strongSwan - Bug #418
keyingtries is ignored with rekey=no
25.09.2013 18:01 - c b

Status: Closed
Priority: Normal
Assignee: Tobias Brunner
Category: starter
Target version: 5.1.1
Affected version: 5.1.0
Resolution: Fixed

Description
I was testing a monitoring script, and noticed it would get stuck when running 'ipsec up xxx' on a connection which is simply not responding.
I could not find any configuration that changes this behavior. keyingtries does not affect it. And it seems like there is supposed to be some way to set it because this shows up in output with things like (2/0 (3/0) as if to say "attempt 2 out of 0":

root@ip-10.10.10.10:/home/chrisb# ipsec up VPN-CONN-NAME
initiating Main Mode IKE_SA VPN-CONN-NAME[27] to 1.2.3.4
generating ID_PROT request 0 [ SA V V V V ]
sending packet: from 10.10.10.10[500] to 1.2.3.4[500] (188 bytes)
sending retransmit 1 of request message ID 0, seq 1
sending packet: from 10.10.10.10[500] to 1.2.3.4[500] (188 bytes)
sending retransmit 2 of request message ID 0, seq 1
sending packet: from 10.10.10.10[500] to 1.2.3.4[500] (188 bytes)
sending retransmit 3 of request message ID 0, seq 1
sending packet: from 10.10.10.10[500] to 1.2.3.4[500] (188 bytes)
sending retransmit 4 of request message ID 0, seq 1
sending packet: from 10.10.10.10[500] to 1.2.3.4[500] (188 bytes)
sending retransmit 5 of request message ID 0, seq 1
sending packet: from 10.10.10.10[500] to 1.2.3.4[500] (188 bytes)
giving up after 5 retransmits
peer not responding, trying again (2/0)
initiating Main Mode IKE_SA VPN-CONN-NAME[27] to 1.2.3.4
generating ID_PROT request 0 [ SA V V V V ]
sending packet: from 10.10.10.10[500] to 1.2.3.4[500] (188 bytes)
sending retransmit 1 of request message ID 0, seq 1
sending packet: from 10.10.10.10[500] to 1.2.3.4[500] (188 bytes)
sending retransmit 2 of request message ID 0, seq 1
sending packet: from 10.10.10.10[500] to 1.2.3.4[500] (188 bytes)
sending retransmit 3 of request message ID 0, seq 1
sending packet: from 10.10.10.10[500] to 1.2.3.4[500] (188 bytes)
sending retransmit 4 of request message ID 0, seq 1
sending packet: from 10.10.10.10[500] to 1.2.3.4[500] (188 bytes)
sending retransmit 5 of request message ID 0, seq 1
sending packet: from 10.10.10.10[500] to 1.2.3.4[500] (188 bytes)
giving up after 5 retransmits
peer not responding, trying again (3/0)
initiating Main Mode IKE_SA VPN-CONN-NAME[27] to 1.2.3.4
generating ID_PROT request 0 [ SA V V V V ]
sending packet: from 10.10.10.10[500] to 1.2.3.4[500] (188 bytes)
sending retransmit 1 of request message ID 0, seq 1
....

Associated revisions
Revision a2cebbe6 - 26.09.2013 10:17 - Tobias Brunner
starter: Don't ignore keyingtries with rekey=no

Since keyingtries also affects the number of retries initially or when reestablishing an SA it should not be affected by the rekey option.

Fixes #418.

20.03.2020
I was testing a monitoring script, and noticed it would get stuck when running 'ipsec up xxx' on a connection which is simply not responding.

You could use ipsec stroke up-nb xxx to start a connection without blocking.

I could not find any configuration that changes this behavior. keyingtries does not affect it. And it seems like there is supposed to be some way to set it because this shows up in output with things like (2/0 (3/0) as if to say "attempt 2 out of 0"

Yes that output is exactly due to keyingtries. If you set keyingtries=%forever, which internally equals 0, you'd get exactly that output. To stop right after the first failure you'd have to set keyingtries=1 (the default is 3).

