strongSwan - Bug #2779

mysql plugin crash in various ways

01.10.2018 15:12 - Jean-Daniel Dupas

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Tobias Brunner	**Estimated time:**	0.00 hour
Category:	libstrongswan		
Target version:	5.7.2		
Affected version:	5.7.0	**Resolution:**	Fixed

Description

I'm using strongswan-swanctl as a VPN server, and I'm using the sql plugin to manage ippool.

My problem is that the server is very instable and crash often with stack trace usually in the libmysqlclient code, which is probably due to a bad usage of the library:

```
#0  0x00007fee60adcd2e in ?? () from /usr/lib/x86_64-linux-gnu/libmysqlclient.so.20
#1  0x00007fee60adcf8f in ?? () from /usr/lib/x86_64-linux-gnu/libmysqlclient.so.20
#2  0x00007fee60add3a9 in ?? () from /usr/lib/x86_64-linux-gnu/libmysqlclient.so.20
#3  0x00007fee60add6d2 in ?? () from /usr/lib/x86_64-linux-gnu/libmysqlclient.so.20
#4  0x00007fee60ad7812 in ?? () from /usr/lib/x86_64-linux-gnu/libmysqlclient.so.20
#5  0x00007fee60ad7b26 in mysql_stmt_prepare () from /usr/lib/x86_64-linux-gnu/libmysqlclient.so.2
0
#6  0x00007fee610c35c0 in run (mysql=0x7fee44051a90, sql=sql@entry=0x7fee60293a68 "SELECT id, star
t, timeout FROM pools WHERE name = ?", args=args@entry=0x7fee5ae03960) at mysql_database.c:292
#7  0x00007fee610c436f in query (this=0x55fb96702b50, sql=0x7fee60293a68 "SELECT id, start, timeou
t FROM pools WHERE name = ?") at mysql_database.c:541
#8  0x00007fee60292f9d in get_pool (name=<optimized out>, family=family@entry=2, timeout=timeout@e
ntry=0x7fee5ae03adc, this=<optimized out>) at attr_sql_provider.c:112
#9  0x00007fee602935dd in acquire_address (this=0x55fb966cfea0, pools=0x7fee200466e0, ike_sa=<opti
mized out>, requested=<optimized out>) at attr_sql_provider.c:265
#10 0x00007fee71c36103 in acquire_address (this=0x55fb96608a70, pools=0x7fee200466e0, ike_sa=0x55f
b9677c490, requested=0x7fee4c068dc0) at attributes/attribute_manager.c:74
#11 0x00007fee71c76fd8 in build_r (this=0x55fb96720c40, message=0x7fee4c0586e0) at sa/ikev2/tasks/
ike_config.c:357
#12 0x00007fee71c6ae6a in build_response (request=0x7fee4c026830, this=0x7fee3005a870) at sa/ikev2
/task_manager_v2.c:864
#13 process_request (message=<optimized out>, this=0x7fee3005a870) at sa/ikev2/task_manager_v2.c:1
225
#14 process_message (this=0x7fee3005a870, msg=<optimized out>) at sa/ikev2/task_manager_v2.c:1554
#15 0x00007fee71c5a007 in process_message (this=0x55fb9677c490, message=0x7fee4c026830) at sa/ike_
sa.c:1569
#16 0x00007fee71c53c04 in execute (this=0x7fee4c04a740) at processing/jobs/process_message_job.c:7
4
#17 0x00007fee71ef64a6 in process_job (worker=0x55fb966ce940, this=0x55fb965fda90) at processing/p
rocessor.c:235
#18 process_jobs (worker=0x55fb966ce940) at processing/processor.c:321
#19 0x00007fee71f07c2b in thread_main (this=0x55fb967109e0) at threading/thread.c:331
#20 0x00007fee7178f6db in start_thread (arg=0x7fee5ae04700) at pthread_create.c:463
#21 0x00007fee714b888f in clone () at ../sysdeps/unix/sysv/linux/x86_64/clone.S:95

#0  0x000056214cc2d620 in ?? ()
#1  0x00007f2ebf96eb83 in mysql_close (mysql=0x56214cd2f590) at ./sql-common/client.c:5220
#2  0x00007f2ebff52f7a in conn_destroy (this=0x56214cd26d30) at mysql_database.c:194
#3  conn_get (this=0x56214cd26b50, trans=<optimized out>) at mysql_database.c:243
#4  0x00007f2ebff53022 in transaction (this=0x56214cd26b50, serializable=<optimized out>) at mysql
_database.c:653
#5  0x00007f2ebf121b6c in get_identity (ike_sa=<optimized out>, this=<optimized out>) at attr_sql_
provider.c:57
#6  0x00007f2ebf12259b in acquire_address (this=0x56214ccf3ea0, pools=0x7f2e7c06fe80, ike_sa=<opti
mized out>, requested=0x7f2eac076e40) at attr_sql_provider.c:257
#7  0x00007f2ed0ac5103 in acquire_address (this=0x56214cc2ca70, pools=0x7f2e7c06fe80, ike_sa=0x7f2
```

```
e9408de50, requested=0x7f2eac076e40) at attributes/attribute_manager.c:74
#8  0x00007f2ed0b05fd8 in build_r (this=0x7f2eac086ca0, message=0x7f2e800297f0) at sa/ikev2/tasks/
ike_config.c:357
#9  0x00007f2ed0af9e6a in build_response (request=0x7f2eac09d4a0, this=0x7f2e94080ef0) at sa/ikev2
/task_manager_v2.c:864
#10 process_request (message=<optimized out>, this=0x7f2e94080ef0) at sa/ikev2/task_manager_v2.c:1
225
#11 process_message (this=0x7f2e94080ef0, msg=<optimized out>) at sa/ikev2/task_manager_v2.c:1554
#12 0x00007f2ed0ae9007 in process_message (this=0x7f2e9408de50, message=0x7f2eac09d4a0) at sa/ike_
sa.c:1569
#13 0x00007f2ed0ae2c04 in execute (this=0x7f2eac0851c0) at processing/jobs/process_message_job.c:7
4
#14 0x00007f2ed0d854a6 in process_job (worker=0x56214cc65660, this=0x56214cc21a90) at processing/p
rocessor.c:235
#15 process_jobs (worker=0x56214cc65660) at processing/processor.c:321
#16 0x00007f2ed0d96c2b in thread_main (this=0x56214cd346d0) at threading/thread.c:331
#17 0x00007f2ed061e6db in start_thread (arg=0x7f2eba494700) at pthread_create.c:463
#18 0x00007f2ed034788f in clone () at ../sysdeps/unix/sysv/linux/x86_64/clone.S:95
```

