strongSwan - Bug #3335

IKE_SA termination request ignored when reestablishing

10.02.2020 17:20 - Krishnamurthy Daulatabad

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Tobias Brunner	**Estimated time:**	0.00 hour
Category:	libcharon		
Target version:	5.8.3		
Affected version:	5.7.2	**Resolution:**	Fixed

Description

We have a 2 situations where VICI terminate of CHILD and IKE SA fails when remote peer is not reachable:

1. When initiator has IKE_SA in half open state and VICI terminate is attempted of IKE_SA. Deleting IKE_SA with force option helped in few attempts we have tried. But we are not sure.
2. When initiator has IKE_SA and CHILD_SA established. But SA is in DPD phase when delete is attempted. We have a DPD timeout of 9 seconds and 2 retries

strongswan.conf:

```
retransmit_timeout = 2
retransmit_tries = 2
retransmit_base = 1.0
```

In both the cases above, tunnel continues to exist and gets established later after terminate fails.

In both the cases we use force=1 and timeout of 1000ms in VICI terminate command;

Why is this the case? This is causing problems in our setup as the tunnel is expected to be deleted. Is there anyway to ensure that tunnel is deleted irrespective of the state?

Attaching the logs here:
IKE_SA in half_open state:

```
2020-02-09T16:57:10.0+0000 09[CFG] added vici connection: server_1
2020-02-09T16:57:10.0+0000 09[CFG] initiating 'server_1'
2020-02-09T16:57:10.0+0000 09[IKE] <server_1|1> initiating IKE_SA server_1[1] to 13.92.86.106*
2020-02-09T16:57:10.0+0000 13[IKE] Queued PB_VICI_CMD_DEL_CONN for server_1
2020-02-09T16:57:10.0+0000 12[CFG] vici terminate CHILD_SA 'server_1'
2020-02-09T16:57:10.0+0000 17[IKE] VICI command failed:
2020-02-09T16:57:10.0+0000 17[IKE] Termination of connection server_1 failed with err -22
2020-02-09T16:57:10.0+0000 17[IKE]
2020-02-09T16:57:10.0+0000 03[CFG] <server_1|1> selected proposal: IKE:AES_GCM_16_256/PRF_HMAC_SHA2_256/ECP_256
2020-02-09T16:57:10.0+0000 03[IKE] <server_1|1> local host is behind NAT, sending keep alives
2020-02-09T16:57:10.0+0000 03[IKE] <server_1|1> remote host is behind NAT
2020-02-09T16:57:10.0+0000 12[IKE] <server_1|1> IKE_SA server_1[1] established between 192.168.10.65[cl.customer1.tenant.int.ves.io]...13.92.86.106[server_1]
2020-02-09T16:57:10.0+0000 12[IKE] <server_1|1> scheduling reauthentication in 13231s
2020-02-09T16:57:10.0+0000 12[IKE] <server_1|1> maximum IKE_SA lifetime 14671s
2020-02-09T16:57:10.0+0000 12[IKE] <server_1|1> received ESP_TFC_PADDING_NOT_SUPPORTED, not using ESPv3 TFC padding
2020-02-09T16:57:10.0+0000 12[CFG] <server_1|1> selected proposal: ESP:AES_GCM_16_256/NO_EXT_SEQ
2020-02-09T16:57:10.0+0000 12[IKE] <server_1|1> CHILD_SA server_1{1} established with SPIs 238cd3ac_i 3eae97f4_o and TS 192.168.10.65/32 === 192.168.9.4/32
2020-02-09T16:57:10.0+0000 12[IKE] <server_1|1> peer supports MOBIKE
2020-02-09T16:57
——Tunnel Get established to server_1
```