That is definitely not the case. I don't have keyingtries defined, so hence this default of 3 is not in place. Then I also set it specifically to 1 or 2 and it still did not make a difference. to implement the change I run ipsec update. then ipsec down on the connection, and then ipsec up.

I will re-test that though.. it's in my conn default section.. maybe I'll try it in the specific section.

But also another thing I'm about to test.. I didn't mention that I have auto=route.. if that makes a difference. However at least with auto=route, my script can just do ipsec down and then test traffic to cause it to come up.

Same results. keyingtries in the specific section or default seems to be totally ignored for the initial conn setup. However, keeping in mind that I'm only issuing 'ipsec update' and I'm not doing 'ipsec restart' because I don't want to affect the other tunnels.. so perhaps if the bug is not ignoring it, the bug is in updating that setting during an update?

I also tested with auto=start and had same results as with auto=route. auto=route, BTW, infinitely more reliable than auto=start for us.. because when the 1st request comes, it will wait for the conn to come up rather than aborting immediately if it happens to be down.

And, as mentioned, I can at least issue 'ipsec down' and then have a traffic test bring it up.

ipsec down and then testing traffic does not work. :) It seems that even though it's still listed as ROUTED, testing traffic after sending 'ipsec down' does not cause it to do anything.

I have to do this, which does work:
ipsec down xx
ipsec unroute xx
ipsec route xx

and then, testing traffic will cause it to want to come up.

Same results. keyingtries in the specific section or default seems to be totally ignored for the initial conn setup. However, keeping in mind that I'm only issuing 'ipsec update' and I'm not doing 'ipsec restart' because I don't want to affect the other tunnels.. so perhaps if the bug is not ignoring it, the bug is in updating that setting during an update?

I can't reproduce this. Could you attach your complete ipsec.conf file (and all included files)? Did you build strongSwan from the 5.1.0 tarball? No patches?

ipsec down and then testing traffic does not work. :) It seems that even though it's still listed as ROUTED, testing traffic after sending 'ipsec down' does not cause it to do anything.

That's probably due to the kernel. When traffic matches an IPsec policy for which no IPsec SA is known then the kernel creates a temporary SA and
sends an acquire to the keying daemon (i.e. strongSwan), so that such an SA gets established. This SA has a lifetime of
/proc/sys/net/core/xfrm_acq_expires, which we set to 165 seconds on startup (the default retransmission timeout). While that SA is there, the kernel
does not send any further acquires. When you do ipsec unroute xxx; ipsec route xxx, the new policy gets a different reqid, so the kernel will once
again create a reqid for new traffic (at that point multiple temporary SAs for the same traffic could be there). If a conn section did actually change
ipsec update will automatically unroute and route that connection.

#6 - 25.09.2013 21:33 - c b
I wanted to do more testing rather than filter out sensitive stuff in the config file.. we have about 15 connections configured, and will get up to about
50.

And fortunately I may have saved you some time also. I deduced that if a connection has 'rekey=no' then the 'keyingtries' setting is ignored and set to
0. That might be the "bug" then. Because from my point of view, 'rekey=no' means don't renegotiate the key. While 'keyingtries' indicates the
attempts to renegotiate if we are renegotiating. But of course, I did not think 'keyingtries' also controlled how many initial connection attempts were
made. I was presuming those were 2 separate actions.

When I removed the 'rekey=no' from that connection's configuration, the 'keyingtries' value indeed controlled how many times it would try when
issuing 'ipsec up'.

#7 - 26.09.2013 11:21 - Tobias Brunner
- Tracker changed from Issue to Bug
- Subject changed from ipsec up tries indefinitely if remote does not respond to keyingtries is ignored with rekey=no
- Category set to starter
- Status changed from Feedback to Closed
- Target version set to 5.1.1
- Resolution set to Fixed

Good catch! That really seems like a strange limitation. The associated commit fixes this issue.

#8 - 26.09.2013 16:08 - c b
Super.

