FreeSWITCH OSP Module
User Guide

May 4, 2012

Revision History

Revision	Date of Issue	Description
0.1	5 May 2010	Draft
0.2	1 July 2010	Add work mode option and support for outbound proxy
0.3	8 July 2010	Change OSP TCP port from 1080 to 5045
0.4	4 May 2012	Update

1 Introduction

This document provides instructions on how to build and configure FreeSWITCH with OSP Toolkit to enable secure, multi-lateral peering. OSP Toolkit is an open source implementation of the OSP peering protocol and is freely available from https://sourceforge.net/projects/osp-toolkit. The OSP standard defined by the European Telecommunications Standards Institute (ETSI TS 101 321) http://www.etsi.org. Additional information on the OSP protocol is available at http://en.wikipedia.org/wiki/Open_Settlement_Protocol. If you have questions or need help to build FreeSWITCH with OSP Toolkit, please post your question on the OSP mailing list at https://lists.sourceforge.net/lists/listinfo/osp-toolkit-client.

2 OSP Toolkit

Please reference the OSP Toolkit document "How to Build and Test the OSP Toolkit" available from https://sourceforge.net/projects/osp-toolkit.

The software listed below is required to build and use OSP Toolkit:

- **OpenSSL** (required for building) - Open Source SSL protocol and Cryptographic Algorithms (version 0.9.8i recommended) from http://www.openssl.org. Pre-compiled OpenSSL binary packages are not recommended because of the binary compatibility issue.
- **Perl** (required for building) - A programming language used by OpenSSL for compilation. Any version of Perl should work. One version of Perl is available from http://www.activestate.com/activeperl. If pre-compiled OpenSSL packages are used, Perl package is not required.
- **C compiler** (required for building) - Any C compiler should work. The GNU Compiler Collection from http://www.gnu.org is routinely used for building OSP Toolkit for testing.
- **OSP Server** (required for testing) - Access to any OSP server should work. An open source reference OSP server developed by Cisco System is available at https://sourceforge.net/projects/openosp. RAMS, a java based open source OSP server is available at https://sourceforge.net/projects/rams. A free version of the TransNexus commercial OSP server may be downloaded from http://www.transnexus.com/OSP%20Toolkit/Peering_Server/VoIP_Peering_Server.htm.

2.1 Unpack OSP Toolkit

After downloading OSP Toolkit (version 4.0.3 or later release) from https://sourceforge.net/projects/osp-toolkit, perform the following steps in order:

1) Copy the OSP Toolkit distribution into the directory where it will reside, say ***/usr/src***.
2) Un-package the distribution file by executing the following command:

```
gunzip –c OSPToolkit-###.tar.gz | tar xvf –
```

Where ### is the version number separated by dots. For example, if the version is 4.0.3, then the above command would be:

```
gunzip –c OSPToolkit-4.0.3.tar.gz | tar xvf –
```

A new directory (***TK-4_0_3-20120120***) will be created within the same directory as the tarball file.

3) Go to the ***TK-4_0_3-20120120*** directory by running this command:

```
cd TK-4_0_3-20120120
```

Within this directory, you will find directories and files similar to what is listed below if the command "ls -F" is executed):

```
ls –F
bin/  crypto/  enroll/  include/  lib/  LICENSE.txt  README.txt  RELNOTES.txt  src/
test/
```

2.2 Prepare to Build

4) Compile OpenSSL according to the instructions provided with the OpenSSL distribution (You would need to do this only if you don't have openssl already).
5) Copy the OpenSSL header files (the *.h files) into the OSP Toolkit ***crypto/openssl*** directory. The OpenSSL header files are located under the ***openssl/include/openssl*** directory.
6) Copy the OpenSSL library files (libcrypto.a and libssl.a) into the OSP Toolkit ***lib*** directory. The OpenSSL library files are located under the openssl directory.
   Note: Since the FreeSWITCH requires the OpenSSL package. If the OpenSSL package has been installed, steps 4~6 are not necessary.
7) Optionally, change the install directory of OSP Toolkit. Open Makefile in the OSP Toolkit ***src*** directory, look for the install path variable – **INSTALL_PATH**, and edit it to be anywhere you want (defaults ***/usr/local***).
   Note: Please change the install path variable only if you are familiar with both OSP Toolkit and FreeSWITCH. Otherwise, it may case FreeSWITCH does not support the OSP protocol.

