

Setting Up Cisco 7911 for FreePBX & Asterisk (SIP)

Published by Bernardo Pinto on JUNE 15, 2025






This guide provides a detailed walkthrough on how to successfully integrate and configure a **Cisco 7911 IP phone** with your **FreePBX and Asterisk** VoIP system. It covers everything from preparing the phone's firmware and configuration files to setting up the extension within FreePBX, getting your Cisco 7911 up and running on FreePBX and Asterisk.

Prerequisites

To follow this guide, you'll need a few things set up. Here's a breakdown of the **hardware** and **software** you'll need to have ready:

Hardware You'll Need

- **Cisco 7911 IP Phone (AKA 7911G):** This is our star! Make sure you've got your phone ready to roll. 
- **PoE Switch:** You'll need a Power over Ethernet switch to give your 7911 some juice and network access.
 - *Quick heads-up:* We used a **Cisco Catalyst 3560-CX Series** switch for this guide. Depending on your switch, you might need console access for some initial tweaks or if things get tricky. Just a fair warning! 
 - **Desktop/Laptop PC:** This will be your control center. Make sure it has an **Ethernet port** for network connectivity to your FreePBX/Asterisk server and the phone's TFTP server. 

Software & Operating System Specs

- **Asterisk:** I'm rolling with version [18.26.2](#) here. If you have a slightly different version, don't worry, it'll likely still work just fine!
- **FreePBX:** My guide uses [16.0.40.13](#) sitting on top of Asterisk. Again, close versions should be okay.
- **Operating System:** I'm doing this on **Debian 12 (Bookworm)**.
 - *Our Setup:* Just for context, I ran our Debian instance on **VMware Workstation 17** which was hosted on a Windows 10 Pro machine. Your setup might vary, but the principles remain the same!
- **SIP – chan_pjsip:** For this guide I am using the now recommended SIP for Asterisk: [chan_pjsip](#).
- **Key Debian Utilities/Packages:**
 - First things first, let's make sure your system knows what's up. Run an update:

```
sudo apt update
```

- Then, go ahead and install these vital tools. They're super helpful for what we're about to do:

```
apt install nano dnsmasq sudo
```

- **nano**: Our trusty text editor for tweaking files. Feel free to use your favorite if it's not nano! 🛠️
- **dnsmasq**: This is super important! It'll act as our TFTP server and as a DHCP server for provisioning the phone. 📡
- **sudo**: Totally optional if you're living life as root, but generally a good idea for managing permissions safely. 👍

Setting Up

Obtaining and Preparing Cisco 7911 Configuration Files

For your Cisco 7911 IP phone to successfully connect and function with FreePBX and Asterisk, it needs a specific set of configuration files. These files tell the phone how to behave, where to find its SIP server (your FreePBX/Asterisk), how its buttons should work, and even how to ring. This includes the necessary SIP firmware itself, along with various XML configuration files, ringtones, and more.

Step 1: Download My Pre-Built Configuration Package

To save you the headache of hunting down individual files and ensure everything works together seamlessly, I've compiled a comprehensive package for you! This package includes:

- **SCCP Firmware**: Essential for the initial upgrade path.
- **SIP Firmware**: The firmware that allows your phone to register with FreePBX/Asterisk.
- **Configuration File Templates**: Ready-to-use templates for files like `SEP(MAC_ADDRESS).cnf.xml`, `dialplan.xml` and others.
- **Ringtones**: Some Cisco ringtones for your phone (If you find more later you can set it up after).
- **Dial Plans**: Both Portuguese (PT) and U.S. A. (USA) dial plan examples to get you started
 - You will need to rename the file you want from `dialplan - USA.xml` or `dialplan PT.xml` to `dialplan.xml`
- **Pre-configured SEP template**: Simplified setup template for SIP.

Download my pre-built setup package here:

- **My Pre-Built Setup For You:** [Here!](#) 

Once downloaded, **extract the entire contents of the archive**. Keep these files organized, as you'll be moving them to your TFTP server in the next steps. This package aims to streamline the most frustrating part of the 7911 setup!

