клієнтом при встановленні з'єднання TLS з сервером Postfix для надсилання пошти.

3.Налаштування поштового клієнта Microsoft Outlook.

Параметри налаштування поштового облікового запису mailadmin@cs.networkacad.net показано на рисунку 3.6-3.7

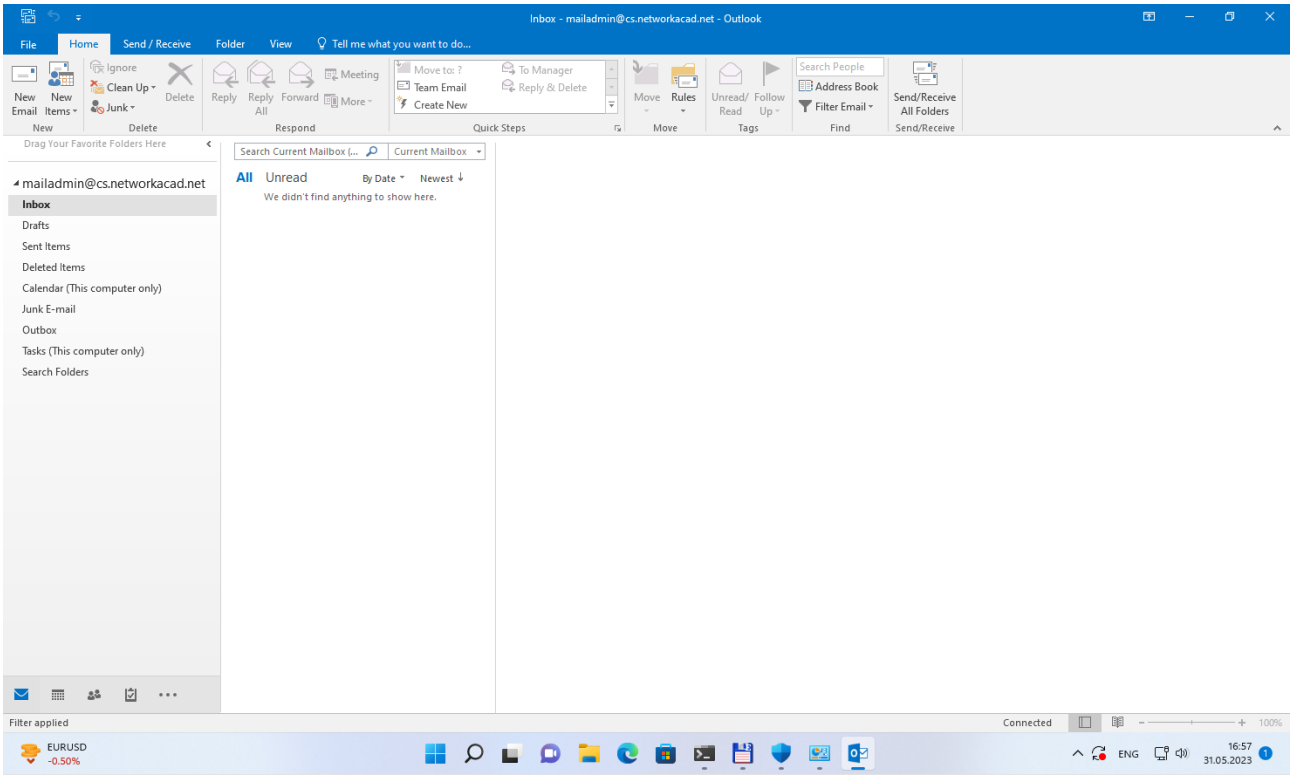


Рисунок 3.6 – Налаштований поштовий обліковий запис mailadmin@cs.networkacad.net

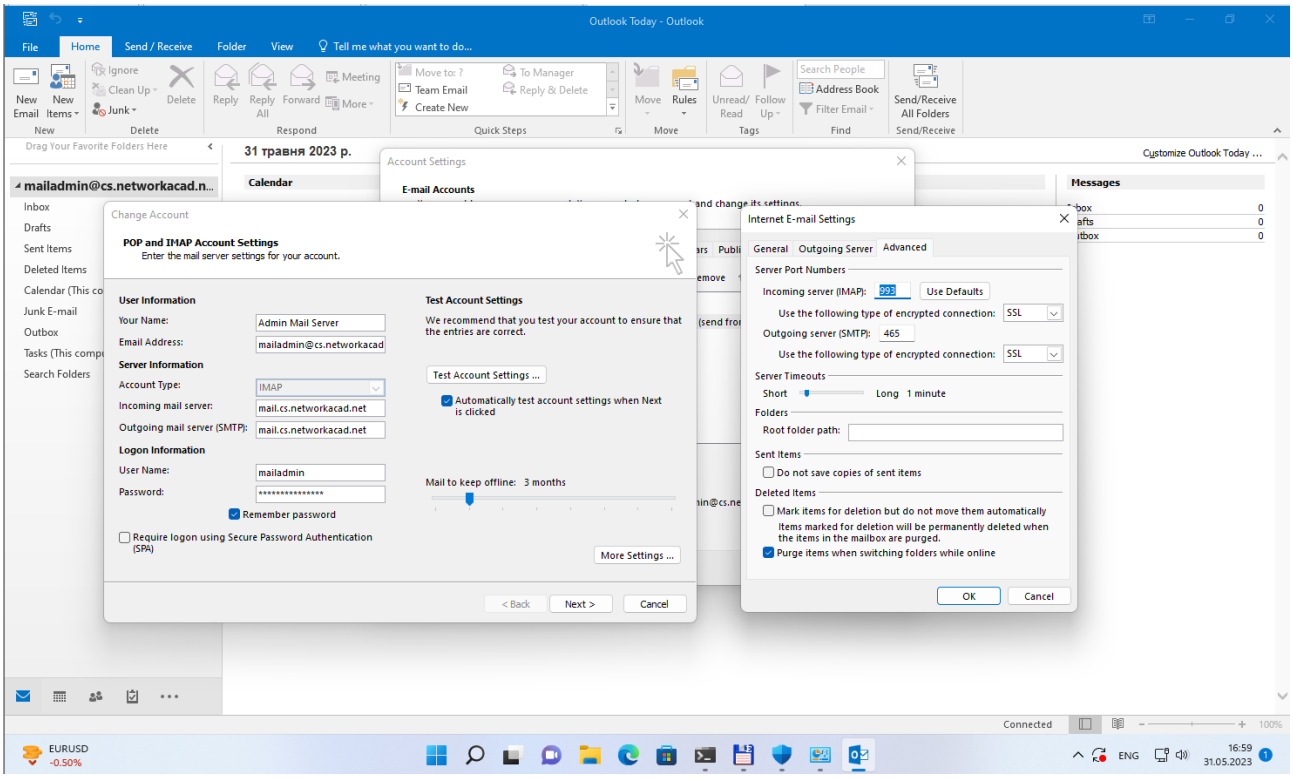


Рисунок 3.7 – Параметри налаштування сервера вхідної та вихідної пошти для mailadmin@cs.networkacad.net

4. Відправимо тестовий лис з поштової скриньки mailadmin@cs.networkacad.net на поштову скриньку pavlo@cs.networkacad.net.

Як можна побачити з рисунка 3.8 лист доставлено в поштовий ящик pavlo@cs.networkacad.net.