Do the sql plugin properly take care of the concurrency restriction when using Mysql library ?

Especially:

Multiple threads cannot send a query to the MySQL server at the same time on the same connection. In particular, you must ensure that between calls to mysql_query() and mysql_store_result() in one thread,
no other thread uses the same connection. You must have a mutex lock around your pair of mysql_query() and mysql_store_result() calls. After mysql_store_result() returns, the lock can be released and other threads may query the same connection.

Associated revisions

Revision a61b1a6e - 26.10.2018 11:18 - Tobias Brunner

mysql: Don't release the connection if transactions are still using it

Fixes #2779.

History

#1 - 01.10.2018 15:36 - Jean-Daniel Dupas

By the way, I'm using strongswan 5.6.3, and this issue rarely occurs when the number of client is very low. That what make me think this is a threading issue.

#2 - 01.10.2018 15:39 - Tobias Brunner

- Status changed from New to Feedback

> My problem is that the server is very instable and crash often with stack trace usually in the libmysqlclient code, which is probably due to a bad usage of the library:

It apparently worked for years without problems. So why couldn't it be a bug in the client library?

> Do the sql plugin properly take care of the concurrency restriction when using Mysql library ?

Yes, each connection is only used by one thread concurrently (see source:src/libstrongswan/plugins/mysql/mysql_database.c).

#3 - 02.10.2018 17:25 - Jean-Daniel Dupas

I'm not sure yet, but there is something dubious.

I start tracing the mysql connection usage and sometimes, I got both mysql_enumerator_destroy() and transaction_destroy() that try to release the same connection.

A simple way to reveal the issue is to break when is_use is false at the entry of conn_release().

This means that just after the call to mysql_enumerator_destroy, the connection can be take from the pool by an other thread, and the is_use flag may then be reset to false by transaction_destroy, while the other thread own the connection.

#4 - 02.10.2018 17:38 - Jean-Daniel Dupas

I did just try to stop releasing the connection in enumerator_destroy when a transaction is active, and it looks like the system is far more stable.

#5 - 02.10.2018 18:52 - Noel Kuntze

I am working on the same problem for some time now and your observation is completely correct. The problem is caused because the functions that call conn_release() are not aware of the transactions using the connection.
For example, in the function get_identity()[1] of the attr_sql plugin, the call to e->destroy(e) already releases the connection, but the next call to commit()(still uses it to send a command and then releases it [2].

[1] https://github.com/strongswan/strongswan/blob/master/src/libcharon/plugins/attr_sql/attr_sql_provider.c#L65
[2] https://github.com/strongswan/strongswan/blob/master/src/libstrongswan/plugins/mysql/mysql_database.c#L695

#6 - 02.10.2018 19:39 - Noel Kuntze

Working on a patch right now and testing it.

#7 - 02.10.2018 20:23 - Noel Kuntze

- File check-transaction.patch added

I tested the following patch using the load-tester with several initiators for 500 and 5000 initiations and an otherwise known-good mysql client library against a patched host acting as responder. I observed no crashes or other failures.

Patch is attached as simple git diff.

#8 - 03.10.2018 09:36 - Tobias Brunner

- Tracker changed from Issue to Bug

- Target version set to 5.7.2

Good catch! I've pushed the fix to the *2779-mysql-conns* branch. Thanks Noel and Jean-Daniel.

#9 - 26.10.2018 11:19 - Tobias Brunner

- Status changed from Feedback to Closed

- Assignee set to Tobias Brunner

- Resolution set to Fixed

Files

check-transaction.patch	720 Bytes	02.10.2018	Noel Kuntze