strongSwan - Issue #431

rekeying of tunnels sometimes fail with "unable to install policy ... for reqid Y, the same policy for reqid X exists"

11.10.2013 17:34 - Wolfgang Walter

Status:	Closed		
Priority:	Normal		
Assignee:	Tobias Brunner		
Category:	charon		
Affected version:	5.1.0	**Resolution:**	Fixed

Description

After some time - usually a day, ipsec-tunnels fail.

In the log I find:

charon: 15[CFG] unable to install policy xxxxx/29 === yyyyy/23 out (mark 0/0x00000000) for reqid X, the same policy for reqid Y exists

It seems that this starts when if a new IKE_SA is negotiated:

charon: 13[IKE] IKE_SA deleted
charon: 14[IKE] aaaaaaa is initiating an IKE_SA
charon: 11[CFG] looking for peer configs matching ....
charon: 11[CFG] selected peer config ....
charon: 11[CFG] using trusted certificate ....
charon: 11[IKE] IKE_SA .... established between bbbbbb ... ...aaaaaaa
charon: 11[CFG] unable to install policy ....

restarting ipsec fixes the issue (it doesn't matter on which side)

Associated revisions

Revision 85489d39 - 20.02.2015 11:07 - Martin Willi

Merge branch 'reqid-alloc'

With these changes, charon dynamically allocates reqids for CHILD_SAs. This
allows the reuse of reqids for identical policies, and basically allows multiple
CHILD_SAs with the same selectors. As reqids do not uniquely define a CHILD_SA,
a new unique identifier for CHILD_SAs is introduced, and the kernel backends
use a proto/dst/SPI tuple to identify CHILD_SAs.

charon-tkm is not yet updated and expires are actually broken with this merge.
As some significant refactorings are required, this is fixed using a separate
merge.

References #422, #431, #463.

Revision 94eb09ac - 20.02.2015 13:34 - Martin Willi

Merge branch 'reqid-alloc'

With these changes, charon dynamically allocates reqids for CHILD_SAs. This

allows the reuse of reqids for identical policies, and basically allows multiple
CHILD_SAs with the same selectors. As reqids do not uniquely define a CHILD_SA,
a new unique identifier for CHILD_SAs is introduced, and the kernel backends
use a proto/dst/SPI tuple to identify CHILD_SAs.

charon-tkm is not yet updated and expires are actually broken with this merge.
As some significant refactorings are required, this is fixed using a separate
merge.

References #422, #431, #463.

History

#1 - 14.10.2013 12:55 - Tobias Brunner

- Tracker changed from Bug to Issue

- Status changed from New to Feedback

- Assignee set to Tobias Brunner

A bit more context would help. Like your ipsec.conf files and longer excerpts of the log files (of both peers).

#2 - 17.10.2013 10:14 - Tobias Brunner

According to the log files you sent me it happens during the reauthentication of an IKE_SA with lots of CHILD_SAs (IPsec tunnels).

During a reauthentication the previous IKE_SA and implicitly all CHILD_SAs are deleted and recreated from scratch.

My assumption is that there is a race condition between the thread that deletes all the IPsec policies of the old CHILD_SAs and the threads that install the policies for the newly created CHILD_SAs. I did some tests and they confirm the possibility of such race conditions, although, the deletion of IPsec policies has to be delayed quite a bit.

So if you don't really need reauthentication (e.g. to verify the revocation state of certificates) then you should try to set *reauth=no* in your configs as a workaround. Then the IKE_SA will be rekeyed inline without affecting the established CHILD_SAs.

#3 - 21.10.2013 17:46 - Wolfgang Walter

Yes, I think it is a race.

The workaround with reauth=no works and is good enough for us. I'm happy to test a fix, though, once you have one.

#4 - 14.02.2014 16:31 - John F

I have experienced this frustrating issue too.

