WebRTC

About

WebRTC provides Real-Time Communications directly from better web browsers and devices without requiring plug-ins such as Adobe Flash nor Silverlight. WebRTC **always** operates in secure mode.FreeSWITCH provides a WebRTC portal to its public conference bridge to demonstrate the possibilities for handling telephony via a web page; join us for our weekly conference calls.

The process for configuring FreeSWITCH with WSS certificates is the same whether for use with classic WebRTC or the FreeSWITCH Verto endpoint.

Installation

The configuration for Secure Web Sockets is slightly different than for TLS over SIP. This guide covers WSS certificate setup.

Debian 7 (Wheezy)

Install Debian 7 (Wheezy) minimal.

Building FreeSWITCH

Building

```
apt-get install git build-essential automake autoconf libtool wget python zlib1g-dev libjpeg-dev libncurses5-dev libssl-dev libpcre3-dev libcurl4-openssl-dev libldns-dev libedit-dev libspeexdsp-dev libspeexdsp-dev libsqlite3-dev apache2

cd /usr/src/
git clone https://freeswitch.org/stash/scm/fs/freeswitch.git

cd freeswitch
./bootstrap.sh -j
./configure -C
make
make install cd-sounds-install cd-moh-install
mkdir -p /usr/local/freeswitch/certs

edit /usr/local/freeswitch/conf/sip_profiles/internal.xml
# Set these params and save the file:
<param name="tls-cert-dir" value="/usr/local/freeswitch/certs"/>
<param name="wss-binding" value=":7443"/>
```

If behind N.A.T. make sure to set the ext-sip-ip and ext-rtp-ip in vars.xml to the public IP address of your FreeSWITCH.

If talking to clients both inside and outside the N.A.T. you must set the local-network-acl rfc1918.auto, and prefix the ext-sip-ip and ext-rtp-ip to autonat:X.X.X.X

Install Certificates

Layout of /usr/local/freeswitch/certs/wss.pem:

/usr/local/freeswitch/certs/wss.pem

```
Cert, Key and Chain(s) are all contained in a single file in this order:

-----BEGIN CERTIFICATE-----
<cert>
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
<key>
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
<chain>
-----END CERTIFICATE-----
```

Start FreeSWITCH

`/usr/local/freeswitch/bin/freeswitch -ncwait`

Once started execute:

```
fs_cli -x 'sofia status profile internal' | grep WSS-BIND-URL
```

If you get output then your wss on FreeSWITCH is setup correctly.

Setting up Apache:

`a2enmod ssl`

`a2ensite`

(Configure your certificates/chain/keys per standard https docs; this is beyond the scope of this document so not covered here.)

Now verify your certs using http://www.sslshopper.com/ssl-checker.html

Verify the hostname and hostname:7443 so you can verify that your certificate chain is intact.

nginx

Notes captured from the freeswitch-users mailing list:

From: Dan Edwards
Sent: Monday, 01 February, 2016 09:35
Subject: Re: [Freeswitch-users] WebSocket behind NGINX

I'm also running behind Nginx and what I found worked was to proxy to the actual IP address (192.168.1.1 vs. 127.0.0.1), then explicitly removing 192.168.1.1 from the localnet ACL in acl.conf. I had to remove 192.168.1.1 from localnet so FS will offer external IP addresses for RTP.

-----Original Message-----

From: Anton
Sent: Saturday, January 30, 2016 2:20 PM
Subject: [Freeswitch-users] WebSocket behind NGINX

Hello All,

I have to proxy all websocket requests though a nginx server. Right now I am using next configuration:

```
map $http_upgrade $connection_upgrade {
    default upgrade;
    ''      close;
}

server {
    listen 443;
    server_name  wss.somedomain.com.ua;

    ssl on;
    ssl_certificate      /etc/nginx/cert.pem;
    ssl_certificate_key  /etc/nginx/private.key;

    location / {
        proxy_pass http://127.0.0.1:5066;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection $connection_upgrade;
        proxy_read_timeout 86400s;
   }

    access_log /var/log/nginx/wss_access;
    error_log /var/log/nginx/wss_error debug; }
```

I dumped traffic from nginx and found out that "switching protocol"
phrase was successful but INVITE message from my browser in pending state.
Maybe FreeSWITCH wants real IP not loopback? Who have faced with similar problem?

