Open Source Software

Student Details

Name:	KORRAPATI MAHI VENKAT PAVAN
Roll Number:	2400040166
Department:	ELECTRONIC AND COMMUNICATION ENGINEERING
University:	KL University
Course:	Open Source Engineering
Semester:	ODD SEMESTER

Submitted to:

Dr. Sripath Roy
Department of Computer Science and Engineering
KL University

Contents

1 Linux Distribution

1.1 Distribution Used: Ubuntu 22.04 LTS

For my open-source engineering work, I selected **Ubuntu 22.04 LTS** as the primary operating system. Its stability, long-term support, and wide community adoption make it an ideal choice for development, testing, and deployment tasks.

1.2 Reasons for Choosing Ubuntu

Ubuntu stands out as one of the most reliable and beginner-friendly Linux distributions. Some of the major advantages that influenced my choice include:

- **Ease of Use:** Smooth user interface and simple navigation suitable for both beginners and advanced users.
- **LTS Updates:** Long Term Support ensures continuous security updates and bug fixes for 5 years.
- **Strong Community:** Large global community offering tutorials, troubleshooting help, and documentation.
- **Rich Package Ecosystem:** Availability of thousands of development tools through APT.
- **Stable and Reliable:** Highly suitable for software development and everyday usage.

1.3 Important Features of Ubuntu 22.04 LTS

1. **GNOME 42 Desktop:** Clean, modern, and highly responsive.
2. **Linux Kernel 5.15 LTS:** Ensures better performance and hardware compatibility.
3. **APT Package Manager:** Efficient for installing and managing software packages.
4. **Default Applications:** Includes essential apps like Firefox, LibreOffice, and terminal utilities.
5. **Snap Integration:** Enables installation of sandboxed applications easily.

1.4 System Specifications

My system configuration consists of:

- **Operating System:** Ubuntu 22.04 LTS
- **Architecture:** x86_64 (64-bit)
- **Desktop Environment:** GNOME 42
- **Shell:** Bash 5.1

Figure 1: UBUNTU

1.5 Installation Steps Followed

To set up Ubuntu successfully, I performed the following steps:

1. Downloaded the official Ubuntu 22.04 LTS ISO from the Ubuntu website.
2. Created a bootable USB drive using Rufus/Etcher.
3. Configured the system for dual-boot mode without affecting existing data.
4. Completed the installation process and performed the initial setup.
5. Installed essential development packages, including Git, GCC, Python, and build tools.
6. Customized the environment for open-source development.

2 Encryption and GPG

2.1 What is Encryption?

Encryption is the process of converting plaintext into ciphertext to protect data confidentiality. It ensures that only authorized parties can access the information.

2.2 Types of Encryption

2.2.1 Symmetric Encryption

Uses the same key for encryption and decryption. Examples: AES, DES.

2.2.2 Asymmetric Encryption

Uses a public-private key pair. Examples: RSA, ECC.

2.3 GNU Privacy Guard (GPG)

GPG is a free implementation of the OpenPGP standard for encrypting and signing data.

2.4 Installing GPG

```
sudo apt update
sudo apt install gnupg
gpg --version
```

2.5 Generating GPG Keys

```
gpg --full-generate-key
```

Steps followed:

1. Selected RSA and RSA (default)
2. Key size: 4096 bits
3. Key validity: 1 year
4. Entered name and email
5. Created strong passphrase

2.6 Listing Keys

```
gpg --list-keys
gpg --list-secret-keys
```

2.7 Exporting Public Key

```
gpg --armor --export your-email@example.com > public-key.asc
```

2.8 Encrypting Files

```
gpg --encrypt --recipient your-email@example.com document.txt
```

2.9 Decrypting Files

```
gpg --decrypt document.txt.gpg > document.txt
```

3 Sending Encrypted Email

3.1 Email Encryption Overview

Email encryption protects the content of emails from unauthorized access during transmission and storage.

