OPEN SOURCE ENGINEERING REPORT

Submitted by

KANDULA KOMALI
2400032813

Y24 Batch HTE

Submitted To: DR.SRIPATH ROY

Department of Computer Science (HTE)
KLUNIVERSITY

November 26, 2025

1 About the Linux Distro Used

A Linux distribution is a complete operating system built on the **Linux kernel** and bundled with system software, libraries, and tools to meet specific user needs. Hundreds of different distributions (distros) exist, each offering a unique assortment of software packages, configuration tools, and default settings.

Key characteristics and components of Linux distributions:

Kernel: The core of the operating system, responsible for managing hardware resources, memory, and processes.

GNU Tools and Libraries: Most distros use essential software from the GNU project, such as the GNU C Library (glibc) and the Bash shell, providing core functionalities.

Package Manager: A system for installing, updating, and removing software packages from online repositories. Common examples include apt (Debian, Ubuntu), dnf/yum (Fedora, RHEL), and pacman (Arch Linux).

Init System: The first process launched after the kernel loads, managing system services (daemons). systemd is the most common modern init system.

User Interface: Can be a command-line interface (CLI) for advanced users and servers, or a graphical user interface (GUI) provided by a desktop environment (DE) such as GNOME, KDE Plasma, or Xfce.

Why So Many Distros?

The open-source nature of the Linux kernel and most accompanying software allows anyone to modify and redistribute the code, leading to diverse distributions tailored for specific use cases. These variations address different priorities:

Target Users: Some are beginner-friendly (Ubuntu, Linux Mint), while others are for experienced users or "power users" (Arch Linux, Gentoo).

Purpose: Distros are optimized for different environments, including desktops, servers (RHEL, Debian, CentOS), cybersecurity (Kali Linux, Parrot OS), or older hardware (Puppy Linux, antiX).

Release Model:

Standard Release (Fixed): Provide stability with less frequent updates and long-term support (LTS) versions, common for enterprise environments.

Rolling Release: Offer the latest software and features continuously, requiring more maintenance but staying on the cutting edge (Arch Linux, openSUSE Tumbleweed).

Philosophy: Some distributions adhere strictly to free software principles, while others include proprietary software for convenience (e.g., specific drivers or codecs).

2 Encryption and GPG

GPG (GNU Privacy Guard) is an encryption tool that provides secure communication using public-key cryptography.

Commands Used

```
gpg --generate-key gpg --list-
keys gpg -e -r <email> file.txt
gpg --output myfile.txt --decrypt myfile.txt.gpg
```

GPG generates a pair of keys: **Public Key** (shared) and **Private Key** (kept secret). Using these keys, I encrypted files and verified signatures securely. Encryption is the process of converting data into an unreadable, encoded format that can only be decrypted by authorized parties. GNU Privacy Guard (GPG) is a free, open-source tool that implements the **OpenPGP standard** for robust data encryption and digital signatures, widely used to secure emails and files.

How Encryption Works

Encryption methods generally fall into two categories, often used in combination:

- **Symmetric Encryption:** Uses a single, shared "secret key" for both encryption and decryption. This method is efficient for large amounts of data but requires a secure way to share the key.

- **Asymmetric (Public-Key) Encryption:** Uses a pair of mathematically linked keys: a **public key** for encryption (which can be shared freely) and a corresponding **private key** for decryption (which must be kept secret by the owner). GPG primarily uses this method, often employing a hybrid approach where a symmetric key for the data itself is encrypted using the recipient's public key.

What is GPG (GNU Privacy Guard)?

GPG (or GnuPG) is a command-line tool and a complete implementation of the OpenPGP standard. It provides:

- **Confidentiality:** Ensures only the intended recipient can read the data.
- **Integrity:** Verifies that the data has not been tampered with in transit.
- **Authentication & Non-repudiation:** Confirms the sender's identity through digital signatures.

Key Management

The core of GPG is the use of key pairs.

- **Public Key:** Given to anyone you want to send encrypted information to. They use it to encrypt a message that only you can open.
- **Private Key:** Kept secret and secure by you. It is essential for decrypting incoming messages and creating digital signatures.

Common GPG Uses & Commands

GPG is highly versatile and available for various operating systems including Linux, Windows (Gpg4win), and macOS.

Common actions are performed via the command line (e.g., gpg or gpg2):

- **Generate a key pair:** gpg --gen-key or gpg --full-generate-key.
- **List public keys:** gpg --list-keys.
- **Encrypt a file for a recipient:** gpg --encrypt --recipient [User ID/Email] [filename].
- **Decrypt a file:** gpg --decrypt [encrypted_filename.gpg] (uses your private key).
- **Symmetric-key encryption (password protected, no public key needed):** gpg --symmetric [filename].

By utilizing GPG, individuals and organizations can protect sensitive data both in transit and at rest from unauthorized access.