Steps to reproduce:

1. There's a tunnel between HostA and HostB. Reboot host A.
2. After Linux has finished booting, the tunnel is not brought up automatically.
3. I cannot bring the HostA-HostB tunnel up manually either by doing an **ipsec up AB**. Here's what I get:

```
2014-02-02T13:10:18.659519+00:00 HostA charon: [info] 14[CHD] adding inbound ESP SA
2014-02-02T13:10:18.659524+00:00 HostA charon: [info] 14[CHD]   SPI 0xc0b6564f, src 4.34.144.70 dst 219.144.14
```

```
5.80
2014-02-02T13:10:18.659538+00:00 HostA charon: [info] 14[CHD] adding outbound ESP SA
2014-02-02T13:10:18.659543+00:00 HostA charon: [info] 14[CHD]   SPI 0xc7b2c477, src 217.144.145.70 dst 54.34.1
44.80
2014-02-02T13:10:18.659593+00:00 HostA charon: [info] 14[CFG] unable to install policy 10.2.0.0/15 === 10.10.0
.0/15 out (mark 0/0x00000000) for reqid 138, the same policy for reqid 122 exists
2014-02-02T13:10:18.659618+00:00 HostA charon: [info] 14[CFG] unable to install policy 10.10.0.0/15 === 10.2.0
.0/15 in (mark 0/0x00000000) for reqid 138, the same policy for reqid 122 exists
2014-02-02T13:10:18.659638+00:00 HostA charon: [info] 14[CFG] unable to install policy 10.10.0.0/15 === 10.2.0
.0/15 fwd (mark 0/0x00000000) for reqid 138, the same policy for reqid 122 exists
2014-02-02T13:10:18.659657+00:00 HostA charon: [info] 14[CFG] unable to install policy 10.2.0.0/15 === 10.10.0
.0/15 out (mark 0/0x00000000) for reqid 138, the same policy for reqid 122 exists
2014-02-02T13:10:18.659677+00:00 HostA charon: [info] 14[CFG] unable to install policy 10.10.0.0/15 === 10.2.0
.0/15 in (mark 0/0x00000000) for reqid 138, the same policy for reqid 122 exists
2014-02-02T13:10:18.659697+00:00 HostA charon: [info] 14[CFG] unable to install policy 10.10.0.0/15 === 10.2.0
.0/15 fwd (mark 0/0x00000000) for reqid 138, the same policy for reqid 122 exists
2014-02-02T13:10:18.659711+00:00 HostA charon: [info] 14[IKE] unable to install IPsec policies (SPD) in kernel
2014-02-02T13:10:18.659730+00:00 HostA charon: [info] 14[IKE] failed to establish CHILD_SA, keeping IKE_SA
2014-02-02T13:10:18.659790+00:00 HostA charon: [info] 14[KNL] deleting policy 10.2.0.0/15 === 10.10.0.0/15 out
 failed, not found
2014-02-02T13:10:18.659806+00:00 HostA charon: [info] 14[KNL] deleting policy 10.10.0.0/15 === 10.2.0.0/15 in
failed, not found
2014-02-02T13:10:18.659829+00:00 HostA charon: [info] 14[KNL] deleting policy 10.10.0.0/15 === 10.2.0.0/15 fwd
 failed, not found
2014-02-02T13:10:18.659844+00:00 HostA charon: [info] 14[KNL] deleting policy 10.2.0.0/15 === 10.10.0.0/15 out
 failed, not found
2014-02-02T13:10:18.659862+00:00 HostA charon: [info] 14[KNL] deleting policy 10.10.0.0/15 === 10.2.0.0/15 in
failed, not found
2014-02-02T13:10:18.659881+00:00 HostA charon: [info] 14[KNL] deleting policy 10.10.0.0/15 === 10.2.0.0/15 fwd
 failed, not found
2014-02-02T13:10:18.659922+00:00 HostA charon: [info] 14[ENC] generating IKE_AUTH response 1 [ IDr CERT AUTH N
(AUTH_LFT) N(TS_UNACCEPT) ]
```