Step 2: Configure the `SEP(MAC_ADDRESS).cnf.xml` File

This is the main configuration file for your specific Cisco 7911 phone. It tells the phone about your FreePBX server, your extension details, and how its softkeys should behave.

1. **Locate the Template:** After extracting the package from Step 1, find the template file for your phone. It's named `SEP(MAC_ADDRESS).cnf.xml` and its inside of the folder named `SIP`.
2. **Rename the File:**
 - **Crucially**, rename this template file to `SEP(MAC_ADDRESS).cnf.xml`.
 - **Replace** `(MAC_ADDRESS)` with the actual **MAC address** of your Cisco 7911 phone. You can find the MAC address printed on a sticker on the bottom or back of the phone (e.g., `SEP001A2B3C4D5E.cnf.xml`).
3. **Edit the File Content:**
 - Open your newly renamed `SEP(MAC_ADDRESS).cnf.xml` file using a text editor (like nano on your Debian system).
 - Inside the file, you'll find several placeholders that look like `[===PLACEHOLDER_NAME===]`. You need to replace these with your specific FreePBX and phone details:
 - `[===FreePBX IP===]`: Replace this with the **IP address** of your FreePBX server (e.g., `192.168.1.100`). Make sure to replace *all instances* of this placeholder throughout the file.
 - `[===Phone Name===]`: Replace this with a descriptive name for your phone (e.g., `Office Main, Bernardo's Desk`). This name will often appear on the phone's display.
 - `[===Line Name===]`: Replace this with the label you want to see next to the line button on the phone's screen (e.g., `Ext 101, Main Line` or just the exten).
 - `[===Extension===]`: Replace this with the **SIP extension number** you've configured for this phone in FreePBX (e.g., `101`).
 - `[===Password===]`: Replace this with the **SIP secret/password** for that extension, as defined in your FreePBX extension settings.
 - **TimeZone:** You can also setup the timezone, but the phone will get the time and date from the FreePBX server.
4. **Save the File:** Save the changes to `SEP(MAC_ADDRESS).cnf.xml`.

Step 3: Setting Up `dnsmasq` for DHCP

This is a critical step where your FreePBX server will act as both a DHCP server (to assign IP addresses to your Cisco 7911 phones) and a TFTP server (to deliver the configuration files and firmware).

Important Network Isolation Warning! ⚠ Before proceeding, it's absolutely crucial that the network interface you're about to configure for [dnsmasq](#) is **isolated**.

- **Disconnect the Server:** Disconnect the Ethernet/Wi-Fi connection from your FreePBX server that leads to your main network (internet, other DHCP servers).
- **VM Users:** If your FreePBX is a VM, ensure its network adapter is set to **Bridged mode** (or the equivalent in your virtualization software) and that the host machine's physical adapter connected to this bridge is *also* isolated from your main network.
- **Dedicated Switch:** Now, connect *only* your FreePBX server, your **Desktop/Laptop PC**, and the Cisco 7911 phones to a **dedicated network switch**. This switch **must NOT be connected to your internet router** or any other DHCP server. This setup prevents IP conflicts and ensures your devices pick up the correct IP addresses and configuration from [dnsmasq](#).

3.1. Configure Your Server's Static IP Address

First, we need to assign a static IP address to your server's Ethernet interface. This will be the IP that dnsmasq serves from.

- **Identify Your Ethernet Interface:** Open a terminal on your FreePBX server and run:

```
ip a
```

Look for the name of your Ethernet interface. Common names are [eth0](#), [ens33](#), [enp0s3](#), [eno1](#), etc. Note down its exact name.

- **Edit Network Interfaces File:** Open the network interfaces configuration file using nano:

```
sudo nano /etc/network/interfaces
```

Comment Out and Add Configuration:

- Find the lines related to your server's current Ethernet interface (e.g., `iface eth0 inet dhcp` or `iface ens33 inet dhcp`).
- **Comment out** these lines by putting a `#` at the beginning of each.
- **Add the static IP configuration** below your commented lines. Replace `[YOUR_ETHERNET_INTERFACE]` with the actual interface name you found in step 1 (e.g., `ens33`).

