

Proxy Alchemist Installation Guide

This comprehensive guide provides detailed installation instructions for Proxy Alchemist across different platforms, with special emphasis on Termux (Android) and Linux environments.

Table of Contents

1. [System Requirements](#)
2. [Termux Installation \(Android\)](#)
3. [Linux Installation](#)
4. [Package Dependencies](#)
5. [Optional Components](#)
6. [Troubleshooting](#)
7. [Post-Installation Setup](#)
8. [Verification](#)

System Requirements

Minimum Requirements

- **Operating System:** Android 7.0+ (for Termux), Linux (Ubuntu 18.04+, Debian 9+, CentOS 7+, Fedora 28+)
- **Python Version:** Python 3.6 or higher
- **RAM:** 512 MB available
- **Storage:** 100 MB free space
- **Network:** Internet connection for proxy fetching and updates

Recommended Requirements

- **RAM:** 1 GB or more
- **Storage:** 500 MB free space (for logs, cache, and optional GeoIP database)
- **Network:** Stable internet connection with good bandwidth

Termux Installation (Android)

Termux is a powerful terminal emulator for Android that provides a Linux environment. This section covers the complete installation process for Proxy Alchemist on Termux.

Step 1: Install Termux

1. **Download Termux from F-Droid** (Recommended):
2. Visit [F-Droid](#)
3. Search for "Termux" and install the official version
4. **Note:** The Google Play Store version is outdated and not recommended
5. **Alternative:** Download from GitHub Releases:
6. Visit [Termux GitHub Releases](#)
7. Download the latest APK file
8. Enable "Install from Unknown Sources" in Android settings
9. Install the APK

Step 2: Initial Termux Setup

1. **Open Termux** and wait for the initial setup to complete
2. **Update package lists:**

```
bash  
pkg update && pkg upgrade
```
3. **Install essential packages:**

```
bash  
pkg install python git curl wget
```
4. **Grant storage permissions** (optional but recommended):

```
bash  
termux-setup-storage
```

This allows Termux to access your device's storage.

Step 3: Install Build Dependencies

Some Python packages require compilation. Install the necessary build tools:

```
pkg install build-essential python-dev libffi-dev openssl-dev
```

Step 4: Clone and Install Proxy Alchemist

1. **Clone the repository:**

```
bash
```

```
git clone https://github.com/your-username/proxy-alchemist.git
cd proxy-alchemist
```

2. Install Python dependencies:

```
bash
pip install -r requirements.txt
```

If requirements.txt doesn't exist, install manually:

```
bash
pip install requests pyfiglet colorama geoip2 qrcode psutil
```

1. Make the script executable: bash chmod +x proxy_alchemist.py

Step 5: Termux-Specific Configuration

1. Create a startup script (optional):

```
bash
echo '#!/bin/bash
cd ~/proxy-alchemist
python proxy_alchemist.py' > ~/start-proxy-alchemist.sh
chmod +x ~/start-proxy-alchemist.sh
```

2. Add to PATH (optional):

```
bash
echo 'export PATH=$PATH:~/proxy-alchemist' >> ~/.bashrc
source ~/.bashrc
```

Linux Installation

Ubuntu/Debian Installation

1. Update system packages:

```
bash
sudo apt update && sudo apt upgrade -y
```

2. Install Python and development tools:

```
bash
sudo apt install python3 python3-pip python3-dev git build-essential libffi-dev
libssl-dev
```

3. Clone the repository:

```
bash
```

```
git clone https://github.com/your-username/proxy-alchemist.git
cd proxy-alchemist
```

4. Install Python dependencies:

```
bash
pip3 install requests pyfiglet colorama geoip2 qrcode psutil
```

5. Create a symbolic link (optional):

```
bash
sudo ln -s $(pwd)/proxy_alchemist.py /usr/local/bin/proxy-alchemist
sudo chmod +x /usr/local/bin/proxy-alchemist
```