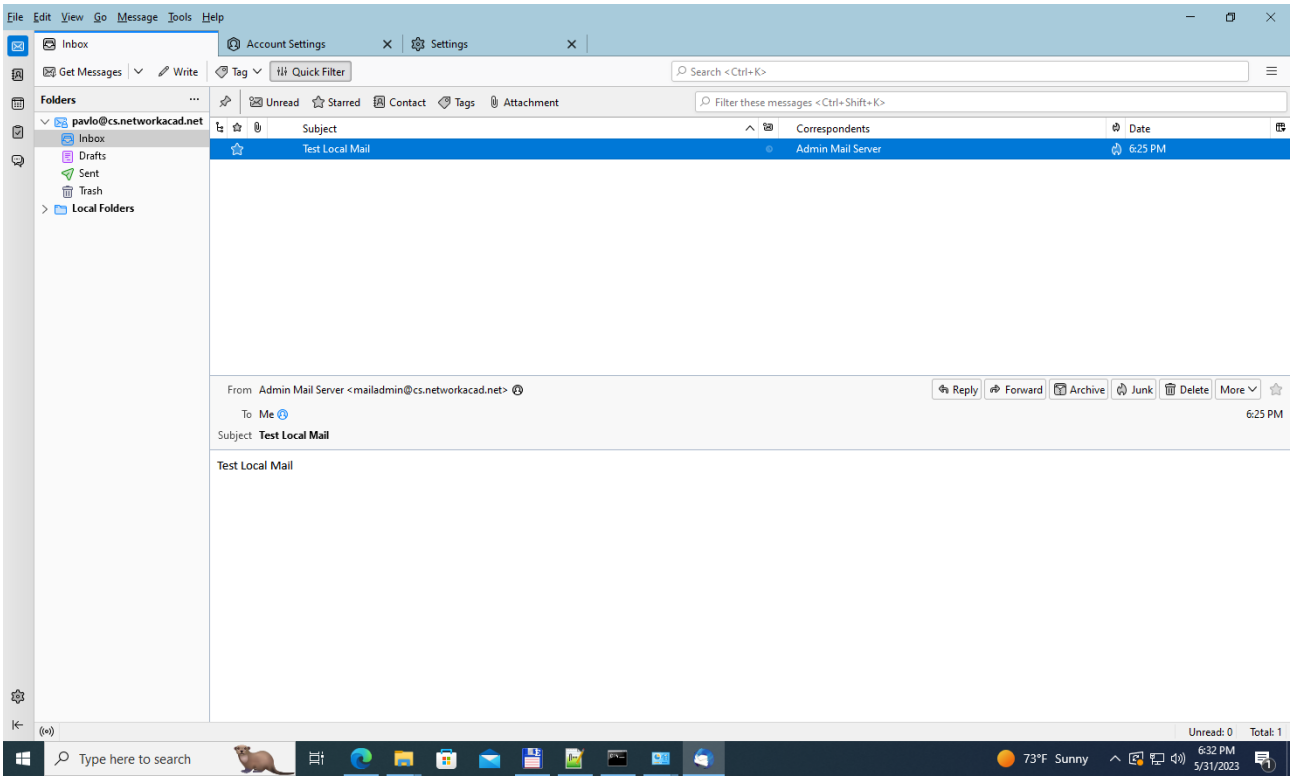


Рисунок 3.8 – Тестовий лист від mailadmin@cs.networkacad.net

В оригіналі (source) (Рис.3.9) даного повідомлення можна побачити що використано версію TLS1.2, шифр ECDHE-RSA-AES256-GCM-SHA384 (256/256 bits) при надсиланні повідомлення поштовим клієнтом Microsoft Outlook через сервер mail.cs.networkacad.net. Також можна побачити що повідомлення перевірено антивірусом ClamAV.

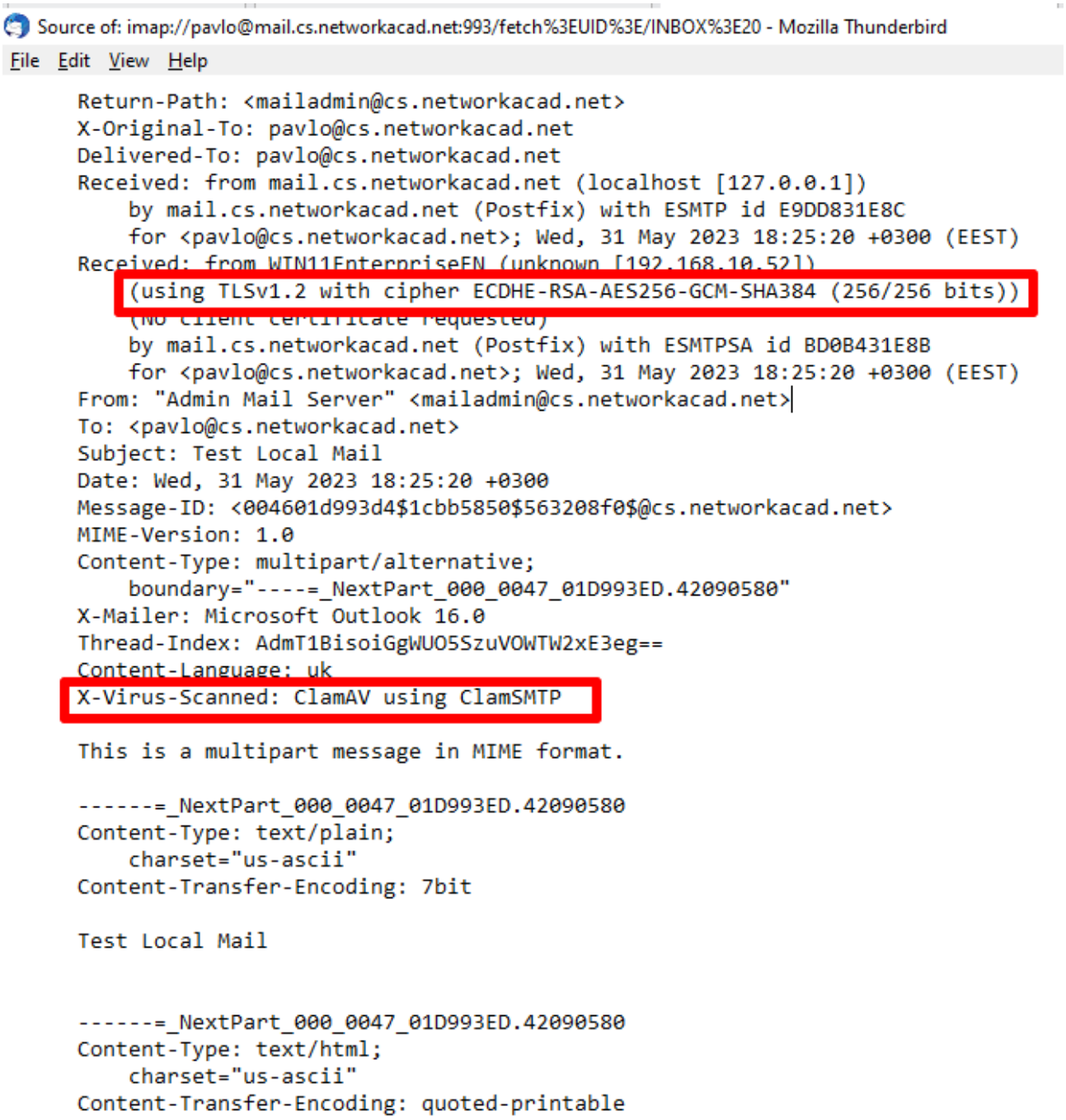


Рисунок 3.9 – Оригінали (source) тестового повідомлення з темою Test Local Mail.

В файлі журналу /var/log/maillog можна відслідкувати етапи обробки повідомлення та побачити підтвердження доставки тестового повідомлення (Рис.3.10).