2.3 Build OSP Toolkit

8) From within the OSP Toolkit directory (***/usr/src/TK-4_0_3-20120120***), start the compilation script by executing the following commands:

```
cd src
make clean; make build
```

2.4 Install OSP Toolkit

The header files and the library of OSP Toolkit should be installed.

9) Use the make script to install OSP Toolkit.

```
make install
```

The make script is also used to install the OSP Toolkit header files and the library into the **INSTALL_PATH** directory specified in Makefile.

Note:

- Please make sure you have the rights to access the **INSTALL_PATH** directory. For example, in order to access ***/usr/local*** directory, root privileges are required.
- By default, OSP Toolkit is compiled in the production mode. The following table identifies which default features are activated with each compile option:

Default Feature	Production	Development
Debug Information Displayed	No	Yes

The "Development" option is recommended for a first time build. The **CFLAGS** definition in Makefile must be modified to build in development mode.

3 FreeSWITCH

In FreeSWITCH, OSP support is implemented as both application and dialplan. The detailed descriptions of the OSP module implementation are listed in section 4. This section describes the steps to add OSP support to FreeSWITCH:

- Build FreeSWITCH with OSP module
- Configure FreeSWITCH with OSP module
- Run Freeswitch

3.1 Build FreeSWITCH with OSP Module

We assume the audiences are familiar with the FreeSWITCH installation. We only briefly discuss the installation steps.

1) Obtain FreeSWITCH source code. Since the OSP module is in beta test, you can only download FreeSWITCH with the OSP module from GIT or SVN repository.

```
git clone git://git.freeswitch.org/freeswitfh.git freeswitch-git
```

Or

```
svn checkout http://svn.freeswitch.org/svn/freeswitch/trunk freeswitch-svn
```

All FreeSWITCH native building steps should be run under FreeSWITCH source directory.

2) Build all necessary packages. The instructions to build OSP Toolkit are listed in section 2.
3) Change build/modules.conf.in to allow the OSP module.

```
…
#applications/mod_nibblebill
applications/mod_osp
#applications/mod_redis
…
```

4) Compile and install FreeSWITCH

```
./bootstrap.sh
./configure
make all
make install
```

Note: root privileges are required to install FreeSWITCH to default folders.

5) Install FreeSWITCH default configuration files

```
make samples
```

Now, FreeSWITCH has been installed under ***/usr/local/freeswitch*** by default. Some configuration files should be changed for the running environment.

3.2 Configure FreeSWITCH with OSP Module

The default FreeSWITCH configuration directory is ***/usr/local/freeswitch/conf***.

3.2.1 modules.conf

/usr/local/freeswitch/conf/autoload_configs/modules.conf.xml contains the modules will be loaded.

```
<configuration name="modules.conf" description="Modules">
  <modules>

    <!-- Loggers (I'd load these first) -->
    <load module="mod_console"/>
    <load module="mod_logfile"/>

    <!-- Multi-Faceted -->
    <!-- mod_enum is a dialplan interface, an application interface and an api
command interface -->
    <load module="mod_enum"/>

    <!-- XML Interfaces -->
    <!-- <load module="mod_xml_cdr"/> -->

    <!-- Event Handlers -->
    <!-- <load module="mod_cdr_csv"/> -->
    <load module="mod_event_socket"/>
    <!-- <load module="mod_radius_cdr"/> -->

    <!-- Directory Interfaces -->

    <!-- Endpoints -->
    <load module="mod_sofia"/>

    <!-- Applications -->
    <load module="mod_commands"/>
    <load module="mod_dptools"/>
    <!-- <load module="mod_voicemail"/> -->

    <!-- Dialplan Interfaces -->
    <load module="mod_dialplan_xml"/>

    <!-- Codec Interfaces -->
    <load module="mod_spandsp"/>
    <load module="mod_g723_1"/>
    <load module="mod_g729"/>
    <load module="mod_amr"/>
    <load module="mod_ilbc"/>
    <load module="mod_speex"/>
    <load module="mod_h26x"/>
    <load module="mod_siren"/>
```

```
    <!-- File Format Interfaces -->
    <load module="mod_sndfile"/>
    <load module="mod_native_file"/>

    <!-- Timers -->

    <!-- Languages -->

    <!-- ASR /TTS -->

    <!-- Say -->
    <load module="mod_say_en"/>

    <!-- Third party modules -->
    <load module="mod_osp"/>

  </modules>
</configuration>
```

Note: Other modules may also be loaded if they are needed.

