

strongSwan - Bug #2199

charon hangs when `parallel_route` is set to yes

28.12.2016 17:04 - Alan Yang

Status:	Closed	Start date:	28.12.2016
Priority:	Normal	Due date:	
Assignee:	Tobias Brunner	Estimated time:	0.00 hour
Category:	kernel-interface	Resolution:	Fixed
Target version:	5.5.3		
Affected version:	5.5.1		

Description

With `charon.plugins.kernel-netlink.parallel_route` set to `yes`, charon hangs at startup..

I'm not sure if this bug is on the kernel side, and I don't know what is the point of performing "concurrent Netlink ROUTE queries on a single socket". But anyway, simply setting it to `yes` makes charon hang at startup, and that sounds like a bug to me.

```
# ipsec start --nofork --debug-all
Starting strongSwan 5.5.1 IPsec [starter]...
Loading config setup
[...]
found netkey IPsec stack
Attempting to start charon...
00[DMN] Starting IKE charon daemon (strongSwan 5.5.1, Linux 4.8.13-1-ARCH, x86_64)
00[KNL] sending XFRM_MSG_GETSPDINFO 201: => 20 bytes @ 0x7fff44dec940
00[KNL]   0: 14 00 00 00 25 00 01 00 C9 00 00 00 F7 0A 00 00   ....%.....
00[KNL]   16: 00 00 00 00   ....
00[KNL] received XFRM_MSG_NEWSPDINFO 201: => 76 bytes @ 0x170ea70
00[KNL]   0: 4C 00 00 00 24 00 00 00 C9 00 00 00 F7 0A 00 00   L...$......
00[KNL]   16: 00 00 00 00 1C 00 01 00 00 00 00 00 00 00 00 00   .....
00[KNL]   32: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   .....
00[KNL]   48: 0C 00 02 00 07 00 00 00 00 00 10 00 06 00 03 00   .....
00[KNL]   64: 20 20 00 00 06 00 04 00 80 80 00 00   .....
00[KNL] sending XFRM_MSG_GETSPDINFO 202: => 20 bytes @ 0x7fff44dec940
00[KNL]   0: 14 00 00 00 25 00 01 00 CA 00 00 00 F7 0A 00 00   ....%.....
00[KNL]   16: 00 00 00 00   ....
00[KNL] received XFRM_MSG_NEWSPDINFO 202: => 76 bytes @ 0x170ea70
00[KNL]   0: 4C 00 00 00 24 00 00 00 CA 00 00 00 F7 0A 00 00   L...$......
00[KNL]   16: 00 00 00 00 1C 00 01 00 00 00 00 00 00 00 00 00   .....
00[KNL]   32: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   .....
00[KNL]   48: 0C 00 02 00 07 00 00 00 00 00 10 00 06 00 03 00   .....
00[KNL]   64: 20 20 00 00 06 00 04 00 80 80 00 00   .....
00[KNL] known interfaces and IP addresses:
00[KNL] sending RTM_GETLINK 201: => 17 bytes @ 0x7fff44deca10
00[KNL]   0: 11 00 00 00 12 00 01 03 C9 00 00 00 F7 0A 00 00   .....
00[KNL]   16: 00   .
charon too long to start... - kill kill
child 2807 (charon) has been killed by sig 9

charon has died -- restart scheduled (5sec)
```

Associated revisions

Revision 7caec9e4 - 23.05.2017 16:49 - Tobias Brunner

kernel-netlink: Directly handle Netlink messages if thread pool is empty

During initialization of the plugins the thread pool is not yet initialized so there is no watcher thread that could handle the queued Netlink message and the main thread will wait indefinitely for a response.

Fixes #2199.

History

#1 - 28.12.2016 17:10 - Alan Yang

Oh, I found this in the comment section of [#1491](#):

No, these were added for a third-party implementation of the Netlink interface. Using them on vanilla Linux kernels does not improve performance (could even deteriorate it).

Why not document that along the configuration file? Plus a "breaks charon on mainline kernel" warning.

#2 - 10.04.2017 19:59 - Noel Kuntze

- *Tracker changed from Issue to Bug*

- *Start date set to 28.12.2016*

Reproduced this with 5.5.2 on 4.9.20-lts.
It hangs right after "known interfaces and IP addresses".

#3 - 12.04.2017 14:47 - Tobias Brunner

Reproduced this with 5.5.2 on 4.9.20-lts.
It hangs right after "known interfaces and IP addresses".

Well, it's not meant to work, so not sure what you expected :)

#4 - 12.04.2017 14:50 - Noel Kuntze

Well, the man page for strongswan.conf says this:

```
On vanilla Linux, DUMP queries fail with EBUSY and must be retried, further decreasing performance.
```

I understood this as that the kernel-netlink plugin retries the query then, instead of waiting. To me it seems it doesn't do that.
I'd expect just performance to be degraded.

#5 - 12.04.2017 15:32 - Tobias Brunner

- *Category set to kernel-interface*

- *Status changed from New to Feedback*

- *Target version set to 5.5.3*

OK, I had a closer look at this and it is actually a bug that causes that lockup. I pushed a fix to the *2199-kernel-netlink-parallel* branch.

#6 - 24.05.2017 15:35 - Tobias Brunner

- *Status changed from Feedback to Closed*

- *Assignee set to Tobias Brunner*

- *Resolution set to Fixed*