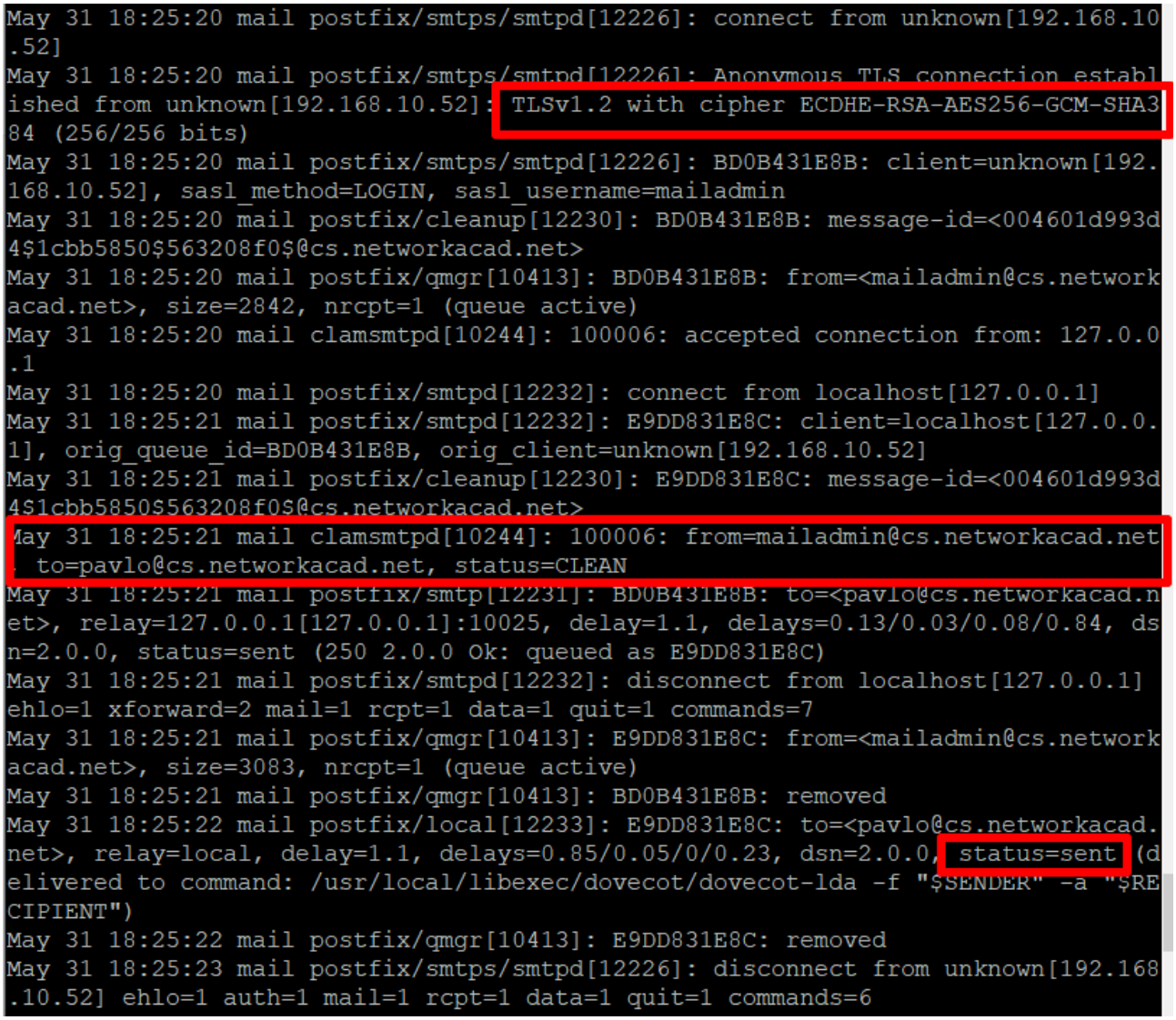


Рисунок 3.10 – Записи в файлі журналу /var/log/maillog

Отже перевірка пройшла успішно. Ми можемо успішно відправити та отримати повідомлення в межах поштового сервера та доставити їх до відповідних скриньок. Можна приступити до наступного етапу перевірки.

3.1.2 Тестування доставки та отримання пошти з зовнішнього поштового сервера.

Щоб перевірити, як поштовий сервер пересилає повідомлення до зовнішніх поштових серверів розглянемо процес доставки листа з поштової скриньки pavlo@cs.networkacad.net на поштову скриньку ternopil.ix@gmail.com.

В файлі журналу /var/log/maillog (Рис.3.11) можна відслідкувати етапи обробки повідомлення, побачити що використано версію TLS1.3, шифр

TLS_AES_128_GCM_SHA256 (128/128 bits) при надсиланні повідомлення поштовим клієнтом Mozilla Thunderbird через сервер mail.cs.networkacad.net. Також можна побачити успішне сканування на віруси тестового повідомлення та підтвердження доставки повідомлення в поштову скриньку ternopil.ix@gmail.com.

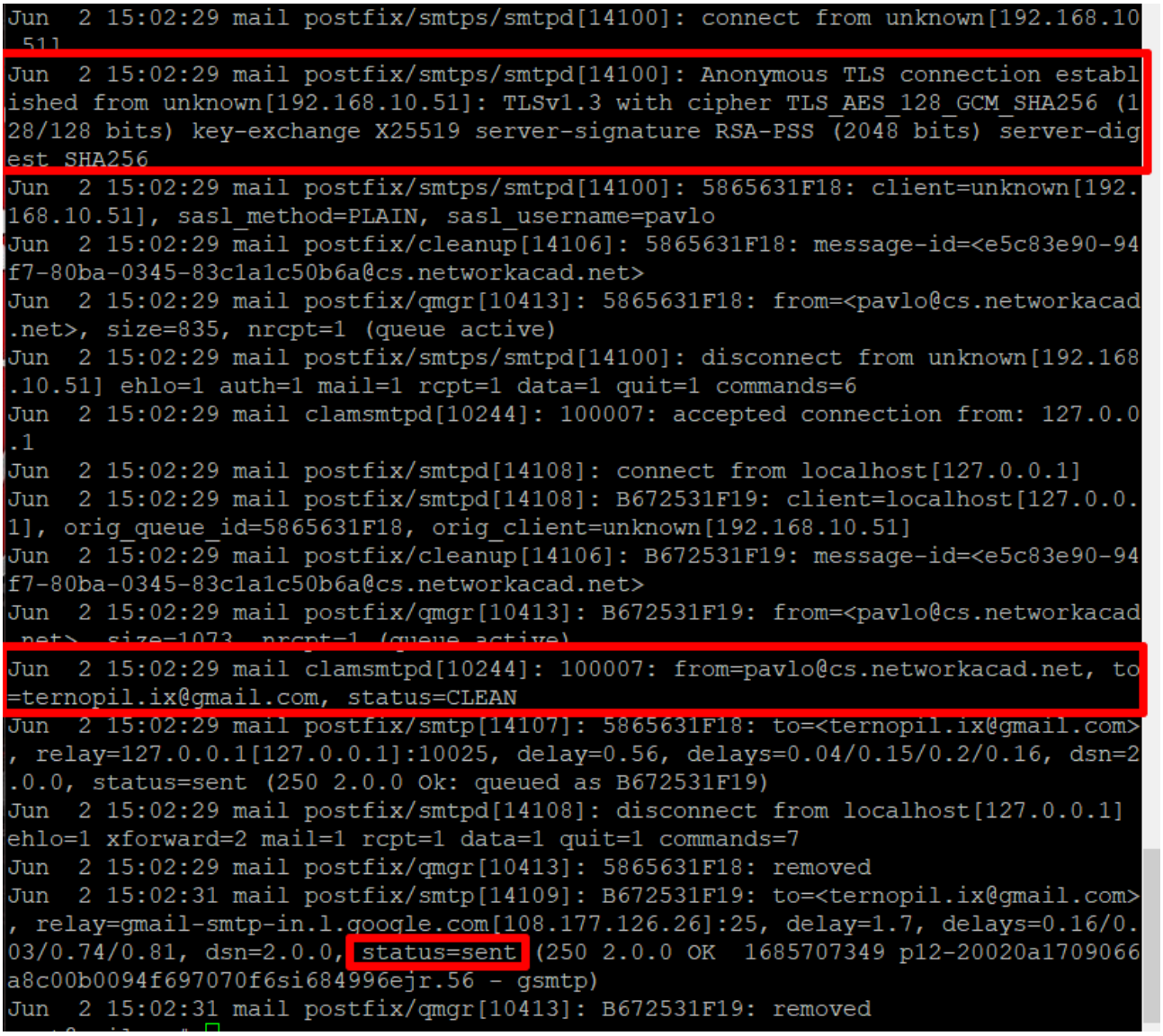


Рисунок 3.11 – етапи обробки поштового повідомлення надісланого до ternopil.ix@gmail.com

В оригіналі (source) даного повідомлення також можна побачити що використано версію TLS1.3, шифр TLS_AES_256_GCM_SHA384 (256/256 bits) для доставки повідомлення з сервера mail.cs.networkacad.net до mx.google.com, повідомлення перевірено антивірусом ClamAV та успішно пройдено етап перевірки SPF (Рис.3.12).