I have identified something else which also makes for a difficult scripting life.
This is very hard to duplicate, but the fact that it sometimes happens kills the connection it happens to until someone kicks it.

I just ran this command line, which I had in my script until this happened:
ipsec down VPN-CONN-NAME; ipsec unroute VPN-CONN-NAME; ipsec route VPN-CONN-NAME

And this was the output:

closing CHILD_SA VPN-CONN-NAME{9} with SPIs c6445126_i (700 bytes) d1308ab4_o (1148 bytes) and TS 1.1.1.1/32 ==
10.1.1.1/32
IKE_SA [13] closed successfully
configuration 'VPN-CONN-NAME' not found
'VPN-CONN-NAME' routed

No one was doing anything else to the ipsec.conf file or running any other 'ipsec' command. Now in this case, it did not harm it because in the end, it
was routed. However the time I discovered it from the script, it managed to 'unroute' but when it was time to 'route' that's when it returned the error
that it could not find the configuration.

Since these commands are blocking, I would not expect to have to sleep in between, if that's the problem.
I have repeated this several times after, and it did not happen... so far it happened twice. Also with the sleep it hasn't happened, but no clue if it would
later or not.

This is the normal output I'd always expect:

closing CHILD_SA VPN-CONN-NAME{9} with SPIs c6445126_i (700 bytes) d1308ab4_o (1148 bytes) and TS 1.1.1.1/32 ==
10.1.1.1/32
configuration 'VPN-CONN-NAME' unrouted
'VPN-CONN-NAME' routed

log data:

Sep 26 08:52:24 ip-10-200-101-11 charon: 01[iKE] closing CHILD_SA VPN-CONN-NAME{9} with SPIs c6445126_i (700 bytes) d1308ab4_o (1148 bytes) and TS OUR-PEER/32 == THEIR-LAN-IP/32

20.03.2020 3/6
configuration 'VPN-CONN-NAME' not found

This is logged if the connection was not routed (i.e. no routed connection with that name was found). Did you call those commands after changing the connection and running ipsec update? As I mentioned earlier ipsec update does also unroute and route changed connections, and ipsec update is an asynchronous call, i.e. non-blocking.

However the time I discovered it from the script, it managed to 'unroute' but when it was time to 'route' that's when it returned the error that it could not find the configuration.

This sounds like a race condition as well. The reason could be the same as above, a call to ipsec update before that command sequence.

Yes. I can repeat it by throwing "ipsec update" into the mix.

So then followup is how can I use 'ipsec' to determine if a connection is routed? Because the standard 'ipsec statusxxx' may indicate "ROUTED" in there but it might not actually be.

Yes. I can repeat it by throwing "ipsec update" into the mix.

Let me just repeat what I said before: ipsec update does call unroute and route if required (i.e. if the config has changed). There is no need for you to do so after calling ipsec update! ipsec down on the other hand is not called so that may be something you want to call depending on your requirements.

So then followup is how can I use 'ipsec' to determine if a connection is routed? Because the standard 'ipsec statusxxx' may indicate "ROUTED" in there but it might not actually be.

Well, due to the multi-threaded nature of the daemon the output of ipsec status is just a snapshot, so whatever is listed there is only true for that exact moment in time.

Yes.. thank you :) I was merely confirming your repeated comments about 'ipsec update' explain what I saw.

Ah, I see. Sorry about that :)

May I propose that there might be something going wrong in the ipsec update sequence? :) Which has helped cause my confusions.

In summary, it seems that ipsec update calls unroute if the connection is ROUTED, and does not call a route afterwards. But if the connection is NOT ROUTED, then it only calls route.

I would expect the correct behavior should be, that if the conn is ROUTED, and it decides to unroute that it should also route after it un-routes. :) Otherwise, the end state is FLIPPED based on the beginning state, rather than always resulting in the same state.

Let's start with the situation where ipsec statusall xx returns to indicate that a conn is ROUTED and also ESTABLISHED. If I call ipsec update nothing happens... expected.