3.2 Tools Used

- **Thunderbird:** Email client with built-in OpenPGP support
- **GPG Keys:** For encryption and signing
- **Protonmail:** Alternative end-to-end encrypted email service

3.3 Setting up Thunderbird with GPG

3.3.1 Installation

```
sudo apt install thunderbird
```

3.3.2 Configuring OpenPGP

Steps followed:

1. Open Thunderbird
2. Go to Account Settings
3. Select End-to-End Encryption
4. Add existing GPG key or generate new one
5. Import recipient's public key

3.4 Sending Encrypted Email

Process:

1. Compose new email
2. Click on Security button
3. Select ”Require Encryption”
4. Optionally add digital signature
5. Send email

3.5 Receiving Encrypted Email

When receiving:

1. Email appears encrypted
2. Thunderbird automatically detects encryption
3. Enter GPG passphrase
4. Email content is decrypted and displayed

3.6 Best Practices

- Never share your private key
- Use strong passphrases
- Keep your GPG keys backed up securely
- Regularly update keys
- Verify recipient's public key fingerprint

4 Privacy Tools from prism-break.org

4.1 Overview of PRISM-Break

PRISM-Break is an online platform that provides recommendations for privacy-focused and open source alternatives to mainstream software. The website encourages users to move away from applications that may engage in data collection, surveillance, or tracking, and instead adopt tools that prioritize user freedom and digital privacy.

4.2 Tool 1: Signal – Secure Messaging Application

Description: Signal is a highly trusted, open source messaging platform that offers end-to-end encryption for text messages, voice calls, and video communication.

Key Advantages:

- All communication is protected with end-to-end encryption.
- Open source and publicly audited, ensuring transparency.
- No advertisements, trackers, or user profiling.
- Minimal metadata retention policies.
- Features such as disappearing messages enhance privacy.

Why It Matters: Signal protects users from mass surveillance and unauthorized data access by ensuring that only the sender and receiver can read the messages.

4.3 Tool 2: Mozilla Firefox – Privacy-Focused Web Browser

Description: Firefox is a fully open source browser known for its strong privacy protections, customizable interface, and large extension ecosystem.

Privacy Features:

- Enhanced Tracking Protection against cookies, fingerprinting, and scripts.
- DNS-over-HTTPS (DoH) for secure domain lookups.
- Telemetry disabled by default in many builds.
- Support for privacy extensions like uBlock Origin and Privacy Badger.

Recommended Settings:

- Enable "Strict" Tracking Protection.
- Use HTTPS-Only Mode.
- Disable all telemetry data collection.
- Install privacy extensions for improved security.

4.4 Tool 3: ProtonMail – Encrypted Email Service

Description: ProtonMail is a privacy-respecting email provider offering end-to-end encrypted email based in Switzerland, a country known for strong privacy regulations.

Notable Features:

- End-to-end and zero-access encryption.
- No phone number or personal details required to sign up.
- Protected under Swiss data protection laws.
- Open source applications for web and mobile.

Ideal For:

- Private personal communication.
- Journalists, activists, and professionals handling sensitive data.
- Business emails requiring confidentiality.

4.5 Tool 4: Tor Browser – Anonymous Internet Browsing

Description: Tor Browser enables anonymous browsing by routing traffic through a network of decentralized volunteer-operated relays.

How It Protects Users:

- Multi-layer encryption ("onion routing").
- Completely hides IP address and location.

- Prevents tracking by advertisers and network observers.
- Helps bypass censorship and geo-restrictions.

Common Use Cases:

- Accessing restricted information securely.
- Conducting anonymous research.
- Whistleblowing and sensitive reporting.
- Avoiding surveillance and profiling.

4.6 Tool 5: VeraCrypt – Advanced Disk Encryption

Description: VeraCrypt is a powerful open source encryption utility designed to secure files, folders, and entire drives through strong cryptographic algorithms.

Key Features:

- Full disk and partition encryption.
- Support for hidden volumes and plausible deniability.
- Cross-platform compatibility (Windows, Linux, macOS).
- Strong encryption standards such as AES, Serpent, and Twofish.

Best Uses:

- Safeguarding confidential files on laptops and PCs.
- Encrypting external drives and portable storage devices.
- Protecting sensitive project data and system partitions.

5 Open Source License

5.1 License Used: MIT License

For my open source contributions and projects, I primarily work with the **MIT License**.

