**Install WebRTC2SIP**

**Link Reference:**

<https://code.google.com/p/webrtc2sip/wiki/Building_Source_v2_0>

<https://geekforum.wordpress.com/2013/06/06/build-and-install-doubango-webrtc2sip/>

<https://code.google.com/p/doubango/issues/detail?id=153>

<https://code.google.com/p/doubango/issues/attachmentText?id=153&aid=1530001000&name=libyuv-include.patch&token=ABZ6GAdF6D1NMzijL3Ws-bVijJxZiKthLQ%3A1440134748611>

<https://code.google.com/p/libyuv/wiki/GettingStarted>

<https://gist.github.com/ghafran/20e339c4bf9e2bfc3652>

**Install necessary packages:**

|  |
| --- |
| *apt-get install make libtool autoconf subversion git wget cmake gcc nasm pkg-config libogg-dev libvorbis-dev libtheora-dev cmake python2.7-dev g++* |

**Building libsrtp**

libsrtp is required.

git clone https://github.com/cisco/libsrtp/

cd libsrtp && git checkout v1.5.0

CFLAGS="-fPIC" ./configure --enable-pic && make && make install

check OpenSSL version and make sure version 1.0c is installed: ***openssl version***

wget http://www.openssl.org/source/openssl-1.0.1c.tar.gz

tar -xvzf openssl-1.0.1c.tar.gz && cd openssl-1.0.1c

./config shared --prefix=/usr/local --openssldir=/usr/local/openssl

|  |
| --- |
| <http://superuser.com/questions/735736/no-version-information-available-required-by-usr-bin-ssh>  n the directory where you ended up extracting the source code for your version of openssl 1.0.1h (should work for other versions too.) I create a file called openssl.ld  In this file put this...  OPENSSL\_1.0.0 { global: \*; };  save it. Now type in...  make clean  (Just to be sure we are starting fresh.)  Now for the really mind boggling part... |

./config --prefix=/usr/local --openssldir=/usr/local/openssl shared -Wl,--version-script=openssl.ld -Wl,-Bsymbolic-functions

&& make && make install

install libspeex-dev and libspeexdsp-dev

apt-get install libspeex-dev libspeexdsp-dev

**Building YASM**

**YASM** is only required if you want to enable **VPX** (VP8 video codec) or **x264** (H.264 codec).

wget http://www.tortall.net/projects/yasm/releases/yasm-1.2.0.tar.gz

tar -xvzf yasm-1.2.0.tar.gz && cd yasm-1.2.0

./configure && make && make install

**Building libvpx**

**libvpx** adds support for **VP8** and is optional but highly recommended if you want support for video when using Google Chrome or Mozilla Firefox.

You can install the devel packages: apt-get install libvpx-dev

**Building opencore-amr**

**opencore-amr** is optional. Adds support for **AMR** audio codec.

git clone git://opencore-amr.git.sourceforge.net/gitroot/opencore-amr/opencore-amr

cd opencore-amr

autoreconf --install && ./configure && make && make install

**Building libopus**

**libopus** is optional but highly recommended as it’s an MTI codec for WebRTC. Adds support for [Opus audio codec](http://www.opus-codec.org/).

wget http://downloads.xiph.org/releases/opus/opus-1.0.2.tar.gz

tar -xvzf opus-1.0.2.tar.gz && cd opus-1.0.2

./configure --with-pic --enable-float-approx && make && make install

**Building libgsm**

**libgsm** is optional. Adds support for **GSM** audio codec.

You can install the devel packages (**recommended**):

apt-get install libgsm1-dev

**Building iLBC**

**iLBC** is optional. Adds support for **iLBC** audio codec.

svn co http://doubango.googlecode.com/svn/branches/2.0/doubango/thirdparties/scripts/ilbc

cd ilbc

wget http://www.ietf.org/rfc/rfc3951.txt

awk -f extract.awk rfc3951.txt

./autogen.sh && ./configure

make && make install

**Building x264**

**x264** is optional and adds support for **H.264** video codec (requires FFmpeg).

wget ftp://[ftp.videolan.org/pub/x264/snapshots/last\_x264.tar.bz2](http://ftp.videolan.org/pub/x264/snapshots/last_x264.tar.bz2) && tar -xvjf last\_x264.tar.bz2

