strongSwan - Feature #243

Configure routing table in peer

07.10.2012 20:50 - Andre Valentin

Status:	Feedback	**Start date:**	07.10.2012
Priority:	Normal	**Due date:**	
Assignee:	Tobias Brunner	**Estimated time:**	0.00 hour
Category:	libhydra		
Target version:			
Resolution:			

Description

Hi!

StrongSWAN has support for a fwmark in a peer configuration. This is perfect to separate customers from each other on shared platforms. If used in this case, every customer has it's own routing table. In the documentation I have not found a possibility to set the routing table used for a peer.
Please add support to allow to specify a routing table ID in the peer configuration.

History

#1 - 08.10.2012 10:57 - Tobias Brunner

- Category set to libhydra

- Status changed from New to Feedback

Currently, the kernel interface plugins that install the routes have no peer information. I suppose it would be possible to add a connection specific routing table number as argument somehow, but that would require quite some refactoring in our kernel interfaces. There are currently two options to change the routing table to be used, first with the --with-routing-table ./configure option, second with the charon.routing_table strongswan.conf option, obviously these are both *global*.

Anyway, you could probably do something like this if you install the routes manually in a custom updown script (disable the default route installation with charon.install_routes).

#2 - 08.10.2012 19:35 - Andre Valentin

Hi!

If I our chiefs create a project out of this, I would like to write a patch to allow the definition via a peer.. (Will take 1..2 month until I can start). Do you have any advice for me, as I would like to get it upstream?

Thanks!

#3 - 09.10.2012 11:44 - Tobias Brunner

A *real* peer-specific config option will probably not be possible, but connection specific should work (a connection, e.g. with right=%any and without rightid, might be used for multiple peers).

How a new keyword may be added to ipsec.conf and transmitted to charon via the stroke interface is illustrated by commit 17319aa (since this option is not left|right specific, e129168b might fit even better). You will then have to add the new config value to either *child_cfg_t* or *peer_cfg_t*. Later when CHILD_SAs are installed the value has to be supplied to the kernel interface plugins, that is, extending *kernel_net_t* and *kernel_ipsec_t* - plus *kernel_interface_t* - will be required. And of course the actual implementation in the plugins would have to be adapted. Since the *kernel_ipsec_t* implementation of the *kernel-netlink* plugin uses the *kernel_net_t* implementation to install the routes the parameter had to be provided to both parts, which is not that nice. Refactoring the route installation somehow might result in a cleaner solution.

Overall, I still think it will be easier for you to implement this with a custom updown script. You may even supply the routing table to the script when configuring it with leftupdown, e.g.

```
leftupdown="/path/to/script <routing table>"
```

#4 - 10.10.2012 19:58 - Andre Valentin

Thanks for you hints. The script is of course a good alternative. I will evaluate what to do if I'm goint to be assigned!

#5 - 23.05.2013 20:03 - Andreas Steffen

- Assignee set to Tobias Brunner