CentOS/RHEL/Fedora Installation

1. Update system packages:

```
bash
# For CentOS/RHEL
sudo yum update -y
# For Fedora
sudo dnf update -y
```

2. Install Python and development tools:

```
bash
# For CentOS/RHEL
sudo yum install python3 python3-pip python3-devel git gcc gcc-c++ make
openssl-devel libffi-devel
# For Fedora
sudo dnf install python3 python3-pip python3-devel git gcc gcc-c++ make
openssl-devel libffi-devel
```

3. Follow the same steps as Ubuntu for cloning and installing dependencies.

Arch Linux Installation

1. Update system:

```
bash
sudo pacman -Syu
```

2. Install dependencies:

```
bash
sudo pacman -S python python-pip git base-devel openssl libffi
```

3. Follow the same steps as Ubuntu for cloning and installing dependencies.

Package Dependencies

Core Dependencies

These packages are essential for Proxy Alchemist to function:

- **requests:** HTTP library for making API calls and proxy testing
- **psutil:** System and process utilities for monitoring

Optional Dependencies

These packages provide additional features but are not required:

- **pyfiglet:** ASCII art for the banner display
- **colorama:** Colored terminal output
- **geoip2:** Geographic IP information and filtering
- **qrcode:** QR code generation for easy configuration sharing

Installation Commands by Package

```
# Core packages (required)
```

```
pip install requests psutil
```

```
# Optional packages (recommended)
```

```
pip install pyfiglet colorama geoip2 qrcode
```

```
# All packages at once
```

```
pip install requests psutil pyfiglet colorama geoip2 qrcode
```

Optional Components

GeoIP Database Setup

The GeoIP database provides geographic information about IP addresses and proxies:

1. **Register for a MaxMind account:**
2. Visit [MaxMind GeoLite2](#)
3. Create a free account
4. **Generate a license key:**
5. Log in to your MaxMind account

6. Go to "My License Key" section

7. Generate a new license key

8. **Download the database:**

```
bash
cd proxy-alchemist
wget "https://download.maxmind.com/app/geoip_download?
edition_id=GeoLite2-City&license_key=YOUR_LICENSE_KEY&suffix=tar.gz" -O
GeoLite2-City.tar.gz
tar -xzf GeoLite2-City.tar.gz
mv GeoLite2-City_*/GeoLite2-City.mmdb .
rm -rf GeoLite2-City_* GeoLite2-City.tar.gz
```

Tor Installation

For enhanced anonymity features:

Termux:

```
pkg install tor
```

Ubuntu/Debian:

```
sudo apt install tor
```

CentOS/RHEL/Fedora:

```
# CentOS/RHEL (enable EPEL first)
sudo yum install epel-release
sudo yum install tor
```

```
# Fedora
sudo dnf install tor
```

Troubleshooting

Common Issues and Solutions

Issue 1: Python Package Installation Failures

Problem: `pip install` fails with compilation errors.

Solution:

1. Ensure development tools are installed:

```
```bash
```

```
Termux
```

```
pkg install build-essential python-dev
```

```
Ubuntu/Debian
```

```
sudo apt install build-essential python3-dev
```

```
CentOS/RHEL
```

```
sudo yum groupinstall "Development Tools"
```

```
sudo yum install python3-devel
```

```
```
```

1. Upgrade pip:

```
bash
```

```
pip install --upgrade pip
```

2. Install packages individually to identify problematic ones:

```
bash
```

```
pip install requests
```

```
pip install psutil
```

```
pip install pyfiglet
```

```
pip install colorama
```

```
pip install qrcode
```

```
pip install geoip2
```

Issue 2: Permission Denied Errors

Problem: Cannot execute script or access certain directories.

Solution:

1. Make script executable:

```
bash
```

```
chmod +x proxy_alchemist.py
```

1. Check file permissions:

```
bash
```

```
ls -la proxy_alchemist.py
```

2. For Termux storage access:

```
bash
```

```
termux-setup-storage
```

Issue 3: Network Connection Issues

Problem: Cannot fetch proxies or connect to external services.

