INDIVIDUAL ASSIGNMENT

TECHNOLOGY PARK MALAYSIA

CT106-3-2-SNA

SYSTEMS AND NETWORK ADMINISTRATION

APU2F2502CS(DA)

HAND OUT DATE: WEEK 2 MARCH 2025

HAND IN DATE: WEEK 10 MAY 2025

WEIGHTAGE: 30%

STUDENT DETAILS: OOI DUN TZI TP071308

INSTRUCTIONS TO CANDIDATES:

1 Submit your assignment at the administrative counter.

2 Students are advised to underpin their answers with the use of references (cited using the Harvard Name System of Referencing).

3 Late submission will be awarded zero (0) unless Extenuating Circumstances (EC) are upheld.

4 Cases of plagiarism will be penalized.

5 The assignment should be bound in an appropriate style (comb bound or stapled).

6 Where the assignment should be submitted in both hardcopy and softcopy, the softcopy of the written assignment and source code (where appropriate) should be on a CD in an envelope / CD cover and attached to the hardcopy.

7 You must obtain 50% overall to pass this module.

Contents

1.0 Introduction

This report details the complete process of setting up a fully functional email server using Rocky Linux, focusing on the integration of Postfix, Dovecot, and Thunderbird. The primary goal is to demonstrate how to configure a secure and reliable email system capable of sending, receiving, and filtering messages effectively. Postfix serves as the Mail Transfer Agent (MTA), while Dovecot handles user authentication and email retrieval via IMAP and POP3 protocols. Thunderbird is utilized as the client application to validate server communication. The report also covers essential security features such as TLS encryption, SMTP authentication, and firewall configuration. Additional functionalities, including SpamAssassin for spam filtering and MIME-based attachment filtering, are implemented to enhance protection against malicious content. Each configuration step is supported with command-line instructions, file modifications, and testing procedures. This hands-on approach not only illustrates best practices in Linux server administration but also builds technical competency in managing secure email infrastructures.

2.0 Setting Up a Fully Functional Email Using Rocky Linux

2.1 Postfix Installation & Configuration

2.1.1 Postfix Installation

```
[ooiduntzi@benserver ~]$ sudo dnf install postfix
[sudo] password for ooiduntzi:
Rocky Linux 9 - BaseOS                          1.2 kB/s | 4.1 kB     00:03
Rocky Linux 9 - AppStream                       2.1 kB/s | 4.5 kB     00:02
Rocky Linux 9 - AppStream                       3.7 MB/s | 8.4 MB     00:02
Rocky Linux 9 - Extras                          3.7 kB/s | 2.9 kB     00:00
Dependencies resolved.
================================================================================
 Package         Architecture   Version                 Repository         Size
================================================================================
```

Figure 1: Install Postfix

Step 1: Installing Postfix on Rocky Linux

Command:

- sudo dnf install postfix

Explanation and Purpose:

- This command installs Postfix, a widely used open-source Mail Transfer Agent (MTA) that manages the sending and receiving of emails on Linux systems. The dnf package manager handles the installation, ensuring all necessary dependencies are installed.

Role in the System:

- Postfix plays a crucial role in email communication by performing two main functions:

Email Routing and Delivery:

- It accepts incoming emails from local applications or external sources (like your Ubuntu client) and ensures they are delivered to the correct mailbox or forwarded appropriately.

Security and Spam Mitigation:

- Postfix supports encryption protocols such as TLS/SSL for secure mail transmission and can work with various spam filtering tools (like SpamAssassin) to help identify and reduce unwanted email traffic.

2.1.2 Postfix Configuration (main.cf)

```
myhostname = benserver.bungkus.org

# The mydomain parameter specifies the local internet domain name.
# The default is to use $myhostname minus the first component.
# $mydomain is used as a default value for many other configuration
# parameters.
#
mydomain = bungkus.org
```

```
mydestination = $myhostname, localhost.$mydomain, $mydomain
#mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
#mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain,
#       mail.$mydomain, www.$mydomain, ftp.$mydomain
```

```
#home_mailbox = Mailbox
home_mailbox = Maildir/
```

Figure 2: Configure /etc/postfix/main.cf

Step 2: Configuring Postfix Settings

Command:

- sudo nano /etc/postfix/main.cf

Explanation and Purpose:

- This command opens the Postfix main configuration file for editing, allowing you to adjust key settings to define how your mail server behaves.

Key Configuration Parameters and Their Roles:

- **smtpd_use_tls = yes**
  - **Role:** Activates TLS encryption for incoming SMTP connections.
  - **Purpose:** Ensures emails are transmitted securely, protecting them from interception or tampering.
- **myhostname = benserver.bungkus.org**
  - **Role:** Specifies the fully qualified domain name (FQDN) of your mail server.

  - **Purpose:** Helps identify your server in SMTP communications and in email headers.

- **mydomain = bungkus.org**
  - **Role:** Defines your server's domain.
  - **Purpose:** Used in conjunction with other parameters (like mydestination) to determine which domains are considered local.
- **mydestination = $myhostname, localhost.$mydomain, $mydomain**
  - **Role:** Lists the domains for which your server will accept and deliver email.
  - **Purpose:** Ensures that Postfix only handles messages destined for recognized local domains.
- **home_mailbox = Maildir/**
  - **Role:** Specifies the format and location of user mailboxes.
  - **Purpose:** Directs Postfix to store incoming messages in Maildir format within each user's home directory.

2.1.3 Postfix TLS Configuration

```
Smtpd_use_tls = yes

smtpd_tls_cert_file = /etc/pki/tls/certs/server.crt

# The full pathname of a file with the Postfix SMTP server RSA private key
# in PEM format. The private key must be accessible without a pass-phrase,
# i.e. it must not be encrypted.
#
smtpd_tls_key_file = /etc/pki/tls/private/server.key

# Announce STARTTLS support to remote SMTP clients, but do not require that
# clients use TLS encryption (opportunistic TLS inbound).
#
smtpd_tls_security_level = may
```

Figure 3: Enable Encryption

Step 3: Securing Postfix with TLS Configuration

Configuration File:

Edit /etc/postfix/main.cf to add or update the following lines:

- smtpd_tls_cert_file = /etc/pki/tls/certs/server.pem
- smtpd_tls_key_file = /etc/pki/tls/private/server.pem
- smtpd_tls_security_level = may

Explanation and Purpose:

- **smtpd_tls_cert_file = /etc/pki/tls/certs/server.pem**
  - **Role:** Defines the location of the TLS certificate used by the mail server.
  - **Purpose:** Authenticates the server during TLS handshakes, allowing email clients to verify the server's identity and establish trust.
- **smtpd_tls_key_file = /etc/pki/tls/private/server.pem**
  - **Role:** Points to the private key corresponding to the TLS certificate.
  - **Purpose:** Facilitates the encryption and decryption process during secure email transmission.
- **smtpd_tls_security_level = may**
  - **Role:** Sets TLS usage as optional for incoming SMTP connections.
  - **Purpose:** Encourages encryption when supported by clients but still allows plain text connections for compatibility with older systems.

Reference: (Hat, 2019)

2.1.4 Create TLS Certificate (OpenSSL)

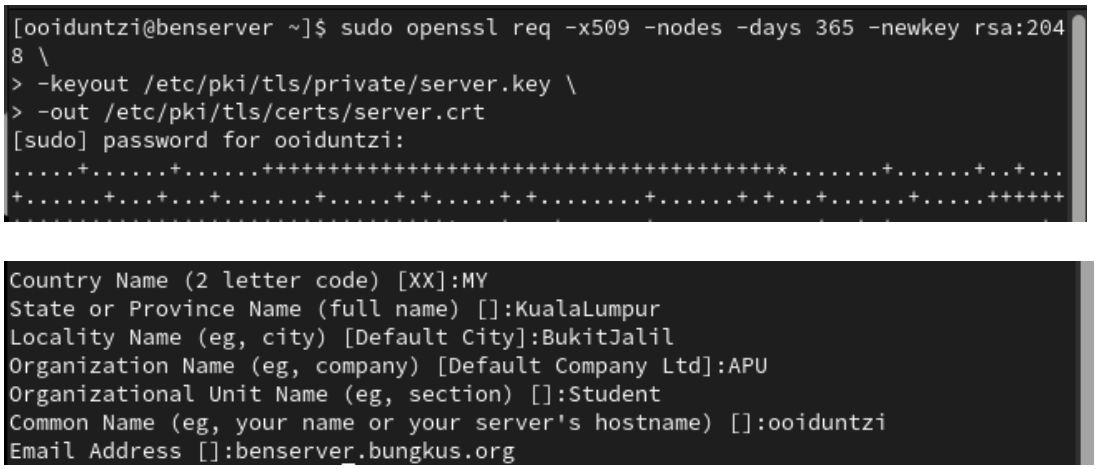


Figure 4: Generate Self Certificate

Step 4: Generating a TLS Certificate with OpenSSL

Command:

- sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
- -keyout /etc/pki/tls/private/server.key \
- -out /etc/pki/tls/certs/server.crt

Explanation and Purpose:

- This command uses OpenSSL to generate a 2048-bit RSA private key and a self-signed X.509 TLS certificate, valid for 365 days. The -nodes option ensures the private key is created without a passphrase, which is required for services like Postfix to access the key non-interactively.

Role in the System:

- **Private Key (server.key):** Used to decrypt secure messages and verify server ownership during TLS handshakes.
- **Certificate (server.crt):** Serves as the public identity of your server, allowing clients to verify they are connecting to the correct host.
- **Purpose:** This self-signed certificate enables encrypted email transmission using TLS, protecting data from unauthorized access or tampering as it travels across networks.

Reference: (Hat, Red Hat, 2025)

2.1.5 Create Test User

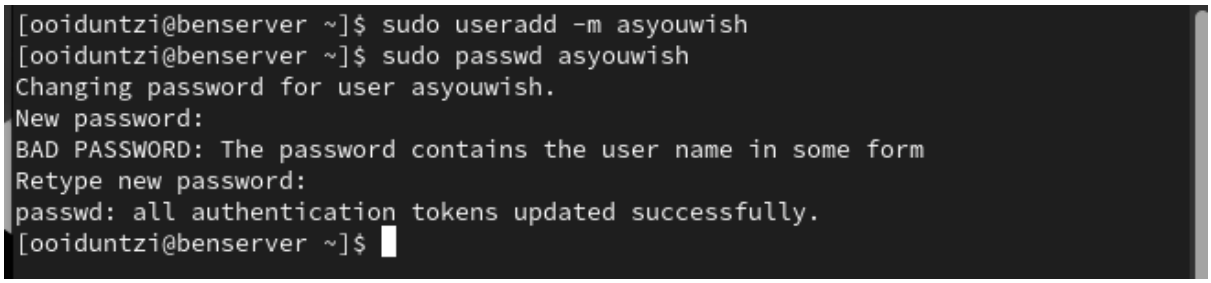


Figure 5: Create a Test User

Step 5: Creating a Test User Account

Commands:

- sudo useradd -m asyouwish
- sudo passwd asyouwish

Explanation:

- These commands create a new system user named asyouwish with a home directory (-m flag) and then set a password for the account using passwd.

Role in the System:

- User Account (asyouwish):
  - Represents a local email recipient on the server.
  - The user's home directory will include a Maildir/ folder where incoming messages are stored (as defined by the home_mailbox setting in Postfix).

Purpose:

- This account is used to test your mail server setup by sending and receiving messages. It helps verify that:
  - Maildir format is correctly implemented.
  - Messages are properly routed and delivered by Postfix.
  - Mailbox permissions and user environment are set up for email access.