If I make a change to that conn config (in my example I added rekey=no) and then run ipsec update, the statusall for that conn indicates that the conn
is still ESTABLISHED but NO LONGER ROUTED.
Subsequent ipsec update calls change nothing.
If I run ipsec down xx, then statusall of course indicates the connection is not listed as up, but still also NOT ROUTED.
Hence, followup traffic requests fail.

Now.. if I make another change to that config, say commenting out the rekey=no, then run ipsec update, that connection ends up being ROUTED.
And so traffic coming will bring it up.

So this causes issues.. because when we run ipsec update ALL tunnels which had configs changed and that are ROUTED, end up NOT ROUTED, and all tunnels that were NOT ROUTED, end up ROUTED.
EDIT: Actually I've just whitnessed that a conn which was not changed, and was ROUTED, no longer is ROUTED.
And another call to ipsec update without changing anything, brought only that handful back as ROUTED. Still, 100% of them should be ROUTED but only 3 of them are at this moment.

#15 - 26.09.2013 19:12 - c b
I think I'm on to something.

When ipsec update detects a change, and it's a conn that's configured with auto=route, as all mine are because it's the default, it has a problem in actually removing the routes from iptables (I can still see them using the command ip xfrm policy list dst THEIR-PROTECTED even though ipsec does not indicate it's ROUTED). So errors show up in the logs (at end of this comment). Subsequent config change and ipsec update actually in this case did not fix, and also ipsec unrout returns that configuration is "not found", but ipsec route throws errors in the log.

Only after doing ipsec down xx; ipsec unrout xx; ipsec route xx does the end state come back to normal.

Here is more data.

```
ipsec route VPN-CONN-NAME
routing 'VPN-CONN-NAME' failed
/32 out (mark 0/0x00000000) for reqid 52, the same policy for reqid 2 exists
/32 in (mark0/0x00000000) for reqid 52, the same policy for reqid 2 exists
/32 fwd (mark 0/0x00000000) for reqid 52, the same policy for reqid 2 exists
```

#16 - 26.09.2013 19:15 - Tobias Brunner
Ah, yes. That's a duplicate issue, see #400. You may try the two patches in the route-reqid branch.

#17 - 26.09.2013 19:50 - c b
Simplified observation.

This causes problem if a conn is currently ROUTED and ESTABLISHED:
ipsec unrout xx
This removes the ROUTED from ipsec point of view, but does NOT remove the iptables rules. You can still see them.

The connection has to first be brought down before you can issue ipsec unrout xx and have it work completely.

oh.. something else.

If a connection is currently NOT ROUTED and DOWN, ipsec up xx will bring up the connection, add the iptables rules, but leave the connection marked as "NOT ROUTED" meaning that it does not show up as ROUTED.

This also then causes confusion for ipsec update and the route/unroute commands.

#19 - 27.09.2013 11:10 - Tobias Brunner
This causes problem if a conn is currently ROUTED and ESTABLISHED:
  ipsec unroutexx
This removes the ROUTED from ipsec point of view, but does NOT remove the iptables rules. You can still see them.

Yes, because those IPsec policies (not iptables rules) are still used by the active IPsec tunnel.

  The connection has to first be brought down before you can issue ipsec unroutexx and have it work completely.

The other way around should work too (i.e. ipsec down after ipsec unroutex). To work around the bug described in #400 it is simply important that the connection is not established when ipsec route is called (i.e. ipsec down before ipsec route is the important bit).

  If a connection is currently NOT ROUTED and DOWN, ipsec up xx will bring up the connection, add the iptables rules, but leave the connection marked as "NOT ROUTED" meaning that it does not show up as ROUTED.

Connections listed under ROUTED are trap policies explicitly created with auto=route or ipsec route. IPsec policies installed for established connections are not listed there.

Also, starter does not refer to the current state of a connection, it simply uses the information in the config file. So if the existing config has auto=route it tries to unroutex, if the updated config has auto=route it routes it. Any manual changes via ipsec route|unroute are not considered. Which I think makes sense, but is currently not working properly due to #400 if a connection is already established.