Koneru Lakshmaiah University

Open Source Engineering Report

Name: B. Arun Kumar

Reg No: 2400032926

Department: CSE – HTE

Course: Open Source Engineering

Faculty: Sripath Roy

Contents

Chapter 1

About the Linux Distribution Used

The Linux distribution chosen for this project is **Ubuntu 22.04 LTS**.

- Based on Debian
- Default Desktop Environment: GNOME
- Package Manager: APT
- Kernel Version: 5.x

Reason for choosing: It is beginner-friendly, stable, and widely supported with extensive community resources.

Chapter 2

Encryption and GPG

GPG (GNU Privacy Guard) is an implementation of the OpenPGP encryption standard, used to secure communication and protect sensitive data using public and private key pairs.

Key Concepts

- **Public Key** — Shared with others to allow message encryption.
- **Private Key** — Secret key used to decrypt received messages.

Commands Used

```
gpg --full-generate-key
gpg --list-keys
gpg --export --armor "arunbenuku118@gmail.com" > publickey.asc
```

Chapter 3

Sending Encrypted Email

The encrypted email process ensures privacy and data security and prevents unauthorized access to messages.

Procedure

1. Import Receiver Public Key

   ```
   gpg --import receiver_public.asc
   ```

2. Encrypt the message

   ```
   gpg --encrypt --armor -r receiver@email.com message.txt > encrypted.asc
   ```

3. Send encrypted file via Thunderbird Email Client

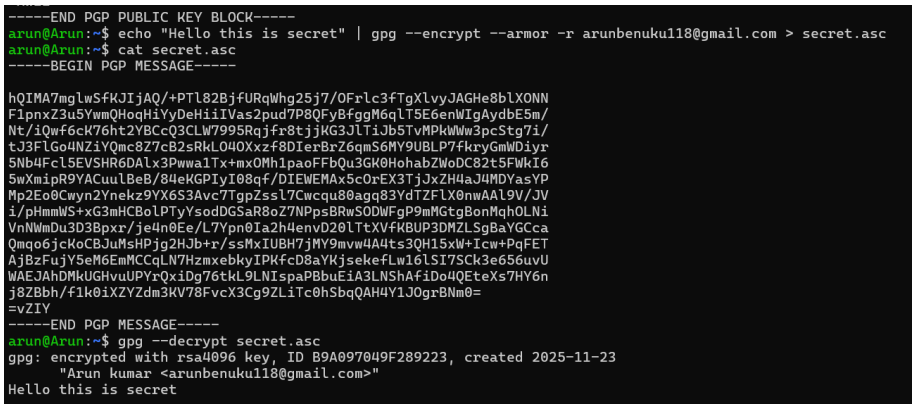

Figure 3.1: Encrypted Email Sent Using Thunderbird + GPG

Chapter 4

Privacy Tools from Prism-Break.org

1. **Signal** — End-to-end encrypted messaging
2. **Tor Browser** — Anonymous and privacy-focused browsing
3. **KeePassXC** — Secure open-source password manager
4. **ProtonMail** — Zero-access encrypted email service
5. **DuckDuckGo** — Private alternative search engine

Chapter 5

Open Source License Used

The selected license for contributions is the **MIT License**.

- Allows usage, modification, and redistribution
- Minimal restrictions and business-friendly

```
MIT License
Copyright (c) 2025 B Arun Kumar
```

Chapter 6

Self-Hosted Server Setup — Jellyfin

Jellyfin is a free and open-source media server used to organize and stream personal media on a local network.