2.1.6 Configuration Firewall (Mail Ports)

```
[ooiduntzi@benserver ~]$ sudo firewall-cmd --permanent --add-port={25/tcp,465/tc
p,587/tcp}
success
[ooiduntzi@benserver ~]$ sudo firewall-cmd --reload
success
[ooiduntzi@benserver ~]$
```

Figure 6: For SMTP/SMTPS/Submission Ports

Step 6: Configuring the Firewall for Mail Ports

Commands:

- sudo firewall-cmd --permanent --add-port={25/tcp,465/tcp,587/tcp}
- sudo firewall-cmd --reload

Explanation and Purpose:

- These commands open the standard SMTP ports on the server's firewall to allow inbound and outbound email traffic. The --permanent flag ensures the changes persist after a reboot, and --reload applies the new rules immediately.

Role in the System:

- **Port 25 (SMTP):**
  - Used for server-to-server email delivery across the internet.
- **Port 465 (SMTPS):**
  - A legacy port for SMTP over SSL, still used by some clients for secure mail submission.
- **Port 587 (SMTP with STARTTLS):**
  - The modern standard port for client-to-server email submission using encryption.
- **Purpose:** Opening these ports enables your mail server to send and receive emails securely and reliably, ensuring compatibility with a wide range of email clients and other mail servers.

2.1.7 Enable & Start Postfix

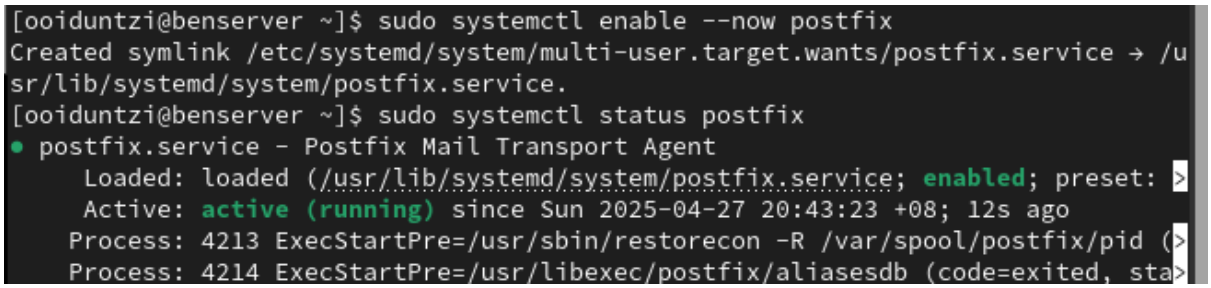


Figure 7: Enable Postfix

Step 7: Enabling and Starting the Postfix Service

Command:

- sudo systemctl enable --now postfix

Explanation and Purpose:

- This command enables the Postfix service to start automatically at boot (enable) and starts it immediately (--now).

Role in the System:

- Ensures that the Postfix mail server is always running and ready to handle email transmission and delivery tasks as soon as the system boots.

Purpose:

- Activates the core mail service so that your server can begin sending, receiving, and routing emails as configured.

2.1.8 Test Postfix Email Delivery

```
[ooiduntzi@benserver ~]$ echo "Test email body" | mail -s "Test Subject" asyouwi
sh@bungkus.org
[ooiduntzi@benserver ~]$ sudo ls /home/asyouwish/Maildir/new
1745757995.Vfd00I302780aM858032.benserver.bungkus.org
1745758000.Vfd00I302780eM379489.benserver.bungkus.org
[ooiduntzi@benserver ~]$
```

Figure 8: Send a Test Email Locally & Verify

Step 8: Testing Postfix Email Delivery

Commands:

- echo "Test email body" | mail -s "Test Subject" asyouwish@bungkus.org
- sudo ls /home/asyouwish/Maildir/new

Explanation and Purpose:

Sending a Test Email:

- echo "Test email body" | mail -s "Test Subject" asyouwish@bungkus.org
    - Purpose: Sends a simple test email to the local user asyouwish using the mail command, confirming that Postfix is capable of handling and delivering messages.

Verifying Email Delivery:

- sudo ls /home/asyouwish/Maildir/new
    - Purpose: Checks the new directory in the user's Maildir, where newly delivered emails are stored. A file here indicates successful delivery.

Outcome: If a new email file appears in /home/asyouwish/Maildir/new, your Postfix setup is working correctly for local delivery.

2.1.9 Integration Postfix with Dovecot (Postfix)

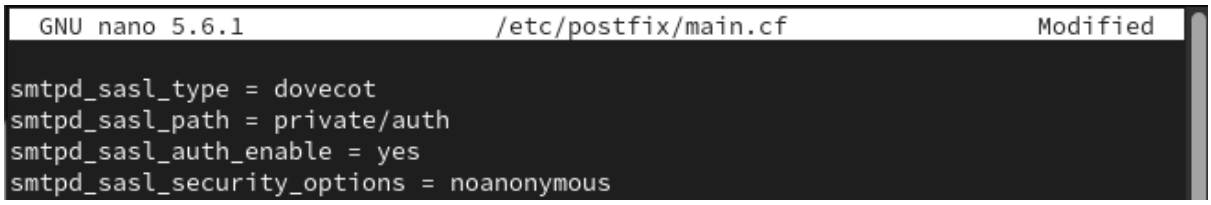


Figure 9: Integration with Dovecot

Step 9: Integrating Postfix with Dovecot for Authentication

To enable SMTP authentication via Dovecot, add the following lines to your Postfix configuration file (/etc/postfix/main.cf):

- smtpd_sasl_type = dovecot
- smtpd_sasl_path = private/auth
- smtpd_sasl_auth_enable = yes
- smtpd_sasl_security_options = noanonymous

Explanation and Purpose:

- **smtpd_sasl_type = dovecot**
  - Role: Specifies Dovecot as the SASL provider for Postfix.
  - Purpose: Enables Postfix to delegate user authentication to Dovecot, which is commonly used for secure and efficient authentication.
- **smtpd_sasl_path = private/auth**
  - Role: Tells Postfix where to find Dovecot's authentication socket.
  - Purpose: Postfix connects to /var/spool/postfix/private/auth to communicate with Dovecot for verifying user credentials.
- **smtpd_sasl_auth_enable = yes**
  - Role: Activates SMTP authentication in Postfix.
  - Purpose: Required for users to authenticate themselves before sending emails, especially when submitting mail from clients.
- **smtpd_sasl_security_options = noanonymous**
  - Role: Enforces secure authentication rules.
  - Purpose: Prevents anonymous logins, requiring valid usernames and passwords for SMTP access.

2.2 Dovecot Installation & Configuration

2.2.1 Dovecot Installation

```
[ooiduntzi@benserver ~]$ sudo dnf install dovecot
Last metadata expiration check: 0:42:23 ago on Sun 27 Apr 2025 08:20:33 PM.
Dependencies resolved.
================================================================================
 Package        Arch    Version                                 Repository  Size
================================================================================
Installing:
 dovecot        x86_64  1:2.3.16-14.el9                         appstream  4.7 M
```

Figure 10: Install Dovecot

Step 10: Installing Dovecot

Command:

- sudo dnf install dovecot

Explanation and Purpose:

- This command installs Dovecot, a widely adopted and secure mail server that supports IMAP and POP3 protocols. Using dnf, the default package manager for Rocky Linux, ensures that Dovecot and its dependencies are downloaded and set up properly.

Role in the System:

- Dovecot plays a crucial role in email delivery and user authentication:
  - Email Access (IMAP/POP3):
    - ❖ Dovecot enables users to retrieve and manage their email using:
      - ➢ IMAP: Emails remain on the server, offering synchronization across multiple devices.
      - ➢ POP3: Emails are typically downloaded to a device and then removed from the server.
  - SMTP Authentication Backend:
    - ❖ Dovecot can serve as the authentication mechanism for Postfix, requiring users to log in before sending emails.
      - ➢ Integrated with: smtpd_sasl_type = dovecot
      - ➢ Provides secure authentication via Unix sockets.

2.2.2 Doveot Configuration

```
protocols = imap pop3

mail_location = maildir:~/Maildir

disable_plaintext_auth = yes

ssl = required
ssl_cert = </etc/pki/tls/certs/server.crt
ssl_key = </etc/pki/tls/private/server.key
```

Figure 11: Configure /etc/dovecot/dovecot.conf

Step 11: Configuring Dovecot Settings

Command:

- sudo nano /etc/dovecot/dovecot.conf

Explanation and Purpose:

- This command opens the main Dovecot configuration file in the Nano text editor, allowing you to adjust critical settings related to protocol support, mail storage, and security. These changes tailor how Dovecot handles email retrieval and authentication.

Key Configuration Directives and Their Roles:

- **protocols = imap pop3**
  - Role: Declares which email retrieval protocols Dovecot should enable.
  - Purpose: Allows clients to connect using either:
    - ❖ IMAP (messages stay on the server and sync across devices)
    - ❖ POP3 (messages are downloaded and typically deleted from the server)
- **mail_location = maildir:~/Maildir**
  - Role: Defines where user emails are stored.
  - Purpose: Sets Maildir as the storage format in each user's home directory. Each email is saved as an individual file in organized folders (new, cur, tmp) for better performance and manageability.
- **disable_plaintext_auth = yes**
  - Role: Blocks unencrypted username and password transmissions.
  - Purpose: Ensures users cannot log in using plain text credentials unless the connection is encrypted, enhancing security.
- **ssl = required**

  - Role: Forces secure communication between clients and the server.
  - Purpose: Mandates the use of SSL/TLS encryption for all connections, protecting email data from interception.
- **ssl_cert = </etc/pki/tls/certs/server.crt**
  - Role: Points to the SSL certificate file used by the server.
  - Purpose: Provides a trusted certificate to identify the server to clients during encrypted sessions.
- **ssl_key = </etc/pki/tls/private/server.key**
  - Role: Indicates the location of the SSL private key.
  - Purpose: Used in conjunction with the SSL certificate to establish and maintain secure, encrypted communication.

2.2.3 Authentication Methods Configuration

```
auth_mechanisms = plain login
!include auth-system.conf.ext
```

Figure 12: Configure /etc/dovecot/conf.d/10-auth.conf

Step 12: Configuring Authentication Methods in Dovecot

Command:

- sudo nano /etc/dovecot/conf.d/10-auth.conf

Explanation and Purpose:

- This command opens the Dovecot authentication configuration file, where you can define which authentication mechanisms the server accepts. Adjusting this setting ensures compatibility with most mail clients and maintains secure login practices, especially when paired with encrypted connections.

Key Configuration Directive and Its Role:

- auth_mechanisms = plain login
    - Role: Specifies the allowed methods for user authentication.
    - Purpose: Enables two commonly used authentication types:
        - ❖ PLAIN: Transmits the username and password in plain text, typically secured within an SSL/TLS encrypted session.
        - ❖ LOGIN: A widely supported method similar to PLAIN, often used by desktop and mobile email clients for compatibility.

2.2.4 Integrating Postfix with Dovecot (Dovecot)

```
# Postfix smtp-auth
unix_listener /var/spool/postfix/private/auth {
  mode = 0660
  user = postfix
  group = postfix
}
```

Figure 13: Postfix Integration /etc/dovecot/conf.d/10-master.conf

Step 13: Integrating Postfix with Dovecot for SMTP Authentication

Command:

- sudo nano /etc/dovecot/conf.d/10-master.conf

Explanation and Purpose:

- This command opens the Dovecot master configuration file, where you define services and their interfaces. The configuration enables communication between Postfix and Dovecot for authenticating users during email submission.

Key Configuration Block and Its Role:

- service auth {
  unix_listener /var/spool/postfix/private/auth {
  mode = 0660
  user = postfix
  group = postfix
  }
  }

Role: Establishes a UNIX socket in a directory accessible to Postfix, allowing it to communicate with Dovecot's authentication daemon.

Purpose: Allows Postfix to use Dovecot for SMTP AUTH, meaning users must log in (authenticate) before sending emails. This enhances mail server security and prevents unauthorized message relaying.