Example `/etc/network/interfaces` content:

```
1 # This file describes the network interfaces available on your system
2 # and how to activate them. For more information, see interfaces(5).
3
4 source /etc/network/interfaces.d/*
5
6 # The loopback network interface
7 auto lo
8 iface lo inet loopback
9
10 # The primary network interface
11 #allow-hotplug [YOUR_ETHERNET_INTERFACE]
12 #iface [YOUR_ETHERNET_INTERFACE] inet dhcp
13
14 # DHCP and TFTP using dnsmasq
15 auto [YOUR_ETHERNET_INTERFACE]
16 iface [YOUR_ETHERNET_INTERFACE] inet static
17 address 192.168.1.1
18 netmask 255.255.255.0
```

- **Important:** The `address` should be `192.168.1.1` as this is the IP the `dnsmasq` configuration will expect for the TFTP server.
- **Save and Exit:** Press `Ctrl+X`, then `Y` to confirm saving, and `Enter` to write to the file.
- **Restart Networking Service:** Apply the new network configuration:

```
sudo systemctl restart networking
```

3.2. Configure `dnsmasq` for DHCP and TFTP

Now we'll configure `dnsmasq` to assign IP addresses to your phones and tell them where to find the TFTP server.

- **Edit `dnsmasq` Configuration File:** Open the `dnsmasq` configuration file:

```
sudo nano /etc/dnsmasq.conf
```

1. **Add dnsmasq Configuration:** Scroll to the bottom of the file (or clear its contents if you want a clean start, but make a backup first). Add the following lines. Remember to replace `[YOUR_ETHERNET_INTERFACE]` with your actual interface name.

Example `/etc/dnsmasq.conf` content:

```
1 interface=[YOUR_ETHERNET_INTERFACE]
2 bind-interfaces
3 dhcp-range=192.168.1.1,192.168.1.250,48h
4 dhcp-option=3,192.168.1.1
5 dhcp-option=6,8.8.8.8,8.8.4.4
6 enable-tftp
7 tftp-root=/srv/tftp
8 dhcp-option=150,192.168.1.1
9 dhcp-option=66,192.168.1.1
```

TFTP Root Directory : As a reminder, ensure all the files you downloaded and prepared in Step 1 and Step 2 (firmware, `SEP(MAC_ADDRESS).cnf.xml`, `dialplan.xml`, etc.) are copied into the `/srv/tftp/` directory on your FreePBX server. This is where dnsmasq will look for them when phones request them. You can use a software like [MobaXterm](#) that when ssh with the root user allows you to send files using SFTP to the folder we want. **You can use other dir but you have to mach the dnsmasq.conf first! You can also use the default dir i had allredy the `/srv/tftp/` setup**

3. **Save and Exit:** Press `Ctrl+X`, then `Y` to confirm saving, and `Enter` to write to the file.
4. **Restart dnsmasq Service:** Apply the new `dnsmasq` configuration:

```
sudo systemctl restart dnsmasq
```

3.3. Verify Your Network Configuration

- **Check Server IP Address:** Run `ip a` again to confirm your server's IP address is set to `192.168.1.1`.

Expected `ip a` Output (Example – details may vary):

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: [YOUR_ETHERNET_INTERFACE]: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:ab:cd:ef brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.1/24 brd 192.168.1.255 scope global
[YOUR_ETHERNET_INTERFACE]
    valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:feab:cdef/64 scope link
        valid_lft forever preferred_lft forever
```

You should see `inet 192.168.1.1/24` listed under your Ethernet interface.

- **Monitor dnsmasq Logs:** To watch dnsmasq actively and see if phones are requesting IPs and TFTP files, run:

```
watch systemctl status dnsmasq
```

When you plug in a Cisco 7911 phone, you should see messages indicating DHCP requests and TFTP file transfers. This is how you confirm dnsmasq is working as expected.

Step 4: Transferring Firmware via TFTP and Initial Phone Setup

This step involves transferring the necessary SCCP and then SIP firmware files to your Cisco 7911 using the dnsmasq TFTP server you've set up. This is a multi-stage process because the SIP firmware update often requires a recent version of SCCP firmware to be installed first.

Preparation Note: Ensure your FreePBX server (with `dnsmasq` running as DHCP and TFTP) and your Cisco 7911 phone are connected to the **isolated network switch**, along with your desktop/laptop PC. Your phone needs to get an IP address from `dnsmasq` to pull files.

4.1. Flashing SCCP Firmware (Initial Upgrade)

We begin by updating the phone to a recent version of SCCP firmware. This often acts as a bridge for the SIP firmware update.

1. **Clear TFTP Directory (Optional but Recommended):** For clarity and to ensure no old files interfere, it's a good practice to empty your TFTP root directory before copying the SCCP files.

```
sudo rm -rf /srv/tftp/*
```

1. **Copy SCCP Firmware to TFTP Root:** You have two main options for this:
 - **Option A: Command Line (if files are already on the server):** Navigate to the extracted directory of your pre-built configuration package (from Step 1) on your FreePBX server. Locate the folder containing the **SCCP firmware files**. Copy **all** the contents of this SCCP firmware folder into your `/srv/tftp/` directory:

```
# Assuming you are in the directory where you extracted the package on
the server
# Replace 'path/to/your/extracted/package/sccp_firmware_folder' with the
actual path
sudo cp -r path/to/your/extracted/package/SCCP/* /srv/tftp/
```

Ensure the files have correct permissions:

```
sudo chmod -R 755 /srv/tftp/
```

- **Option B: Using MobaXterm (Recommended for ease of use):**
 1. Open MobaXterm on your Desktop/Laptop PC, connect to our isolated network.
 2. Start an SSH session to your FreePBX server (using its new static IP: `192.168.1.1`) and as the root user – [Enable ssh root login](#).
 3. Once connected, MobaXterm's sidebar will show an SFTP browser. Navigate to the `/srv/tftp/` directory on the server.
 4. On your local machine, find the extracted SCCP firmware folder.

5. **Drag and drop** all the files from your local SCCP firmware folder directly into the [/srv/tftp/](#) directory in MobaXterm's SFTP browser.

- **Place Cisco 7911 into Upgrade Mode:** This tells the phone to look for new firmware from the TFTP server.
- **Unplug** the Ethernet cable (or power) from your Cisco 7911
- **Plug the Ethernet cable back in** to power on the phone.
- Immediately and repeatedly **press and hold the # button** until the headset light starts blinking red. This typically takes about 5-10 seconds.
- Once the headset light is blinking red, **release the # button**.
- Within 60 seconds, **press the sequence 123456789*0#**.

The phone should display a message indicating it's "Upgrading" .

[Here is a video of that on Youtube](#) – Credits

BSNetworking – Dont Forget to support the ones that help you! In the video he is using SIP but the reset setup is the same!

4. Monitor the Upgrade Process:

- **On your server, watch dnsmasq status:**

```
watch systemctl status dnsmasq
```

You should see entries indicating that the phone is requesting an IP address (DHCP activity) and then requesting firmware files via TFTP.

- **On the Phone:** The phone's screen will show progress messages as it downloads and installs the SCCP firmware. This process can take a few minutes.
- **Completion:** When the SCCP firmware upgrade is complete, the phone will typically restart. It might eventually display a message like "Unprovisioned" or "Registering" (and then fail to register) because it now has SCCP firmware but no matching configuration for your FreePBX SIP server. **This is normal and expected.**

Do NOT proceed to the next step until your phone has successfully completed the SCCP firmware upgrade and restarted.

4.2. Flashing SIP Firmware and Loading Configuration

Now that the SCCP firmware is updated, we can push the SIP firmware and your custom `SEP(MAC_ADDRESS).cnf.xml` file.

- **Clear TFTP Directory Again:** It's best to clear out the SCCP firmware files before adding the SIP ones to avoid confusion:

```
sudo rm -rf /srv/tftp/ *
```

- **Copy SIP Firmware and Configuration Files to TFTP Root:** Again, you have two options:
 - **Option A: Command Line:** Navigate to the extracted directory of your pre-built configuration package on your server. Locate the folder containing the **SIP firmware files**. Copy **all** the contents of this SIP firmware folder into your `/srv/tftp/` directory. **Crucially, ensure your custom `SEP(MAC_ADDRESS).cnf.xml` file (from Step 2) and your `dialplan.xml` file are also present in `/srv/tftp/` at this stage.**

```
# Assuming you are in the directory where you extracted the package
# Replace 'path/to/your/extracted/package/sip_firmware_folder' with the
actual path
sudo cp -r path/to/your/extracted/package/SIP/* /srv/tftp/
```

Ensure permissions are correct:

```
sudo chmod -R 755 /srv/tftp/
```

- **Option B: Using MobaXterm (Recommended):**
 1. In your existing MobaXterm SSH session to `192.168.1.1`, navigate to the `/srv/tftp/` directory in the SFTP browser.
 2. On your local machine, find the extracted **SIP firmware folder**.
 3. **Drag and drop** all the files from our **SIP** folder directly into the `/srv/tftp/` directory.
- **Place Cisco 7911 into Upgrade Mode (Again):** You need to repeat the process to force the phone to look for new firmware and configuration:

- **Unplug** the Ethernet cable (or power) from your Cisco 7911.
- **Plug the Ethernet cable back in** to power on the phone.
- Immediately and repeatedly **press and hold the # button** until the headset light starts blinking red.
- Once the headset light is blinking red, **release the # button**.
- Within 60 seconds, **press the sequence 123456789*0#**.

[Here is a video of that on Youtube](#) – Credits

BSNetworking – Dont Forget to support the ones that help you!

4. Monitor SIP Upgrade and Configuration Loading:

- On your server, keep watching **dnsmasq** status:

```
watch systemctl status dnsmasq
```

You will see more DHCP requests and TFTP transfers as the phone downloads the SIP firmware and then your [SEP\(MAC_ADDRESS\).cnf.xml](#) file.

- **On the Phone:** The phone may restart several times. It's common for Cisco phones to upgrade in stages, sometimes downloading one SIP file, restarting, then downloading another. Be patient.
- **Completion:** The phone will eventually finish upgrading. It will then attempt to register with FreePBX using the details in your [SEP\(MAC_ADDRESS\).cnf.xml](#). Since your FreePBX server is currently on the isolated [192.168.1.x](#) network and likely doesn't have its "normal" production IP, the registration attempt will fail, and the phone might display "Registering" forever **This is normal for now**.

At this point, your Cisco 7911 has the correct SIP firmware and has loaded its FreePBX configuration. You can now unplug the phone from the isolated network. It's ready for the final step: restoring your server's network and setting up a dedicated TFTP server for continued phone operations.

This is the final stretch! Now we'll restore your server's network to its normal operation and configure the phones to find their configuration files on your main network.

Step 5: Finalizing Setup and Connecting to Main Network 🔌 ✨

Now that your Cisco 7911 phones are running SIP firmware and have received their initial configuration, it's time to put your FreePBX server and phones back onto your main network for full functionality.

5.1. Restore Your FreePBX Server's Network Configuration

First, we need to revert the static IP changes made to your FreePBX server's network interface so it can reconnect to your main network and obtain its regular IP address (likely via DHCP from your main router, or a static IP you've previously assigned).

1. **Edit Network Interfaces File:** Open the network interfaces configuration file:

```
sudo nano /etc/network/interfaces
```

1. **Revert Changes:**

- **Comment out** the static IP configuration lines you added in Step 3.1.
- **Uncomment** or re-add the original configuration lines for your Ethernet interface that allowed it to obtain an IP via DHCP (or its previous static configuration).

Example Reverted `/etc/network/interfaces` content (assuming original DHCP):