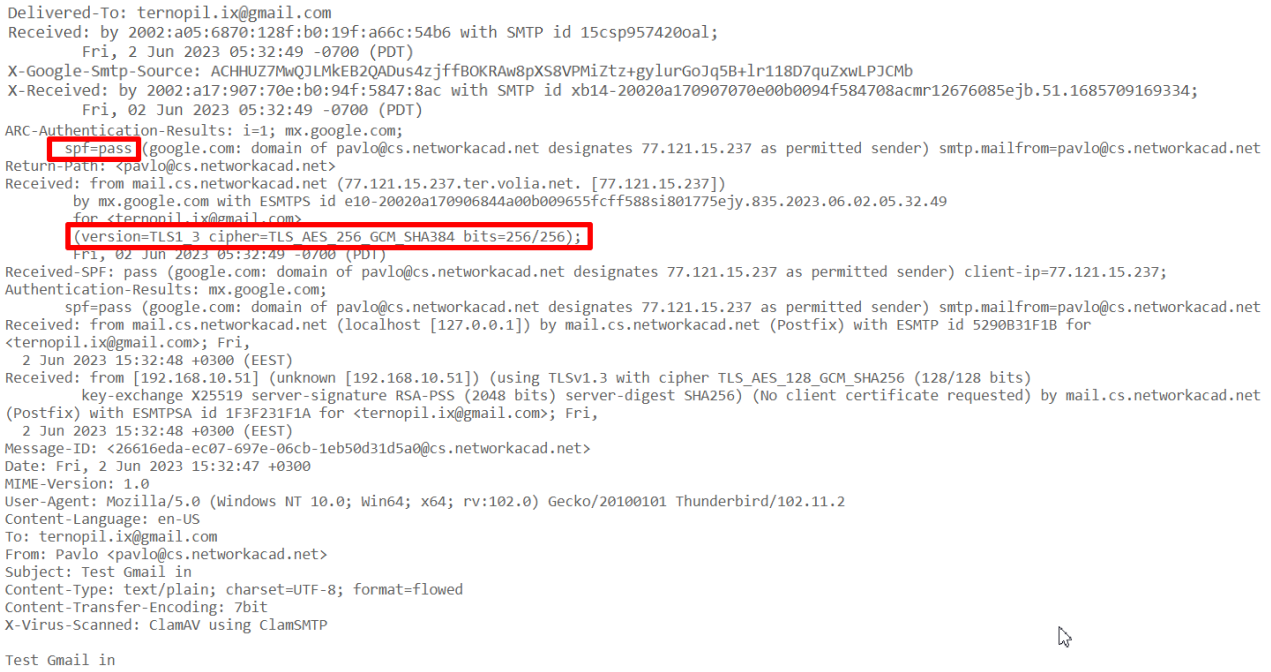


Рисунок 3.12 – Оригінали (source) тестового повідомлення темою Test Gmail in

Щоб перевірити чи коректно поштовий сервер отримує повідомлення з зовнішніх поштових серверів розглянемо процес доставки листа з поштової скриньки ternopil.ix@gmail.com на поштову скриньку pavlo@cs.networkacad.net.

В файлі журналу /var/log/maillog (Рис.3.13) можна відслідкувати етапи обробки повідомлення, побачити що використано версію TLS1.3, шифр TLS_AES_128_GCM_SHA256 (128/128 bits) для доставки повідомлення з сервера mail-oa1-f46.google.com до mail.cs.networkacad.net. Також можна побачити успішне сканування на віруси тестового повідомлення та підтвердження доставки повідомлення в поштову скриньку pavlo@cs.networkacad.net.

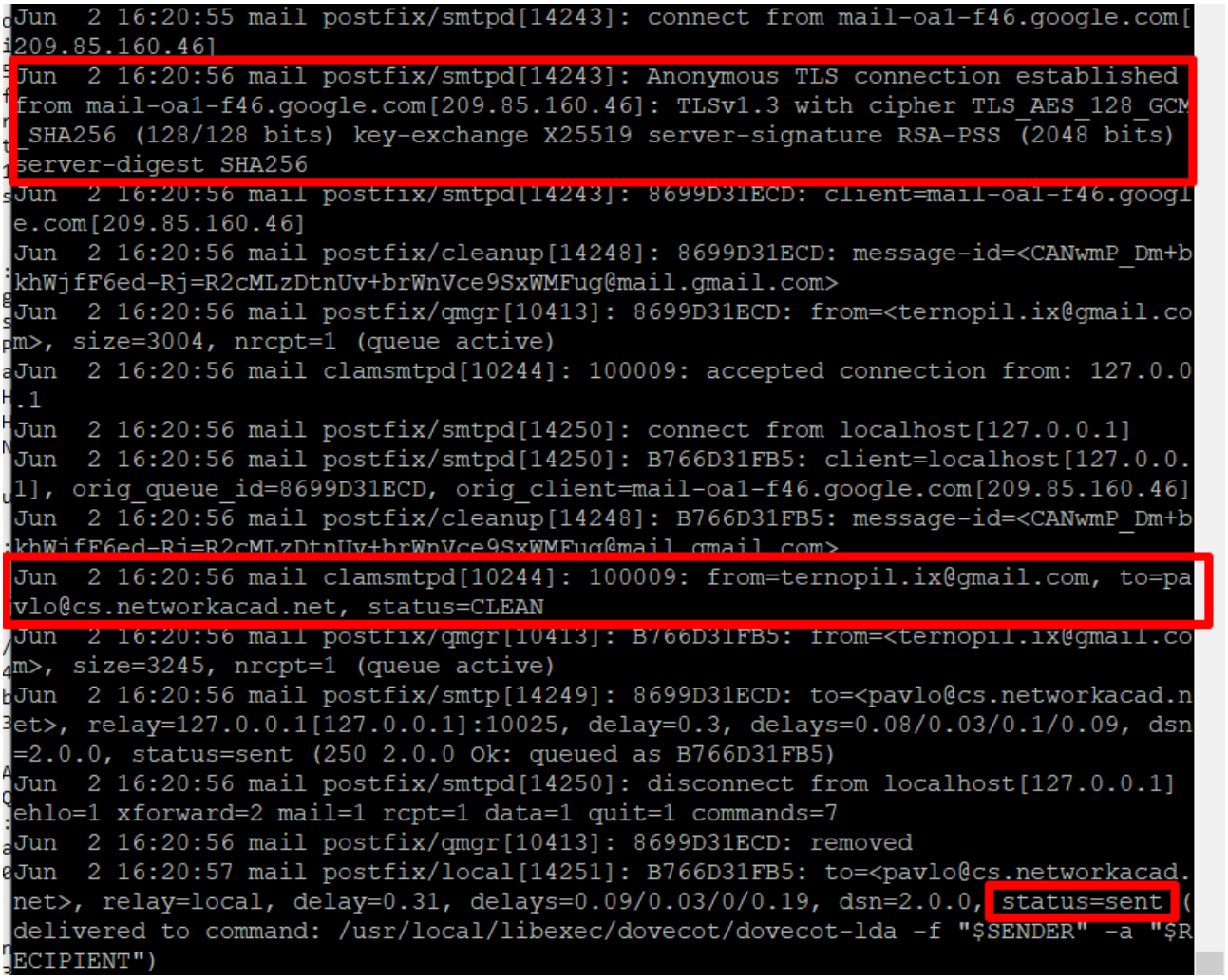


Рисунок 3.13 – Етапи обробки поштового повідомлення при доставці з gmail.com

Отже перевірка пройшла успішно. Ми можемо успішно надіслати повідомлення на зовнішні поштові сервера. Також ми можемо успішно отримати повідомлення з зовнішнього поштового сервера. Можна приступити до наступного етапу перевірки.

3.2 Оцінка безпеки поштового сервера.

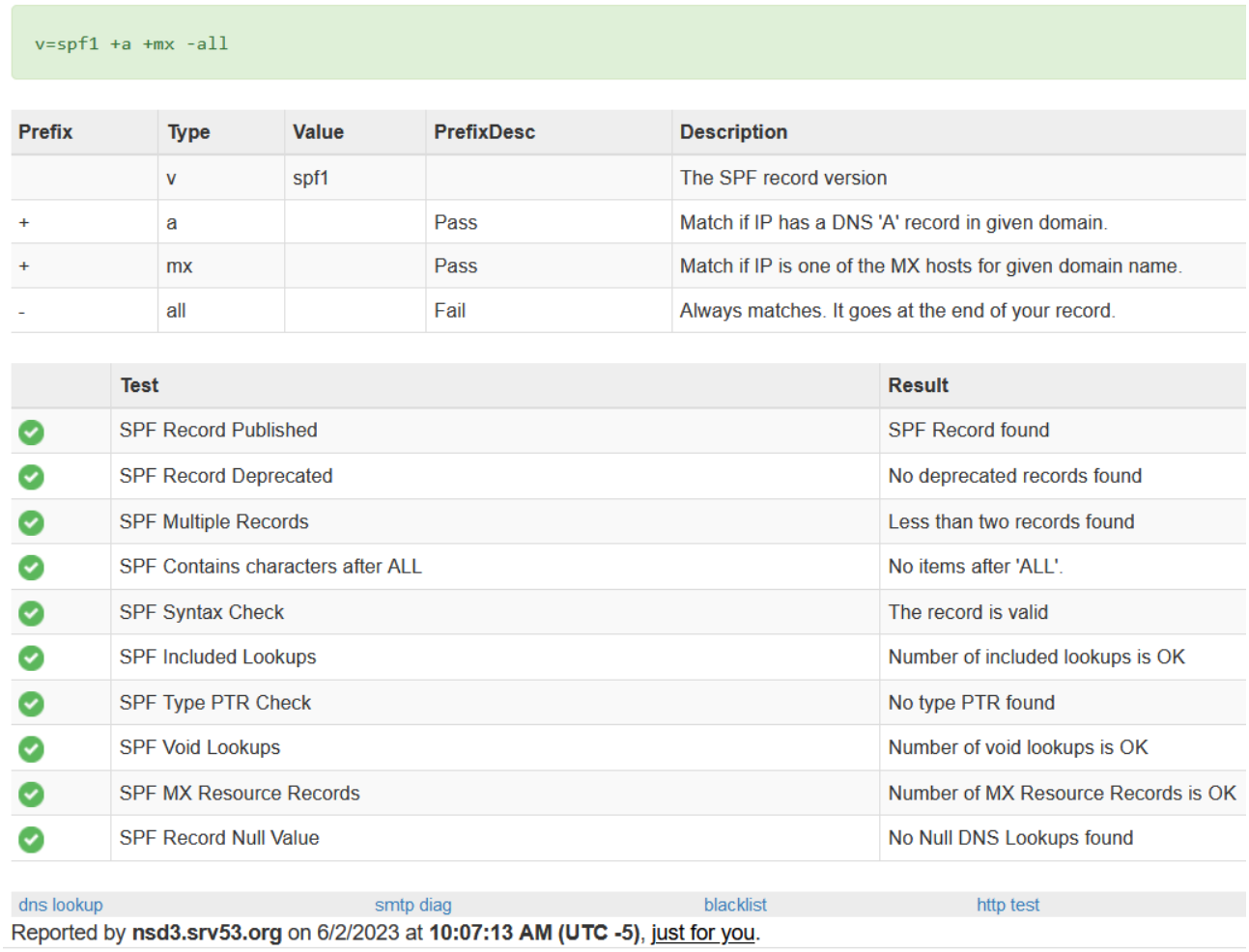
Оцінка безпеки поштового сервера є критично важливим аспектом. Потрібно переконатися, що сервер належним чином захищений від потенційних загроз та вразливостей і також що сервер налаштований коректно.