5.2 What is the MIT License?

The MIT License is a permissive free software license that allows users to:

- Use the software commercially
- Modify the software
- Distribute the software
- Use the software privately
- Sublicense the software

5.3 MIT License Text

```
MIT License

Copyright (c) 2025 Duvvu Venkata Ramana

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND...
```

5.4 Why Choose MIT License?

1. **Simple and Easy:** Short and easy to understand
2. **Permissive:** Minimal restrictions on reuse
3. **Business-Friendly:** Can be used in proprietary software
4. **Popular:** Widely used and recognized
5. **Compatible:** Works well with other licenses

5.5 Other Common Open Source Licenses

5.5.1 GPL (GNU General Public License)

- Copyleft license
- Requires derivative works to be open source
- Used by Linux kernel

5.5.2 Apache License 2.0

- Permissive like MIT
- Includes patent grant
- Used by Apache projects

5.5.3 BSD License

- Very permissive
- Similar to MIT
- Used by FreeBSD

6 Self-Hosted Server: PairDrop

6.1 What is PairDrop?

PairDrop is an open-source, browser-based peer-to-peer (P2P) file sharing application that lets devices on the same local network — or, optionally, over the internet — transfer files directly, without relying on third-party cloud servers. :contentReference[oaicite:2]index=2

Unlike many conventional file-sharing services, PairDrop requires no installation or account creation: just open a modern web browser, and it runs on Windows, Linux, macOS, Android, or iOS. :contentReference[oaicite:3]index=3

6.2 Why Self-Host PairDrop?

- **Privacy and Data Control:** Since files are transferred directly between devices, and not uploaded to third-party servers, you retain full control over your data. :contentReference[oaicite:4]index=4
- **Cross-Platform Compatibility:** Works across operating systems and devices via the browser. :contentReference[oaicite:5]index=5
- **Simplicity and No Setup for End-Users:** Recipients do not need to install any software — they just open the browser. :contentReference[oaicite:6]index=6
- **Flexibility:** Because PairDrop is open-source (GPL-3.0), you can self-host it — giving you independence from public instances. :contentReference[oaicite:7]index=7

6.3 Self-Hosting Instructions (Basic Setup)

Below is a simplified guide to self-host PairDrop on a server of your choice (e.g., a VPS or a local machine):

1. Clone the PairDrop repository:

```
git clone https://github.com/schlagmichdoch/PairDrop.git
cd PairDrop
```

2. Install dependencies using Node.js / npm:

```
npm install
npm run build     # or as per repo instructions
```

3. Optionally set up a reverse proxy (e.g. Nginx) and SSL/TLS (e.g. Let's Encrypt) to serve over HTTPS — especially if you want to access it over the internet or avoid browser warnings.

4. If behind NAT or for remote connections: configure a TURN/STUN server (or use the default configuration) so peers can connect. :contentReference[oaicite:8]index=8

5. Start the server (e.g. 'npm start' or via Docker — as suggested in the PairDrop documentation). Once running, open the server's URL in a browser; devices on the same network (or paired via internet) will see each other and you can start transferring files. :contentReference[oaicite:9]index=9

6.4 Use Cases

PairDrop self-hosting is especially useful when:

- You want private file sharing within local network (home / office), without uploading to public servers.
- You need cross-OS file transfer (e.g. from Android phone to Linux laptop, or Windows PC to iOS device).
- You want full control of the service (no tracking, no ads).
- You want a lightweight alternative to bulky cloud drives, for occasional file transfers among trusted devices.

6.5 Security Privacy Considerations

- PairDrop uses WebRTC for peer-to-peer transfers. Once the connection is established, file data flows directly between devices, not through third-party servers — which reduces risk of leak. :contentReference[oaicite:10]index=10
- If you expose the server to the internet (for remote access), it is strongly recommended to secure it via HTTPS and, optionally, restrict access (firewall, VPN, or password protection) to trusted users.
- Always ensure you trust the devices you're pairing with — as with any P2P tool, sending files blindly may pose risk if the other device is compromised.

7 My Open Source Contributions

Over the past few weeks, I have actively contributed to multiple open-source repositories. These contributions helped me understand collaboration, code review practices, and the open-source workflow. Below is a structured list of my merged pull requests, presented in the style of my LinkedIn post.

Merged Pull Requests

- **fineanmol/hacktoberfest**
  Title: contributed my branch1
  Status: Successfully merged
  Timeline: Merged 4 days ago

- **yfosp/start-here**
  Title: Add my GitHub username to CONTRIBUTORS.md
  Status: Successfully merged, Earned "First Open Source Contribution Badge"
  Timeline: Merged 3 days ago

- **zero-to-mastery/start-here-guidelines**
  Title: Add my branch number 3
  Status: Approved and merged
  Timeline: Merged last week

- **firstcontributions/first-contributions**
  Title: added name and profile
  Status: Successfully merged
  Timeline: Merged 3 weeks ago

Reflection

These contributions strengthened my understanding of collaborative development, documentation improvement, and beginner-friendly open-source workflows. I am excited to continue contributing and learning through real-world community projects.