```
1  # This file describes the network interfaces available on your system
2  # and how to activate them. For more information, see interfaces(5).
3
4  source /etc/network/interfaces.d/*
5
6  # The loopback network interface
7  auto lo
8  iface lo inet loopback
9
10 # The primary network interface
11 allow-hotplug [YOUR_ETHERNET_INTERFACE]
12 iface [YOUR_ETHERNET_INTERFACE] inet dhcp
13
14 # DHCP and TFTP using dnsmasq
15 #auto [YOUR_ETHERNET_INTERFACE]
16 #iface [YOUR_ETHERNET_INTERFACE] inet static
17 # address 192.168.1.1
18 # netmask 255.255.255.0
```

3. **Save and Exit:** Press `Ctrl+X`, then `Y` to confirm saving, and `Enter` to write to the file.

5.2. Configure `dnsmasq` for TFTP Only

Now that your FreePBX server is back on the main network, `dnsmasq` should **no longer provide DHCP services** (your router or main DHCP server will handle this). We will reconfigure `dnsmasq` to act *only* as a TFTP server.

- **Edit dnsmasq Configuration File:** Open the `dnsmasq` configuration file:

```
sudo nano /etc/dnsmasq.conf
```

- **Modify `dnsmasq` Configuration for TFTP Only:**
 - **Comment** the `dhcp-range` line.
 - **Comment** the `dhcp-option=66` and `dhcp-option=150` lines.
 - **Ensure `enable-tftp` and `tftp-root=/srv/tftp/` are still present and uncommented.**
 - Ensure the `interface` directive still points to the correct Ethernet interface that is now connected to your main network.

Example `/etc/dnsmasq.conf` content (modified):

```
1 #interface=[YOUR_ETHERNET_INTERFACE]
2 #bind-interfaces
3 #dhcp-range=192.168.1.1,192.168.1.250,48h
4 #dhcp-option=3,192.168.1.1
5 #dhcp-option=6,8.8.8.8,8.8.4.4
6 enable-tftp
7 tftp-root=/srv/tftp
8 #dhcp-option=150,192.168.1.1
9 #dhcp-option=66,192.168.1.1
```

- **Save and Exit:** Press `Ctrl+X`, then `Y` to confirm saving, and `Enter` to write to the file.
- **Restart dnsmasq Service:** Apply the new `dnsmasq` configuration:

```
sudo systemctl restart dnsmasq
```

- **Restart Networking Service:** Apply the new network configuration:

```
sudo systemctl restart networking
```

Your server should now obtain its regular IP address from your main network's DHCP server or connect with its previous static IP. Verify its new IP by running `ip a`.

5.3. Connect Everything to Your Main Network

1. **Connect Server:** Ensure your FreePBX server is now fully connected to your main network (where your internet router and other network devices are).
2. **Connect Switch:** Connect the dedicated PoE switch (with your Cisco 7911 phones plugged in) to your main network.
3. **Power On Phones:** Plug in your Cisco 7911 phones. They should now receive an IP address from your main network's DHCP server (your router).

5.4. Configure Alternative TFTP Server on Cisco 7911 Phones

Unless your main router or DHCP server specifically supports and is configured to send **DHCP Option 66 (TFTP Server Name)** or **Option 150 (Cisco TFTP Server Address)**, your phones won't automatically know where to get their configuration files from your FreePBX server. For most home/small office routers, this is not supported.

Therefore, you'll need to manually tell each Cisco 7911 phone the IP address of your FreePBX server (which is now also your TFTP server) as an "Alternative TFTP Server."

1. **Access Phone Settings:**
 - On your Cisco 7911 phone, press the **Settings button** (the physical globe-like button).
 - Scroll down and select **3. Settings**.
 - Scroll down and select **2. Network Configuration**.
 - Scroll down and select **1. IPv4 Configuration**.
2. **Navigate to Alternative TFTP:**
 - Scroll down to option **16. Alternative TFTP**.
 - To enable it, press ****#** and wait a moment. The option to "Yes" should appear as a choice.
 - Press the **Yes softkey**.
3. **Set TFTP Server IP:**
 - Scroll down to option **17. TFTP Server 1**.
 - Press the **Edit softkey**.

- Enter the **IP address of your FreePBX server** (your server's normal IP on the main network, not [192.168.1.1](#)). Use the phone's keypad. The ***** button often acts as a **.** (dot) for IP addresses.
- Press the **Validate softkey** when done.

4. Save and Exit:

- Press the **Exit softkey** repeatedly until you are prompted to save your changes. Confirm to save.
- The phone should restart and attempt to re-register with your FreePBX server, this time successfully, as it now has an IP from your main network and knows where to find its configuration files.

Enjoy! Your Cisco 7911 phone should now fully register with your FreePBX system, allowing you to make and receive calls.

Common Issues & Troubleshooting Tips

Even with a detailed guide, setting up older VoIP phones like the Cisco 7911 can sometimes present challenges. Here are some common issues you might encounter and how to troubleshoot them, along with an important notice regarding the software.

1. Typos and Configuration Errors

This is, by far, the most frequent culprit! A single misplaced character or incorrect value in your configuration files can prevent the phone from working.

- **Double-Check [SEP\(MAC_ADDRESS\).cnf.xml](#):**
 - **MAC Address:** Ensure the filename [SEP\(MAC_ADDRESS\).cnf.xml](#) precisely matches your phone's MAC address.
 - **Placeholders:** Verify that every instance of [\[===FreePBX IP===\]](#), [\[===Extension===\]](#), [\[===Password===\]](#), [\[===Phone Name===\]](#), and [\[===Line Name===\]](#) has been correctly replaced with your specific details.
 - **Syntax:** Even small things like missing angle brackets ([<](#), [>](#)), incorrect quotes (["](#)), or extra spaces can break the XML. Use a plain text editor (like Nano or Notepad++) that doesn't add hidden formatting.
- **Verify [dialplan.xml](#):** Ensure your [dialplan.xml](#) is present in [/srv/tftp/](#) and correctly configured for your dialing patterns. A malformed dial plan can prevent outbound calls.
- **dnsmasq.conf Scrutiny:**
 - **Interface Name:** Is [interface=\[YOUR_ETHERNET_INTERFACE\]](#) correct? A typo here means [dnsmasq](#) won't listen on the right network.
 - **Paths:** Is [tftp-root=/srv/tftp/](#) correctly specified?

- **Commented Lines:** When you commented out lines for DHCP, did you miss any, or did you accidentally comment out essential TFTP lines?

2. NAT Settings in Asterisk/FreePBX SIP Configuration

Incorrect Network Address Translation (NAT) settings are a very common source of one-way audio, no audio, or registration issues with SIP phones, especially if your FreePBX server is behind a NAT (e.g., your router) or if phones are connecting from outside your local network.

- **Asterisk SIP Settings (FreePBX GUI):**
 - In your FreePBX administrative interface, go to **Settings > Asterisk SIP Settings**.
 - **NAT:** Look for the **NAT** setting. While some older guides might suggest “No” or “Yes”, for most modern FreePBX/Asterisk deployments, especially if your server is local, setting this to “**Never**” is generally recommended.

3. Permissions Issues

If the phone isn't pulling files, even if [dnsmasq](#) appears to be running, it might be a file permission problem.

- **TFTP Root Permissions:** Ensure the [/srv/tftp/](#) directory and all files within it have read permissions for the TFTP server process. The `sudo chmod -R 755 /srv/tftp/` command should generally handle this, but if you're manually moving files, double-check.

Important Notice Regarding Cisco 7911 Software

Please note that **all software and firmware provided in the pre-built package for the Cisco 7911 is from Cisco Systems, Inc.** The distribution of this software is solely for educational and personal use in configuring your own equipment.

The Cisco 7911 IP Phone was last updated with official firmware in **2007**, and as such, it is considered an **old and unsupported device** by Cisco. This means Cisco no longer provides official support, updates, or direct assistance for issues related to this model or its firmware.

Should you encounter any problems with the phone's operation, please refer to this guide, community forums, or other unofficial resources for assistance. **If Cisco Systems, Inc. has any concerns regarding the distribution or use of this firmware, please contact me immediately, and it will be removed.** This guide is provided as a helpful community resource for users leveraging legacy hardware with open-source VoIP solutions.