# the output directory may be difference depending on the version and date

cd x264-snapshot-20121201-2245

./configure --enable-shared --enable-pic && make && make install

**Building FFmpeg**

**FFmpeg** is optional and adds support for **H.263**, **H.264** (requires **x264**) and **MP4V-ES** video codecs.

git clone git://source.ffmpeg.org/ffmpeg.git ffmpeg && cd ffmpeg

# grap a release branch

git checkout n1.2

# configure source code

./configure --extra-cflags="-fPIC" --extra-ldflags="-lpthread" --enable-pic --enable-memalign-hack --enable-pthreads --enable-shared --disable-static --disable-network --enable-pthreads --disable-ffmpeg --disable-ffplay --disable-ffserver --disable-ffprobe --enable-gpl --disable-debug --enable-libx264 --enable-encoder=libx264 --enable-decoder=h264 --enable-encoder=h263 --enable-encoder=h263p --enable-decoder=h263

# to force enabling h264, append to the configure command: --enable-libx264 --enable-encoder=libx264 --enable-decoder=h264

# to force enabling h263 and h263+, append to the configure command: --enable-encoder=h263 --enable-encoder=h263p --enable-decoder=h263

# build and install

make && make install

**Building OpenH264**

**OpenH264** is optional. Adds support for **H.264** constrained baseline video codec.

git clone <https://github.com/cisco/openh264.git> && cd openh264

git checkout v1.1

make ENABLE64BIT=Yes # Use ENABLE64BIT=No for 32bit platforms

make install

**Building libyuv**

**libyuv** is optional. Adds support for video scaling and chroma conversion.

mkdir libyuv && cd libyuv

svn co http://src.chromium.org/svn/trunk/tools/depot\_tools .

./gclient config http://libyuv.googlecode.com/svn/trunk

./gclient sync && cd trunk

mkdir out  
 cd out  
 cmake -DCMAKE\_INSTALL\_PREFIX="/usr" -DCMAKE\_BUILD\_TYPE="Release" -DCMAKE\_CXX\_FLAGS="-fPIC" ..  
 cmake --build . --config Release  
 sudo cmake --build . --target install --config Release

**Building Doubango**

svn checkout https://doubango.googlecode.com/svn/branches/2.0/doubango && cd doubango

./autogen.sh

Apply this patch:

|  |
| --- |
| --- doubango.orig/tinyDAV/src/video/tdav\_converter\_video.cxx 2012-12-05 22:08:42.957619566 +0100  +++ doubango/tinyDAV/src/video/tdav\_converter\_video.cxx 2012-12-05 15:27:53.730101557 +0100  @@ -36,7 +36,7 @@    #if HAVE\_LIBYUV    -#include <libyuv/libyuv.h>  +#include <libyuv.h>    using namespace libyuv; |

vi configure, change libyuv/libyuv.h to libyuv.h

./configure --with-ssl --with-srtp --with-vpx --with-yuv --with-amr --with-speex --with-speexdsp --with-opus --with-gsm --with-ilbc --with-ffmpeg

make && make install

**Building webrtc2sip and 3rd-party libraries**

**webrtc2sip** depends on Doubango IMS Framework v2.0 and **libxml2**.

The first step is to checkout the source code:

svn co https://webrtc2sip.googlecode.com/svn/trunk/ webrtc2sip

**Installing libxml2**

apt-get install libxml2-dev

**Building webrtc2sip**

export PREFIX=/opt/webrtc2sip

cd webrtc2sip && ./autogen.sh

./configure --prefix=$PREFIX CFLAGS='-lpthread' LDFLAGS='-ldl' LIBS='-ldl'

make clean && make && make install

cp -f ./config.xml $PREFIX/sbin/config.xml

The gateway is configured using **config.xml**. Please check the [technical guide](http://webrtc2sip.org/technical-guide-1.0.pdf) for more information about this file.

