

# Phone Registration

## Enable Logging

First set the log level for the Proxy and the Registrar to at least INFO level. This is enough to ensure that SIP messages are logged. In rare cases, it may be helpful to set the Registrar logging to DEBUG level; this will log *lots* of additional information, and can consume considerable disk space.

See [Logging](#) for how to change these settings.

## Certificates

Check if certificates are properly installed, run:

```
service sipxpbx configtest
```

Make sure that httpd is **not** configured to start independently of sipXpbx. To prevent httpd from starting automatically see [previous section](#).

## Basics

First: **The fact that you can make a call does *not* indicate that your phone is registered.** Registration only controls how a call is sent *to* the phone.

In the administration console, under the Diagnostics heading there is a Registrations page that displays all registrations.

In the user portal, there is a Phones tab that displays the registration status of any phone that has a line for the user on it.

## SIP Registrar Log Messages

Turn the logging level to debug on Registrar and tail the log file (`/var/log/sipxpbx/sipregistrar.log`). It will report what was not valid about the registration.

## Hostname resolution

? Unknown Attachment

Check `/etc/hosts` for proper name resolution

```
http://track.sipfoundry.org/browse/XPB-523
```

## Multiple Network Cards

If your machine has 2 networks cards, normally the right IP address is used. To verify this, make sure both of these values are identical  
Code: IP address resolution check

```
/usr/bin/get_def_addr
```

1. Second IP address (e.g. Allow from 127.0.0.1 **YOUR IP HERE**)  
grep "Allow from" `/etc/sipxpbx/httpd-sipxchange-common.conf`

## SIP password misconfigured

In order for the phone to register correctly SIP password known to registrar must match the one configured on the phone. sipXconfig ensures that this is the case for all managed phones. However if you configure your phone manually, you have to make sure that you use the same SIP password when adding user and when configuring the phone.

Important: **Release 3.0** and higher of sipxcom automatically generate a SIP password when a new user is created. The SIP password has to be programmed into the phone and is required for the line to register. This password is **different from the PIN** entered on the "New User" page. In order to see or change the assigned SIP password, click on the "Show Advanced Settings" link.

## General failure sending user information to SIP Registrar

Assuming you've added the user information into the web admin UI correctly, it's possible the information did not get sent to appropriate sipX servers.

**TEST:** See if your user is listed as a valid user of the PBX

```
grep {username} /var/sipxdata/sipdb/credential.xml
```

If they are, see if your user has registered by checking to the SIP Registrar database

```
grep {username} /var/sipxdata/sipdb/registration.xml
```

**SOLUTION:** Edit users PIN in web ui and look for detailed error messages in sipxconfig.log and resolve accordingly

## Mismatch of REGISTER Domain and SipX Domain

There is a good reason for the [Set up DNS Entries](#) section at the **very beginning** of the sipX Installation Manual.

sipX is **particularly** strict about who can register. \*It is not enough that your *realm*, *userid*, and *password* are correct in a REGISTER request. \* In addition, the [@domain.com](#) portion of the "TO:" field of your REGISTER packet must match the value of "Domain Name" in System -> Domain of your SipX installation.

If the values do not match, an endpoint that is trying to register will continue to get "401 Unauthorized" messages even though their REGISTER packets look as if they contain all of the correct values. Technically, this is a DNS issue and not a problem with SipX, but since most other SIP based systems are not as vigilant in their enforcement of this, the issue may not be obvious to even an experienced SIP administrator.

A good way to test this is to inspect your failed REGISTER packets using [Wireshark](#) or the contents of your sipproxy.log. Compare the TO: field of the SIP REGISTER message to the "uri" field from the output from "sipdbtest display credential" which can be run from the command prompt of your SipX installation. The fields must match.

### CONFIGURING ENDPOINTS

#### ? Unknown Attachment

It appears that there are two schools of thought among endpoint manufacturers with regard to how the [@domain.com](#) portion of the "To:" field should be populated.

**School 1:** The value found in the "SIP Proxy" setting is added to the value found in the "UserID" field with an "@" symbol in the middle.

This approach pretty much requires you to have properly configured DNS to get even a basic test installation of SipX functioning because the phone must resolve the IP Address of the Proxy. One way around this in a lab environment is to put the IP Address of the SipX box into the SIPXCHANGE\_DOMAIN\_NAME field (Don't forget to restart your SipX services) and then put the same value into the "SIP Proxy" setting of the phone. Most of the time with this configuration, the resulting "To:" field will end up being, "UserID@IPAddress". Since IPAddress will match the value of SIPXCHANGE\_DOMAIN\_NAME, SipX will allow the registration to take place.

**School 2:** In addition to the "SIP Proxy" setting, a separate "SIP Domain" field (not to be confused with the "SIP Realm") is provided and whatever value is in that field is placed after the "@" symbol in the "TO:" field.