4. No matter how many times I down and up the tunnel, the same error appears. It fails from both sides. An **ipsec restart** gives the same result! The only workaround I found to bring the tunnel up in a timely manner was the following:

```
HostA ~ # ipsec down AB
deleting IKE_SA g01[20] between 219.144.145.80[HostA]...54.34.144.80[HostB]
sending DELETE for IKE_SA g01[20]
generating INFORMATIONAL request 0 [ D ]
sending packet: from 219.144.145.80[500] to 54.34.144.80[500] (76 bytes)
received packet: from 54.34.144.80[500] to 219.144.145.80[500] (76 bytes)
parsed INFORMATIONAL response 0 [ ]
IKE_SA deleted
IKE_SA [20] closed successfully
```

The same a second time...

```
HostA ~ # ipsec down AB
deleting IKE_SA g01[12] between 219.144.145.80[HostA]...54.34.144.80[HostB]
sending DELETE for IKE_SA g01[12]
generating INFORMATIONAL request 2 [ D ]
sending packet: from 219.144.145.80[500] to 54.34.144.80[500] (76 bytes)
retransmit 1 of request with message ID 2
sending packet: from 219.144.145.80[500] to 54.34.144.80[500] (76 bytes)

[ here I press Ctrl + C ]

retransmit 2 of request with message ID 2
sending packet: from 219.144.145.80[500] to 54.34.144.80[500] (76 bytes)
```

Then a third time...

```
HostA ~ # ipsec down AB
destroying IKE_SA in state DELETING without notification
IKE_SA [12] closed successfully
```

And then finally the tunnel can be brought up successfully...

```
HostA ~ # ipsec up g01
initiating IKE_SA g01[21] to 54.34.144.80
generating IKE_SA_INIT request 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) ]
sending packet: from 219.144.145.80[500] to 54.34.144.80[500] (708 bytes)
received packet: from 54.34.144.80[500] to 219.144.145.80[500] (465 bytes)
parsed IKE_SA_INIT response 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) CERTREQ N(MULT_AUTH) ]
received cert request for "C=GB, O=strongSwan, CN=strongSwan CA"
sending cert request for "C=GB, O=strongSwan, CN=strongSwan CA"
authentication of 'HostA' (myself) with RSA signature successful
sending end entity cert "C=GB, O=strongSwan, CN=HostA"
establishing CHILD_SA g01
generating IKE_AUTH request 1 [ IDi CERT CERTREQ IDr AUTH SA TSi TSr N(MULT_AUTH) N(EAP_ONLY) ]
sending packet: from 219.144.145.80[500] to 54.34.144.80[500] (1516 bytes)
received packet: from 54.34.144.80[500] to 219.144.145.80[500] (1324 bytes)
parsed IKE_AUTH response 1 [ IDr CERT AUTH SA TSi TSr N(AUTH_LFT) ]
received end entity cert "C=GB, O=strongSwan, CN=HostB"
  using certificate "C=GB, O=strongSwan, CN=HostB"
  using trusted ca certificate "C=GB, O=strongSwan, CN=strongSwan CA"
checking certificate status of "C=GB, O=strongSwan, CN=HostB"
  using trusted certificate "C=GB, O=strongSwan, CN=strongSwan CA"
  crl correctly signed by "C=GB, O=strongSwan, CN=strongSwan CA"
  crl is stale: since Aug 22 12:10:37 2013
  fetching crl from 'file:///etc/ourcompany.crl' ...
  using trusted certificate "C=GB, O=strongSwan, CN=strongSwan CA"
  crl correctly signed by "C=GB, O=strongSwan, CN=strongSwan CA"
  crl #01 is not newer - existing crl #01 retained
certificate status is unknown, crl is stale
  reached self-signed root ca with a path length of 0
authentication of 'HostB' with RSA signature successful
IKE_SA g01[21] established between 219.144.145.80[HostA]...54.34.144.80[HostB]
scheduling reauthentication in 9827s
maximum IKE_SA lifetime 10367s
CHILD_SA g01{23} established with SPIs c26e6429_i ca72313c_o and TS 10.14.0.0/15 === 10.10.0.0/15
connection 'g01' established successfully
```

This bug needs to be addressed as this seriously affects the reliability of this VPN solution.
The strongSwan version in use is the 5.1.1

#5 - 19.02.2014 18:21 - Tobias Brunner

1. There's a tunnel between HostA and HostB. Reboot host A.

Rebooting HostA doesn't cause a proper deletion of the existing IKE_SA?