CHILD SA in DPD phase - retransmit count is 2 in our case:

```
2020-02-10T14:58:42.0+0000 13[CFG] <server_1|9> selected proposal: ESP:AES_GCM_16_256/NO_EXT_SEQ
2020-02-10T14:58:42.0+0000 13[IKE] <server_1|9> CHILD_SA server_1{9} established with SPIs 6fc47e4d_i 312b0ec0_o and TS 50.228.178.69/32 === 84.54.61.126/32
```

```
2020-02-10T14:58:42.0+0000 13[IKE] <server_1|9> peer supports MOBIKE

2020-02-10T14:58:56.0+0000 12[IKE] Queued PB_VICI_CMD_DEL_CONN for server_1
2020-02-10T14:58:56.0+0000 14[CFG] vici terminate CHILD_SA 'server_1'
2020-02-10T14:58:57.0+0000 17[IKE] VICI command failed:
2020-02-10T14:58:57.0+0000 17[IKE] Termination of connection server_1 failed with err -22
2020-02-10T14:58:57.0+0000 17[IKE]
2020-02-10T14:58:57.0+0000 04[CFG] vici terminate IKE_SA 'server_1'
2020-02-10T14:58:57.0+0000 12[IKE] <server_1|9> giving up after 2 retransmits
2020-02-10T14:58:57.0+0000 12[IKE] <server_1|9> restarting CHILD_SA server_1
2020-02-10T14:58:57.0+0000 12[IKE] <server_1|9> initiating IKE_SA server_1[12] to 84.54.61.126
server_12020-02-10T14:58:57.0+0000 17[IKE] Terminated connection server_1
2020-02-10T14:58:57.0+0000 17[IKE]
2020-02-10T14:58:57.0+0000 11[CFG] <server_1|12> selected proposal:
IKE:AES_GCM_16_256/PRF_HMAC_SHA2_256/ECP_256
2020-02-10T14:58:57.0+0000 04[IKE] <server_1|12> IKE_SA server_1[12] established between
50.228.178.69[ver.santa-clara-office.volterra-corp-csthxgny.tenant.int.ves.io]...84.54.61.126[server_1]
2020-02-10T14:58:57.0+0000 04[IKE] <server_1|12> scheduling reauthentication in 13321s
2020-02-10T14:58:57.0+0000 04[IKE] <server_1|12> maximum IKE_SA lifetime 14761s
2020-02-10T14:58:57.0+0000 04[IKE] <server_1|12> received ESP_TFC_PADDING_NOT_SUPPORTED, not using ESPv3 TFC
padding
2020-02-10T14:58:57.0+0000 04[CFG] <server_1|12> selected proposal: ESP:AES_GCM_16_256/NO_EXT_SEQ
2020-02-10T14:58:57.0+0000 04[IKE] <server_1|12> CHILD_SA server_1{11} established with SPIs 1e09e652_i 2496e641_o and
TS 50.228.178.69/32 === 84.54.61.126/32
```

Associated revisions

Revision 96b61792 - 21.02.2020 10:38 - Tobias Brunner

ike: Don't reestablish IKE_SAs for which a deletion is queued

If an IKE_SA is terminated while a task is active, the delete task is simply queued (unless the deletion is forced). If the active task times out before any optional timeout associated with the termination hits, the IKE_SA previously was reestablished without considering the termination request.

Fixes #3335.

History

#1 - 11.02.2020 19:10 - Tobias Brunner

- Category set to vici

- Status changed from New to Feedback

```
2020-02-09T16:57:10.0+0000 12[CFG] vici terminate CHILD_SA 'server_1'
```

Please note that you attempted to terminate a CHILD_SA, not an IKE_SA.

```
2020-02-10T14:58:56.0+0000 14[CFG] vici terminate CHILD_SA 'server_1'
```

Same thing here.

#2 - 12.02.2020 04:28 - Krishnamurthy Daulatabad

In the case when SA is in DPD phase, I have sent both CHILD SA and IKE SA terminate commands. Even then the SA is not terminated. You can see that in the logs above and I have pasted them below again in 1). After deletion command of IKE SA, IKE SA and Child SA is again established. after this I have unloaded the connection too. Somehow the configuration is still there with charon.

