

Build Moonshot from source on Debian

The Moonshot source code is composed of several modules, all of them available from our GIT repositories, and they all can be built by hand relatively easier, assuming you have all of the prerequisite packages installed. This page has instructions for getting those packages and building the software itself for Debian 8. For other Debian and Ubuntu versions, the process is very similar and only a few package names may differ.

Contents

- [1. System Preparation](#)
- [2. Build and install Moonshot from source](#)
 - [2.1. libradsec](#)
 - [2.2. moonshot-ui](#)
 - [2.3. mech_eap](#)
 - [2.4. trust_router](#)
 - [2.5. freeradius](#)
- [3. Building Moonshot binary packages from source](#)

1. 1. System Preparation

To build all of the Moonshot components, you need various packages installed. To install all of these, see below:

Debian 8

```
apt-get install build-essential automake autoconf libtool autopoint pkg-config libgtk-3-dev libgee-dev\
  libnewt-dev valac desktop-file-utils libssl-dev libdbus-glib-1-dev git dh-autoreconf devscripts\
  debhelper autotools-dev libshibsp-dev libradsec-dev libkrb5-dev libshibresolver-dev libjansson-dev libssl-
dev\
  libboost-dev liblog4shib-dev dh-autoreconf devscripts libconfuse-dev libevent-dev libtalloc-dev libsqlite3-
dev\
  quilt dh-systemd libcap-dev libgdbm-dev libiodbc2-dev libjson0 libjson0-dev libldap2-dev libpam0g-dev
libpcap-dev\
  libperl-dev libmysqlclient-dev libpq-dev libreadline-dev libsasl2-dev libwbclient-dev libyubikey-dev
libykclient-dev\
  libmemcached-dev libhiredis-dev python-dev samba-dev wget libxml2-dev libsecret-1-dev libgcr-3-dev
libsystemd-dev \
  libkeyutils-dev sqlite3 dbus-x11
```

2. 2. Build and install Moonshot from source

In order to build Moonshot you should build and install its components in the given order (as they depend on each other).

2.1. 2.1. libradsec

```
git clone https://github.com/janetuk/libradsec.git
cd libradsec
sh autogen.sh
./configure
make
make install
```

2.2. 2.2. moonshot-ui

```
git clone https://github.com/janetuk/moonshot-ui.git
cd moonshot-ui
sh autogen.sh
./configure
make
make install
```

2.3. 2.3. mech_eap

```
git clone https://github.com/janetuk/mech_eap.git
cd mech_eap
sh autogen.sh
./configure
make
make install
# Make install does not install everything in the right place, so a couple of adjustments are required
ln -s /usr/local/lib/gss/ /usr/lib/x86_64-linux-gnu/
cp ./mech_eap/mech /etc/gss/mech.d/moonshot.conf
```

2.4. 2.4. trust_router

```
git clone https://github.com/janetuk/trust_router.git
cd trust_router
autoreconf -fi
./configure
make
make install
```

2.5. 2.5. freeradius

```
git clone https://github.com/janetuk/freeradius.git
cd freeradius
./configure
make
make install
```

3. 3. Building Moonshot binary packages from source

Instead of building and installing from source, you can build Debian binary packages from source. This method is preferred over the previous one unless you want to actively develop some of the Moonshot modules, since it will install all the files in the right location (eg. libraries, systemd units...) as well performing the pre and post install operations.

The installation order is the same as described in the previous section. The process is identical for all the modules. In the instructions, we use the variable \$MODULE_NAME to refer to a module's name.

```
git clone https://github.com/janetuk/$MODULE_NAME.git
cd $MODULE_NAME
debuild -us -uc
# Note: packages are generated in ..
dpkg -i ../*.deb
```