3 Sending Encrypted Email

Email encryption is a vital security measure for protecting sensitive information such as financial, health, or personally identifiable data from unauthorized access during transit and storage. Common methods include **S/MIME**, **PGP**, and **built-in services** like those offered by Outlook (Microsoft Purview Message Encryption) and Gmail (Confidential Mode).

Why Encrypt Emails?

Standard email (Transport Layer Security or TLS) only encrypts the connection between servers, not the message content itself, meaning it can potentially be read if a server is compromised. Email encryption secures the message contents, ensuring that only the intended recipient with the correct key or passcode can view the information.

Common Methods & How They Work

- **S/MIME (Secure/Multipurpose Internet Mail Extensions):** This method uses digital certificates (public and private keys) to encrypt and digitally sign messages. The sender uses the recipient's public key to encrypt the email, and the recipient uses their private key to decrypt it. It requires both sender and recipient to have S/MIME configured and certificates installed.
- **PGP (Pretty Good Privacy):** Similar to S/MIME, PGP uses public-key cryptography. It operates on a decentralized trust model, allowing users to manage their own keys.
- **Built-in/Managed Encryption Services (e.g., Microsoft Purview Message Encryption, Gmail Confidential Mode):** These services offer a simpler user experience without requiring manual key management.
    - **Microsoft Outlook/Purview:** The sender applies encryption, often by selecting an "Encrypt" or "Do Not Forward" option. Recipients (especially external ones) often receive a link to a secure web portal where they can sign in with a one-time passcode or an existing email account (Google, Yahoo, Microsoft) to view the message.
    - **Gmail Confidential Mode:** This allows users to set an expiration date for the message and require a passcode (sent via email or SMS) for access. It also disables options to forward, copy, print, or download the email/attachments.

General Steps to Send an Encrypted Email (Varies by Provider)

1. **Compose** a new email as you normally would.
2. **Locate the encryption option.** This is typically found in the ribbon, under "Options," or represented by a shield/lock icon in the message composition window.
3. **Select the desired level of encryption or sensitivity label.** Options might include Encrypt Only, Do Not Forward, or Confidential Mode. Some systems also automatically encrypt if a specific keyword like "Secure" is in the subject line.

4. **Finish composing** your email and send it.
5. **Recipient Access:** The recipient's experience will depend on their email provider and the method used. They may need to use a dedicated portal, enter a passcode, or have S/MIME configured to view the message directly in their inbox.

For detailed, platform-specific instructions, consult the official support pages for Gmail or Microsoft Outlook.

4 Five Privacy Tools (prism-break.org)

I explored several privacy tools from PRISM-Break:

1. Signal – Secure Messaging App

Category: Communication Use: End-to-end encrypted messaging, voice calls, video calls. Why: No ads, no tracking, open-source, strong encryption. Platforms: Android, iOS, Windows, MacOS, Linux.

2. Tor Browser – Anonymous Web Browsing

Category: Web Browsers Use: Hides your IP and identity by routing traffic through Tor nodes. Why: Prevents tracking, fingerprinting, surveillance. Platforms: Windows, Linux, Mac, Android.

3. KeePassXC – Password Manager

Category: Security Use: Stores all your passwords locally in an encrypted vault. Why: No cloud sync by default, avoids data breaches, open-source. Platforms: Windows, Linux, MacOS.

4. ProtonMail – Encrypted Email

Category: Email Use: End-to-end encrypted email service. Why: Protects emails from unauthorized access; Swiss privacy laws. Platforms: Web, Android, iOS.

5. DuckDuckGo – Privacy Search Engine

Category: Search Engines Use: Search without collecting personal data. Why: No tracking, no profiling, shows unbiased results. Platforms: Web, Android, iOS.

5 Open Source Licence Used

I used the **HomeGallery is an open-source self-hosted photo and video management software. The project is officially licensed under the MIT License.)**

About the MIT License

- The MIT License is one of the most widely used open-source licenses. It is:

  —> Simple and permissive

  —> Allows anyone to use, copy, modify, merge, publish, distribute

  —> Allows private or commercial use

  —> Requires only one condition: include the original copyright notice

  This makes the MIT License ideal for open-source projects because it gives users complete freedom while still protecting the copyright holder

6 Self-Hosted Server

I self-hosted the following server:

HOME GALLERY .

About

HomeGallery is a free, open-source, self-hosted photo and video gallery designed for people who want complete privacy and control over their media. It automatically scans your images and videos, generates previews, detects faces, allows smart search, and helps you organize everything—without depending on cloud services like Google Photos or iCloud.

HomeGallery runs on your own system (Ubuntu/Docker/Windows), meaning your photos never leave your device, making it a secure and privacy-friendly alternative.