I added a retry logic to delete in the tunnel with 10 retries. What I observed was that the tunnel actually got deleted only after it came out of DPD phase and started initiating a new tunnel. I have attached the logs for this too below 2. In this case 2 below both CHILD_SA and IKE_SA exists when the delete attempt is made. But still delete fails and eventually succeeds only of it restarts tunnel.

```
1) ---------
2020-02-10T14:58:56.0+0000 14[CFG] vici terminate CHILD_SA 'server_1'
2020-02-10T14:58:57.0+0000 17[IKE] VICI command failed:
2020-02-10T14:58:57.0+0000 17[IKE] Termination of connection server_1 failed with err -22
```

```
2020-02-10T14:58:57.0+0000 17[IKE]
2020-02-10T14:58:57.0+0000 04[CFG] vici terminate IKE_SA 'server_1'
2020-02-10T14:58:57.0+0000 12[IKE] <server_1|9> giving up after 2 retransmits
2020-02-10T14:58:57.0+0000 12[IKE] <server_1|9> restarting CHILD_SA server_1
2020-02-10T14:58:57.0+0000 12[IKE] <server_1|9> initiating IKE_SA server_112 to 84.54.61.126
server_12020-02-10T14:58:57.0+0000 17[IKE] Terminated connection server_1
2020-02-10T14:58:57.0+0000 17[IKE]
2020-02-10T14:58:57.0+0000 11[CFG] <server_1|12> selected proposal: IKE:AES_GCM_16_256/PRF_HMAC_SHA2_256/ECP_256
2020-02-10T14:58:57.0+0000 04[IKE] <server_1|12> IKE_SA server_112 established between 50.228.178.69[client1]...84.54.61.126[server_1]

2)---------
Queued PB_VICI_CMD_DEL_CONN for server_1
2020-02-11T07:05:58.0+0000 10[CFG] vici terminate CHILD_SA 'server_1'
2020-02-11T07:05:58.0+0000 13[IKE] <server_1|2> retransmit 2 of request with message ID 48
2020-02-11T07:05:59.0+0000 17[IKE] VICI command failed:
2020-02-11T07:05:59.0+0000 17[IKE] Termination of connection server_1 failed with err -22
2020-02-11T07:05:59.0+0000 17[IKE]
2020-02-11T07:05:59.0+0000 17[IKE] Retrying[1] termination of CHILD server_1
2020-02-11T07:05:59.0+0000 11[CFG] vici terminate CHILD_SA 'server_1'
2020-02-11T07:05:59.0+0000 14[IKE] VICI: Loading new configuration for server_1
2020-02-11T07:05:59.0+0000 14[IKE]
2020-02-11T07:06:00.0+0000 17[IKE] VICI command failed:
2020-02-11T07:06:00.0+0000 17[IKE] Termination of connection server_1 failed with err -22
2020-02-11T07:06:00.0+0000 17[IKE]
2020-02-11T07:06:00.0+0000 17[IKE] Retrying[2] termination of CHILD server_1
2020-02-11T07:06:00.0+0000 08[CFG] vici terminate CHILD_SA 'server_1'
2020-02-11T07:06:00.0+0000 05[IKE] <server_1|2> giving up after 2 retransmits
2020-02-11T07:06:00.0+0000 05[IKE] <server_1|2> restarting CHILD_SA server_1
2020-02-11T07:06:00.0+0000 05[IKE] <server_1|2> initiating IKE_SA server_1[4] to 13.92.86.106
2020-02-11T07:06:00.0+0000 17[IKE] Terminated connection server_1
2020-02-11T07:06:00.0+0000 17[IKE]
2020-02-11T07:06:00.0+0000 13[CFG] vici terminate IKE_SA 'server_1'
2020-02-11T07:06:00.0+0000 06[IKE] <server_1|4> destroying IKE_SA in state CONNECTING without notification
2020-02-11T07:06:00.0+0000 17[IKE] Terminated connection server_1
2020-02-11T07:06:00.0+0000 17[IKE]
2020-02-11T07:06:00.0+0000 17[IKE] Unloaded connection server_1
2020-02-11T07:06:00.0+0000 17[IKE]
```

#3 - 12.02.2020 10:48 - Tobias Brunner

> In the case when SA is in DPD phase, I have sent both CHILD SA and IKE SA terminate commands. Even then the SA is not terminated. You can see that in the logs above and I have pasted them below again in 1).

Probably a timing issue. If the retransmit job (or any other) is actively working on an IKE_SA, the *vici* plugin and later the terminate job can't lock the IKE_SA. And if a new IKE_SA was created just then, the terminate command doesn't know anything about it (i.e. the new IKE_SA won't be terminated). So with your low retransmit settings that might just be that.

You can see more about the locking of IKE_SAs if you increase the log level for the *mgr* log group.

> after this I have unloaded the connection too. Somehow the configuration is still there with charon.

That won't have any effect on currently active IKE_SAs and their reestablishment.