BR,
Anton

A quick how to from bkw (Brian K. West):

Certificates

```
# Create certificates:

wget http://files.freeswitch.org/downloads/ssl.ca-0.1.tar.gz
tar zxfv ssl.ca-0.1.tar.gz
cd ssl.ca-0.1/
perl -i -pe 's/md5/sha256/g' *.sh
perl -i -pe 's/1024/4096/g' *.sh
./new-root-ca.sh
./new-server-cert.sh self.bkw.org
./sign-server-cert.sh self.bkw.org
cat self.bkw.org.crt self.bkw.org.key > /usr/local/freeswitch/certs/wss.pem

# Setup Apache:

# default-ssl:

SSLCertificateFile    /usr/local/freeswitch/certs/wss.pem
SSLCertificateKeyFile /usr/local/freeswitch/certs/wss.pem
SSLCertificateChainFile /usr/local/freeswitch/certs/wss.pem

# Setup Sofia TLS:

cat self.bkw.org.crt self.bkw.org.key > /usr/local/freeswitch/certs/agent.pem
cat ca.crt > /usr/local/freeswitch/certs/cafile.pem

# vars.xml:

<X-PRE-PROCESS cmd="set" data="internal_ssl_enable=true"/>
<X-PRE-PROCESS cmd="set" data="external_ssl_enable=true"/>

# Restart FreeSWITCH.

## Now make sure your system has ca.crt imported so it will trust your new found hotness.

# TEST:

openssl s_client -connect self.bkw.org:443
openssl s_client -connect self.bkw.org:5061
openssl s_client -connect self.bkw.org:5081
openssl s_client -connect self.bkw.org:8082


# Depending on what you've setup you'll see:

subject=/C=US/ST=Oklahoma/L=McAlester/O=Tonka Truck/OU=Secure Web Server/CN=self.bkw.org/emailAddress=brian@bkw.
org
issuer=/C=US/ST=Oklahoma/L=McAlester/O=Whizzzzzzy Bang Bang/OU=Certification Services Division/CN=WBB Root CA
/emailAddress=brian@bkw.org

# Or there abouts.
```

Caveats

The latest version of Freeswitch should automatically generate self-signed certificates. However, self-signed certs often don't work very well, since you will need to induce the prompt to allow an untrusted certificate. You should use a trusted certificate, just as you would your website. If you take your WebRTC url, such as wss://foo.bar.com:7443 and change it to https://foo.bar.com:7443 you can visit it in the browser and give it a permanent exception. On macOS and Windows you may import the ca.crt into your trust store.

Errors

If you see the following:

Profile Error

```
[ERR] sofia.c:2855 Error Creating SIP UA for profile: external (sip:host@127.0.0.1:5090;maddr=10.10.1.150;
transport=udp,tcp) ATTEMPT 1 (RETRY IN 5 SEC)
```

The likely causes for this are:

1) Another application is already listening on the specified address.

2) The IP address to which the profile is attempting to bind is not local to this system.

There could be a certificate problem and the wss directive will cause the whole profile to fail.

Next

Now Proceed to configure sipjs/sipml5.

Clients

- SIP.js
- sipml5 – World's first HTML5 SIP client from Doubango
- JsSIP – Written by the authors of **RFC 7118** and OverSIP

Tips

If you want you can use Opus codec for high audio quality. If you do, be careful with testing with software SIP clients, because SIP clients which implement it according to the RFC's are currently rare (possibly non-existent). You'd better call between two WebRTC peers. If you for example want to use Jitsi, my current experience is that you can call with Jitsi with the Opus codec to Freeswitch (probably because Freeswitch accepts the not 100% correct SDP sent by Jitsi), but when Freeswitch originates a call it won't work.

Use this to see if ws and wss work:

```
sofia status profile internal
```

References

WebRTC Glossary

Test firewall for proper WebRTC ports