

Asterisk SIP Trunk Settings – Vestalink

[Vestalink](#) is a new SIP trunk provider that has sprung up as a replacement for Google Voice trunking within Asterisk servers. They offer a very attractive pricing plan with 2000 mins/month going for \$39.99 per year, and unlimited plans at \$49.99 per year! This provides a single DID along with two SIP channels for the possibility of holding two simultaneous calls from one account (*call waiting*). The folks over to [Nerd Vittles](#) have written up a great guide to getting the service working with FreePBX, however I was hard pressed to find any documentation on how to get this working under a standard Asterisk server using only the CLI. Applying the Nerd Vittles FreePBX configuration to Asterisk CLI did not work out of the box. After some messing around I was able to successfully register my Asterisk server to my Vestalink account, and I have provided generic configurations that should help anyone looking to integrate a Vestalink SIP trunk with a vanilla Asterisk installation.

In `sip.conf` the following entries are necessary in order to register your Asterisk server to your Vestalink account. Note that the registry entry needs to be placed under the `[general]` section. Replace `YOURPHONENUMBER` with the 10 digit DID that you chose when signing up for an account [here](#).

Now in `extensions.conf` the following entries need to be made for inbound and outbound calling over your SIP trunk.

Endian Firewall – Getting SIP Phones To Work

There are a few things that need to be done on a new Endian Firewall (Community or UTM Appliance) installation in order to get it to play nice with SIP based voice over IP phones. If your phones are registering to an internal VoIP server you should not have any issues, however if your server or SIP provider are external to your network then some settings will need to be modified.

First in order to even get your phones to register to the external SIP provider or VoIP server you will need to disable the outbound firewall, or add rules to allow traffic over UDP port 5060 (SIP) and the UDP port range 10000:20000 (RTP). Once this is done you will notice that your SIP phones will register properly to the VoIP service and normal outbound and inbound calls will work properly. However if you try to dial another extension on your internal LAN the extension will ring and the call will be setup, but there will be no audio.

To understand why this issue occurs first we need to visualize what is happening when the first extension dials out to the second and what happens to the packet as it travels over the Internet. In very simple terms the packet leaves the first phone destined for the SIP provider or external VoIP server. Once it gets there the server attempts to establish the connection to the second extension which is located behind the same router/firewall as the first extension. Since both phones are located behind the same router they both appear to be coming from the same external IP address. The router does not understand where to send the packets from there since the packet header information only contains the public IP address, not the address of the phone it is destined for on the internal network. This is where port forwarding or NAT usually

comes in and fixes the issue, but in normal practice this is a one to one relation and since you probably have many phones on your internal network it does not apply.

In order to fix the issue we need to allow the packets coming from and going to the phones on the internal network to still retain information that will allow them to be routed properly. To do this create two new rules in the firewall menu on the Endian under Port Forwarding/NAT. These rules need to be coming from the RED interface over the SIP and RTP ports destined for the entire LAN subnet not a single node like a normal port forwarding/NAT rule.

```
RED UDP 5060 -> 192.168.0.0/24 UDP 5060 #SIP
```

```
RED UDP 10000:20000 -> 192.168.0.0/24 UDP 10000:20000 #RTP
```

Once these rules are established you need to SSH into the Endian and modify some of the netfilter kernel modules to allow something called SIP Inspection so that the packets can be routed to the correct devices. SSH into your Endian and place the following script in /etc/cron.cyclic so that it is run every time the device is rebooted.

Name the script sip_fix and make sure you run the command

To make it executable.

Once this script is in place reboot your EFW and enjoy your now fully functional VoIP phones!

AsteriskNow – IPTables Firewall Configuration

In a [previous guide](#) I discussed how to setup an [AsteriskNow](#) server with Polycom phone support. In this guide I will illustrate how to tighten up your server's security by using the IPTables firewall already installed in the distribution.