> I added a retry logic to delete in the tunnel with 10 retries.

What do you mean? Sending repeated terminate commands via VICI?

#4 - 12.02.2020 11:05 - Krishnamurthy Daulatabad

We are using a retransmit time of 2 seconds. That should be sufficient for VICI command to get the required lock right? Or is the IKE SA lock held across the entire DPD timeout phase?

Yes I am retrying the VICI command to terminate the connection with force=1 and timeout=1000ms. So retry after every 1 second.

So is there any way to ensure that IKE SA and Child SA is deleted for sure without using force=1 and timeout=-1

#5 - 12.02.2020 11:18 - Tobias Brunner

> We are using a retransmit time of 2 seconds. That should be sufficient for VICI command to get the required lock right?

Depends on how many SAs there are and if they are locked etc. There is a loop over all IKE_SAs and if that is blocked by one of them it obviously delays acquiring other IKE_SAs and so on.

> Or is the IKE SA lock held across the entire DPD timeout phase?

No, only when retransmits are sent etc.

> So is there any way to ensure that IKE SA and Child SA is deleted for sure without using force=1 and timeout=-1

Ah, I forgot about the timeout aspect. If the IKE_SA is recreated during the timeout, the termination will not be enforced after the timeout as it will only affect the current IKE_SA (which might already be destroyed due to the recreation) not the new one. There is no code that checks if a delete task is queued/active when considering to reestablish an IKE_SA (not sure if such a task is actually queued in your case, you'd have to check).

#6 - 13.02.2020 09:58 - Krishnamurthy Daulatabad

There are only 2 SAs from this initiator. So too much time should not be spent in looking over all SAs.

I enabled log level 3 and captured some logs. Here CHILD SA was established normally and stable before I simulated DPD with iptable rule to block packets from responder. Here SA entered into DPD phase by 06:41:06. So tunnel got deleted only after new IKE_SA was re-initiated.

From the logs it looks like the delete task is delayed due to INFORMATIONAL exchange in progress. So does this mean that one has to wait for this phase to complete before delete event gets handled? Can't it be queued and delete later? The issue here is that if delete is skipped, the tunnel gets reestablished which is not supposed to happen as it caused other problems in our datapath.

What if DPD timeout is more (say > 100s)? Waiting for this long and retrying will block other events in our case.