Solution:

1. Check internet connectivity:

```
bash
```

```
ping google.com
```

1. Verify DNS resolution:

```
bash
```

```
nslookup google.com
```

2. Test with different proxy sources in the configuration.

Issue 4: Import Errors

Problem: `ModuleNotFoundError` when running the script.

Solution:

1. Verify Python version:

```
bash
```

```
python3 --version
```

1. Check installed packages:

```
bash
```

```
pip list
```

2. Reinstall missing packages:

```
bash
```

```
pip install --force-reinstall package_name
```


Platform-Specific Issues

Termux-Specific Issues

1. **Package not found:**

```
bash
```

```
pkg update
```

```
pkg search package_name
```

2. **Storage access issues:**

```
bash
```

```
termux-setup-storage
```

3. **Background execution:**

4. Install Termux:Boot for auto-start

5. Use `nohup` for background processes

Linux-Specific Issues

1. **Sudo requirements:**

2. Some features may require root access

3. Use `sudo` where necessary

4. **Firewall issues:**

```
```bash
```

```
Ubuntu/Debian
```

```
sudo ufw allow 8080
```

```
CentOS/RHEL
```

```
sudo firewall-cmd --add-port=8080/tcp --permanent
```

```
sudo firewall-cmd --reload
```

```
```
```

Post-Installation Setup

Configuration

1. **Run the initial setup:**

```
bash
```

```
python3 proxy_alchemist.py
```

2. **Configure settings through the menu:**

3. Access "Configuration" option
4. Set preferred protocols
5. Adjust latency thresholds
6. Configure favorite countries
7. **Test functionality:**
8. Try "Change IP" option
9. Verify proxy connectivity
10. Test local proxy server

Security Considerations

1. File permissions:

```
bash
chmod 600 proxy_config.json
chmod 600 favorites.json
chmod 600 history.json
```

2. Log file management:

```
bash
# Set up log rotation
echo "proxy_alchemist.log {
    daily
    rotate 7
    compress
    missingok
    notifempty
}" | sudo tee /etc/logrotate.d/proxy-alchemist
```

Performance Optimization

1. Cache management:

2. Regularly clean proxy cache
3. Monitor disk usage

4. Memory optimization:

5. Adjust proxy list size in configuration
6. Limit concurrent connections

Verification

Basic Functionality Test

1. **Start the application:**

```
bash
```

```
python3 proxy_alchemist.py
```

2. **Test proxy fetching:**

3. Select option 1 (Change IP)

4. Verify proxy list is fetched

5. Check if working proxy is found

6. **Test local proxy server:**

7. Select option 9 (Local Proxy Server)

8. Start local proxy

9. Verify server is listening on port 8080

Advanced Testing

1. **Tor integration** (if Tor is installed):

2. Select option 8 (Tor Integration)

3. Start Tor service

4. Test circuit changes

5. **Configuration persistence:**

6. Change settings in Configuration menu

7. Restart application

8. Verify settings are preserved

9. **QR code generation:**

10. Set up a proxy

11. Generate QR code

12. Verify QR code contains correct information

Troubleshooting Verification Issues

If verification fails:

1. **Check logs:**

```
bash  
tail -f proxy_alchemist.log
```

2. **Verify network connectivity:**

```
bash  
curl -I http://google.com
```

3. **Test with verbose output:**

```
bash  
python3 -v proxy_alchemist.py
```

4. **Check system resources:**

```
bash  
free -h  
df -h
```

Next Steps

After successful installation and verification:

1. **Customize configuration** to match your needs
2. **Set up favorites** for frequently used proxies
3. **Configure auto-start** if desired
4. **Explore advanced features** like proxy chaining
5. **Set up monitoring** for long-term usage

For additional support, please refer to the main README.md file or open an issue on the GitHub repository.