This makes things easy regardless of how your DNS is configured because the endpoint pretty much takes your word for it. No DNS is required to get this type of endpoint registered. As long as the "SIP Domain" field and the SIPXCHANGE\_DOMAIN\_NAME match, and all of the other credentials are valid, SipX handles the REGISTER request perfectly.

--Spr 05:46, 14 Apr 2005 (CEST)

## Late Registration Refresh

If your phone has this problem, you'll see these symptoms:

- Soon after booting, the phone will appear in the Registrations display, its own user interface will display that it is registered. Calls will ring normally on the phone.
- Some time later, the entry in the Registrations display will be marked 'expired' or be removed, but the user interface of the phone indicates that it is still registered. Calls do not ring on the phone (they are routed to any contacts in the forwarding plan or to voicemail).
- After some more time, the phone reappears in the Registrations display and calls once again ring on the phone.

The time during which the phone is not registered will vary, sometimes considerably.

Those symptoms indicate that that phone is not correctly interpreting the expiration time in the response from the registrar.

### Why does this happen?

The sipxcom registrar attempts to spread registrations over time to smooth load (both on itself and on the network) caused by registration refreshes. It does this by shortening the granted registration time from the requested time by a random factor:

- Normal length requests (for  $\geq$  twice the allowed minimum) are randomized between twice the minimum and the configured maximum.

- Short requests (for < twice the minimum) are randomized between the minimum and the requested amount.
- Minimum length requests are not shortened (that would be hypocritical, since we would not accept a request that short).
- Requests less than the minimum are rejected (with a header that tells the phone what the minimum allowed value is).

When a phone requests a registration, it asks for a specific duration (the 'expires' time) in seconds. In a successful registration response, the 'expires' field parameter on the Contact header is set to the actual time the server has granted for the registration, which may (at the servers option) be less than the requested duration. If the phone does not refresh by that time, the registration will become inactive (it may still be visible in the database for other reasons, but will be marked 'expired' on the registrations display and will not be used to route calls).

## What can I do about it?

You should alert the phone manufacturer to the defect (feel free to log an issue for tracking purposes at [track.sipfoundry.org](http://track.sipfoundry.org) in the External project and then send them the URL there).

If this is big problem for you, you can configure the minimum and maximum registration times to be the same as that you configure the phones to use, so the randomization won't have any room to make the granted time different. This workaround is **not recommended**, as it can cause very large load spikes to develop in your registrar (SIP systems in general behave poorly in the presence of very spiky load, especially if the UDP transport is being used).

## localhost misconfigured

**TEST:** Try pinging localhost

```
ping localhost
```

sipXconfig will try to connect to SIPXCONFIG database located on 'localhost'. In order for that to work you have to be able to resolve localhost to loopback interface.

**SOLUTION** Make sure you have the following line in your /etc/hosts file

```
127.0.0.1 localhost.localdomain localhost
```

## TFTP server not started or misconfigured

**TEST:** Try to download from from your TFTP server. You may need to install tftp client software and your tftp client may have slightly different command syntax

```
tftp localhost -c get SIPDefault.cnf  
less SIPDefault.cnf
```

**SOLUTION:** Walk through installation notes for TFTP server installation and look in /var/log/messages for TFTP error messages.

## Phones do not pick up profiles

### Make sure all the components are installed

Verify that you installed the following RPMs:

```
rpm -q sipxconfig-tftp  
rpm -q sipxconfig-ftp
```

### Make sure profiles are generated

Verify that you generated the profiles for the phones in question inside Config Server. Go to the Job Status page to verify that the profile generation succeeded.

Using console access to the sipX machine, go to the directory /var/sipxdata/configserver/phone/profile/tftp/ and verify that the phone profile files got generated. For Snom phones go to /var/sipxdata/configserver/phone/profile/docroot/.

### Verify that the TFTP server is working

From any Linux machine try this (assuming your sipX host is [sipx.example.com](http://sipx.example.com)):

```
1. tftp sipx.example.com
tftp> verbose
tftp> status
tftp> get sipx7912.logo (or some other file you know exists)
Received 309 bytes in 0.3 seconds
tftp> quit
```

Installing tftp: `yum install tftp`

## Verify that the FTP server is working

Use a browser and navigate to:

```
ftp://PlcmSpIp:PlcmSpIp@sipx.example.com/
```

This is the default FTP account for Polycom phones. You should see the content of the tftpboot directory.

## Verify DHCP settings

Make sure your DHCP server is configured to distribute the TFTP Server address.

Download dhcping e.g. from [here](#). Install the RPM.

For info on how to use: `man dhcping`

```
e.g. dhcping -v -s <your DHCP server name> -c <your monitoring host IP address> -i
```