Installation Steps Example

commands

```
sudo apt update sudo apt install docker.io
sudo systemctl enable docker --now mkdir
-p ~/homegallery/data mkdir -p
~/homegallery/config mkdir -p
~/homegallery/photos sudo docker run -d
\
   -p 3000:3000 \
   -v ~/homegallery/photos:/input \
   -v ~/homegallery/data:/data \
   -v ~/homegallery/config:/config \
   --name homegallery \ xypine/homegallery:latest
http://localhost:3000
```

Localized (Translated) Document

I translated the documentation into **Telugu**:

HomeGallery పరిచయం

HomeGallery అనేది మీ ఫోటోలు మరియు వీడియోలను మీ స్వంత సర్వర్లో సురక్షితంగా నిల్వచేసుకోవడానికి ఉపయోగించే ఒక సెల్ఫ్హోస్టెడ్ గ్యాలరీ సిస్టమ్.

ఇది పూర్తిగా ఓపెన్ సోర్స్మరియు మీ ప్రైవసీని కాపాడే విధంగా రూపొందించబడింది.

ప్రధాన లక్షణాలు (తెలుగులో)

మీ ఫోటోలు, వీడియోలు క్లౌడ్కివెళ్లవు

ఆటోమేటిక్ సార్టింగ్ మరియు గ్యాలరీ రూపొందింపు

ముఖాల గుర్తింపు, ఆబ్జెక్ట్ సెర్చ్

బ్రౌజర్ లేదా మొబైల్ నుండి యాక్సెస్

పూర్తిగా ఉచితం మరియు MIT లైసెన్స్

ఇన్స్టాలేషన్ (సంక్షిప్త తెలుగు వివరణ)

Docker ఇన్స్టాల్ చేయండి

homegallery ఫోల్డర్లు సృష్టించండి

Docker కమాండ్తో సర్వర్ రన్ చేయండి

బ్రౌజర్లో localhost:3000 ను ఓపెన్ చేయండి

Poster

Poster Attached Below:

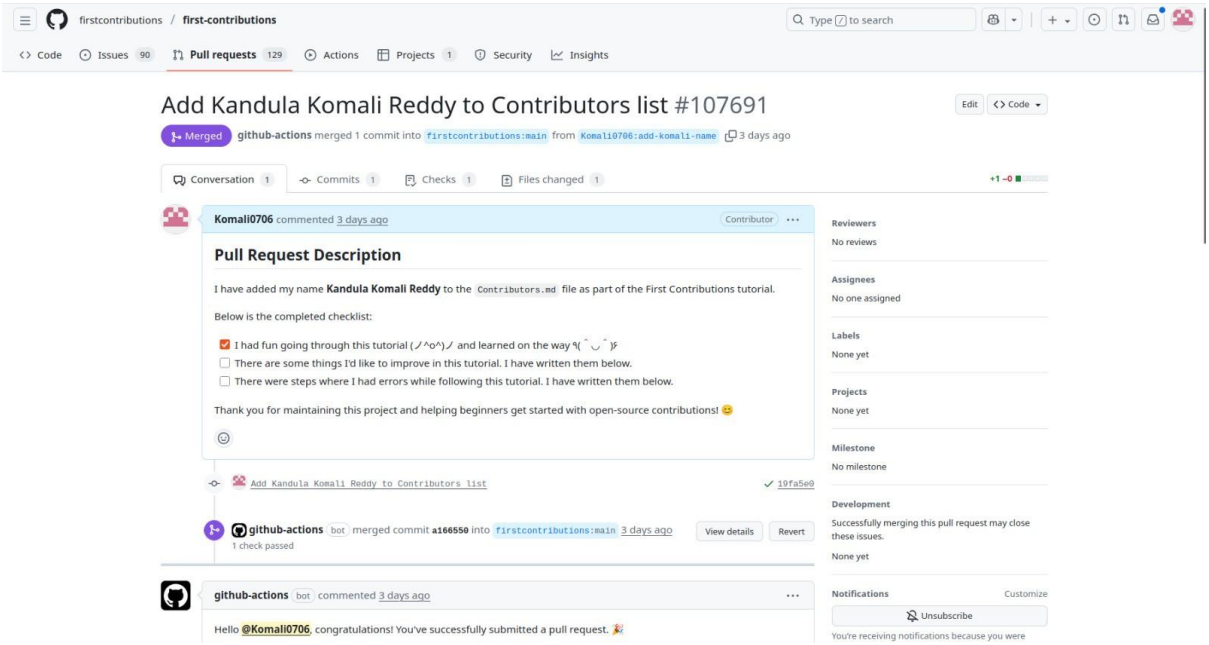
————————————————————-

8 Open-Source Contributions and PR Status

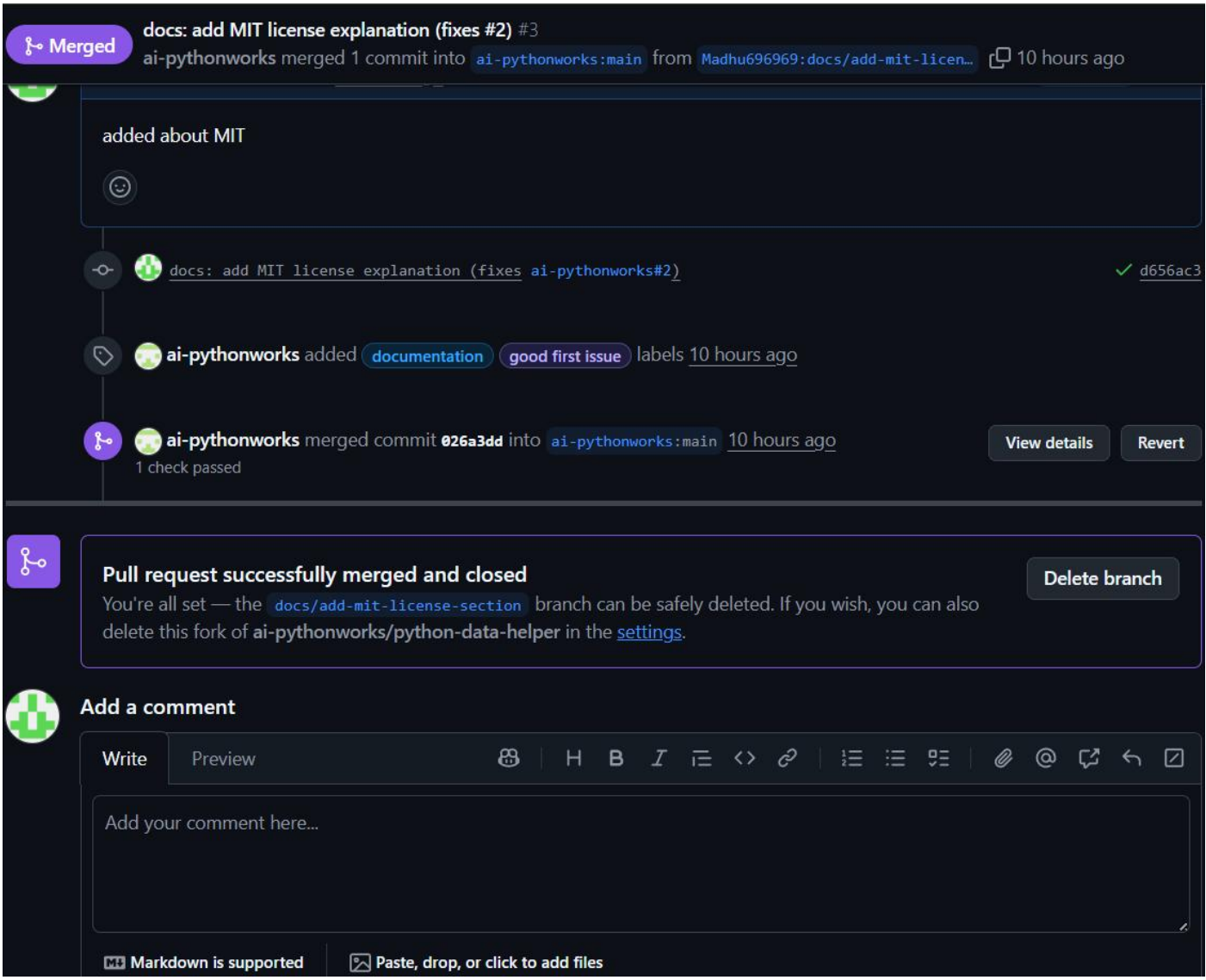
Below are my open-source contributions:

Pull Request 1

Project: first-contributions **Issue:** Add my name to Contributors.md **Status:** Merged **Description:** I added my name as part of the beginner tutorial.



Pull Request 2



9 LinkedIn Post Links

- Self Hosting Post: https://www.linkedin.com/posts/kandula-komali-9a5b02353_ opensource-engineering-selfhosting-activity-7399116358657015808-gmw1?utm_source=social_share_video_v2&utm_medium=android_app&rcm=ACoAAFroOD4BqGFsfqUcO utm_campaign=whatsapp

- PR Merge Post: https://www.linkedin.com/posts/kandula-komali-9a5b02353_ opensource-github-firstcontributions-share-7399118145887739905-DkMa?utm_source=social_share_send&utm_medium=android_app&rcm=ACoAAFhDRw0BglYgflCQdZy1bXnKj utm_campaign=whatsapp

- Blog Post: https://www.linkedin.com/pulse/my-journey-through-linux-self-hosting-op utm_source=share&utm_medium=member_android&utm_campaign=share_via