

## 3.1.1. Setting up SIP voice services for an institution with OpenSER (fabio-work in progress)

### Goal

This guide helps you setup a simple, mature and flexible open source SIP server. OpenSER can be used on systems with limited resources as well as on carrier grade servers, scaling to up to thousands call setups per second.

### Applicability

OpenSER includes all major entity of a SIP Network. It is a registrar, location server, proxy server, redirect server, gateway to SMS/XMPP, or advanced VoIP application server.

### Prerequisites (OS, dependencies on other software)

- Linux/Unix OS

### Installation (OS agnostic)

Read first carefully the [INSTALL file](#), paying especially attention to section 3C) and 3D).



#### Use packages

There are several packages ready to be installed. Packages are available for Debian, Fedora, FreeBSD, Gentoo, NetBSD, OpenBSD, Solaris.



#### Make packages

If you need a newer version. Make packages by typing 'make deb' or 'make rpm' instead of 'make install'.

### Configuration (OS agnostic)

The configuration is located in `/etc/openser`. The default configuration is ready for a very simple setup and for some very basic tests. You can already try to register to the SIP server.

### Troubleshooting with ngrep

Before you try to register to your SIP server make sure udp and tcp port 5060 are open. Use ngrep for tracing SIP messages on the SIP server. Useful commands are:

```
ngrep port 5060
ngrep -d eth0 port 5060
ngrep -q 'test_user_name'
```



#### Message too big

If you get the following error when registering: **<Message too big>** then probably your host forwards the SIP Message to itself. So trace on localhost to see if this is the case. If your server has more than one DNS entry, make sure that you have all hostnames in `/etc/hosts` and that you have the following line in your `openser.cfg` `alias="yoursipdomain.ch:5060"`

### Troubleshooting with the xlog module

For debugging purposes, it may be very helpful to use the xlog module, which logs directly with syslog.

Add the following line to the `/etc/syslog.conf` file:

```
# OpenSER messages
# make sure this spacing is done with tabs, spaces may cause errors
local7.* /var/log/openser

# remove local7 from syslog and messages
*.*;auth,authpriv.none,local7.none -/var/log/syslog
*.=info;*.=notice;*.=warn;\
mail,news.none,local7.none -/var/log/messages
```

And add the following line to the openser.cfg:

```
log_facility=LOG_LOCAL7
;Add the xlog module:
loadmodule "/usr/lib/openser/modules/xlog.so"
```

It is particularly helpful for logging variables. You can add e.g. following line at the beginning of the default route.

```
xlog("L_INFO", "request method: [$rm], from URI: [$fu], To URI: [$tu]");
```

The full set of variables is available on <http://openser.org/docs/pseudo-variables.html>

## Control server with openserctl

With `sudo openserctl moni` you can monitor the state of your openser (whether it is actually running after the installation).

`openserctl` is located in `/usr/sbin`, which is not by default in the user PATH. To use `openserctl` as a non-privileged user you need to add the path to the environment `export $PATH=PATH:/usr/sbin` or call with the full path `/usr/sbin/openserctl`. Additionally you have to change permissions for the fifosockets. Add the following line in `/etc/openser/openser.cfg`.

```
sock_group="staff" # openser socket/fifo group, e.g. staff
```

If you don't want to type the read/write password every time you make a change via `openserctl`, you can add the following lines to your `/etc/openser/openserctlrc`. This is also a good moment to configure the SIP domain.

```
## password for database read/write user
DBRWPW=openserrw
## your SIP domain
SIP_DOMAIN=yoursipdomain.ch
```



### Plain text passwords

Be aware that this can be a security problem, as your password is unencrypted in a file. However, this might be anyway the case in `/etc/openser/openser.cfg`.