Для оцінки безпеки поштового сервера використаємо такі інструментами, як **mxtoolbox.com**, **checktls.com** та **nessus** сканер.

Сервіс ***mxtoolbox.com*** надає різноманітну інформацію про домен та поштовий сервер.

Перевірка наявності записів SPF (Sender Policy Framework) для домену є важливим етапом тестування поштового сервера. SPF визначає список IP-адрес, які мають дозвіл надсилати листи від імені домену, що допомагає запобігти підробці листів. Проведемо перевірку та коректність записів SPF.

Результати перевірки SPF запису для домену cs.networkacad.net показано на рисунку 3.14.



```
v=spf1 +a +mx -all
```

Prefix	Type	Value	PrefixDesc	Description
	v	spf1		The SPF record version
+	a		Pass	Match if IP has a DNS 'A' record in given domain.
+	mx		Pass	Match if IP is one of the MX hosts for given domain name.
-	all		Fail	Always matches. It goes at the end of your record.

	Test	Result
✓	SPF Record Published	SPF Record found
✓	SPF Record Deprecated	No deprecated records found
✓	SPF Multiple Records	Less than two records found
✓	SPF Contains characters after ALL	No items after 'ALL'.
✓	SPF Syntax Check	The record is valid
✓	SPF Included Lookups	Number of included lookups is OK
✓	SPF Type PTR Check	No type PTR found
✓	SPF Void Lookups	Number of void lookups is OK
✓	SPF MX Resource Records	Number of MX Resource Records is OK
✓	SPF Record Null Value	No Null DNS Lookups found

dns lookup smtp diag blacklist http test

Reported by **nsd3.srv53.org** on 6/2/2023 at **10:07:13 AM (UTC -5)**, just for you.

Рисунок 3.14 – Перевірка SPF запису для домену cs.networkacad.net

Записи MX (Mail Exchanger) для домену вказують на які поштові сервери повинні надсилатись електронні листи для даного домену.

Тест SMTP перевіряє з'єднання з поштовим сервером за допомогою протоколу SMTP і перевіряє доступність сервера та коректність його налаштування.

Результати тесту SMTP з попередньою перевіркою MX для хоста mail.cs.networkacad.net показано на рисунку 3.15.

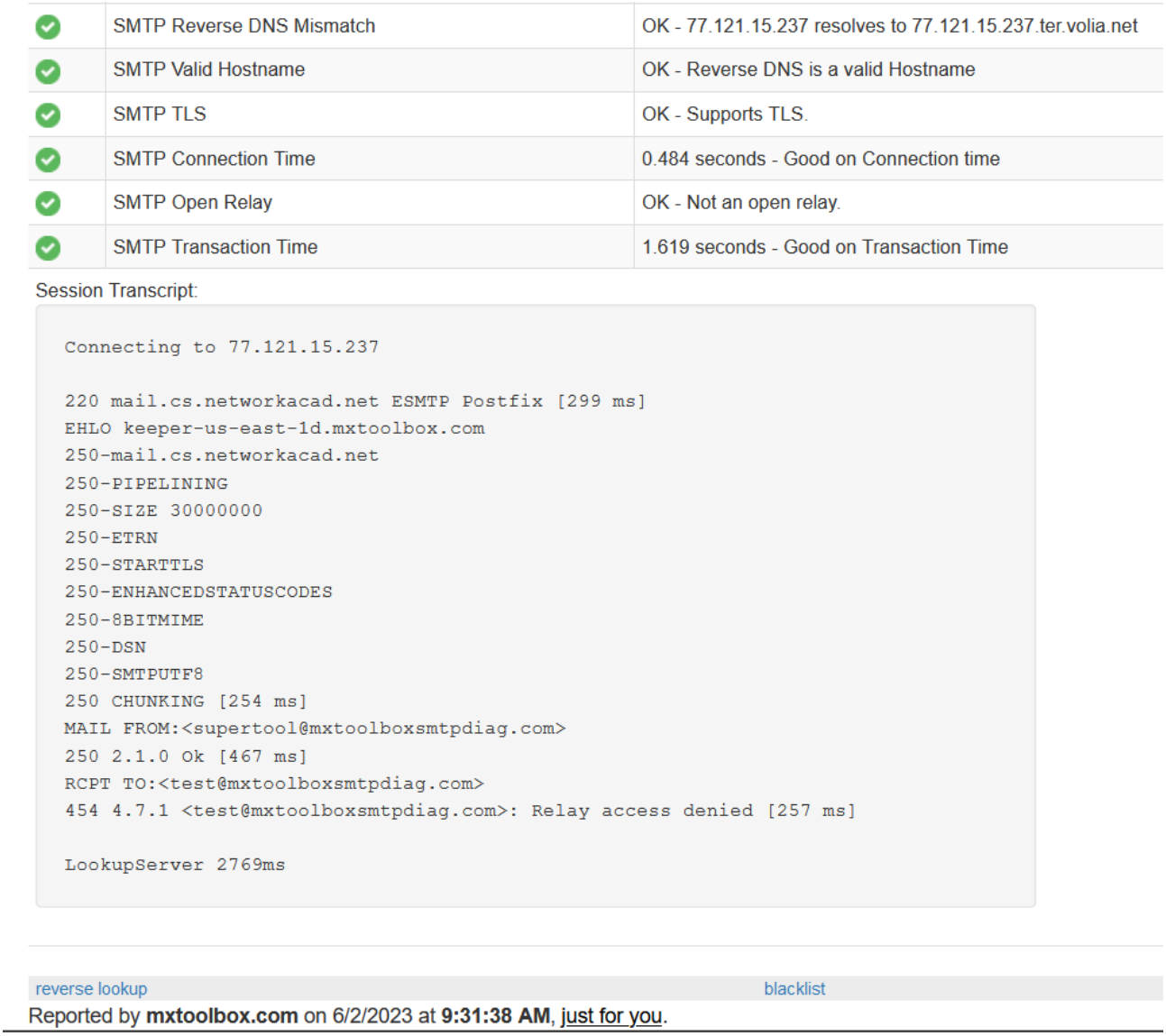


Рисунок 3.15 – Результати тесту SMTP для хоста mail.cs.networkacad.net

Позитивний результат перевірки MX записів свідчать про те , що для домену налаштовані вірні записи MX. Це означає, що визначені поштові сервери, які призначені для отримання електронних листів для цього домену, коректно налаштовані.