IPTables should already be setup and running on the server, however no rules have been applied. You can verify this by doing the following as the root user:

This should report the following:

Verify there are no rules present:

You should see:

Now it's time to add some rules. You can copy the following text to a file and import it into IPTables:

Save the file as iptables.bak and copy it to /etc/iptables.bak

Now import the file into IPTables:

And verify that the rules have been committed:

You should now see:

Now save the new IPTables settings:

That's it! Your server is now blocking all incoming traffic by

default, and only allowing connections to the ports that are necessary to do it's job. Specifically:

Port 123 UDP for NTP (Time)

Port 69 UDP for TFTP (Phone provisioning)

Port 5060 UDP for SIP (Phone Calls)

Port 10000-20000 UDP for RTP (Phone Calls)

Port 22 TCP for SSH (SSH Connection)

Port 80 & 443 TCP for HTTP/HTTPS (Web)

If you need to open another port just use the following syntax at the command line:

example for SSH over TCP port 22

To specify a range of ports do the following:

example for RTP over UDP ports 10000-20000



You can then save the new configuration by doing:

And if your completely satisfied and want to back up the configuration do:

AsteriskNow – Polycom SoundPoint IP 335 & 550 Provisioning In FreePBX

[AsteriskNow](#) is a free and powerful turnkey open source PBX system that can be combined with high quality Polycom phones to create an enterprise level VoIP solution. In this guide I

will outline the steps needed in order to install AsteriskNow and setup automatic configuration and firmware provisioning for your Polycom SoundPoint IP 335 and 550 SIP phones.

Polycom SoundPoint IP 335	Polycom SoundPoint IP 550
	

The first step is to download the ISO image from [here](#). Choose 32 bit or 64 bit depending on the hardware your installing on. Burn the ISO to a CD then boot the computer. You should then see the following screen:



Type 1 and press Enter to install Asterisk 1.6 with the FreePBX gui.

The installer will begin to load and if the hard disk has not been formatted yet it will ask if you would like to initialize the hard drive. Choose **Yes**. You will then see a screen that gives you partitioning options for your hard drive. If you are only using this computer for AsteriskNow then choose: **“Remove all partitions on selected drives and create default layout”**. Click next, then click Yes when it asks: “Are sure you want to do this”.



Choose your region and click next:



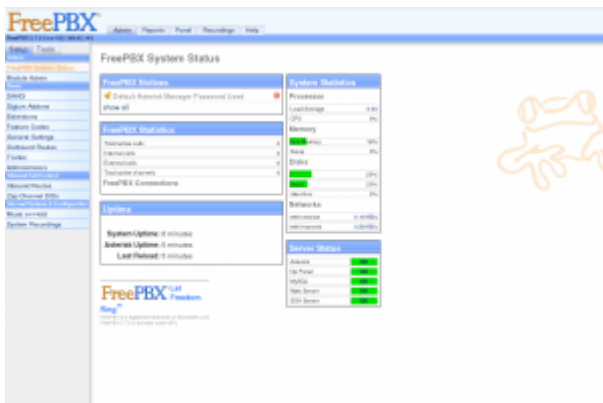
Then set the root password, click next and wait for the installation to complete. Once the installer completes press Reboot and remove the CD from the drive.

When the machine comes back up you will be presented with the Setup Agent menu. You can use this tool to configure your network interface card if you need to supply a static IP address. For now we will assume DHCP and just let it time out or exit to proceed. You can call up the network setup utility at any time later and add a static IP, configure the hostname, or add DNS entries by using the command:

The system should now be at a login prompt. Log in as the user root with the password you setup during the installation. Now it is time to update the system with the most current software.

This command will download all of the latest software updates and then ask you if you want to install them. Type y during any prompts. Once the updates are complete reboot your computer by issuing the command:

After the server comes back up from the reboot you will see the login prompt. Above the prompt there should be a line that tells you where to point your web browser to configure AsteriskNow with FreePBX. Point your web browser to the address, click on the **FreePBX Administration** link, and log in with the user **freepbx** and the password **fpbx**. You should now see the FreePBX Status Page.



Now for security purposes we need to change the admin user's password. This will also prevent you from being locked out after we upgrade FreePBX, since for some reason the freepbx user becomes inaccessible after the upgrade. To do this click on the **Administrators** link in the Setup menu. You will see a small list in the top left, one button says **Add User** and the other says **admin**. Click on **admin** then change the password in the password box and hit **Submit Changes**.