2.2.5 Secure Email Ports Through Firewall

```
[ooiduntzi@benserver ~]$ sudo firewall-cmd --permanent --add-port={993/tcp,995/t
cp}
success
[ooiduntzi@benserver ~]$ sudo firewall --reload
sudo: firewall: command not found
[ooiduntzi@benserver ~]$ sudo firewall-cmd --reload
success
```

Figure 14: Open Ports 993 and 995

Step 14: Allowing Secure Email Ports Through the Firewall

Commands:

- sudo firewall-cmd --permanent --add-port={993/tcp,995/tcp}
- sudo firewall-cmd --reload

Explanation and Purpose:

- These commands configure the system's firewall to allow encrypted email traffic by permanently opening the necessary ports and reloading the firewall rules to apply the changes.

Details:

- sudo firewall-cmd --permanent --add-port={993/tcp,995/tcp}
    - Role: Adds exceptions in the firewall for secure IMAP and POP3 traffic.
    - Purpose:
        - ❖ Port 993 (IMAPS): Enables secure access to email via the IMAP protocol over SSL/TLS.
        - ❖ Port 995 (POP3S): Enables secure access to email via the POP3 protocol over SSL/TLS.
- sudo firewall-cmd --reload
    - Role: Applies the new firewall settings.
    - Purpose: Reloads the firewall to enforce the newly added permanent rules without rebooting the system.

Reference: (Hat, Red Hat, 2019)

2.2.6 Enable & Start Dovecot

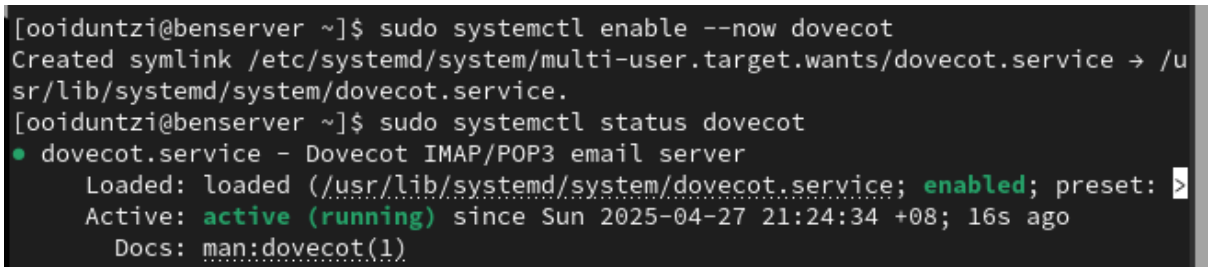


Figure 15: Enable Dovecot

Step 15: Enabling and Starting the Dovecot Service

Command:

- sudo systemctl enable --now dovecot

Explanation and Purpose:

- This command ensures that the Dovecot service is both started immediately and configured to launch automatically at system boot.

Details:

- --now: Starts the Dovecot service right away.
- enable: Configures the system to start Dovecot automatically on every reboot.

Role in the System: Activates Dovecot to handle secure email retrieval and authentication services without requiring manual startup each time the server restarts.

2.2.7 Test IMAP SSL Authentication (OpenSSL)

```
[ooiduntzi@benserver ~]$ openssl s_client -connect localhost:993 -quiet
Connecting to ::1
Can't use SSL_get_servername
depth=0 OU=IMAP server, CN=imap.example.com, emailAddress=postmaster@example.com
verify error:num=18:self-signed certificate
verify return:1
depth=0 OU=IMAP server, CN=imap.example.com, emailAddress=postmaster@example.com
verify return:1
* OK [CAPABILITY IMAP4rev1 SASL-IR LOGIN-REFERRALS ID ENABLE IDLE LITERAL+ AUTH=
PLAIN AUTH=LOGIN] Dovecot ready.
a login asyouwish asyouwish
a OK [CAPABILITY IMAP4rev1 SASL-IR LOGIN-REFERRALS ID ENABLE IDLE SORT SORT=DISP
LAY THREAD=REFERENCES THREAD=REFS THREAD=ORDEREDSUBJECT MULTIAPPEND URL-PARTIAL
CATENATE UNSELECT CHILDREN NAMESPACE UIDPLUS LIST-EXTENDED I18NLEVEL=1 CONDSTORE
 QRESYNC ESEARCH ESORT SEARCHRES WITHIN CONTEXT=SEARCH LIST-STATUS BINARY MOVE S
NIPPET=FUZZY PREVIEW=FUZZY PREVIEW STATUS=SIZE SAVEDATE LITERAL+ NOTIFY SPECIAL-
USE] Logged in
a logout
* BYE Logging out
a OK Logout completed (0.001 + 0.000 secs).
[ooiduntzi@benserver ~]$
```

Figure 16: Test IMAP via Telnet

Step 16: Testing IMAP SSL Authentication Using OpenSSL

Command:

- openssl s_client -connect localhost:993 -quiet

Explanation and Purpose:

- This command initiates a secure connection to the local Dovecot IMAP server over port 993 using OpenSSL, a powerful tool for testing encrypted services.

Role in the System:

- **Verifies that Dovecot's IMAP service is:**
  - Listening on port 993
  - Accepting SSL/TLS-secured connections

Follow-up Command (Inside OpenSSL Prompt):

- a login asyouwish asyouwish
  - Role: Sends a raw IMAP login request using a test username and password.
  - Purpose: Confirms that Dovecot correctly processes authentication requests and allows valid user logins through its IMAP SSL interface.

2.2.8 Verify Delivery of New Mail

```
[asyouwish@benserver ~]$ ls /home/asyouwish/Maildir/new
1745757995.Vfd00I302780aM858032.benserver.bungkus.org
1745758000.Vfd00I302780eM379489.benserver.bungkus.org
1745758731.Vfd00I30276a1M184698.benserver.bungkus.org
[asyouwish@benserver ~]$
```

Figure 17: Check Emails are Stored

Step 17: Verifying Delivery of New Mail in Maildir Format

Command:

- ls /home/asyouwish/Maildir/new

Explanation and Purpose:

- This command lists all files in the new directory of the specified user's Maildir, which is where unread email messages are initially stored.

Role in the System:

- Checks whether any new emails have been delivered to the user’s mailbox.

Details:

- In the Maildir storage structure, incoming messages are first saved in the new folder.
- Once accessed by a mail client, messages are typically moved to the cur folder.

2.3 Thunderbird Installation & Configuration

2.3.1 Thunderbird Installation (Ubuntu Client)

```
Benooi@benclient:~/Desktop$ sudo apt update && sudo apt install thunderbird
[sudo] password for Benooi:
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:2 http://my.archive.ubuntu.com/ubuntu noble InRelease
Get:3 http://my.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
```

Figure 18: Install Thunderbird

Step 18: Installing Thunderbird Email Client on Ubuntu

Command:

- sudo apt update && sudo apt install thunderbird

Explanation and Purpose:

- This command updates the package index and installs Thunderbird, a widely-used graphical email client that supports IMAP, POP3, and SMTP.

Role in the System:

- Provides a user-friendly interface to send, receive, and manage emails, making it easier to interact with your configured mail server.

Details:

- Thunderbird allows real-world testing of your email setup using secure protocols.
- It helps confirm that services like Dovecot (for receiving mail) and Postfix (for sending mail) are working properly from the end-user's perspective.

2.3.2 Configuration Thunderbird (IMAP & SMTP)

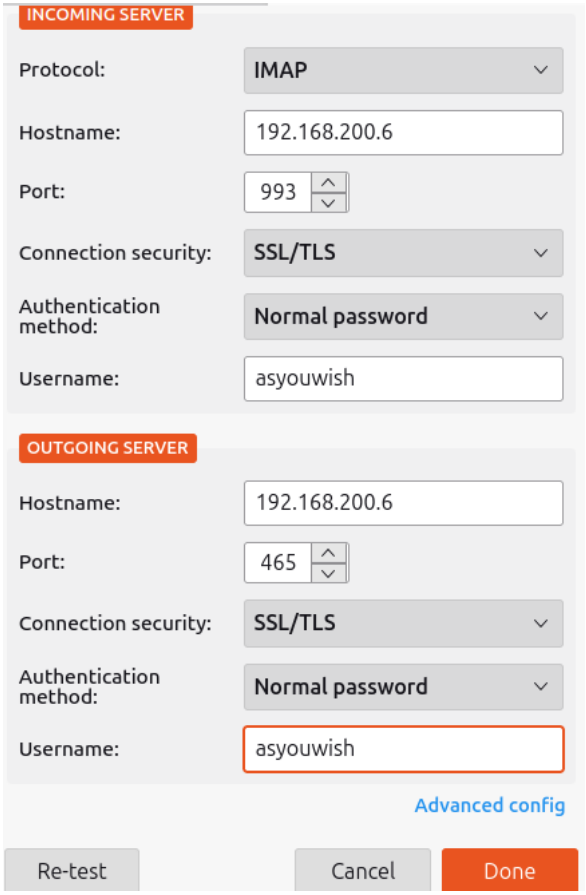


Figure 19: Account Creation

Step 19: Configuring Thunderbird with Secure IMAP and SMTP Settings

Email Client Settings (Thunderbird)

Incoming Mail (IMAP):

- Server Address: 192.168.200.6 (internal mail server IP)
- Port: 993
- Connection Security: SSL/TLS
- Authentication Method: Normal password
- Username: asyouwish

Outgoing Mail (SMTP):

- Server Address: 192.168.200.6
- Port: 465
- Connection Security: SSL/TLS

- Authentication Method: Normal password
- Username: asyouwish

Explanation and Purpose:

- This configuration sets up Thunderbird to securely communicate with your mail server over encrypted channels. Both IMAP and SMTP connections are protected using SSL/TLS to ensure the confidentiality of your credentials and email content.

Role in the System:

- IMAP over SSL (port 993): Retrieves emails securely from the Dovecot server.
- SMTP over SSL (port 465): Sends emails securely through Postfix with user authentication.

2.3.3 Add Security Exception

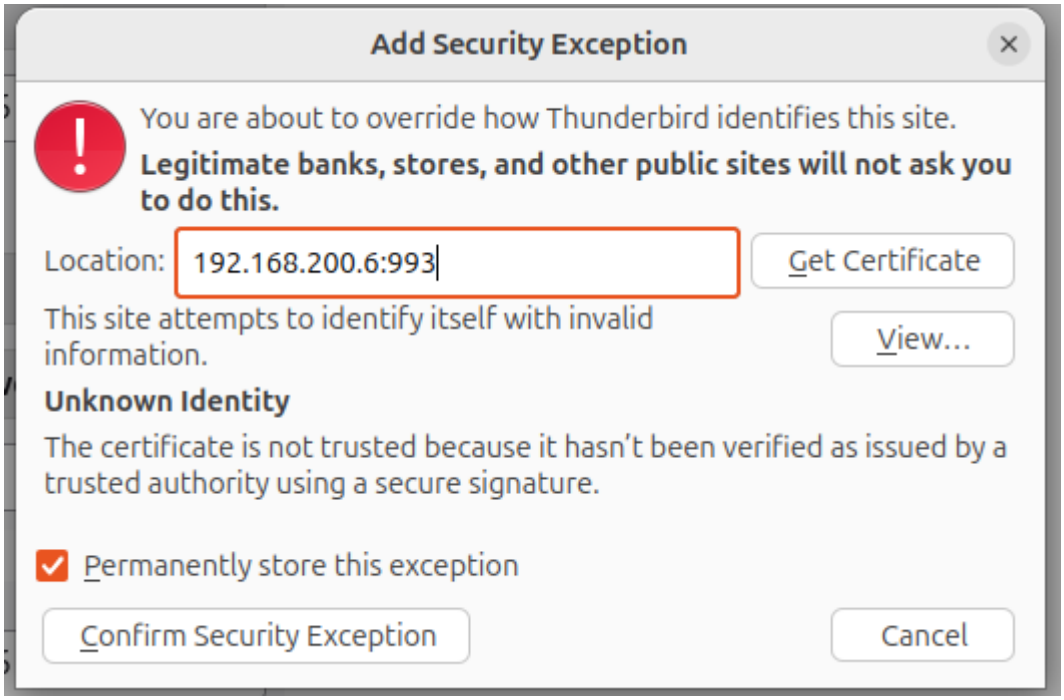


Figure 20: SSL/TLS Encryption

Step 20: Allow Security Exception

Action: Click on "Confirm Security Exception"

Purpose: This step allows the email client to trust and communicate with the mail server despite the use of a self-signed or unverified SSL certificate.