2. After Linux has finished booting, the tunnel is not brought up automatically.

What are your DPD settings? Why would you expect it to be brought up? Did you configure *auto=start* or *auto=route*? Or *dpdaction=restart* on HostB?

3. I cannot bring the HostA-HostB tunnel up manually either by doing an **ipsec up AB**.

Apparently, the old SA still exists on HostB, configuring DPD settings there might help (depending on how long it takes for HostA to reboot).

When HostA reconnects, the old IKE_SA is deleted/replaced due to *uniqueids=yes* (the default, unless you changed it - in which case the error you are seeing is expected, as there would be duplicate IPsec policies).
The problem: The SA is deleted using a DELETE payload within an INFORMATIONAL exchange sent under the old IKE_SA - but HostA has no knowledge of this SA, so it doesn't respond. Therefore, the regular retransmission timeouts apply, that is, it takes 2.75 minutes, by default, until the SA is finally destroyed and the installed IPsec policies are removed. Until that happens the new policies are declined with the observed error message.

One improvement in this situation might be to uninstall the IPsec policies right when sending the DELETE (currently this is done after a response has been received), or at least to allow duplicate IPsec policies, if the associated IKE_SA (or IPsec SA) is currently being deleted. Could be tricky to implement this, though, with the current architecture.

As mentioned in #455, in general, RFC 6290 might help in such a situation. Also, returning INVALID_SPI for the INFORMATIONAL request or DPDs sent by HostB, and adapting the retransmission timeouts upon receipt of such a notify would probably help too (at least it would reduce the time it takes to recover from this situation). But both these things are currently not implemented.

4. No matter how many times I down and up the tunnel, the same error appears. It fails from both sides. An **ipsec restart** gives the same result! The only workaround I found to bring the tunnel up in a timely manner was the following:

Naturally, you can't remove state about the old SAs by calling ipsec down or ipsec restart on HostA (it can only delete known SAs). And ipsec down on HostB will cause it to delete the SA using DELETE payloads, resulting in the same behavior described above (with ipsec restart or stop only one DELETE is sent, so that should be faster - but is obviously undesired if there are other SAs).

#6 - 02.05.2014 14:46 - Mark Thebridge

I'm also hitting this race condition with long-standing tunnels failing when re-authenticated, using strongSwan 5.1.1.

We've integrated strongSwan into our Linux-based telecoms application for securing VoIP signaling with IPsec. We chose to use strongSwan based on its general reputation as a reliable, secure, up-to-date IPsec solution. Overall we've been pleased and impressed - but this one issue is a showstopper for us. Intermittent tunnel failure means this is not deployable in a telecoms environment.

The *reauth=no* workaround suggested in earlier comment 2 is not enough for us, as we need to be able to support IKEv1 as well as IKEv2, where that option is not available.

We're experimenting with adding polling code to our application to spot and recover from this condition, but even if that works we still expect multi-second outages.

Any chance of a fix for the next release?

#7 - 20.05.2014 17:18 - Mark Thebridge

Mark Thebridge wrote:

> I'm also hitting this race condition with long-standing tunnels failing when re-authenticated, using strongSwan 5.1.1.

We've now managed to work around this, with a slightly better understanding of what's going on.

Previously, we were using the auto=start option in ipsec.conf for both ends of the tunnel. We've managed to avoid problems by instead using auto=route, and sending regular pings over the tunnel. If we do hit this race on a reauthentication exchange, then any subsequent traffic sent over the tunnel - or the next ping - will cause a new IKE exchange and so the outage, if any, is very brief.