3.2.2 osp.conf.xml

/usr/local/freeswitch/conf/autoload_configs/osp.conf.xml contains OSP module configuration parameters. Two parameters, "service-point-url" and "device-ip" must be configured. The default values for all other parameters will work well for standard OSP implementations. The details of the OSP module configuration parameters are listed in section 5.1.

```
<configuration name="osp.conf" description="OSP Module Configuration">
    <settings>
           <!-- Debug info flag -->
           <param name="debug-info" value="disabled"/>
           <!-- Log level for debug info -->
           <param name="log-level" value="info"/>
           <!-- Crypto hareware accelerate is disabled by default -->
           <param name="crypto-hardware" value="disabled"/>
           <!-- SIP settings -->
           <param name="sip" module="sofia" profile="external"/>
           <!-- H.323 settings -->
           <!-- <param name="h323" module="h323" profile="external"/> -->
           <!-- IAX settings -->
           <!-- <param name="iax" module="iax" profile="external"/> -->
           <!-- Skype settings -->
           <!-- <param name="skype" module="skypopen" profile="external"/> -->
           <!-- Default destination protocol -->
           <param name="default-protocol" value="sip"/>
    </settings>

    <profiles>
    <!-- Default OSP profile -->
           <profile name="default">
                  <!-- Service point URLs, up to 8 allowed -->
                  <!-- <param name="service-point-url"
value="http://osptestserver.transnexus.com:5045/osp"/> -->
```

```
                  <!-- <param name="service-point-url"
value="https://127.0.0.1:1443/osp"/> -->
                  <param name="service-point-url" value="http://127.0.0.1:5045/osp"/>

                  <!-- FreeSWITCH IP address for OSP -->
                  <param name="device-ip" value="127.0.0.1:5080"/>

                  <!-- SSL lifetime in seconds -->
                  <param name="ssl-lifetime" value="300"/>
                  <!-- HTTP max connections, 1~1000 -->
                  <param name="http-max-connections" value="20"/>
                  <!-- HTTP persistence in seconds -->
                  <param name="http-persistence" value="60"/>
                  <!-- HTTP retry delay in seconds, 0~10 -->
                  <param name="http-retry-delay" value="0"/>
                  <!-- HTTP retry limit, 0~100 -->
                  <param name="http-retry-limit" value="2"/>
                  <!-- HTTP timeout in milliseconds, 200~60000 -->
                  <param name="http-timeout" value="10000"/>

                  <!-- OSP work mode, direct or indirect -->
                  <param name="work-mode" value="direct"/>
                  <!-- OSP service type, voice or npquery -->
                  <param name="service-type" value="voice"/>
                  <!-- Max number of destinations -->
                  <param name="max-destinations" value="12"/>
           </profile>
    </profiles>
</configuration>
```

3.2.3 01_txnx.conf.xml

OSP functions are implemented as both application and dialplan. 01_txnx.conf.xml is an example about how to configure the dial plan to use OSP application and dialplan. The details of the OSP application and dialplan configuration parameters are listed in section 4.2 and section 4.3.

```
<include>
  <extension name="app.osp.txnx.com">
    <condition field="destination_number" expression="^(1404002\d{4})$">
      <action application="osp" data="default"/>
      <action application="bridge" data="${osp_auto_route}"/>
    </condition>
  </extension>

  <extension name="dialplan.osp.txnx.com">
    <condition field="destination_number" expression="^(1404003\d{4})$">
      <action application="set" data="continue_on_fail=true"/>
      <action application="transfer" data="$1 osp:default"/>
    </condition>
  </extension>
</include>
```

Note: This configuration example uses OSP application for called number 1404002xxxx and uses OSP dialplan for called number 1404003xxxx.