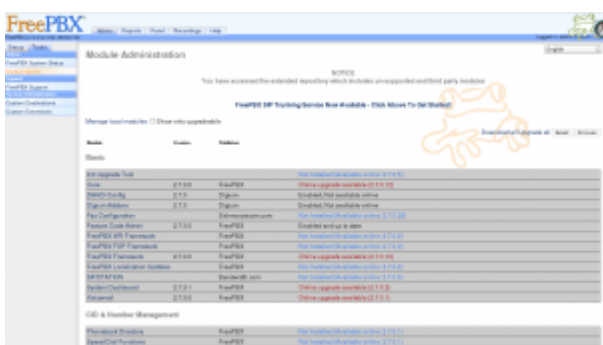
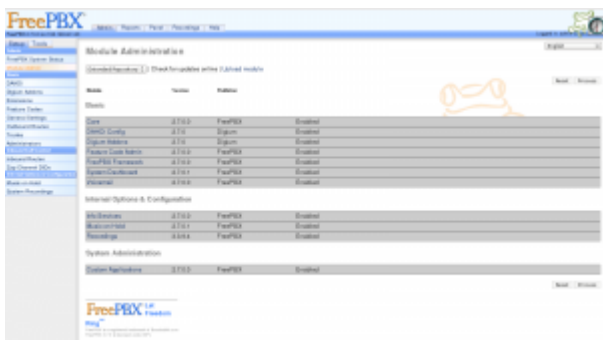
Notice the orange button that now appears on top of the page. You need to click that every time you make a configuration change so that the change can be committed to the system, and Asterisk can be reloaded so the changes can take effect.



Now you can log out and log back in as admin with your new password. Use this account to log in from now on instead of the freepbx user.

Now it is time to update FreePBX from 2.7 to 2.9. There are a few steps to this and they **must** be performed in the correct order. If you are doing this installation in a Virtual Machine I highly suggest you make a snapshot now that you can revert back to if something goes wrong. If you don't know what a Virtual Machine is then just disregard that last statement!

To upgrade FreePBX select **Module Admin** from the Setup menu. In the drop down box select **extended repository**, click Ok on the prompt then click on **Check for updates online**.



You will now see a list of packages, some of them will be

marked with an available update, and some will be marked as not installed. The only package we are concerned with at the moment is the **2.8 Upgrade Tool**. Click on it and then select **Download and Install**. Then click on **Process**.



Confirm the installation, click return when the orange box pops up, then click the orange **Apply Configuration Changes** button at the top. Another orange box will pop up, click **Continue with reload**.

You will now have a new menu item in the **Setup** menu called **2.8 Upgrade Tool**. Click on it to use the upgrade tool.



Follow the instructions on the page **TO THE LETTER!**. First you will press the **Upgrade Now** button on the page to update the database. Then you will go back over to the module admin page, click on check for updates online, and **ONLY UPDATE** the FreePBX Framework module. After the FreePBX Framework is updated select extended repository from the drop down list and check for updates online again. This time click on the **Upgrade All** link to select all modules that need to be updated, change the **2.9 Upgrade tool** to **"No Action"**, and then click **process**. Apply the configuration changes and reload.

Now we can proceed to the 2.9 Upgrade which is basically the same exact process as the 2.8 upgrade except we need to install a few dependency modules first. In the module admin click on check for updates online then select the following 2 modules and set them to **“Download and Install”**:

FreePBX ARI Framework
FreePBX FOP Framework



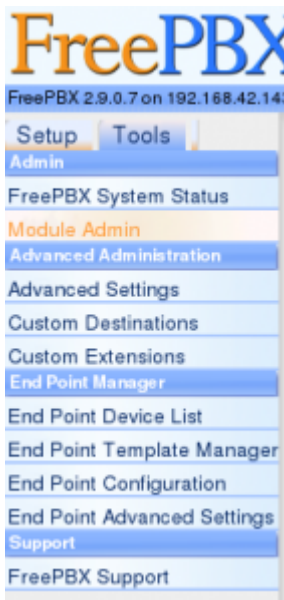
After they are installed apply the configuration changes and reload Asterisk. Then in module admin click check for updates online again, select the 2.9 Upgrade Tool and choose **“Download and Install”**, then click on **process** to install it. When it is finished installing apply the configuration changes and reload. Then select **2.9 Upgrade Tool** from the Setup menu.