2.3.4 Access Thunderbird Interface

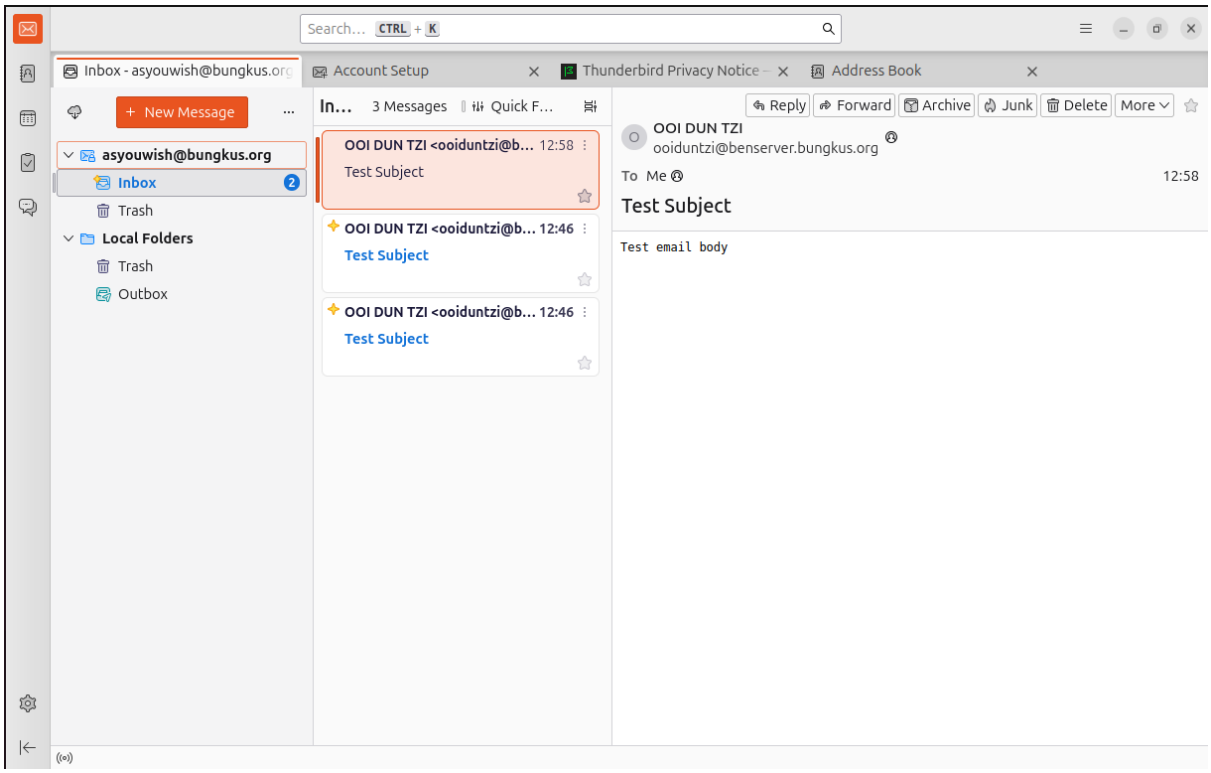


Figure 21: Email Receive Evidence

Step 21: Accessing Thunderbird Interface

Action: Open Thunderbird and log in with the email account asyouwish@bungkus.org.

Purpose: This confirms successful configuration of the email client. You should be able to view messages in the inbox, including those previously sent from benserver.bungkus.org.

2.3.5 Enable SMTP Authentication

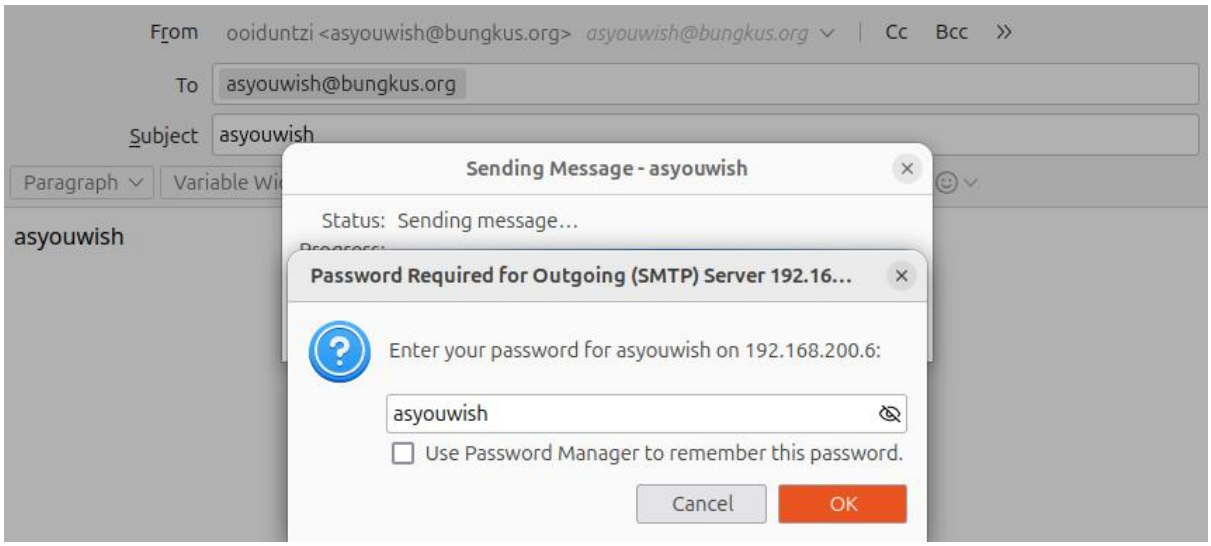


Figure 22: SMTP Requirement

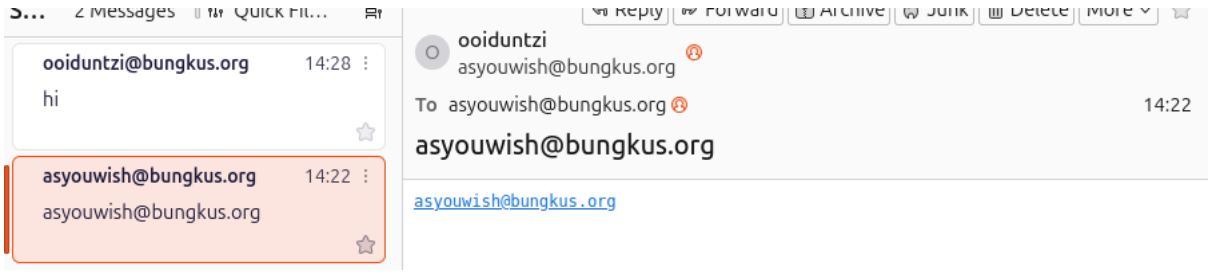


Figure 23: Send Message to Client

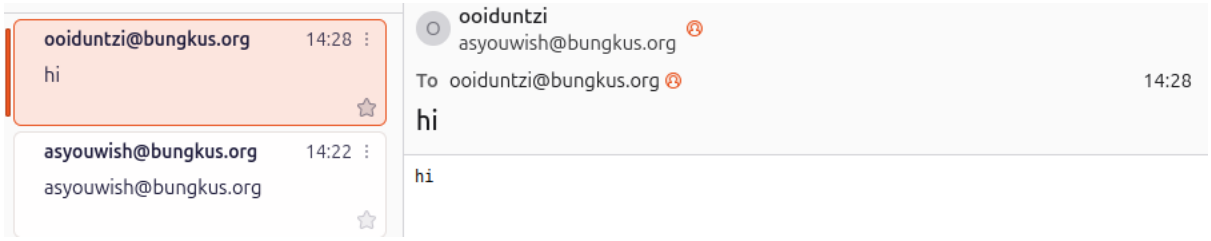


Figure 24: Send Message to Server

Step 22: Enable SMTP Authentication for Outgoing Mail

Role in the System: This ensures the Thunderbird client securely authenticates with the Postfix SMTP server before sending any emails.

Purpose: To confirm that only authorized users are permitted to send emails, thereby enhancing security and preventing unauthorized usage of the mail server.

2.3.6 Verity Email Delivery (Rocky Linux Server)

```
[ooiduntzi@benserver ~]$ sudo ls /home/ooiduntzi/Maildir/new
[sudo] password for ooiduntzi:
1745764131.Vfd00I30546b0M587614.benserver.bungkus.org
[ooiduntzi@benserver ~]$
```

Figure 25: Check Emails are Stored

Step 23: Verify Email Delivery on Server

Command:

- sudo ls /home/ooiduntzi/Maildir/new

Purpose: To check that emails sent from the client have successfully arrived on the server and are stored in the correct Maildir location.

2.3.7 Check Validation Secure Transmission (Ubuntu Client)

```
Received: from [192.168.200.80] (unknown [192.168.200.80])
    by benserver.bungkus.org (Postfix) with ESMTPSA id B5E6321B89E0
    for <asyouwish@bungkus.org>; Sun, 27 Apr 2025 22:22:54 +0800 (+08)
```

Figure 26: ESMTPSA Header Indicating Secure Authenticated Email

Step 24: Validate Secure Transmission of Emails

Action: In Thunderbird, go to the Inbox > select the sent message > click "More" > choose "View Source".

Purpose: To inspect the email headers and confirm that the message was encrypted and transmitted securely.

3.0 Extra Feature Spam Assassin

3.1 Spam Assassin Installation & Configuration

3.1.1 Spam Assassin Installation

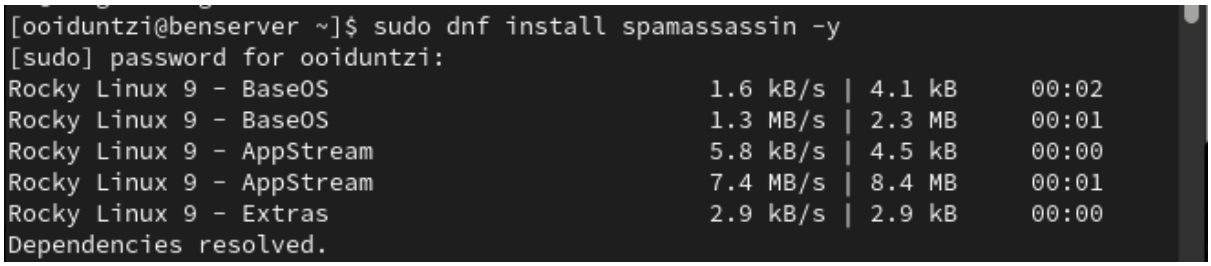


Figure 26: Install Spam Assassin

Step 1: Installing SpamAssassin

Command:

- sudo dnf install spamassassin -y

Explanation and Purpose:

- This command installs SpamAssassin, an open-source and widely-used spam filtering software designed to analyze and classify email messages based on their likelihood of being spam. By using the dnf package manager with the -y flag, the installation proceeds without prompting for confirmation.

Role in the System:

- SpamAssassin serves as the core engine of your spam detection framework. It employs a variety of sophisticated techniques such as:
    - **Text analysis** to identify common spam patterns or keywords.
    - **Bayesian filtering** to learn from previously categorized spam and non-spam messages.
    - **DNS blocklists** to assess the reputation of sender domains.
    - **Collaborative filtering** through shared databases of known spam content.

Once integrated, SpamAssassin assigns a spam score to each email and enables automated actions like flagging, tagging, moving, or discarding spam messages based on configurable thresholds.