This is still not ideal, so if any fix might be forthcoming we'd like to take it - but this is good enough for us for now, so posting this in case it helps anyone else hitting this issue!

#8 - 31.10.2014 14:28 - Ilya Ermolin

Hi,

we hit to this issue in corporate lan, wich uses Cisco server with IKEv1.
Charon version: Linux strongSwan U5.2.1rc1/K3.13.0-37-generic

config:

```
config setup
    uniqueids=yes
    charondebug="lib 2"

# Add connections here
```

```
conn cisco
        keyexchange=ikev1
        aggressive=yes
        forceencaps=yes
        dpddelay=15 # Dead peer detection - 30 секунд - интервал между keep-alive пакетами
        dpdtimeout=180 # dpd таймаут 120 секунд, после которого хост будет объявлен недоступным
        dpdaction=restart # перезапустить процесс подключения к хосту
        ike=aes256-sha1-modp1024
        esp=aes256-sha1
        ikelifetime=90000s
        lifetime=1000s
        margintime=1m
        authby=xauthpsk
        xauth=client
        type=tunnel
        left=%defaultroute # OpenSWAN side
        leftid=@er_ipsec
        leftsourceip=%config
        leftauth=psk
        leftauth2=xauth
        xauth_identity=login_name
        right=1.2.0.2 #CISCO side
        rightsubnet=10.68.0.0/16,172.16.0.0/12,192.168.220.0/24,192.168.223.0/24,192.168.226.0/24,10.151.2.0/2
4,10.120.2.0/24,10.62.0.0/16,10.63.0.0/16,192.168.251.0/24,192.168.192.0/24,192.168.195.0/24,10.65.0.0/16,192.
168.227.0/24,192.168.228.0/24,172.20.14.0/23,192.168.120.0/24,192.168.224.0/28,192.168.225.0/24,192.168.250.0/
24,10.61.0.0/16,192.168.98.0/24,192.168.230.0/24,192.168.231.0/25,192.168.152.96/28,10.66.0.0/16,192.168.232.0
/24,172.16.104.0/24,172.30.0.0/24,172.20.3.0/24
        rightauth=psk
        auto=add
```