Follow the instructions on the page **TO THE LETTER!**. First you will press the **Upgrade Now** button on the page to update the database. Then you will go back over to the module admin page, click on check for updates online, and **ONLY UPDATE** the FreePBX Framework module. After the FreePBX Framework is updated select the basic and extended repositories, and check for updates online again. This time click on the **Upgrade All** link

to select all modules that need to be updated, and then click **process**. Apply the configuration changes and reload. Now make sure the basic and extended repositories are selected and check for updates online again. Choose **Upgrade All** and then click process. Once it is finished updating the modules apply the changes and reload. Your system should now be fully updated to FreePBX 2.9. Now we can install the Endpoint Manager module that will be used to setup our Polycom phones.

In the module admin make sure the basic and extended repositories are selected then click check for updates online. Now scroll down the list a bit and look for the Endpoint Manager section. Click on **PBX End Point Manager** and choose **“Download and Install”** then click on **process**. Apply the configuration changes and reload Asterisk. You should now have a few **End Point Manager** links in your tools menu.



Now it is time to take a step back from the FreePBX interface and get our hands dirty at the command line! We need to setup and install a few utilities that will be used by the End Point manager to configure the phones. Specifically we need to install nmap, and configure a tftp server that the phones will

use to download their configurations and firmware from. We will also need to setup an NTP server that the phones will synchronize their time with.

In a terminal on the AsteriskNow server do the following as the root user:

First install NMAP, type y for any prompts:

That was easy! Ok now we need to configure a NTP time server, the package is already installed but the service is not running. To start it up and set it to autostart on boot type the following:

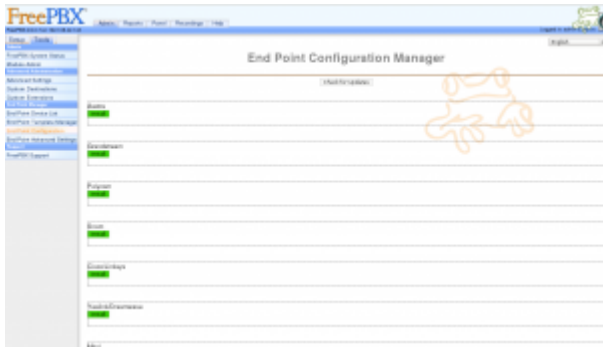
Now let's setup the tftp server:

Thats it for the command line! You can now exit out of the terminal and log back into the FreePBX web interface. Next we will configure the End Point Manager so that it can support the Polycom phones.

Click on the **Tools** menu then click **End Point Manager Advanced Settings**. Click the **Determine for me** link next to the **IP address of phone server** box. Then make sure the NMAP executable path is set to `/usr/bin/nmap`. Then click on the **Update Globals** button.



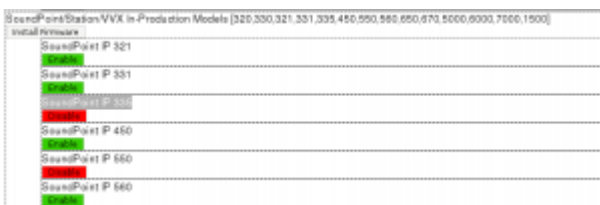
Next click on the **End Point Configuration** link. Then click the **Check for Updates** button, you should now see a list of company names.



Now click the green **install** button under the **Polycom** section to download configuration files for all of the supported polycom phones.



Now click the green **Enable** button under the **SoundPoint IP 335** and **SoundPoint IP 550** listings to enable them to be used when configuring an endpoint.



Now we need to install the firmware for the PolyCom phones to the /tftpboot folder. This is done by clicking the **Install Firmware** button under the **SoundPoint/Station/VVX In-Production Models [320,330,321,331,335,450,550,560,650,670,5000,6000,7000,1500]** heading. This process will take a few moments to download the

files.