Reference: (Vultr, 2025)

3.1.2 Check Spam Assassin Version

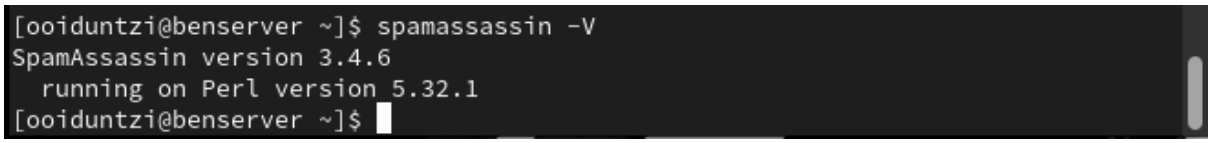


Figure 27: Check Spam Assassin Version

Step 2: Checking the Installed Version of SpamAssassin

Command:

- spamassassin -V

Explanation and Purpose:

- This command retrieves and displays the currently installed version of SpamAssassin. Confirming the version ensures the software has been installed successfully and is functioning as expected.

Role in the System:

- **Verification**: Confirms the presence and accessibility of SpamAssassin.
- **Dependency Management**: Helps in determining compatibility with other software components and rulesets.
- **Troubleshooting**: The version number is essential when seeking support, as solutions often vary between versions.
- **Security Maintenance**: Verifying the version helps assess whether updates or patches are required.
- **Documentation Reference**: Ensures that you consult the correct version of SpamAssassin's documentation for accurate configuration and usage.

3.1.3 Spam Assassin Configuration

Figure 28: Configure /etc/sysconfig/spamassassin

Step 3: Configuring SpamAssassin

Command:

- sudo nano /etc/sysconfig/spamassassin

Explanation and Purpose:

- This command opens the SpamAssassin configuration file using the nano text editor with superuser privileges. The file contains startup parameters that determine how the SpamAssassin daemon (spamd) operates when launched.

Role in the System:

- This step is essential for customizing SpamAssassin's runtime behavior. The default parameters may be adjusted to suit performance, security, and operational needs. Example options:
    - **--create-prefs:** Automatically generates user preference files, enabling personalized spam filtering if needed.
    - **--max-children 5:** Limits the number of simultaneous child processes used for spam filtering. This setting is crucial for balancing performance and resource consumption.
    - **--helper-home-dir:** Specifies a home directory for helper processes, such as those handling Bayesian filtering.
    - **-u nobody:** Ensures that the spamd process runs under an unprivileged user for enhanced security. If compromised, the process has minimal system access, reducing the risk of broader attacks.

Impact: Proper configuration here ensures SpamAssassin runs efficiently and securely, with appropriate limits on system resource usage. It establishes the foundation for robust and scalable spam filtering operations.

3.1.4 Enable & Start Spam Assassin

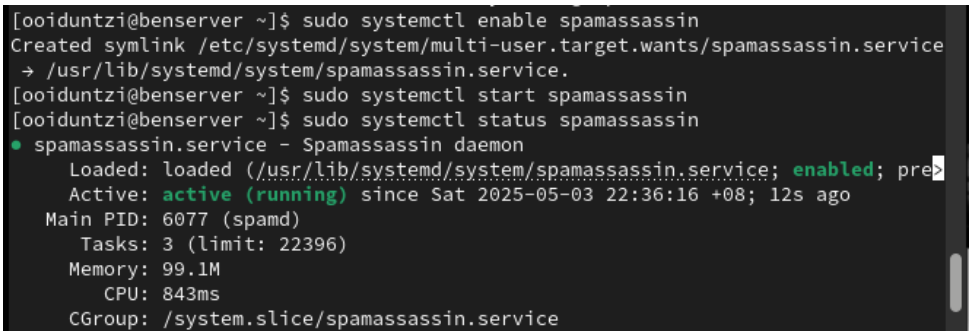


Figure 29: Enable, Start, Start, & Start Spam Assassin

Step 4: Enabling and Starting the SpamAssassin Service

Commands:

- sudo systemctl enable spamassassin
- sudo systemctl start spamassassin
- sudo systemctl status spamassassin

Explanation and Purpose:

- **enable:** Configures SpamAssassin to automatically start at system boot.
- **start:** Launches the SpamAssassin service immediately.
- **status:** Displays the current status of the service, including whether it is running, enabled, and any recent logs.

Role in the System:

- **Automation**: Ensuring SpamAssassin starts on boot guarantees persistent spam protection even after system reboots.
- **Activation**: Starting the service activates the daemon responsible for real-time email filtering.
- **Monitoring and Troubleshooting**: Verifying the service status is a critical checkpoint to ensure that installation and configuration steps have led to a fully operational system. It also aids in early detection of issues through log visibility.

Outcome: These commands transition SpamAssassin from a passive installation to an active system component. It begins monitoring and scoring incoming emails based on your defined rules and configurations, enabling the spam filtering system to operate in real time.

3.1.5 Postfix Configuration (master.cf)

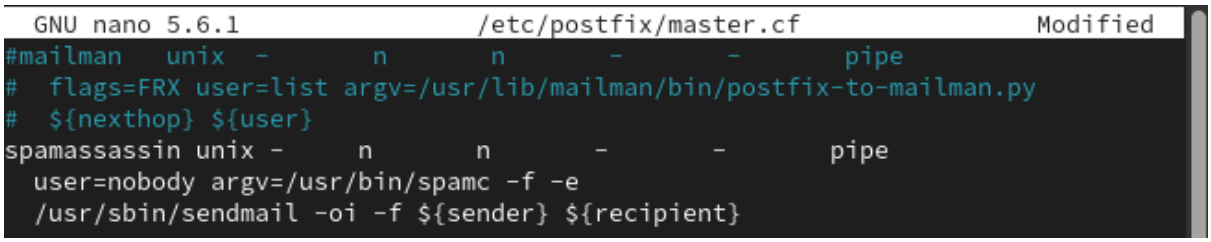


Figure 30: Configure /etc/postfix/master.cf

Step 5: Configure Postfix master.cf for SpamAssassin Integration

Action: Edit the Postfix master configuration file (/etc/postfix/master.cf) and add the following lines at the bottom

- spamassassin unix - n n - - pipe
  user=nobody argv=/usr/bin/spamc -f -e
  /usr/sbin/sendmail -oi -f ${sender} ${recipient}

Explanation and Purpose:

- This step defines a new custom Postfix transport service named spamassassin. The service uses a Unix domain socket and invokes the SpamAssassin client (spamc) to scan the email. The pipe command is used to pass the message through spamc, and then to sendmail, ensuring that the message is analyzed before final delivery.
- **user=nobody:** Ensures the command runs with minimal privileges, reducing security risks.
- **spamc -f -e:** Passes the message through the SpamAssassin client, forwarding it to sendmail only if it's successfully processed.

Role in the System:

- This configuration allows Postfix to offload incoming mail to SpamAssassin before it is accepted for final delivery. It essentially enables real-time spam filtering during the SMTP pipeline.

3.1.6 Postfix Configuration (main.cf)

Figure 31: Configure /etc/postfix/main.cf

Step 6: Configure Postfix main.cf to Use SpamAssassin

Action: Add the following line to the /etc/postfix/main.cf file

- content_filter = spamassassin

Explanation and Purpose:

- This line tells Postfix to use the spamassassin content filter defined in the master.cf file. It links the mail processing pipeline with the scanning engine configured in Step 5.

Role in the System:

- By enabling this content filter, all incoming mail messages will be redirected through SpamAssassin before they are finally delivered. This ensures centralized spam scanning for all emails handled by Postfix.

3.1.7 Reload Postfix

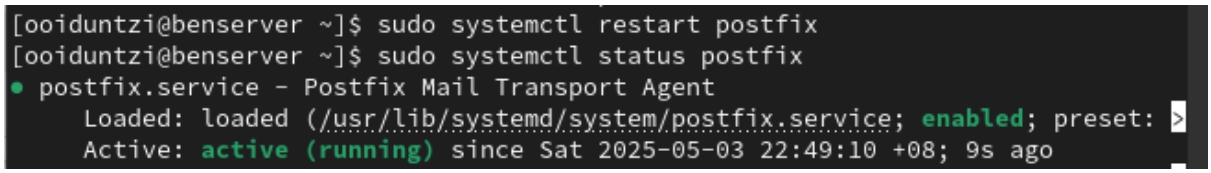


Figure 32: Restart & Status Postfix

Step 7: Reload Postfix Configuration

Command:

- sudo systemctl restart postfix

Explanation and Purpose:

- Restarting Postfix reloads its configuration files (main.cf and master.cf) and applies all changes made in the previous steps.

Role in the System:

- This step activates the newly defined SpamAssassin filtering service. It's essential to ensure that all changes related to content filtering take effect without needing a system reboot.

3.1.8 Apply Spam Assassin Scoring & Rules

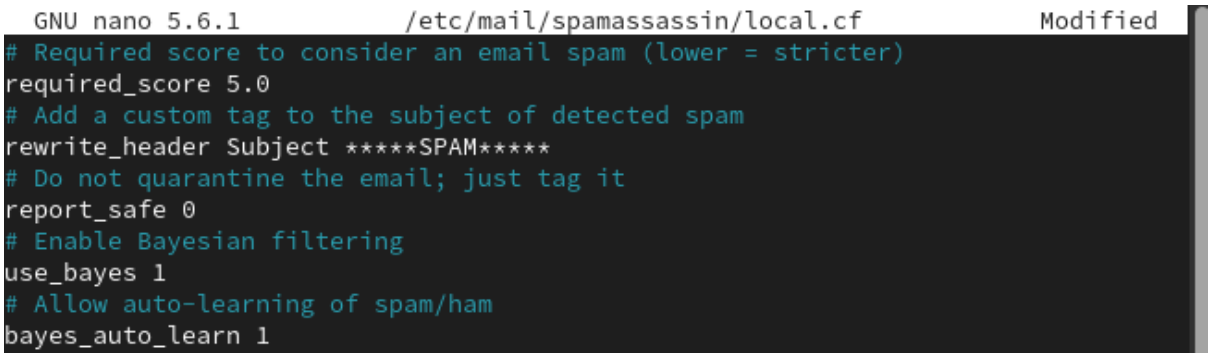


Figure 33: Configure /etc/mail/spamassassin/local.cf

Step 8: Adjust SpamAssassin Scoring and Rules as Needed

Action: Edit the /etc/spamassassin/local.cf file to adjust settings such as

- required_score 5.0
- rewrite_header Subject *****SPAM*****
- report_safe 0
- use_bayes 1
- bayes_auto_learn 1

Explanation and Purpose:

- This step fine-tunes SpamAssassin's behavior to suit your specific needs:
- Adjusting required_score changes how strict the spam filter is.
- Header rewriting helps visually mark spam in users' inboxes.
- Whitelists and blacklists override scoring for specific senders.

Role in the System:

Custom rules enhance detection accuracy and reduce false positives or negatives. They provide administrators with control over how spam is identified and treated based on organizational requirements.

3.1.9 Reload Spam Assassin

```
[ooiduntzi@benserver ~]$ sudo systemctl restart spamassassin
[ooiduntzi@benserver ~]$
```

Figure 34: Restart Spam Assassin

Step 9: Automate SpamAssassin Rule Updates (Optional but Recommended)

Command:

- sudo systemctl restart spamassassin

Explanation and Purpose:

- SpamAssassin rules evolve frequently to keep up with new spam trends. The sa-update command fetches the latest rule updates from official channels. Restarting the service applies these new rules immediately.

Role in the System:

- Keeping SpamAssassin updated ensures it remains effective against the latest spam tactics. Regular updates improve accuracy, maintain relevance, and enhance the overall robustness of the filtering system.