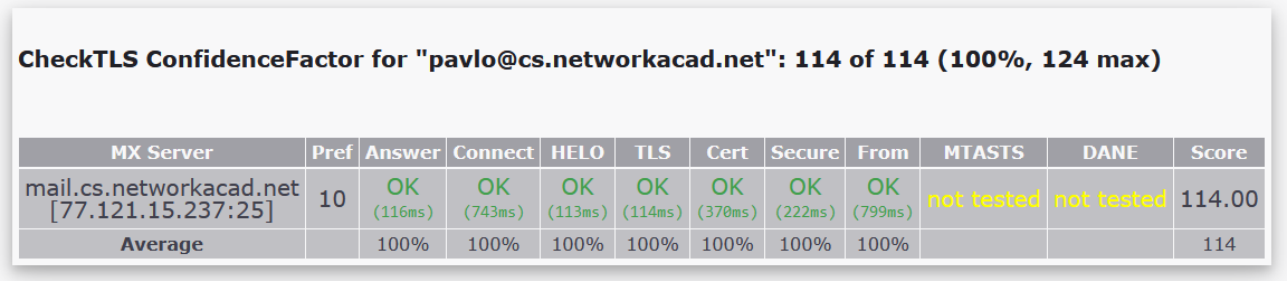
Наявність запису SPF показує що визначено список IP-адрес, які мають дозвіл надсилати листи від імені домену. Якщо SPF запис налаштований правильно, це допомагає запобігти підробці листів та покращує доставку повідомлень.

Позитивний результат тесту SMTP показує, що поштовий сервер домени доступний і готовий приймати та надсилати електронні листи. Це означає, що з'єднання з сервером може бути встановлено та комунікація через протокол SMTP працює належним чином.

Ці позитивні результати свідчать про налаштування поштового сервера та домену, що дозволяє надійно отримувати, аутентифікувати та доставляти електронні листи. Вони показують, що інфраструктура для обробки і доставки повідомлень належним чином налаштована та функціонує коректно.

Сервіс ***checktls.com*** надає інформацію про безпеку і налаштування поштового сервера.

Зведені результати перевірки поштового сервера показано на рисунку 3.16.



CheckTLS ConfidenceFactor for "pavlo@cs.networkacad.net": 114 of 114 (100%, 124 max)

MX Server	Pref	Answer	Connect	HELO	TLS	Cert	Secure	From	MTASTS	DANE	Score
mail.cs.networkacad.net [77.121.15.237:25]	10	OK (116ms)	OK (743ms)	OK (113ms)	OK (114ms)	OK (370ms)	OK (222ms)	OK (799ms)	not tested	not tested	114.00
Average		100%	100%	100%	100%	100%	100%	100%			114

Рисунок 3.16 – Результати перевірки хосту mail.cs.networkacad.net

Розширені результати перевірки наведено в додатку А. Дана перевірка показала коректність підтримки TLS та версію TLS 1.3, доступні шифри та протоколи шифрування, які використовуються на поштовому сервері.

Дана перевірка також підтверджує що на сервері використовується валідний та вірно налаштований сертифіката TLS.

Позитивні результати перевірок на сайті checktls.com свідчать про добру безпеку та налаштування поштового сервера з точки зору шифрування та захисту від перехоплення даних.

Позитивний результат перевірки TLS показує, що поштовий сервер підтримує сучасний протокол шифрування TLS і має налаштовані безпечні параметри. Це вказує на те, що з'єднання між поштовим сервером та клієнтом може бути зашифроване та забезпечує конфіденційність даних. Позитивний результат перевірки шифрування показує, що на поштовому сервері встановлені

сильні шифри та протоколи шифрування. Це важливо для забезпечення безпеки передачі даних та захисту від зловмисників. Позитивний результат перевірки сертифіката показує, що TLS сертифікат, який використовується на поштовому сервері, є валідним і правильно налаштованим. Це означає, що клієнти можуть довіряти сертифікату та впевнено встановлювати зашифроване з'єднання з сервером.

Виконаємо сканування на вразливості, використовуючи сканер ***nessus.*** Це допоможе виявити можливі вразливості та проблеми безпеки, які потребують подальшої уваги та виправлення.

Nessus - це комерційний сканер вразливостей, розроблений компанією Tenable. Він використовується для виявлення потенційних вразливостей і слабких місць в комп'ютерних системах та мережах. Nessus відомий своєю широкою базою даних вразливостей і має велику кількість вбудованих тестів безпеки [19].

Перед скануванням поштового сервера на вразливості зупинимо роботу брандмауера PF. Це дасть можливість отримати максимально повний звіт що до всіх відкритих портів на сервері та сервісів, які ці порти прослуховують.

Зведені результати перевірки поштового сервера показано на рисунку 3.17.

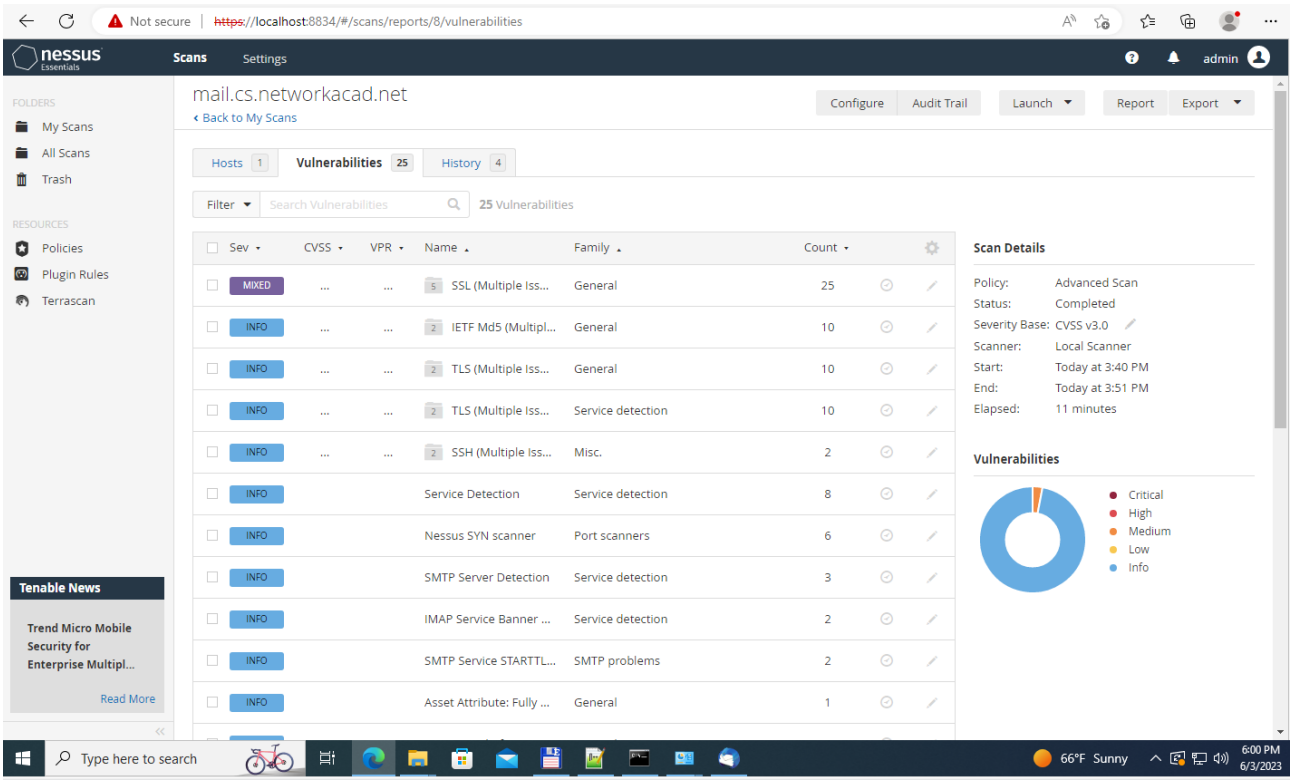


Рисунок 3.17 – Зведені результати перевірки хосту mail.cs.networkacad.net сканером nessus.

Єдиним застереження при перевірці є сертифікат (Рис. 3.18). Це було очікувано, оскільки я не використовував комерційні зовнішній центр сертифікації при створення сертифікатів.