3.2.4 02_txnx.conf.xml for voicemail

02_txnx.conf.xml is an example about how to configure the dial plan to use voicemail.

```
<include>
  <extension name="personal.voicemail.txnx.com">
    <condition field="destination_number" expression="^1678000(\d{4})$">
      <action application="answer"/>
      <action application="voicemail" data="$${vm_profile} $${domain} $1"/>
    </condition>
  </extension>

  <extension name="mail.voicemail.txnx.com">
    <condition field="destination_number" expression="^1770000(\d{4})$">
      <action application="answer"/>
      <action application="set" data="voicemail_authorized=true"/>
      <action application="voicemail" data="check auth $${vm_profile}
$${domain} $1"/>
    </condition>
  </extension>
</include>
```

Note: This configuration is configured for called number 1678000xxxx (xxxx is user/voicemail ID). 1770000xxxx is the maintainer's called numbers. There are several pre-defined user IDs, such as 1001, 1002 to 1014.

3.3 Run FreeSWITCH

FreeSWITCH binary is installed under /usr/local/freeswitch/bin by default.

- Command options.

```
-u [user]        -- specify user to switch to
-g [group] -- specify group to switch to
-waste           -- allow memory waste
-core            -- dump cores
-nosql           -- disable internal sql scoreboard
-nonat           -- disable auto nat detection
-stop            -- stop freeswitch
-nc              -- do not output to a console and background
-c               -- output to a console and stay in the foreground
```

- To start FreeSWITCH with console

```
#/usr/local/freeswitch/bin/freeswitch –core –waste –c
```

- To stop FreeSWITCH with console

```
>fsctl shutdown
```

- To start FreeSWITCH without console

```
#/usr/local/freeswitch/bin/freeswitch –core –waste –nc
```

- To stop FreeSWITCH without console

```
#/usr/local/freeswitch/bin/freeswitch –stop
```

4 FreeSWITCH OSP Implementation

This section provides details of OSP implementation.

4.1 OSP CLI

OSP command line interface is used to show the OSP module status.

```
> osp status

=============== OSP Module Settings & Status ===============
                debug-info: enabled
                 log-level: info
           crypto-hardware: disabled
                       sip: sofia/external
                      h323: unsupported
                       iax: unsupported
                     skype: unsupported
          default-protocol: sip
============== OSP Profile Settings & Status ==============
Profile: default
         service-point-url: http://172.16.4.32:5045/osp
                 device-ip: 172.16.4.63:5080
              ssl-lifetime: 300
      http-max-connections: 20
          http-persistence: 60
          http-retry-dalay: 0
          http-retry-limit: 2
              http-timeout: 10000
                 work-mode: direct
              service-type: voice
          max-destinations: 12
                    status: enabled
```

4.2 OSP Application

The OSP application can be called from the dial plan by executing it within condition statements as follows

```
<condition field="destination_number" expression="^(1404002\d{4})$">
  <action application="osp" data="default"/>
  <action application="bridge" data="${osp_auto_route}"/>
</condition>
```

OSP application accept the following input channel variables:

- osp_source_device: Actual source device IP address variable. It is only for FreeSWITH running in indirect mode.
- osp_source_nid: Source device network ID variable.
- osp_custom_info_N: Up to 8 custom info variables. N is the index starting from 1.
- osp_networkid_userparam: The URI user parameter name that is used to present destination network ID.
- osp_networkid_uriparam: The URI parameter name that is used to present destination network ID.
- osp_user_phone: Flag to add "user=phone" URI parameter. The default is "disabled".
- osp_outbound_proxy: Outbound proxy IP address variable.
- <profile>: OSP service provider name configured in osp.conf.xml. If it is empty, profile "default" is used.

OSP application set the following output channel variables:

- osp_profile: Used OSP profile name. For OSP cookie used by outbound channel.
- osp_transaction_id: OSP transaction ID. For OSP cookie used by outbound channel.
- osp_calling: Original inbound calling number. For OSP cookie used by outbound channel.
- osp_called: Original inbound called number. For OSP cookie used by outbound channel.
- osp_start_time: Inbound call start time. For OSP cookie used by outbound channel.
- osp_source_device: Actual source device. For OSP cookie used by outbound channel. It is only for FreeSWITH running in indirect mode.
- osp_source_nid: Source network ID. For OSP cookie used by outbound channel.
- osp_destination_total: Total number of destinations from OSPrey. For OSP cookie used by outbound channel.
- osp_destination_count: Destination index. For OSP cookie used by outbound channel.
- osp_destination_ip: Destination IP. For OSP cookie used by outbound channel.
- osp_destination_nid: Destination network ID. For OSP cookie used by outbound channel.
- osp_authreq_status: Authorization request result status.
- osp_route_count: Number of supported destinations.
- osp_route_N: Destination route string. N is the index starting from 1.
- osp_auto_route: Bridge route string.

