strongSwan - Feature #2372

Allow swanctl to read socket from strongswan.conf setting rather than default to hard-coded "unix:///var/run/charon.vici"

29.06.2017 00:52 - Gerald Turner

Status:	Closed	**Start date:**	28.06.2017
Priority:	Low	**Due date:**	
Assignee:	Tobias Brunner	**Estimated time:**	0.00 hour
Category:	swanctl		
Target version:	5.6.0		
Resolution:	Fixed		

Description

Disclaimer: Rather than waste any scarce brain power, feel free to close this bug report **WONTFIX** immediately ;-)

While preparing to upgrade several sites running traditional charon/stroke to modern charon-systemd/swanctl, including rewriting a couple rudimentary tunnel monitoring shell scripts, for a brief time I had the **misguided** belief that I would end up specifying *charon.user* and *charon.group*, tweak systemd tmpfiles.d to create a /run/strongswan subdirectory at boot, and specify *charon.plugins.vici.socket* to create it's socket file in that subdirectory. I had discovered that the swanctl command-line tool ignores the *charon.plugins.vici.socket* setting and instead hard-codes *unix:///var/run/charon.vici*. These aforementioned monitoring shell scripts, as well as incidental invocations of swanctl, would require the use of the '-u' switch. I had written a small patch, included here, that changes libvici to use the *charon.plugins.vici.socket* configuration setting before falling back on the hard-coded default.

Later I found that strongSwan makes all the appropriate setuid/setgid calls when creating the socket files and there isn't any need to override the default *charon.plugins.vici.socket* configuration. Thus this bug report and patch is frivolous. However rather than discard my patch, I figured that since it's subtly more correct, that maybe the project would appreciate it?

Associated revisions

Revision 4272a3e9 - 27.07.2017 13:22 - Tobias Brunner

swanctl: Read default socket from swanctl.socket option

Also read from swanctl.plugins.vici.socket so we get
libstrongswan.plugins.vici.socket if it is defined.

Fixes #2372.

History

#1 - 29.06.2017 10:24 - Tobias Brunner

- Status changed from New to Feedback

- Assignee set to Tobias Brunner

> I had discovered that the swanctl command-line tool ignores the *charon.plugins.vici.socket* setting and instead hard-codes *unix:///var/run/charon.vici*.

As you saw it's actually libvici that does that if no URI is specified when a connection is created.

> These aforementioned monitoring shell scripts, as well as incidental invocations of swanctl, would require the use of the '-u' switch.

Correct.

> I had written a small patch, included here, that changes libvici to use the *charon.plugins.vici.socket* configuration setting before falling back on the hard-coded default.

An alternative would be to introduce a *swanctl.socket* option that defines a default value if -u is not passed (implemented in the *2372-swanctl-socket* branch). Using "charon.*" hard-coded in libvici to read settings is potentially problematic as the plugin could be used by different daemons using different sockets. Granted that's less of a problem since 5.4.0 / #1300 because charon-systemd and charon-svc now have a fallback to charon's settings (i.e. it's not required to specifically set *charon-systemd.plugins.vici.socket*), but the socket option could still be set differently only for a single daemon (or globally in *libstrongswan.plugins.vici.socket*, to which there would be no fallback from swanctl if *charon.** was used, unless such a fallback

is explicitly set up e.g. in libvici). When reading the setting we could use %s.* and lib->ns but that's set to *swanctl* in that context so you'd have to configure *swanctl.plugins.vici.socket* (would then have about the same effect as a *swanctl.socket* option, however, the global *libstrongswan.plugins.vici.socket* would automatically be considered), but that might be more confusing because the vici plugin is not actually loaded by swanctl.

#2 - 03.07.2017 19:04 - Gerald Turner

Tobias Brunner wrote:

> An alternative would be to introduce a *swanctl.socket* option that defines a default value if -u is not passed (implemented in the *2372-swanctl-socket* branch). Using "charon.*" hard-coded in libvici to read settings is potentially problematic as the plugin could be used by different daemons using different sockets. Granted that's less of a problem since 5.4.0 / #1300 because charon-systemd and charon-svc now have a fallback to charon's settings (i.e. it's not required to specifically set *charon-systemd.plugins.vici.socket*), but the socket option could still be set differently only for a single daemon (or globally in *libstrongswan.plugins.vici.socket*, to which there would be no fallback from swanctl if *charon.** was used, unless such a fallback is explicitly set up e.g. in libvici). When reading the setting we could use %s.* and lib->ns but that's set to *swanctl* in that context so you'd have to configure *swanctl.plugins.vici.socket* (would then have about the same effect as a *swanctl.socket* option, however, the global *libstrongswan.plugins.vici.socket* would automatically be considered), but that might be more confusing because the vici plugin is not actually loaded by swanctl.

I tested the *2372-swanctl-socket* branch, it's effective if two settings are used: {charon{,-systemd},libstrongswan}.plugins.vici.socket and swanctl.socket. However if I'm parsing your statement correctly, you're saying that swanctl would fallback to libstrongswan.plugins.vici.socket? - that's not the case, only swanctl.socket has any effect on swanctl. Regardless of that, at least having some way of changing the default/hard-coded value is useful, thanks!

Note about testing: I have a personal "fork" of Debian's strongswan git-buildpackge repository in which I diverge from Debian by enabling a few additional plugins, and have applied additional patches like the three bugs I opened last week. As far as testing *2372-swanctl-socket* goes, I merely refreshed the patch with the diff between 2372-swanctl-include and master branches. Other than these patches, I'm actually testing the 5.5.3 release.

#3 - 04.07.2017 09:10 - Tobias Brunner

> However if I'm parsing your statement correctly, you're saying that swanctl would fallback to libstrongswan.plugins.vici.socket? - that's not the case, only swanctl.socket has any effect on swanctl.

Yes, the patch I pushed only read that key. The fallback is only available when reading *swanctl.plugins.vici.socket* (or *charon.plugins.vici.socket* if the fallback is set up manually in either swanctl or libvici). Then the value *libstrongswan.plugins.vici.socket* would be read if the more specific value is not defined. We could easily read *swanctl.plugins.vici.socket* too so we had that fallback. I've updated the patch in the branch.

#4 - 27.07.2017 13:26 - Tobias Brunner

- Subject changed from Allow swanctl to read "charon.plugins.vici.socket" strongswan.conf setting rather than default to hard-coded "unix:///var/run/charon.vici" to Allow swanctl to read socket from strongswan.conf setting rather than default to hard-coded "unix:///var/run/charon.vici"

- Status changed from Feedback to Closed

- Target version set to 5.6.0

- Resolution set to Fixed

Files

07_libvici-vici_connect-default-uri-from-settings.patch	730 Bytes	28.06.2017	Gerald Turner