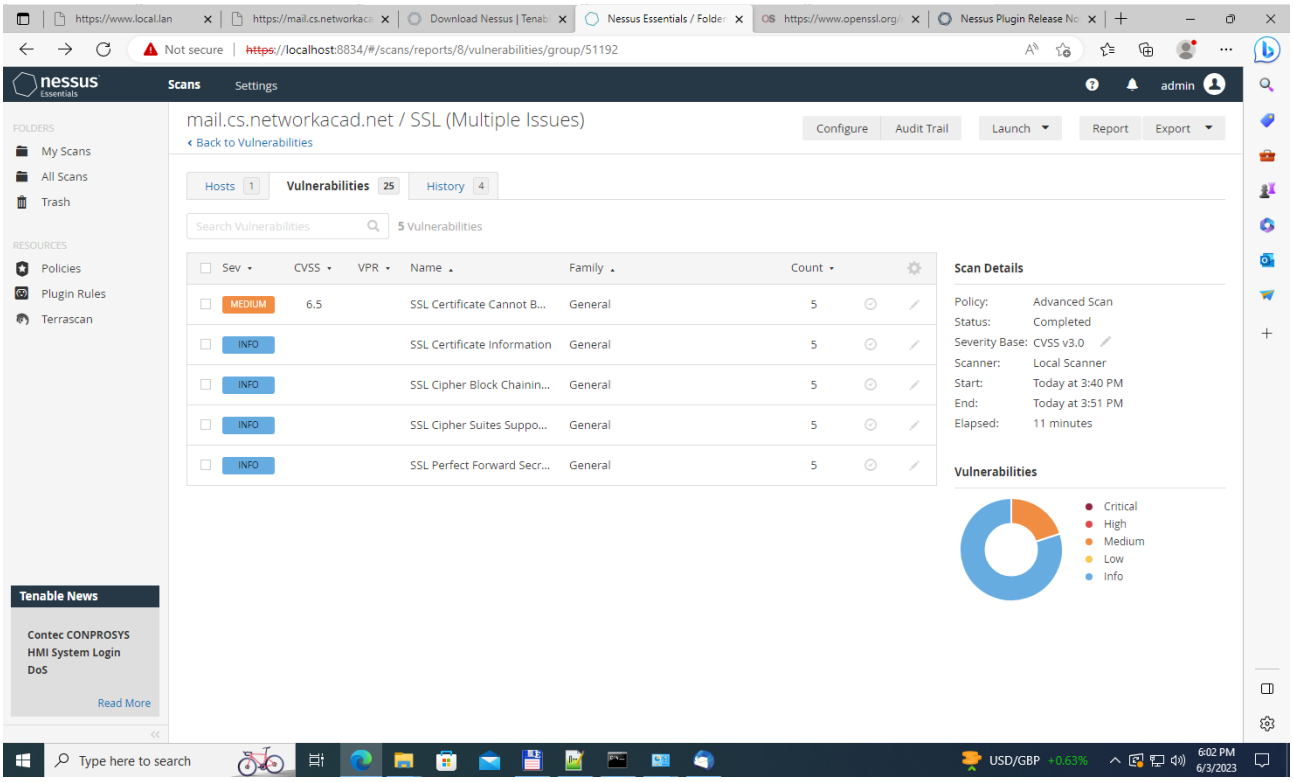


Рисунок 3.18 – Застереження при перевірці сертифікату mail.cs.networkacad.net сканером nessus.

Ці позитивні результати свідчать про те, що поштовий сервер належним чином налаштований з точки зору безпеки і шифрування. Вони підтверджують, що дана інфраструктура забезпечує безпечну комунікацію та захист даних під час передачі електронних повідомлень.

4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Домедична допомога при тепловому ударі.

Тепловий удар - це серйозний медичний стан, який виникає внаслідок перегрівання організму, зазвичай внаслідок тривалого перебування на сонці або у дуже спекотному середовищі. Це може призвести до серйозних ушкоджень органів і навіть може привести до смерті потерпілого.

Симптоми теплового удару можуть включати:

- Висока температура тіла (понад 40°C).
- Запаморочення, слабкість та втома.
- Головний біль.
- Червона, гаряча суха шкіра.
- Швидке та поверхневе дихання.
- Слабкість, запаморочення або втрата свідомості.

Домедична допомога постраждалим при підозрі на тепловий удар є важливою процедурою, яку можуть виконувати особи без медичної освіти.

Наказ Міністерства охорони здоров'я України від 09.03.2022 р. № 441 " Про затвердження порядків надання домедичної допомоги особам при невідкладних станах" встановлює порядки надання домедичної допомоги постраждалим при підозрі на тепловий удар. У цьому порядку термін “тепловий удар” вживаються у такому значенні - невідкладний стан, викликаний дією високої температури навколишнього середовища, що спричиняє системні розлади у постраждалого [20].

Надання домедичної допомоги постраждалим при тепловому ударі передбачає такі кроки:

1.Переконатися, що немає небезпеки для себе, оточуючих та постраждалого, перед тим, як надавати допомогу.

2.Заспокоїти постраждалого та пояснити свої дії.

3.Викликати екстрену медичну допомогу та слухати інструкції диспетчера.

4.Перемістити постраждалого в прохолодне приміщення, щоб припинити дію тепла на нього.

5.Виміряти внутрішню температуру тіла постраждалого.

6.Використовувати методи охолодження, які доступні:

- повністю занурити постраждалого у холодну воду (18-26 °C), якщо його внутрішня температура тіла перевищує 40°C, та продовжувати занурення, поки температура не знизиться до 39°C.

- якщо повне занурення неможливе, можна використовувати пакети з льодом, обгорнуті у рушники, і накласти їх на тіло постраждалого. Можна також обдувати постраждалого вентиляторами або накласти вологі серветки на тіло.

7.Наглядати за постраждалим до прибуття медичної бригади.

8.Якщо постраждалий залишається свідомим, давати йому пити багато рідини.

9.Якщо стан постраждалого погіршується, повторно викликати екстрену медичну допомогу.

10.Зібрати інформацію про обставини виникнення теплового удару і передати її медичним працівникам при прибутті.

11.Якщо постраждалий втратив свідомість до прибуття медичної бригади, перейти до надання домедичної допомоги при раптовій зупинці кровообігу, відповідно до встановлених протоколів.

Це загальна послідовність дій, яку слід виконати, але завжди важливо дотримуватись інструкцій медичних фахівців та адаптувати допомогу до конкретної ситуації. Виконання цих кроків допоможе забезпечити постраждалому першу необхідну допомогу та зберегти його життя до прибуття медичних фахівців.

4.2 Естетичне оформлення робочого місця оператора ПК.

В сучасному світі багато людей проводять значну частину свого часу за робочим столом оператора ПК. Робоче місце є місцем, де проходить багато годин концентрованої праці, комунікації і творчості. Тому важливо не лише забезпечити функціональність та зручність цього простору, але й звернути увагу на його естетичне оформлення.

Естетичне оформлення робочого місця оператора ПК не є просто прикрасою. Воно впливає на настрій, комфорт і продуктивність. Гармонійне та затишне оточення може стимулювати творчість, поліпшувати концентрацію і сприяти ефективній роботі. Крім того, персоналізація робочого простору дозволяє виразити свою індивідуальність та створити потрібну робочу атмосферу в довколишньому середовищі.

При організації естетичного оформлення робочого місця оператора ПК потрібно врахувати наступні моменти:

Оптимальне розташування обладнання: розмістіть комп'ютер, монітор і периферійні пристрої (клавіатура, миша і т.д.) таким чином, щоб було зручно досягати до них і працювати. Уникайте перенавантаження робочої поверхні надмірною кількістю об'єктів.

Організація кабелів: спробуйте зберегти порядок на робочому місці, організувавши кабелі. Використовуйте спеціальні тримачі або кабельні канали, щоб зібрати всі кабелі разом і уникнути безладу.

Регульованість меблів: якщо це можливо, оберіть регульовані меблі, такі як стіл і стілець. Це дозволить вам налаштувати їх на оптимальну висоту і зручніше працювати. Також не забувайте про крісло, яке підтримує вашу спину та посадку, для забезпечення комфорту протягом тривалого робочого дня.

Освітлення: забезпечте достатнє освітлення на робочому місці. Використовуйте природне освітлення, де це можливо, і додаткові лампи, якщо потрібно. Уникайте світлових джерел, які можуть втомлювати очі.

Персоналізація: додайте особистого штриху до свого робочого простору, розташувавши на столі фотографії, рослини, мотиваційні цитати або речі, які надихають вас. Зробіть його комфортним і приємним для вас.

Колірна гамма: використовуйте кольори, які вам подобаються і створюють приємну атмосферу. Наприклад, нейтральні або природні відтінки можуть сприяти спокою і концентрації.

Мінімалізм: розгляньте можливість створення мінімалістичного дизайну. Уникайте зайвих предметів або безладу на робочому столі. Чистота та простота можуть сприяти зосередженості і ефективності.

Правильне використання простору: максимізуйте використання доступного простору, особливо якщо у вас обмежений робочий простір. Використовуйте полички, ящики або стінні органайзери для зберігання речей і важливих документів.

Зонування: якщо ви маєте можливість, створіть зони на робочому місці, наприклад, зона для роботи з комп'ютером або зона для письма. Це допоможе розподілити простір і зберегти організованість.

