Multi-tenant

About

Note: There is also some useful information on the Multiple Companies page, which needs merging.

FreeSWITCH can be configured to host multiple separate companies and clients in a multi-tenant scenario, such as a multi-tenant building served by a single FreeSWITCH instance, or commercial providers serving numerous companies and clients, all of which require isolation from one another regarding extensions, dialplans, gateways, etc.

Such configuration is simple: SIP Domains are the key. Separating each tenant into its own domain using SIP domains offers the advantage of being able to reuse the same dialplan for every tenant. Exceptions to this are when some tenants have an IVR as their front end, and some tenants map DIDs directly to extensions.

Disclaimer: FreeSWITCH is a VERY flexible platform, and with that flexibility, you get many ways to achieve the same task. At some point best practices will emerge, but this is the brute force method I used.

SIP Domain background

To understand the concept of SIP routing, the analogy is email addresses. Consider the email address joe.smith@company-a.com. This address is composed of two parts: the user id, on the left of the @, and the domain name, on the right of the @. The Internet DNS system has a simple but effective way of determining how to get email addressed to user Joe Smith, at company-a.com. It involves asking the DNS hierarchy for a list of host names that will accept mail for company-a.com. DNS responds with a list of host names that will accept the mail. The machines do not have to live in the company-a.com namespace. Any machine that has a valid MX (mail exchanger) record for company-a.com is acceptable to send mail to. The sending host then starts at the top of the list, and starts trying to connect to the hosts, one by one, until one responds. There are more intricate details, but this level of explanation will do. SIP domains do not have DNS MX records, as those are reserved for internet mail. DNS SRV records serve the same purpose. SRV records can be used for routing multi-protocol data around the internet. For more information on DNS SRV and SIP, refer to RFC 3263.

The best way to document the configuration of domains is to dive right into the XML config files. The configuration shown below uses the stock XML default configuration, with domain support added. Several sections of the default dialplan can be ripped out, because they are not needed. Documented is a basic, no frills multi-tenant system with each tenant(domain) determined from where the call originated. Inbound calls are routed directly to extensions with a 1:1 mapping.

Files to edit

We will break down the configuration into four sections: Directory, Inbound and Outbound call routing, SIP profiles, and gateways.

Directory

First, lets start by building the directory. under conf/directory, you’ll notice a default.xml and default/ folder. Copy both of these to company-a.com.xml, and company-a.com, respectively. The names of the files and folder are arbitrary, but it makes sense to name them to match the domain you are creating. The configuration I desired was for each domain to receive and send calls to its own external gateways via upstream ITSPs, with a unique account per domain, but you can just as easily use one upstream ITSP account to terminate and originate all calls.
In conf/directory/company-a.com/., you will have the normal directory entries for each UA/user, just as in the default configuration. There is nothing special about those files.

Inbound Call Routing

The method I used to route inbound calls is very simple: Match the inbound DID, set the domain, then transfer the call into the default dialplan. The advantage of this is a single dialplan handles ALL inbound calls. If you restrict your UAs to specific extension ranges, like the default 1000-1nnn, the dialplan can be very simple. To take advantage of larger scale solutions, such as curl_xml, etc for call routing, that’s best left to your own experimentation. The example here is the most basic that makes use of as much of the sample default dialplan as possible, so as to make it easy for you to get multi-tenant working, then deal with scalability and redundancy issues later.

This example uses the default dialplan context for all domains. It assumes each tenant uses the same dialplan. Separating each tenant into its own dialplan context is a best practice.
Also note, IVRs and DIDs can all be handled in the dialplan with multiple tenants. If domain wishes to use an IVR, just build an extension like:

```
<extension name="company-b.com IVR">
    <condition field="destination_number" expression="1235551213">
        <action application="set" data="domain_name=company-b.com"/>
        <action application="set" data="domain=company-b.com"/>
        <action application="transfer" data="company_b_ivr XML default"/>
    </condition>
</extension>
```

and then in the default dialplan, create an extension called `company_b_ivr` that launches the IVR. Again, this is just one method, your mileage may vary.

### Outbound Call Routing

Outbound call routing is also easy; it's just like the default configuration. A good place to put your outbound dial plans in `conf/dialplan/default/yourdomain.xml`.

### SIP Profiles

In one of the 1.0.4pre releases, the default internal SIP profile (`conf/sip_profiles/internal.xml`) was significantly edited and commented up. In this file you will find the params "force-register-domain" and "force-register-db-domain" as shown here:

```
<!--
These are enabled to make the default config work better out of the box.
If you need more than ONE domain you'll need to not use these options.
-->
<param name="force-register-domain" value="${domain}"/>
<param name="force-register-db-domain" value="${domain}"/>
```

Also, modified

```
<!--force the domain in subscriptions to this value-->
<param name="force-subscription-domain" value="${domain}"/>
```
Go ahead and comment out both of these options to enable multi-tenant/multi-domain registrations. If you fail to do so, your users will register as "1001@", where 1001 is the extension, with no domain after the @ symbol.

Gateways

Gateways are mentioned here, because you have several options when defining connections to other SDPs, or Gateways; let your imagination be your guide. If you 'think' you can do it, you probably can. Gateways can be defined in a profile, or in a domain (in the directory). Take your pick. Defining gateways in a domain allows you to have different gateways per domain. This is helpful for accounting purposes. Some providers allow you to create sub-accounts, and one use case may be to define a sub account per domain, to be able to separate out your CDRs, etc.