3.1.10 Send Test Email (Thunderbird)

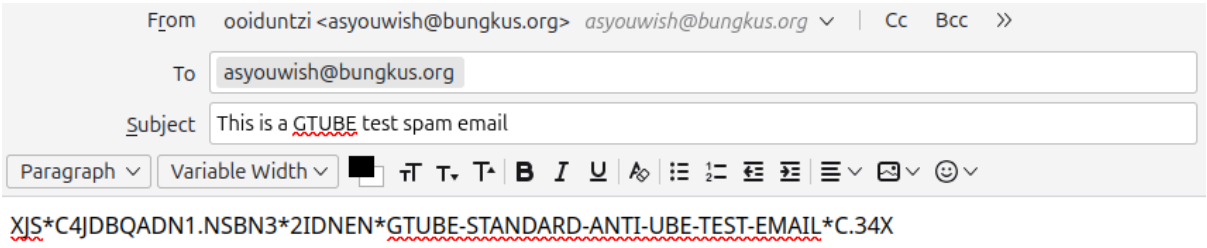


Figure 35: Test Spam Email

Step 10: Send a Test Email Using Thunderbird

Action:

- Open **Thunderbird** and compose a new email.
- Set the **Subject** to: This is a GTUBE test spam email
- In the **email body**, paste the following GTUBE test string (used to trigger SpamAssassin's spam detection): XJS*C4JDBQADN1.NSBN3*2IDNEN*GTUBE-STANDARD-ANTI-UBE-TEST-EMAIL*C.34X
- Send the email to own address (or another test mailbox).

Explanation and Purpose:

- This step uses Thunderbird to send a test email containing the GTUBE (Generic Test for Unsolicited Bulk Email) string—a predefined pattern that simulates a typical spam message. When SpamAssassin is correctly installed and integrated, it should immediately identify the message as spam. Upon detection, SpamAssassin will either mark the email with spam headers.

Role in the System:

This GTUBE test email acts as a reliable benchmark to confirm that your spam filtering system is working end-to-end. It allows administrators to verify that SpamAssassin is fully operational and integrated with the mail server's processing pipeline. If the email is not flagged or moved as expected, it serves as a starting point for troubleshooting, guiding you to review SpamAssassin logs, Postfix settings, or filtering rules.

By simulating a real-world spam scenario in a controlled manner, this step ensures the system is ready to detect and handle actual spam effectively.

Reference: (Eset, 2025)

3.1.11 Spam Checking

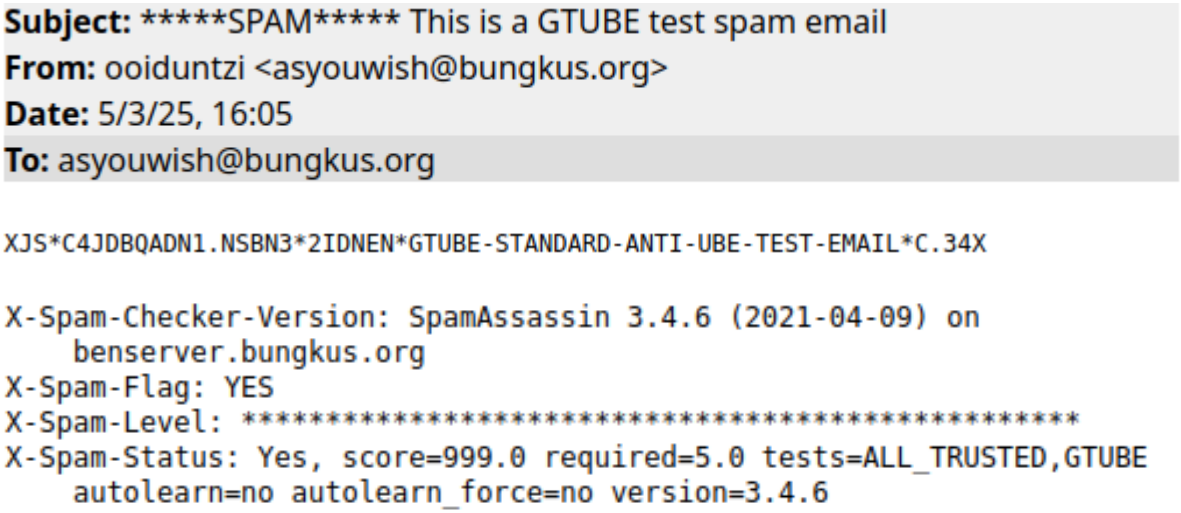


Figure 36: Spam Test Result

Step 11: Check Headers in the Received Email

Action: Open the email received and inspect its headers. You should see lines such as

- X-Spam-Flag: YES
- X-Spam-Score: 999.0
- X-Spam-Status: Yes, score=999.0 required=5.0 tests=...

Explanation and Purpose:

- These headers are inserted by SpamAssassin to indicate how it evaluated the message. Key indicators include whether the message was flagged as spam, the numerical score assigned, and which rules were triggered.

Role in the System:

- These headers provide transparency into SpamAssassin's decision-making process. They are also critical for debugging, tuning spam thresholds, and informing downstream mail processing tools or users.

4.0 Extra Features Attachment Filtering

4.1 Postfix Configuration (Attachment Filtering)

Figure 37: Configure /etc/postfix/main.cf

Step 1: Configuring Postfix for Attachment Filtering

Action:

- mime_header_checks = regexp:/etc/postfix/mime_header_checks

Explanation and Purpose:

- This configuration line is added to the Postfix main configuration file to enable attachment filtering based on MIME headers. By specifying the mime_header_checks directive and linking it to a regular expression file (/etc/postfix/mime_header_checks), Postfix can inspect the MIME headers of incoming emails and apply custom rules to detect and block specific types of attachments.

Role in the System:

- This setting enhances the email security system by allowing administrators to define patterns that identify potentially harmful or unwanted attachments. It plays a crucial role in preventing the delivery of risky file types (such as .exe or .js) by scanning headers for suspicious content. When a match is found, Postfix can reject, quarantine, or flag the message, helping reduce the risk of malware and phishing attacks delivered via email attachments.

4.2 Attachment Filtering Rules Setting

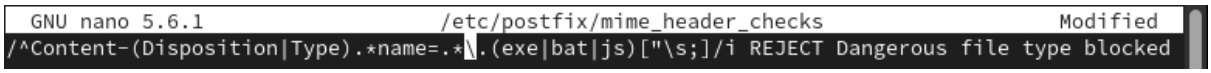


Figure 38: Configure /etc/postfix/mime_header_checks

Step 2: Setting Up Attachment Filtering Rules

Action: Add the following line to /etc/postfix/mime_header_checks:

- /^Content-(Disposition|Type).*name=*\.(exe|bat|js)[“\s;”]/i REJECT Dangerous file type blocked

Explanation and Purpose:

- This rule defines a regular expression that matches email attachments with file extensions commonly associated with potentially harmful content—specifically .exe, .bat, and .js files. The Content-Disposition and Content-Type headers in MIME-encoded emails are scanned for filenames that match these extensions. If such a match is found, the message is immediately rejected with a custom message: “Dangerous file type blocked.”

Role in the System:

- This filter strengthens the Postfix mail server’s defense against malware and scripting attacks delivered via email attachments. By rejecting messages with high-risk file types at the mail gateway level, this rule helps to reduce the chance of users inadvertently executing malicious files and contributes to a more secure email environment.

4.3 Reload Postfix

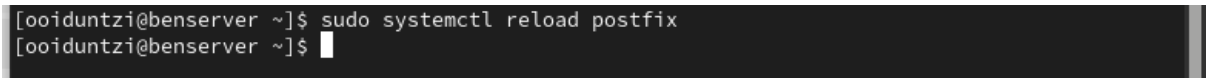


Figure 39: Reload Postfix

Step 3: Reloading Postfix to Apply Changes

Command:

- sudo systemctl reload postfix

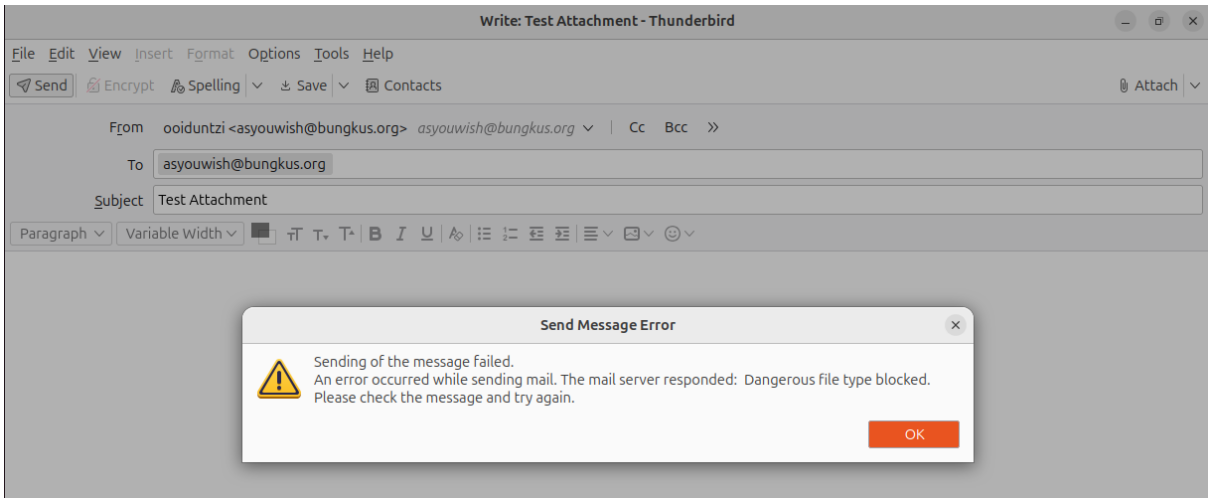
Explanation and Purpose:

- This command reloads the Postfix mail server, applying any configuration changes made—such as the addition of mime_header_checks or updates to its rules file—without fully stopping and restarting the service. It ensures that new settings take effect immediately while maintaining ongoing mail processing.

Role in the System:

- Reloading Postfix is a crucial step to activate the newly implemented attachment filtering rules. Without this, changes made in the configuration files would not be enforced, leaving the system vulnerable to the file types it was meant to block. This action ensures the mail server runs with the latest security enhancements in place.

4.4 Test Attachment Filtering Feature




```
May  4 01:41:52 benserver postfix/cleanup[53971]: B241521B8867: reject: header Content-Type: ap
plication/x-ms-dos-executable; name="test.exe" from unknown[192.168.200.80]; from=<asyouwish@bu
ngkus.org> to=<asyouwish@bungkus.org> proto=ESMTP helo=<[192.168.200.80]>: 5.7.1 Dangerous file
 type blocked
```

Figure 40: Attachment Filtering Result

Step 4: Testing the Attachment Filtering Feature

Action: Open Thunderbird, compose a new email, and attach a file with a .exe extension. Send the email to your Postfix server.

Explanation and Purpose:

- This step is intended to verify that the MIME header filtering rule is working as expected. By attempting to send an email with a potentially dangerous attachment (e.g., a .exe file), the system should trigger the rule defined in /etc/postfix/mime_header_checks.

Role in the System:

- Successful testing confirms that the Postfix mail server correctly detects and blocks restricted file types, providing feedback such as a rejection message stating "Dangerous file type blocked." This ensures that the filtering mechanism is active and functioning properly, enhancing your email security by preventing the delivery of potentially harmful attachments.