Зображення та графіка: розгляньте можливість додавання художніх картин, плакатів або інших видів графіки на стіни робочого простору. Це може створити стимулююче середовище та надихати на творчість.

Потрібні аксесуари: виберіть стильні й корисні аксесуари, які підходять до вашого стилю та потреб. Наприклад, стильна підставка для ноутбука, ергономічна підставка для рук або оригінальні канцелярські засоби.

Правильна вентиляція та комфорт: переконайтеся, що у вас є належна вентиляція та забезпечена комфортна температура в приміщенні.

Загалом, естетичне оформлення робочого місця оператора ПК має бути практичним і зручним для роботи, одночасно створюючи приємну атмосферу, яка сприяє продуктивності і комфорту.

ВИСНОВКИ

У даній кваліфікаційній роботі було проведено налаштування захищеного корпоративного поштового сервера з використанням FreeBSD, Postfix, Dovecot, ClamAV, Clamsmtpd, Cyrus-SASL, Cyrus-SASL-saslauthd та OpenSSL. Правильне налаштування сервера має велике значення для організацій, оскільки ефективний та безпечний обмін електронними листами є ключовим аспектом комунікації в сучасному бізнес-середовищі.

У процесі дослідження та розробки були розглянуті теоретичні основи поштових протоколів та компонентів, необхідних для побудови поштового сервера. Була описана архітектура поштового сервера, яка включає компоненти, такі як MTA (Postfix), MDA (Dovecot), антивірусний сканер (ClamAV) та інші.

Після опису архітектури були проведені кроки для розробки та налаштування поштового сервера. Ці кроки включали встановлення та конфігурацію необхідних компонентів, створення поштових скриньок, налаштування безпеки та фільтрації повідомлень.

Далі було розглянуто тестування поштового сервера. Виконано тестування функціональності, перевірено відправлення та отримання повідомлень, аутентифікацію користувачів та правила фільтрації. Також проведено оцінку безпеки, яка могла виявити потенційні проблеми.

Отримані результати свідчать про успішну реалізацію поставлених завдань та досягнення поставленої мети. Компоненти поштового сервера налаштовані для забезпечення безпеки, аутентифікації, обробки спаму та вірусів, а також шифрування комунікації.

Загалом, розробка та налаштування захищеного корпоративного поштового сервера є складним процесом, який вимагає глибоких знань поштових протоколів, мережевої безпеки та системного адміністрування.

Також, важливо зазначити, що розробка та налаштування поштового сервера - це неперервний процес, оскільки потреби організації можуть змінюватися, а загрози безпеці постійно еволюціонують. Тому потрібно

виконувати регулярну підтримку, оновлення та моніторинг поштового сервера з метою забезпечення його безперебійної та безпечної роботи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Microsoft Exchange documentation [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://learn.microsoft.com/en-us/exchange/
2. FreeBSD Documentation [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://docs.freebsd.org/en/
3. Postfix Documentation [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: http://www.postfix.org/documentation.html
4. Dovecot manual [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://doc.dovecot.org/
5. Pigeonhole Sieve Interpreter [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://doc.dovecot.org/configuration_manual/sieve/pigeonhole_sieve_interpreter/
6. ClamAV Documentation [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://docs.clamav.net/
7. Cyrus SASL [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://www.cyrusimap.org/sasl/
8. OpenSSL Documentation [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://www.openssl.org/docs/
9. Oracle Solaris ZFS Administration Guide [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://docs.oracle.com/cd/E19253-01/819-5461/
10. The Z File System (ZFS) [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://docs.freebsd.org/en/books/handbook/zfs/
11. Postfix Installation [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://www.postfix.org/INSTALL.html

12. Postfix Basic Configuration [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://www.postfix.org/BASIC_CONFIGURATION_README.html
13. Postfix SASL Howto [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://www.postfix.org/SASL_README.html
14. Postfix TLS Support [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://www.postfix.org/TLS_README.html
15. Domain names - concepts and facilities [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://datatracker.ietf.org/doc/html/rfc1034
16. Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1 [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://datatracker.ietf.org/doc/html/rfc7208
17. Dovecot Wiki [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://wiki.dovecot.org/
18. Dovecot LDA [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://doc.dovecot.org/configuration_manual/protocols/lda/#lda
19. Documentation Nessus [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://docs.tenable.com/Nessus.htm
20. Про затвердження порядків надання домедичної допомоги особам при невідкладних станах [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://zakon.rada.gov.ua/laws/show/z0356-22#n769

Додаток А

Розширені результати перевірки домену cs.networkacad.net з сайту checktls.com.

```
Checking pavlo@cs.networkacad.net from www11-do.CheckTLS.com(V03.71.00) at 2023-06-02T18:16:25Z:

seconds    lookup                      result
[000.000]  DNS LOOKUPS
[000.008]  SEARCHLIST                  104.131.108.216,134.209.169.224,1.1.1.1,2001:4860:4860::8888,8.8.8.8,67.207.67.3
[000.116]  MX                          (10) mail.cs.networkacad.net
[000.219]  MX:A-->mail.cs.networkacad.net  77.121.15.237

seconds  test stage and result
[000.000]  Trying TLS on mail.cs.networkacad.net[77.121.15.237:25] (10)
[000.116]  Server answered
[000.858] <--220 mail.cs.networkacad.net ESMTP Postfix
[000.859]  We are allowed to connect
[000.859] -->EHLO www11-do.CheckTLS.com
[000.972] <--250-mail.cs.networkacad.net
           250-PIPELINING
           250-SIZE 30000000
           250-ETRN
           250-STARTTLS
           250-ENHANCEDSTATUSCODES
           250-8BITMIME
           250-DSN
           250-SMTPUTF8
           250 CHUNKING
[000.972]  We can use this server
[000.972]  TLS is an option on this server
[000.973] -->STARTTLS
[001.085] <--220 2.0.0 Ready to start TLS
[001.086]  STARTTLS command works on this server
[001.451]  Connection converted to SSL

[001.451]  Connection converted to SSL
           SSLVersion in use: TLSv1_3
           Cipher in use: TLS_AES_256_GCM_SHA384
           Perfect Forward Secrecy: yes
           Session Algorithm in use: Curve X25519 DHE(253 bits)
           Certificate #1 of 4 (sent by MX):
           Cert VALIDATED: ok
           Cert Hostname VERIFIED (mail.cs.networkacad.net = mail.cs.networkacad.net | DNS:mail.cs.networkacad.net)
           Not Valid Before: Jun  2 16:43:51 2023 GMT
           Not Valid After: Aug 31 16:43:50 2023 GMT
           subject: /CN=mail.cs.networkacad.net
           issuer: /C=US/O=Let's Encrypt/CN=R3
           Certificate #2 of 4 (sent by MX):
           Cert VALIDATED: ok
           Not Valid Before: Sep  4 00:00:00 2020 GMT
           Not Valid After: Sep 15 16:00:00 2025 GMT
           subject: /C=US/O=Let's Encrypt/CN=R3
           issuer: /C=US/O=Internet Security Research Group/CN=ISRG Root X1
           Certificate #3 of 4 (added from CA Root Store):
           Cert VALIDATED: ok
           Not Valid Before: Jun  4 11:04:38 2015 GMT
           Not Valid After: Jun  4 11:04:38 2035 GMT
           subject: /C=US/O=Internet Security Research Group/CN=ISRG Root X1
           issuer: /C=US/O=Internet Security Research Group/CN=ISRG Root X1
           Certificate #4 of 4 (sent by MX):
           Cert VALIDATED:
           Not Valid Before: Jan 20 19:14:03 2021 GMT
           Not Valid After: Sep 30 18:14:03 2024 GMT
           subject: /C=US/O=Internet Security Research Group/CN=ISRG Root X1
           issuer: /O=Digital Signature Trust Co./CN=DST Root CA X3
```

```
[001.459] ~~>EHLO www11-do.CheckTLS.com
[001.678] <~~250-mail.cs.networkacad.net
          250-PIPELINING
          250-SIZE 30000000
          250-ETRN
          250-AUTH PLAIN LOGIN
          250-AUTH=PLAIN LOGIN
          250-ENHANCEDSTATUSCODES
          250-8BITMIME
          250-DSN
          250-SMTPUTF8
          250 CHUNKING
[001.678]    TLS successfully started on this server
[001.678] ~~>MAIL FROM:<test@checktls.com>
[002.476] <~~250 2.1.0 Ok
[002.477]    Sender is OK
[002.477] ~~>QUIT
[002.591] <~~221 2.0.0 Bye
```