# Step-by-Step Setup: TLS Fingerprinting using Fingerproxy + Flask

Generated on: 2025-06-25 04:46:30

## Overview

This document guides you through setting up TLS fingerprinting on an Ubuntu 22.04 EC2 instance using the open-source Fingerproxy tool and a custom Flask server that logs fingerprints to a CSV file.

## Steps

### Step 1: Update and Install Required Packages

sudo apt update -y && sudo apt install -y git curl unzip python3 python3-pip python3-venv openssl build-essential

### Step 2: Install Go 1.22.3

cd ~  
curl -LO https://go.dev/dl/go1.22.3.linux-amd64.tar.gz  
sudo rm -rf /usr/local/go  
sudo tar -C /usr/local -xzf go1.22.3.linux-amd64.tar.gz  
echo 'export PATH=$PATH:/usr/local/go/bin' >> ~/.bashrc  
source ~/.bashrc  
go version

### Step 3: Clone Fingerproxy Repository

git clone https://github.com/wi1dcard/fingerproxy.git  
cd fingerproxy  
make build\_linux\_amd64  
cd ~

### Step 4: Generate Self-Signed TLS Certificates

openssl req -x509 -newkey rsa:2048 -nodes -sha256 -days 365 -keyout tls.key -out tls.crt -subj "/CN=fingerproxy" -addext "subjectAltName=IP:127.0.0.1,IP:$(curl -s ifconfig.me)"

### Step 5: Set Up Flask App in Virtual Environment

python3 -m venv venv  
source venv/bin/activate  
pip install flask httpagentparser

### Step 6: Create Flask App (app.py)

Use the Python code from previous steps to create a Flask app that logs fingerprint data.

### Step 7: Start Flask App

nohup ./venv/bin/python app.py > flask.log 2>&1 &

### Step 8: Start Fingerproxy

cd ~/fingerproxy  
nohup ./bin/fingerproxy\_linux\_amd64 -listen-addr :8443 -cert-filename ~/tls.crt -certkey-filename ~/tls.key -forward-url http://127.0.0.1:8080 > proxy.log 2>&1 &

### Step 9: Test Setup

Open https://<EC2\_PUBLIC\_IP>:8443 in a browser.  
Accept TLS warning and verify entries in fingerprints.csv

### View CSV Logs

cat fingerprints.csv

### Stop Services

pkill -f fingerproxy  
pkill -f app.py

You can now use this document to replicate the setup on any new machine.