5.0 Troubleshoot

5.1 Troubleshooting Email System

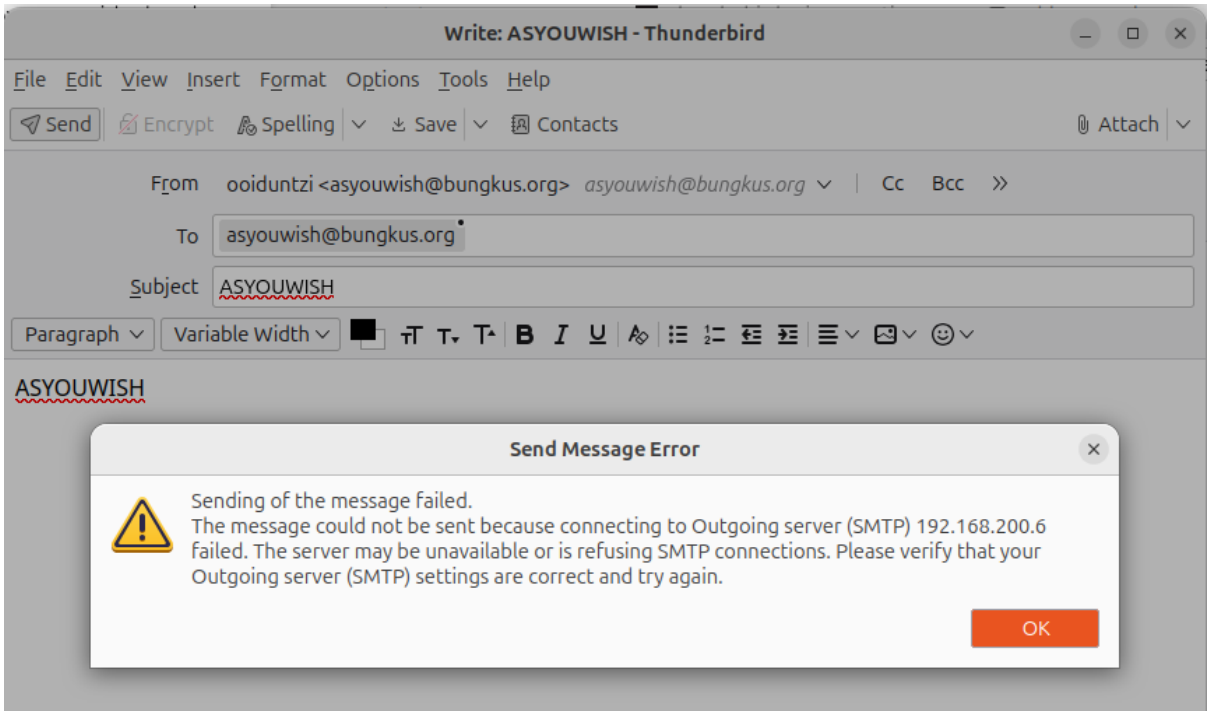


Figure 41: Send Message Error

Troubleshooting: Unable to Send Messages – Connection to Outgoing Server Failed

Issue: When attempting to send an email, a notification appears

- "The message could not be sent because the connection to the outgoing server 192.168.200.6 failed."

Explanation and Purpose:

This error typically indicates that the email client (e.g., Thunderbird) is unable to establish a connection with the Postfix SMTP server at the specified IP address. Common causes include misconfigured server settings, network issues, or Postfix not running properly.

5.1.1 Verify Client-Server Connection

```
Benooi@benclient:~/Desktop$ telnet 192.168.200.6 25
Trying 192.168.200.6...
Connected to 192.168.200.6.
Escape character is '^]'.
220 benserver.bungkus.org ESMTP Postfix
421 4.4.2 benserver.bungkus.org Error: timeout exceeded
Connection closed by foreign host.
```

Figure 42: Check SMTP Port 25

Step 1: Verify Client-Server Connection

Action: On the client Ubuntu machine, run the following commands to test connectivity to the mail server:

- telnet 192.168.200.6 25

Explanation and Purpose:

- These commands check whether the client can establish a connection to the Postfix server on the standard SMTP ports:
    - Port 25 is typically used for mail server-to-server communication or internal sending.

Observation:

- Port 25: Connection is successful – this indicates that the SMTP service is listening and reachable.

Role in the System: This test helps diagnose network-level issues and confirms which ports are open for mail communication. Since port 25 is working, the basic SMTP service is operational.

5.1.2 Adjust Thunderbird SMTP Setting

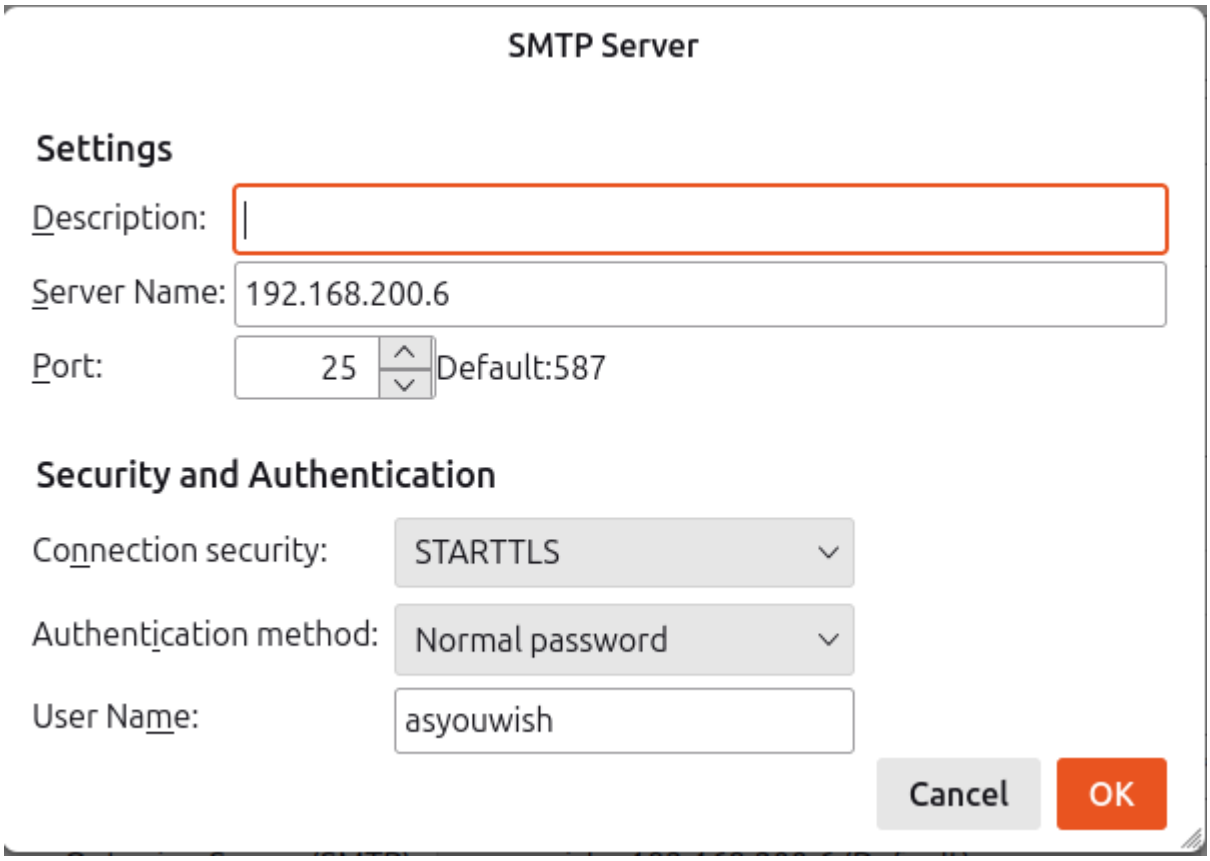


Figure 43: Change to STARTTLS & Port 25

Step 2: Adjust Thunderbird SMTP Settings to Use Port 25 with STARTTLS

Action: In Thunderbird, update the SMTP server settings as follows:

- Port: 25
- Connection Security: STARTTLS
- Then attempt to send an email again.

Explanation and Purpose:

- Using port 25 with STARTTLS allows the client to initiate a secure connection to the Postfix server after the connection is established. This is a commonly supported and secure method for sending emails if port 465 is unavailable.

Role in the System: This step confirms whether the Thunderbird client can securely send emails using the available port. If successful, it demonstrates that STARTTLS is functioning correctly on port 25, enabling encrypted communication without needing port 465.

5.1.3 Add Seciruty Exxception

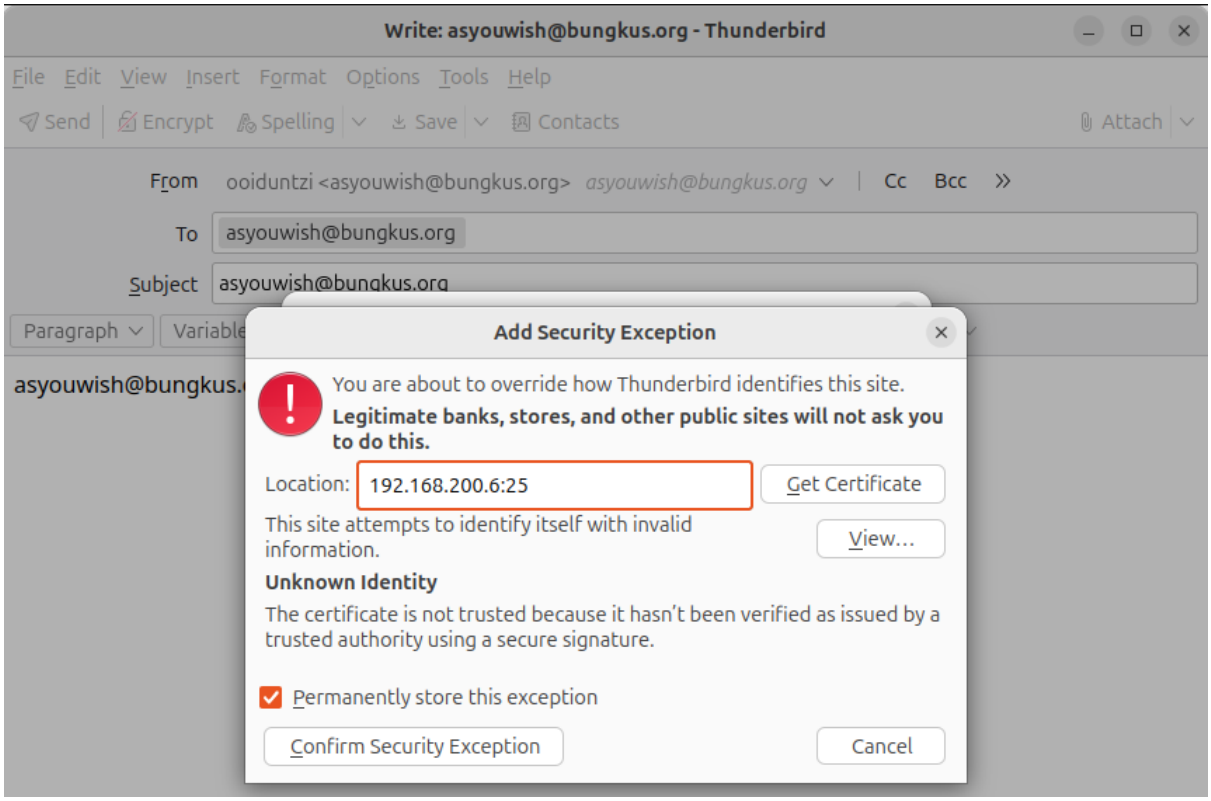


Figure 44: Confirm Security Exception of Port 25

Step 3: Confirm Security Exception for Mail Server

Action: Before sending the email, Thunderbird may prompt you to confirm a security exception for the server at 192.168.200.6:25. Accept and confirm this exception to proceed.

Explanation and Purpose:

- Since you're using STARTTLS with a self-signed or untrusted certificate, Thunderbird treats the connection as potentially insecure. It prompts the user to manually verify and accept the certificate to ensure you're intentionally connecting to the specified mail server.

Role in the System:

- Confirming the security exception is necessary to establish a secure connection with the mail server over STARTTLS. This step ensures the email client trusts the server's certificate and can proceed with encrypted communication.