**Running webrtc2sip: /opt/webrtc2sip/sbin**

**Genarate Cert**

sudo openssl req -new > new.ssl.csr

sudo openssl rsa -in privkey.pem -out new.cert.key

sudo openssl x509 -in new.ssl.csr -out new.cert.cert -req -signkey new.cert.key -days 999999

sudo mkdir -p /etc/ssl/certs/

sudo cp new.cert.cert /etc/ssl/certs/server.crt

sudo chmod 640 /etc/ssl/certs/server.crt

sudo mkdir -p /etc/ssl/private/

sudo cp new.cert.key /etc/ssl/private/server.key

sudo chmod 640 /etc/ssl/private/server.key

vi /opt/webrtc2sip/sbin/config.xml

|  |
| --- |
| <?xml version="1.0" encoding="utf-8" ?>  <!-- Please check the technical guide (http://webrtc2sip.org/technical-guide-1.0.pdf) for more information on how to adjust this file -->  <config>  <debug-level>INFO</debug-level>  <transport>udp;\*;15061</transport>  <transport>ws;\*;15088</transport>  <transport>wss;\*;15089</transport>  <!--transport>tcp;\*;10063</transport-->  <!--transport>tls;\*;10064</transport-->  <enable-rtp-symetric>yes</enable-rtp-symetric>  <enable-100rel>no</enable-100rel>  <enable-media-coder>no</enable-media-coder>  <enable-videojb>yes</enable-videojb>  <video-size-pref>vga</video-size-pref>  <rtp-buffsize>65535</rtp-buffsize>  <avpf-tail-length>100;400</avpf-tail-length>  <srtp-mode>optional</srtp-mode>  <srtp-type>sdes;dtls</srtp-type>  <dtmf-type>rfc4733</dtmf-type>  <codecs>opus;pcma;pcmu;gsm;vp8;h264-bp;h264-mp;h263;h263+</codecs>  <codec-opus-maxrates>48000;48000</codec-opus-maxrates>  <stun-server>stun.l.google.com;19302;;</stun-server>  <enable-icestun>yes</enable-icestun>  <max-fds>-1</max-fds>  <!--nameserver>66.66.66.6</nameserver-->  <ssl-certificates>  /etc/ssl/private/server.key;  /etc/ssl/certs/server.crt;  ;  no  </ssl-certificates>  <!-- \*\*\*CLICK-TO-CALL SERVICE\*\*\* -->  <transport>c2c;\*;10070</transport>  <transport>c2cs;\*;10072</transport>  <database>sqlite;\*</database>  <!--account-mail>smtps;\*;\*;auth.smtp.1and1.fr;465;noreply@example.com;noreply@example.com;mysecret</account-mail-->  <!--account-sip-caller>\*;sip:a@example.com;a;example.com;mysecret</account-sip-caller-->  <!--account-http-domain>click2dial.org</account-http-domain-->  <!--account-recaptcha>https://www.google.com/recaptcha/api/siteverify;your-secret-here</account-recaptcha-->  </config> |

**Installation SIPml5**

\*Install the sipml5 softphone in the webrtc2sip VMs!"

sudo apt-get install nginx git subversion

\*Create the nginx site file */etc/nginx/sites-available/sipml5* and add the nginx config:

server {

listen 443;

listen [::]:443 ipv6only=on;

server\_name webrtc2sip-dev2.vnctalk.zimbra-vnc.de sipml5-dev2.vnctalk.zimbra-vnc.de webrtc2sip-dev2 sipml5-dev2;

root /opt/VNCtalk/www/sipml5;

index index.html index.htm;

ssl on;

ssl\_certificate /etc/ssl/certs/server.crt;

ssl\_certificate\_key /etc/ssl/private/server.key;

location / {

try\_files $uri $uri/ =404;

}

error\_page 404 /404.html;

error\_page 500 502 503 504 /50x.html;

location = /50x.html {

root /usr/share/nginx/html;

}

}

cd /etc/nginx/sites-enabled/

ln -s ../sites-available/sipml5 sipml5

\*Create the folder and get the sources for the sipml5 softphone client.

sudo mkdir -p /opt/VNCtalk/www  
sudo chown $USER:$USER /opt/VNCtalk  
sudo chown $USER:$USER /opt/VNCtalk/www  
cd /opt/VNCtalk/www  
git clone <https://github.com/sipml5/sipml5.git>