# strongSwan - Feature #1300

## Use the same set of plugin config files for charon and charon-systemd

08.02.2016 12:15 - Yves-Alexis Perez

| | | | | |
|---|---|---|---|---|
| **Status:** | Closed | | **Start date:** | 08.02.2016 |
| **Priority:** | Normal | | **Due date:** | |
| **Assignee:** | Tobias Brunner | | **Estimated time:** | 0.00 hour |
| **Category:** | charon-systemd | | | |
| **Target version:** | 5.4.0 | | | |
| **Resolution:** | Fixed | | | |

**Description**

Hi,

I recently started playing with charon-systemd, and was surprised that my charon configuration (here it was advmss in kernel-netlink plugin configuration) was not loaded anymore.

That's because /etc/strongswan.conf only has include lines for charon {} and not the other charon daemon. I think it'd make sense to have all of them here, and by default to have all of them including the same configuration. If an user need to tune it, then she can override that.

Attached is a patch against strongswan.conf doing that.

Another solution would be to actually drop the charon {} block altogether, and put it in /etc/strongswan.d/charon.conf (already included everywhere), and have all other charon daemon ship a relevant charon-foo.conf with the same block.

**Associated revisions**

**Revision 819da83f - 08.03.2016 10:56 - Tobias Brunner**

Merge branch 'charon-conf-fallback'

Makes charon-systemd and charon-svc also load settings from the charon
section in strongswan.conf.

Fixes #1300.

**History**

**#1 - 08.02.2016 17:31 - Tobias Brunner**

*- Tracker changed from Issue to Feature*

*- Subject changed from use the same set of charon config file for all charon daemons to Use the same set of plugin config files for charon and charon-systemd*

*- Category set to charon-systemd*

*- Status changed from New to Feedback*

*- Assignee set to Tobias Brunner*

*- Target version set to 5.4.0*

*charon-cmd* and *charon-nm* are clients, which use a different set of plugins. But I agree that for *charon-systemd* something like this is definitely required, in particular because it uses the same set of default plugins and probably is never used together with the default charon daemon (the same applies to *charon-svc*). So we could add a copy of the *charon* section for *charon-systemd* in the default strongswan.conf file, which would at least fix the plugin config.

However, the problem is that this issue also concerns strongswan.d/charon.conf. Because the options in that file are defined in the *charon* section they won't apply to *charon-systemd* without renaming that section (or copying the file and renaming it then). The already existing strongswan.d/charon-systemd.conf file only contains options specific to *charon-systemd* (logging via systemd's journal logger).

One solution would be to manually patch that file so it also contains the options from the *charon* section. Another would be to change charon.conf to contain the options without parent section, so we would be able to include this file in the *charon* and *charon-systemd* sections in strongswan.conf. Both of these approaches would require some additional hacks in the Makefile that creates these config files ([source:conf/Makefile.am](source:conf/Makefile.am)).

Luckily, we already have an alias/fallback system in our settings framework. We use this for renamed sections and for defaults. For instance, for many settings in *charon* defaults may be defined in *libstrongswan*. Another example is the *libtls* section, which defines defaults that may be overridden in e.g. *charon.tls*. We can use this to inherit settings from the *charon* section in *charon-systemd*. I pushed patches for this to the

*1300-charon-conf-fallback* branch.

**#2 - 10.02.2016 21:21 - Yves-Alexis Perez**

Thanks, the patch seems to work fine indeed here.

**#3 - 11.02.2016 14:14 - Tobias Brunner**

> Thanks, the patch seems to work fine indeed here.

There is one drawback, but not sure how significant it is. Since we currently can only set the alias after the call to library_init() any code that accesses lib->settings during initialization of the library (e.g. *charon.integrity_test* or *charon.crypto_test.*\* or *charon.cert_cache*) will not be influenced by this (i.e. these settings will have to be configured in either the *libstrongswan* or the specific *charon-*\* section). To change that we'd have to add the possibility to add such aliases early on (e.g. via library_init() or some static initialization before that).

**#4 - 08.03.2016 10:59 - Tobias Brunner**

*- Status changed from Feedback to Closed*

*- Resolution set to Fixed*

I now pushed some patches that also work for settings that are read during initialization of libstrongswan.

## Files

| | | | |
|---|---|---|---|
| 0001-use-the-same-charon-configuration-for-all-charon-dae.patch | 1 KB | 08.02.2016 | Yves-Alexis Perez |