4.3 OSP Dialplan

The OSP dialplan requires only the used OSP profile name and an OSP lookup will be done. If results are found, the ones supported will be called in order. The OSP dialplan can be set from one of the Endpoints configuration XML as follows

```
<param name="dialplan" value="set:continue_on_fail=true"/>
<param name="dialplan" value="osp:default"/>
```

It also can be configured in dial plan as follow

```
<condition field="destination_number" expression="^(1404003\d{4})$">
  <action application="set" data="continue_on_fail=true"/>
  <action application="transfer" data="$1 osp:default"/>
```

```
</condition>
```

OSP application accept the following input channel variables:

- osp_source_device: Actual source device IP address variable. It is only for FreeSWITH running in indirect mode.
- osp_source_nid: Source device network ID variable.
- osp_custom_info_N: Up to 8 custom info variables. N is the index starting from 1.
- osp_networkid_userparam: The URI user parameter name that is used to present destination network ID.
- osp_networkid_uriparam: The URI parameter name that is used to present destination network ID.
- osp_user_phone: Flag to add "user=phone" URI parameter. The default is "disabled".
- osp_outbound_proxy: Outbound proxy IP address variable.
- <profile>: OSP service provider name configured in osp.conf.xml. If it is empty, profile "default" is used.

OSP application set the following output channel variables:

- osp_profile: Used OSP profile name. For OSP cookie used by outbound channel.
- osp_transaction_id: OSP transaction ID. For OSP cookie used by outbound channel.
- osp_calling: Original inbound calling number. For OSP cookie used by outbound channel.
- osp_called: Original inbound called number. For OSP cookie used by outbound channel.
- osp_start_time: Inbound call start time. For OSP cookie used by outbound channel.
- osp_source_device: Actual source device. For OSP cookie used by outbound channel. It is only for FreeSWITH running in indirect mode.
- osp_source_nid: Source network ID. For OSP cookie used by outbound channel.
- osp_destination_total: Total number of destinations from OSPrey. For OSP cookie used by outbound channel.
- osp_destination_count: Destination index. For OSP cookie used by outbound channel.
- osp_destination_ip: Destination IP. For OSP cookie used by outbound channel.
- osp_destination_nid: Destination network ID. For OSP cookie used by outbound channel.
- osp_authreq_status: Authorization request result status.
- osp_route_count: Number of supported destinations.
- osp_route_N: Destination route string. N is the index starting from 1.
- osp_auto_route: Bridge route string.

5 Appendix

5.1 OSP Module Configuration Parameters

5.1.1 Global Configuration Parameters

- debug-info: Flag to show OSP module debug information. The default is "disabled".
- log-level: At which log level to show OSP module debug information. The default is "info".
- crypto-hardware: If to use hardware for OpenSSL. The default is "disabled".
- sip: Used SIP module and profile. The default is "sofia" and "external".

- h323: Used H.323 module and profile. The default is "h323" and "external". This option has not been implemented.
- iax: Used IAX2 module and profile. The default is "iax" and "external". This option has not been implemented.
- skype: Used Skype module and profile. The default is "skypopen" and "external". This option has not been implemented.
- default-protocol: The VoIP protocol for destinations with unknown/undefined protocol. The default is "sip".

5.1.2 Profile Configuration Parameters

- profile: OSP profile name.
- service-point-url: OSP service point URL. This parameter must be defined. Up to 8 URLs are allowed.
- device-ip: FreeSWITCH IP for OSP. This parameter must be defined.
- ssl-lifetime: SSL lifetime. The default is 300 in seconds.
- http-max-connections: HTTP max connections. The default is 20.
- http-persistence: HTTP persistence. The default is 60 in seconds.
- http-retry-delay: HTTP retry delay. The default is 0 in seconds.
- http-retry-limit: HTTP retry times. The default is 2.
- http-timeout: HTTP timeout. The default is 10000 in ms.
- work-mode: OSP work mode. The default is "direct".
- service-type: OSP service type. The default is "voice".
- max-destinations: Max destinations OSP server will return. It is up to 12. The default is 12.