LOGS:

```
2020-02-13T06:41:10.0+0000 11[CFG] vici client 1 requests: terminate
2020-02-13T06:41:10.0+0000 11[CFG] vici terminate CHILD_SA 'server_1'
2020-02-13T06:41:10.0+0000 15[MGR] checkout IKEv2 SA with SPIs 4cefbe635810b56e_i 21bb5ed92dd035b5_r
2020-02-13T06:41:10.0+0000 15[MGR] IKE_SA server_1[1] successfully checked out
2020-02-13T06:41:10.0+0000 15[IKE] <server_1|1> queueing CHILD_DELETE task
2020-02-13T06:41:10.0+0000 15[IKE] <server_1|1> delaying task initiation, INFORMATIONAL exchange in progress
2020-02-13T06:41:10.0+0000 15[MGR] <server_1|1> checkin IKE_SA server_1[1]
2020-02-13T06:41:10.0+0000 15[MGR] <server_1|1> checkin of IKE_SA successful

2020-02-13T06:41:11.0+0000 16[CFG] vici client 1 requests: terminate
2020-02-13T06:41:11.0+0000 16[CFG] vici terminate CHILD_SA 'server_1'
2020-02-13T06:41:11.0+0000 14[MGR] checkout IKEv2 SA with SPIs 4cefbe635810b56e_i 21bb5ed92dd035b5_r
2020-02-13T06:41:11.0+0000 14[MGR] IKE_SA server_1[1] successfully checked out
2020-02-13T06:41:11.0+0000 14[IKE] <server_1|1> queueing CHILD_DELETE task
2020-02-13T06:41:11.0+0000 14[IKE] <server_1|1> delaying task initiation, INFORMATIONAL exchange in progress
2020-02-13T06:41:11.0+0000 14[MGR] <server_1|1> checkin IKE_SA server_1[1]
2020-02-13T06:41:11.0+0000 14[MGR] <server_1|1> checkin of IKE_SA successful
2020-02-13T06:41:11.0+0000 02[JOB] watcher got notification, rebuilding

2020-02-13T06:41:12.0+0000 08[CFG] vici client 1 requests: terminate
2020-02-13T06:41:12.0+0000 08[CFG] vici terminate CHILD_SA 'server_1'
2020-02-13T06:41:12.0+0000 11[MGR] checkout IKEv2 SA with SPIs 4cefbe635810b56e_i 21bb5ed92dd035b5_r
2020-02-13T06:41:12.0+0000 11[MGR] IKE_SA server_1[1] successfully checked out
2020-02-13T06:41:12.0+0000 11[IKE] <server_1|1> queueing CHILD_DELETE task
2020-02-13T06:41:12.0+0000 11[IKE] <server_1|1> delaying task initiation, INFORMATIONAL exchange in progress
2020-02-13T06:41:12.0+0000 11[MGR] <server_1|1> checkin IKE_SA server_1[1]
2020-02-13T06:41:12.0+0000 11[MGR] <server_1|1> checkin of IKE_SA successful

2020-02-13T06:41:13.0+0000 14[IKE] <server_1|1> giving up after 2 retransmits
2020-02-13T06:41:13.0+0000 14[MGR] <server_1|1> created IKE_SA (unnamed)[3]
2020-02-13T06:41:13.0+0000 14[IKE] <server_1|1> restarting CHILD_SA server_1
2020-02-13T06:41:13.0+0000 10[CFG] vici client 1 requests: terminate
2020-02-13T06:41:13.0+0000 10[CFG] vici terminate IKE_SA 'server_1'
2020-02-13T06:41:13.0+0000 11[MGR] checkout IKE_SA by unique ID 3
2020-02-13T06:41:13.0+0000 11[MGR] IKE_SA server_1[3] successfully checked out
2020-02-13T06:41:13.0+0000 11[IKE] <server_1|3> destroying IKE_SA in state CONNECTING without notification
2020-02-13T06:41:13.0+0000 11[MGR] <server_1|3> checkin and destroy IKE_SA server_1[3]
2020-02-13T06:41:13.0+0000 11[IKE] <server_1|3> IKE_SA server_1[3] state change: CONNECTING => DESTROYING
2020-02-13T06:41:13.0+0000 11[MGR] checkin and destroy of IKE_SA successful
```

#7 - 13.02.2020 10:53 - Tobias Brunner

> So does this mean that one has to wait for this phase to complete before delete event gets handled? Can't it be queued and delete later?

The DPD task (INFORMATIONAL) is active so the delete is queued. Then when the maximum retransmits for the active exchange are reached before the timeout for the terminate command, the SA is recreated while the queued delete task is ignored. I described this in my last comment.

> The issue here is that if delete is skipped, the tunnel gets reestablished which is not supposed to happen as it caused other problems in our datapath.

Not sure what problems you refer to, or why you use a 1 second timeout for the terminate command in the first place if you want to force it (within 1 second there wouldn't be a retransmit of the DELETE anyway with your retransmission settings).

> What if DPD timeout is more (say > 100s)?

I guess you mean DPD delay (DPD timeout is irrelevant for IKEv2). It will reduce the likelihood of a conflict between DPD and termination (depending on when the latter is triggered) as fewer DPD tasks are queued. But again, if you terminate while the DPD task (or any other exchange really) is active and the maximum retransmissions are reached within the timeout for the terminate command, you'd run into the same issue again.

> Waiting for this long and retrying will block other events in our case.

What do you mean?

Anyway, I pushed a commit to the *3335-ike-delete* branch, which checks if a delete task for the IKE_SA is queued before reestablishing the IKE_SA. That should avoid your issue.

#8 - 18.02.2020 13:00 - Krishnamurthy Daulatabad

Yes this fix, deletes the SA queued before the re-establishment.

Thanks for the fix

#9 - 21.02.2020 10:41 - Tobias Brunner

- Tracker changed from Issue to Bug

- Subject changed from VICI terminate failure to IKE_SA termination request ignored when reestablishing

- Category changed from vici to libcharon

- Status changed from Feedback to Closed

- Assignee set to Tobias Brunner

- Target version set to 5.8.3

- Resolution set to Fixed

Thanks for testing. The fix is now in master.