Log:

```
Oct 31 16:08:06 d00cgiws01 charon: 09[NET] received packet: from 109.207.0.2[4500] to 192.168.6.133[4500] (172
 bytes)
Oct 31 16:08:06 d00cgiws01 charon: 09[ENC] parsed QUICK_MODE request 2406517847 [ HASH SA No ID ID ]
Oct 31 16:08:06 d00cgiws01 charon: 09[IKE] received 2400s lifetime, configured 1000s
Oct 31 16:08:06 d00cgiws01 charon: 09[IKE] received 4608000000 lifebytes, configured 0
Oct 31 16:08:06 d00cgiws01 charon: 09[ENC] generating QUICK_MODE response 2406517847 [ HASH SA No ID ID ]
Oct 31 16:08:06 d00cgiws01 charon: 09[NET] sending packet: from 192.168.6.133[4500] to 109.207.0.2[4500] (188
bytes)
Oct 31 16:08:06 d00cgiws01 charon: 11[NET] received packet: from 109.207.0.2[4500] to 192.168.6.133[4500] (76
bytes)
Oct 31 16:08:06 d00cgiws01 charon: 11[ENC] parsed QUICK_MODE request 2406517847 [ HASH ]
Oct 31 16:08:06 d00cgiws01 charon: 11[CFG] narrowed CHILD_SA to UNITY_SPLIT_INCLUDE 172.16.0.0/12 192.168.220.
0/24 192.168.223.0/24 192.168.226.0/24 10.151.2.0/24 10.120.2.0/24 10.62.0.0/16 10.63.0.0/16 192.168.251.0/24
192.168.192.0/24 192.168.195.0/24 10.65.0.0/16 192.168.227.0/24 192.168.228.0/24 172.20.14.
0/23 192.168.120.0/24 192.168.224.0/28 192.168.225.0/24 192.168.250.0/24 10.61.0.0/16 192.168.98.0/24 192.168.
230.0/24 192.168.231.0/25 192.168.152.96/28 10.66.0.0/16 192.168.232.0/24 172.16.104.0/24 10.68.0.0/16 172.30.
0.0/24 172.20.3.0/24 192.168.224.0/28
Oct 31 16:08:06 d00cgiws01 charon: 11[CFG] unable to install policy 172.16.3.2/32 === 172.16.0.0/12 out (mark
0/0x00000000) for reqid 2, the same policy for reqid 1 exists
Oct 31 16:08:06 d00cgiws01 charon: 11[CFG] unable to install policy 172.16.0.0/12 === 172.16.3.2/32 in (mark 0
/0x00000000) for reqid 2, the same policy for reqid 1 exists
Oct 31 16:08:06 d00cgiws01 charon: 11[CFG] unable to install policy 172.16.0.0/12 === 172.16.3.2/32 fwd (mark
0/0x00000000) for reqid 2, the same policy for reqid 1 exists
Oct 31 16:08:06 d00cgiws01 charon: 11[CFG] unable to install policy 172.16.3.2/32 === 172.16.0.0/12 out (mark
0/0x00000000) for reqid 2, the same policy for reqid 1 exists
Oct 31 16:08:06 d00cgiws01 charon: 11[CFG] unable to install policy 172.16.0.0/12 === 172.16.3.2/32 in (mark 0
/0x00000000) for reqid 2, the same policy for reqid 1 exists
Oct 31 16:08:06 d00cgiws01 charon: 11[CFG] unable to install policy 172.16.0.0/12 === 172.16.3.2/32 fwd (mark
0/0x00000000) for reqid 2, the same policy for reqid 1 exists
Oct 31 16:08:06 d00cgiws01 charon: 11[IKE] unable to install IPsec policies (SPD) in kernel
Oct 31 16:08:06 d00cgiws01 charon: 11[KNL] deleting policy 172.16.3.2/32 === 172.16.0.0/12 out failed, not fou
nd
Oct 31 16:08:06 d00cgiws01 charon: 11[KNL] deleting policy 172.16.0.0/12 === 172.16.3.2/32 in failed, not foun
d
Oct 31 16:08:06 d00cgiws01 charon: 11[KNL] deleting policy 172.16.0.0/12 === 172.16.3.2/32 fwd failed, not fou
nd
```

What warkaround we can use for ikev1?

#9 - 20.02.2015 13:58 - Martin Willi

With the referenced merge of the reqid-alloc branch to master, strongSwan can now handle identical policies by reusing the same reqid. This allows identical CHILD_SAs to the same host, and potentially fixes the issues you see with these conflicts. Feedback welcome.

Regards
Martin

#10 - 06.03.2015 02:54 - richard hu

I did not found reqid-alloc branch on github. where is it?
will it in 5.2.3?

#11 - 06.03.2015 09:09 - Martin Willi

The branch has been merged to master and deleted.

Yes, these changes will be part of the next release, but that will have a 5.3.0 version number.

#12 - 09.03.2015 19:32 - Nick Huanca

Martin Willi wrote:

> With the referenced merge of the reqid-alloc branch to master, strongSwan can now handle identical policies by reusing the same reqid. This allows identical CHILD_SAs to the same host, and potentially fixes the issues you see with these conflicts. Feedback welcome.

Is there an indication that the duplicheck plugin would also avoid this condition? By removing the duplicate SA then the Child_SA wouldn't continually try to associate the routes? (Forgive my understanding of the problem as I'm just starting with StrongSwan).

#13 - 07.07.2015 17:08 - Tobias Brunner

- Status changed from Feedback to Closed

- Resolution set to Fixed