Metric	**Count**
Total Pull Requests	21
Merged PRs	4
Open PRs	13

Table 1: Open Source Contribution Statistics

7.1 Key Learnings from Contributions

1. **Git Workflow:** Mastered branching, committing, and pull request processes
2. **Code Review:** Learned to receive and implement feedback constructively
3. **Documentation:** Understood the importance of clear, comprehensive documentation
4. **Collaboration:** Experienced working with global developer communities
5. **Testing:** Learned to write comprehensive test suites
6. **Localization:** Appreciated the value of making software accessible in regional languages

7.2 Post 1: My Open Source Blog – Journey, Challenges, and Growth

Link: `https://www.linkedin.com/feed/update/urn:li:activity:7398222448963465216/`

Summary: Shared a detailed blog post covering my complete journey into open source, including the initial challenges, learning phases, mindset shift, and the confidence gained through hands-on contributions.

Key Points:

- Documenting my beginner-friendly learning path
- Challenges faced during initial contributions
- Importance of consistency in open source
- Growth in Git, GitHub, Linux, and collaboration skills

7.3 Post 2: My First Open Source Contribution on GitHub

Link: `https://www.linkedin.com/feed/update/urn:li:activity:7398209468167221248/`

Summary: Shared the excitement of successfully making my first open source contribution. Highlighted how this milestone boosted my confidence and motivated me to continue exploring and contributing to the community.

Highlights:

- First PR successfully merged
- Learning how to contribute to beginner-friendly repositories
- Understanding how issues, forks, and branches work
- Receiving support from the open source community

7.4 Post 3: Self-Hosting a PairDrop Server

Link: `https://www.linkedin.com/feed/update/urn:li:activity:7398228622836293632/`

Summary: Shared my experience hosting a self-hosted PairDrop server, a fast and secure local file-sharing solution. Explained the setup process, benefits of self-hosting, and why it is an essential privacy-friendly tool.

Key Points:

- Introduction to PairDrop and local file sharing
- Setting up a private self-hosted server
- Understanding security and privacy advantages
- Deploying tools in a Linux-based environment

8 Conclusion

This report summarizes my complete journey into Open Source Engineering, highlighting my learning experience with Linux, Git, GitHub, privacy tools, self-hosted servers, and meaningful project contributions. Along the way, I also shared my progress through multiple LinkedIn posts to inspire others and document my growth.

Key Achievements

- Successfully set up and worked with Ubuntu Linux as my primary development environment
- Explored encryption, GPG keys, and essential privacy tools
- Self-hosted a **PairDrop server** for secure local file sharing
- Made multiple open source contributions across beginner-friendly repositories
- Published professional updates and reflections on LinkedIn

Skills Gained

1. Technical skills in Linux, Git, GitHub, and open source workflows
2. Understanding of collaborative development and pull request practices
3. Experience with documentation, testing, and community-driven development
4. Practical abilities in server administration and deployment
5. Improved communication and professional networking through social platforms

Self-Hosting PairDrop

The experience of self-hosting **PairDrop** has been particularly valuable, allowing me to gain hands-on experience with:

- Local server configuration and setup
- Web-based file transfer systems
- Understanding security and privacy-focused architectures
- Managing and running services on a Linux environment

LinkedIn Documentation

Throughout my journey, I actively documented my progress on LinkedIn:

- **My Open Source Journey Blog Post:** `https://www.linkedin.com/feed/update/urn:li:activity:7398222448963465216/`

- **My First Open Source Contribution Post:** https://www.linkedin.com/feed/update/urn:li:activity:7398209468167221248/
- **Self-Hosted PairDrop Server Post:** https://www.linkedin.com/feed/update/urn:li:activity:7398228622836293632/

Final Reflection

Overall, my open source journey has been an eye-opening experience. It has connected me with a global community of developers, improved my technical abilities, strengthened my confidence, and motivated me to continue contributing. Open source has taught me the value of collaboration, transparency, and building software that empowers everyone.