SoundPointStation/VVX In-Production Models [320,330,321,331,335,450,550,560,650,670,5000,6000,7000,1500]
Remove Firmware

Now we need to create a SIP extension that we can bind a phone to. Click on the **Setup** menu then choose **Extensions** and then choose **Generic SIP Device** in the drop down box. For now lets just get the extension working and not worry about any of the other settings. Enter in information for the following variables:

User Extension

Display Name

secret

Then apply the configuration changes and reload Asterisk.

Add SIP Extension

Add Extension

User Extension: 123
Display Name: secret
CID Num Alias:
SIP Alias:

Extension Options

Outbound CID:
Ring Time: Default
Call Forward Ring Time: Default
Outbound Concurrency Limit: No Limit
Call Waiting: enable
Call Screening: disable
Pinless Dialing: disable
Emergency CID:

Assigned DID/CD

DID Description:
Add Inbound DID:
Add Inbound CID:

Device Options

This device uses sip technology.
secret: 12234
displayname: RVC 2833

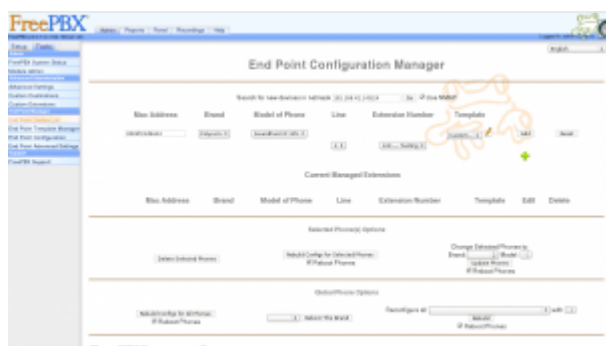
Next click on the newly created extension in the list on the right hand side and change the **nat** value to **Yes**.

Device Options

This device uses sip technology.

secret	tt1234
dtmfmode	RFC 2833
canreinvite	No
context	from-internal
host	dynamic
trustpid	Yes
sendpid	No
type	friend
nat	Yes
port	5060
quality	yes
qualityfreq	60
callgroup	
pickupgroup	
disallow	
allow	
dial	SIP/123
accountcode	
mailbox	123@device
vmexten	
deny	0.0.0.0/0.0.0.0
permit	0.0.0.0/0.0.0.0

Now let's configure an End Point device that will be assigned to this extension. Click on the **Tools** menu and then **End Point Device List**. Enter the MAC Address of the phone (usually located on the bottom), select the brand from the drop down list (Polycom), use line 1, select the extension you just created, then click the **Add** button.



That's it for the End Point Configuration. All that is left to now is setup the phone's boot server option to point to the IP address of the AsteriskNow server.

Power up your PolyCom phone enter the setup menu and use the password **456** when prompted. Make sure **DHCP** is set to **enabled** and in the **DHCP Menu** the **Boot Server** option is set to **Custom+Opt.66**. Next in the **Server Menu** make sure the **Server Type** is set to **TrivialFTP** then set the **Server Address** option to the IP address of your AsteriskNow server. Once this is all set reboot the phone and it should connect to the server,

synchronize it's time, download and install the new firmware files, and update it's configuration with the extension settings.

To setup more phones just create more extensions and corresponding devices in the End Point Manager. Then setup the phone boot options and you should be good to go!

AsteriskNow – How To Change The Default FreePBX Password

[AsteriskNow](#) is a full blow free turnkey PBX Linux distribution based on CentoOS that can be used in many different types of telephony applications. It is very easy to install and initially setup however changing the default freepbx password can be a bit challenging since there is no link or menu anywhere in the interface to do so!

The default username and password for the FreePBX database and Web frontend are freepbx/fpbx. Having no clear way of changing this leaves a very large security hole open on your newly created PBX system! Here is how you can do it manually via the command line.

Log in as root at the command line and start the mysql command line client:

Select the mysql database:

Update the freepbx user password:

Now we need to update a few files with the new password

In `/etc/amportal.conf` edit this line:

In `/etc/asterisk/cdr_mysql.conf` edit this line:

In `/etc/asterisk/extensions_additional.conf` edit this line:

Now restart the `amportal` service:

You should now be able to log into your FreePBX web frontend with the user `freepbx` and your new password.