- Completely open-source and community supported
- Alternative to Plex, Netflix-style streaming at home
- Access media privately without third-party servers

Installation Commands (Ubuntu)

```
sudo apt update
curl https://repo.jellyfin.org/install-debuntu.sh | sudo bash
sudo systemctl enable jellyfin
sudo systemctl start jellyfin
```

Server Access: `http://172.27.234.166:8096/web`

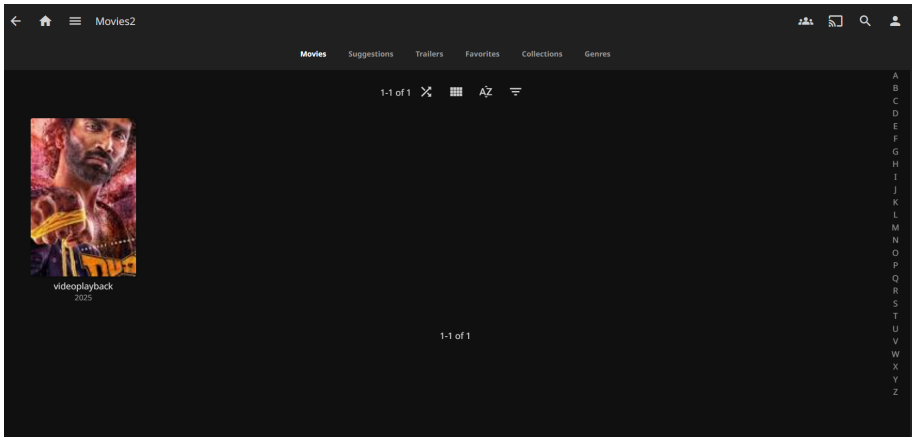

Figure 6.1: Jellyfin Self-Hosted Media Server Running on Local Network

Chapter 7

Open Source Contributions (GitHub Pull Requests)

Total PRs Raised: 8

- **2** PRs Merged
- **2** PRs Under Review
- **2** PRs Closed (Requested Changes)
- **2** Newly Created

Pull Request Summary

- **the-turing-way/the-turing-way** — Simplified glossary reference syntax PR #4409 — Closed `https://github.com/the-turing-way/the-turing-way/pull/4409`
- **HeyPuter/puter** — Traditional Chinese translations PR #1822 — Merged `https://github.com/HeyPuter/puter/pull/1822`
- **firstcontributions/first-contributions** — Gujarati alt-text translation PR #105693 — Merged `https://github.com/firstcontributions/first-contributions/pull/105693`
- **KLGLUG/Y24OpenSourceEngineering** — Added self-hosting documentation PR #130 — Review Required `https://github.com/KLGLUG/Y24OpenSourceEngineering/pull/130`

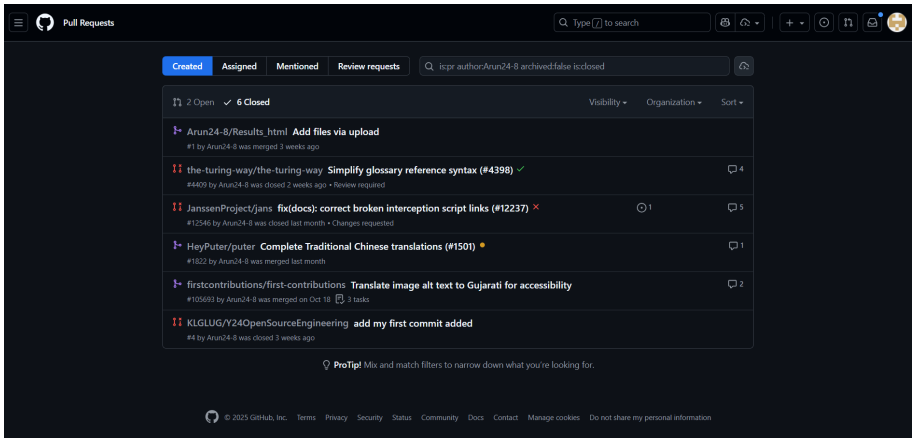

Figure 7.1: Pull Request Activity Screenshot

Chapter 8

LinkedIn Posts

- Jellyfin Server Deployment https://www.linkedin.com/posts/arunkumar2418_self-hosting-journey-jellyfin-media-server-activity-7398757850451214336-qF5V
- Pull Request Merge Update https://www.linkedin.com/posts/arunkumar2418_complete-traditional-chinese-translations-activity-7398736593731178496-RGY5
- Open Source Contribution Blog https://www.linkedin.com/pulse/my-journey-linux-self-h