5.2 Troubleshooting Port 465 Connection

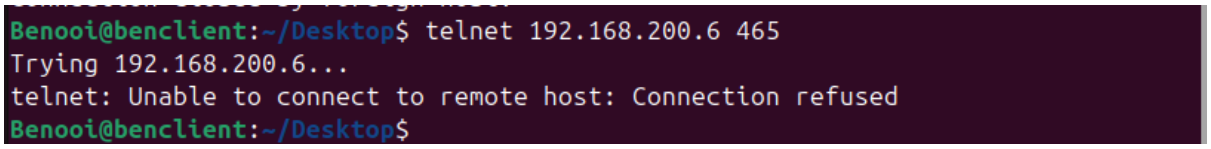


Figure 45: Check SMTP Port 465

Step 1: Verify Client-Server Connection

Command:

- telnet 192.168.200.6 465

Observation:

- Port 465: Connection fails – this suggests that the server is not listening on port 465 or the port is blocked by a firewall.

5.2.1 Troubleshooting Port Connection Issue to Ubuntu Client

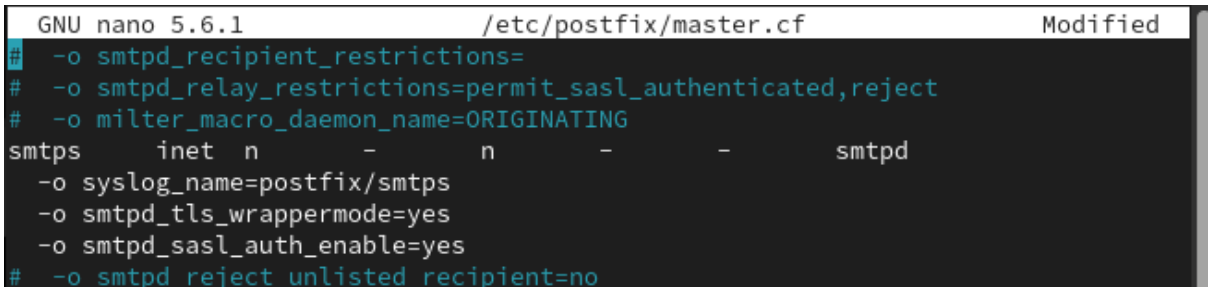


Figure 46: Configured to Accept SMTPS

Step 1: Troubleshooting Port 465 Connection Issue to Ubuntu Client

Action on the Linux Server: Edit the Postfix configuration by running

- sudo nano /etc/postfix/master.cf

Enable the SMTPS Service:

- Locate and activate the smtps (port 465) service by uncommenting or adding the following lines:
  - smtps inet n - n - - smtpd
    -o syslog_name=postfix/smtps
    -o smtpd_tls_wrappermode=yes
    -o smtpd_sasl_auth_enable=yes

Explanation:

- This configuration enables encrypted SMTP communication over port 465. The options ensure that:
  - Logs for this service are labeled appropriately.
  - TLS is used in wrapper mode for secure transmission.
  - SASL authentication is enabled to require valid login credentials.

This change helps resolve connectivity issues between your email server and the Ubuntu client when trying to use SMTPS.

5.2.2 Verirying Port 465 Connectivity (Ubuntu Client)

```
Benooi@benclient:~/Desktop$ telnet 192.168.200.6 465
Trying 192.168.200.6...
Connected to 192.168.200.6.
Escape character is '^]'.
```

Figure 47: Check SMTP Port 465

Step 2: Verifying Port 465 Connectivity from Ubuntu Client

Command (on Ubuntu client):

- telnet 192.168.200.6 465

Description and Function:

- This command attempts to establish a TCP connection from the Ubuntu client to the Rocky Linux server at IP address 192.168.200.6 on port 465, which is typically used for secure SMTP (SMTPS).

Purpose in the System:

- The goal here is to confirm that the SMTPS service is active and accepting connections on the server. A successful response (e.g., showing a greeting from the mail server) indicates that:
- The SMTPS service is properly configured and running.
- Port 465 is open and accessible on the server.
- There are no firewall or network issues blocking the connection.

5.2.3 Reconfirm Security Exception After Switching to Port 465

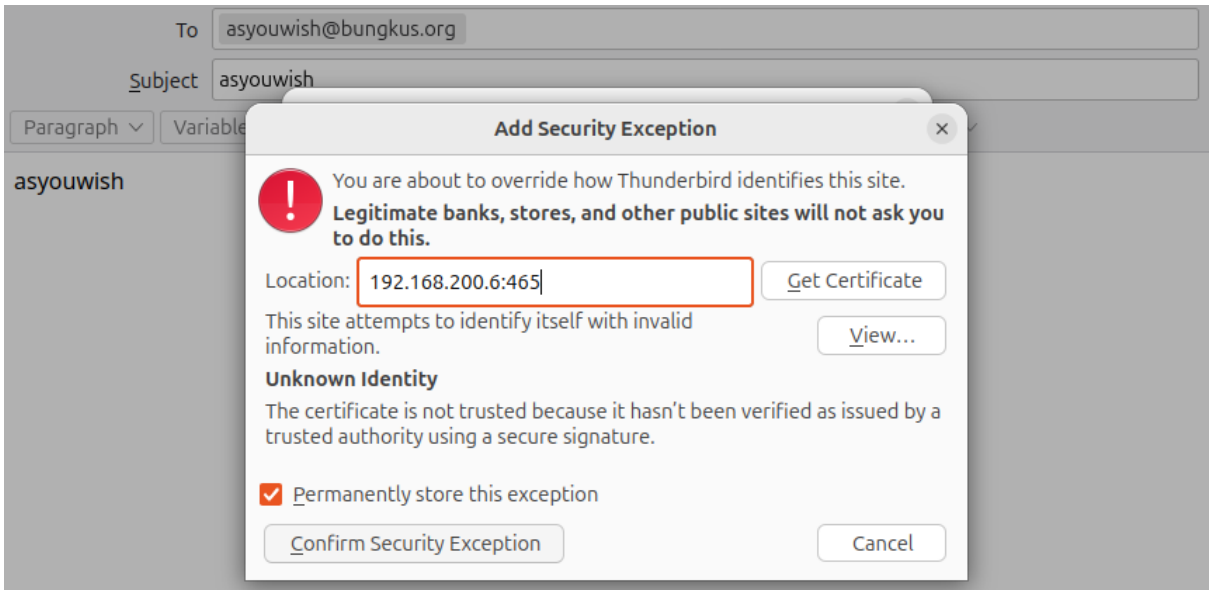


Figure 48: Confirm Security Exception of Port 465

Step 3: Reconfirming Security Exception After Switching to Port 465

Context:

- After changing the SMTP port from the 25 to the secure 465, it's important to ensure the client system accepts the new secure connection settings.

Action:

- Before sending any emails from the client, verify and approve the updated security exception. This typically involves:
- Confirming the server's SSL/TLS certificate is trusted.
- Allowing the email client or to connect securely over port 465.
- Possibly accepting a warning or prompt about an untrusted certificate, especially if using a self-signed certificate.

5.3 Troubleshooting Email Failling to Deliver

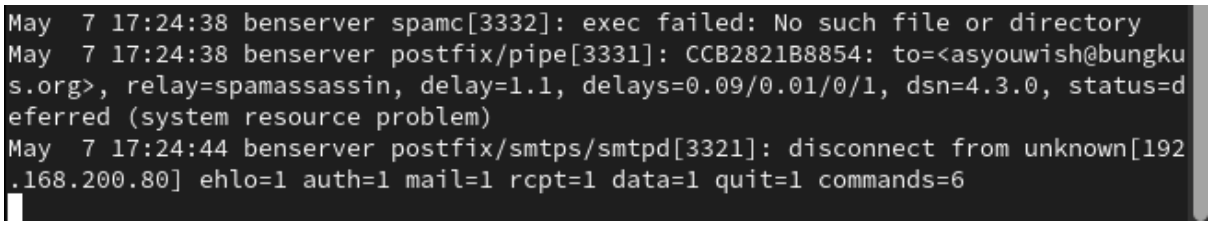


Figure 49: Postfix-SpamAssassin-Failure-Deferred-Email

Issue: Emails Failing to Deliver Due to spamc Error

Command:

- sudo tail -f /var/log/mail.log

Observed Symptom:

- May 7 17:24:38 benserver spamc[3332]: exec failed: No such file or directory

Explanation:

- This error indicates that Postfix attempted to run the spamc command (the client for SpamAssassin), but the system couldn't find it—likely because the binary is missing or not installed correctly.

Postfix Email Deferral:

- May 7 17:24:38 benserver postfix/pipe[3331]: CCB2821B8854: to<asyouwish@bungkus.org>,
- relay=spamaasassin, status=deferred (system resource problem)

Explanation: Since spamc failed to run, Postfix couldn't process the email through the spam filter and temporarily postponed delivery, classifying it as a system-level resource issue.

5.3.1 Correcting a Misconfigured spamc Command (postfix/master.cf)

```
spamassassin unix -     n       n       -       -       pipe
  user=nobody argv=/usr/bin/spamc -f -e \
  /usr/sbin/sendmail -oi -f ${sender} ${recipient}
```

```
spamassassin unix -     n       n       -       -       pipe
  user=nobody argv=/usr/bin/spamc -f -e
  /usr/sbin/sendmail -oi -f ${sender} ${recipient}
```

Figure 50: Configuration Fix /etc/postfix/master.cf

Step 1: Correcting a Misconfigured spamc Command in Postfix Configuration

Command (on the Linux server): sudo nano /etc/postfix/master.cf

What to Do: Find the section related to SpamAssassin that looks like this:

- spamassassin unix - n n - - pipe
  user=nobody argv=/usr/bin/spamc -f -e\
  /usr/sbin/sendmail -oi -f ${sender} ${recipient}
- If the line user=nobody argv=/usr/bin/spamc -f -e\ ends with a backslash (\), remove the backslash so it becomes:
  - user=nobody argv=/usr/bin/spamc -f -e

Explanation and Purpose:

- The argv= line defines how Postfix should execute SpamAssassin via spamc. A stray backslash at the end of this line causes Postfix to misinterpret the command, resulting in execution failure.

Role in the System:

- This section controls how incoming emails are filtered through SpamAssassin. If misconfigured, the filter process breaks, and emails may be deferred instead of delivered. Removing the incorrect backslash ensures spamc runs properly as part of the email handling pipeline.

Key Takeaways:

- A backslash (\) at the end of a line tells the system the command continues—only if it's immediately followed by a newline (with no spaces).
- A misplaced or unintended backslash corrupts the command syntax, causing failures.

6.0 Conclusion

Overall, the correct deployment and configuration of a secure email server with Rocky Linux is a proof of the feasibility and effectiveness of open-source application within enterprise-level communications systems. With Postfix and Dovecot installation and configuration, the system allows encrypted transmission of emails, secure user authentication, and assured message delivery. Thunderbird was employed in testing functionality from the client's side, thereby ensuring seamless integration of server and user interface. Additional features such as Spam Assassin and attachment filtering were included to combat spam and malware attacks, further strengthening the security posture of the system. Common issues that users will encounter in the setup, including port conflicts, authentication errors, and filtering errors, were identified and eliminated through formalized troubleshooting. This project provides a practical foundation for email service configuration and management, with an emphasis on layered security and operational testing. In brief, it provides system administrators with the fundamental knowledge and expertise necessary to implement secure, scalable, and effective mail servers.

7.0 Red Hat Certification

Figure 51: Red Hat Certification

8.0 References

Eset. (19 March, 2025). *eset*. Retrieved from help.eset.com: https://help.eset.com/emsl/10.0/en-US/antispam_test.html

Hat, R. (2019). *Red Hat*. Retrieved from www.redhat.com: https://www.redhat.com/en/blog/install-configure-postfix

Hat, R. (31 October, 2019). *Red Hat*. Retrieved from www.redhat.com: https://www.redhat.com/en/blog/install-configure-dovecot

Hat, R. (February, 2025). *Red Hat*. Retrieved from docs.redhat.com: https://docs.redhat.com/en/documentation/red_hat_amq/6.2/html/security_guide/createcerts#CreateCerts

Vultr. (1 April, 2025). *Vultr*. Retrieved from docs.vultr.com: https://docs.vultr.com/how-to-install-spamassassin-with-postfix-on-ubuntu