strongSwan - Bug #1453

Starter is getting stuck handling ipsec reload

09.05.2016 14:03 - Sankar Penniboyina

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Tobias Brunner	**Estimated time:**	0.00 hour
Category:	starter		
Target version:	5.6.1		
Affected version:	5.5.3	**Resolution:**	Fixed

Description

Hi,

We have several hundred connections defined in ipsec.conf. We have script which calls "ipsec reload" whenever a new connection definition is added. If the script calls "ipsec reload" few times in loop due to any reason we are seeing that starter getting stuck in the following call stack and never recovers. We started seeing this issue after upgraded to 5.3.5. Any help in debugging this issue is highly appreciated.

```
(gdb) bt
#0  0x00007faeee9edcb0 in __write_nocancel () from /lib/x86_64-linux-gnu/libpthread.so.0
#1  0x00007faeeee31c25 in update (this=<optimized out>) at processing/watcher.c:121
#2  0x00007faeeee31cce in remove_ (this=0xf60490, fd=6) at processing/watcher.c:490
#3  0x00007faeeee2cc1e in destroy (this=0xfe2730) at networking/streams/stream.c:263
#4  0x0000000000405dc3 in send_stroke_msg (msg=0xf61840) at starterstroke.c:119
#5  0x0000000000406476 in starter_stroke_add_conn (cfg=0xfd0e30, conn=0xfbe000) at starterstroke.c
:280
#6  0x0000000000403023 in main (argc=<optimized out>, argv=<optimized out>) at starter.c:902

(gdb) info threads
  Id   Target Id         Frame
* 1    Thread 0x7faeef276700 (LWP 25088) "starter" 0x00007faeee9edcb0 in __write_nocancel () from
/lib/x86_64-linux-gnu/libpthread.so.0
```

Related issues:

Related to Issue #2414: Charon sometimes doesn't react to stroke command **Closed**

Associated revisions

Revision 0d08959a - 10.10.2017 11:13 - Tobias Brunner

watcher: Don't notify watcher if removed FD was not found

This can happen if a stream is used blocking exclusively (the FD is
never registered with watcher, but is removed in the stream's destructor
just in case it ever was - doing this conditionally would require an
additional flag in streams). There may be no thread reading from
the read end of the notify pipe (e.g. in starter), causing the write
to the notify pipe to block after it's full. Anyway, doing a relatively
expensive FD update is unnecessary if there were no changes.

Fixes #1453.

History

#1 - 09.05.2016 15:57 - Tobias Brunner

- Description updated

- Category set to starter

- Status changed from New to Feedback

> If the script calls "ipsec reload" few times in loop

That command just sends a SIGUSR1 to the starter daemon. I haven't looked into this in detail yet, but maybe the signal handling/multi threading does not interact very well at that point.

On the other hand you perhaps should just call ipsec reload once from your script if **any** changes occurred and not for **every** single change. Actually, you should probably use ipsec update, which has no effect on unchanged connections. And you might want to have a look at vici/swanctl.

> We started seeing this issue after upgraded to 5.3.5

What version did you use before? (Streams and watcher_t are used since 5.2.0)

#2 - 10.05.2016 08:11 - Sankar Penniboyina

Thanks Tobias. I have updated the script to use "ipsec update" but want to make sure if same issue can occur with "ipsec update" as well as this call stack is hit even when running "ipsec update". I briefly checked the swanctl but don't see any option to reload only the changed connection configs.

We were using 5.1.0 before moving to 5.3.5.

#3 - 10.05.2016 10:30 - Tobias Brunner

> I have updated the script to use "ipsec update" but want to make sure if same issue can occur with "ipsec update" as well as this call stack is hit even when running "ipsec update".

Yes it could still happen, especially, if you call it in rapid succession (that's why I said you should only call it once after doing all your file changes, not for every single update). But with update fewer stroke messages are transmitted, so chances for conflicts are reduced.

> I briefly checked the swanctl but don't see any option to reload only the changed connection configs.

The vici plugin merges them automatically and leaves unchanged configs as they are. And multiple --load-conns calls are serialized.

#4 - 10.05.2016 11:37 - Sankar Penniboyina

Thanks Tobias. I will try swanctl.

#5 - 31.05.2017 00:19 - Noel Kuntze

- Status changed from Feedback to Closed

- Resolution set to No feedback

#6 - 06.09.2017 18:36 - Tomas Paukrt

- File strongswan-5.5.3-watcher-fix.patch added

We have experienced exactly same issue with strongSwan 5.5.3, so I looked into the source code and found out that function **update** in **watcher.c** is sending one character to write-end of notify pipe in blocking mode while there is no thread for extracting data from read-end of notify pipe, so program **starter** stopped working after 65536 writes in our case.

I attached quick fix that switches both ends of notify pipe to non-blocking mode.

#7 - 07.09.2017 10:18 - Tobias Brunner

- Affected version changed from 5.3.5 to 5.5.3

> We have experienced exactly same issue with strongSwan 5.5.3, so I looked into the source code and found out that function **update** in **watcher.c** is sending one character to write-end of notify pipe in blocking mode while there is no thread for extracting data from read-end of notify pipe, so program **starter** stopped working after 65536 writes in our case.

I see. Since there are no threads in starter there is, as you noticed, nobody reading from the other end of the notify pipe. However, while we use a stream to connect to the stroke socket, we don't register any callbacks (i.e. these calls happen blocking, without any interaction with watcher). So there is not really any need for this (source:src/libstrongswan/networking/streams/stream.c#L263) call to remove the FD from watcher (doing this conditionally would require an additional flag, though), or calling update() if there were no changes in watcher (source:src/libstrongswan/processing/watcher.c#L545). I've pushed a commit that changes the latter to the *watcher-no-update* branch.

#8 - 07.09.2017 10:20 - Tobias Brunner

- Has duplicate Issue #2414: Charon sometimes doesn't react to stroke command added

#9 - 07.09.2017 10:21 - Tobias Brunner

- Has duplicate deleted (Issue #2414: Charon sometimes doesn't react to stroke command)

#10 - 07.09.2017 10:21 - Tobias Brunner

- Related to Issue #2414: Charon sometimes doesn't react to stroke command added

#11 - 07.09.2017 12:58 - Tobias Brunner

- Status changed from Closed to Assigned

- Assignee set to Tobias Brunner

- Resolution deleted (No feedback)

#12 - 07.09.2017 12:58 - Tobias Brunner

- Tracker changed from Issue to Bug

- Target version set to 5.6.1

#13 - 10.10.2017 11:16 - Tobias Brunner

- Status changed from Assigned to Closed

- Resolution set to Fixed

Files

strongswan-5.5.3-watcher-fix.patch	1.47 KB	06.09.2017	Tomas Paukrt