strongSwan - Issue #661

Working site-to-site config in 4.6.2 fails in 5.2.0

27.07.2014 15:25 - Marcel Müller

Status:	Closed		
Priority:	Normal		
Assignee:	Tobias Brunner		
Category:	interoperability		
Affected version:	5.2.0	**Resolution:**	Fixed

Description

Hello,

I'm trying to establish a site-to-site connection between strongSwan and a Fritzbox. This is working fine in 4.6.2, but works only partially in 5.2.0.
Details:
1. strongSwan is initator. Everything is working perfectly, all hosts on network 1 can communicate with all hosts on network 2 and the other way around.
2. Fritzbox is initator. Both show the tunnel as established, but no communication between the networks is possible.

Network

```
172.31.1.5   ----   172.31.0.254    ======== Internet =========    10.90.1.1 - (10.90.1.0/24)
(strongSwan Srv)  (Lancom Router,                                (Fritzbox 7360 6.03,
                fixed IP Internet)                               dyn. IP Internet)
```

ipsec.conf (for 5.2.0)

```
conn 9001
        keyexchange=ikev1
        left=%defaultroute
        leftid=<publicIPLancom>
        leftsubnet=172.31.0.0/16
        #
        ike=aes256-sha1-modp1024
        esp=aes256-sha1-modp1024
        #
        right=<dynDNSFritzbox>
        rightid="@<dynDNSFritzbox>"
        rightsubnet=10.90.1.0/24
        rightallowany=yes
        #
        ikelifetime=1h
        keylife=1h
        #
        authby=secret
        auto=add
```

ipsec.conf (for 4.6.2)

```
config setup
        charonstart=no
        plutostart=yes
        nat_traversal=yes

conn 9001
    keyexchange=ikev1
        left=%defaultroute
        leftid=<publicIPLancom>
        leftsubnet=172.31.0.0/16
```

```
        #
        ike=aes256-sha1-modp1024
        esp=aes256-sha1
        pfs=yes
        pfsgroup=modp1024
        #
        right=<dynDNSFritzbox>
        rightid="@<dynDNSFritzbox>"
        rightsubnet=10.90.1.0/24
        rightallowany=yes
        #
        ikelifetime=1h
        keylife=1h
        #
        authby=secret
        auto=add
```

ipsec status

```
Security Associations (1 up, 0 connecting):
        9001[13]: ESTABLISHED 34 minutes ago, 172.31.1.5[<publicIPLancom>]...<publicIPFritzbox>[<d
ynDNSFritzbox>]
        9001{2}:  INSTALLED, TUNNEL, ESP in UDP SPIs: c25c1685_i a847f1c3_o
        9001{2}:   172.31.0.0/16 === 10.90.1.0/24
```

ip route list table 220

```
10.90.1.0/24 via 172.31.0.254 dev eth0  proto static  src 172.31.1.5
```

ip xfrm policy show

```
src 10.90.1.0/24 dst 172.31.0.0/16
        dir fwd priority 2915 ptype main
        tmpl src <publicIPFritzbox> dst 172.31.1.5
                proto esp reqid 2 mode tunnel
src 10.90.1.0/24 dst 172.31.0.0/16
        dir in priority 2915 ptype main
        tmpl src <publicIPFritzbox> dst 172.31.1.5
                proto esp reqid 2 mode tunnel
src 172.31.0.0/16 dst 10.90.1.0/24
        dir out priority 2915 ptype main
        tmpl src 172.31.1.5 dst <publicIPFritzbox>
                proto esp reqid 2 mode tunnel
src 0.0.0.0/0 dst 0.0.0.0/0
        socket in priority 0 ptype main
src 0.0.0.0/0 dst 0.0.0.0/0
        socket out priority 0 ptype main
src 0.0.0.0/0 dst 0.0.0.0/0
        socket in priority 0 ptype main
src 0.0.0.0/0 dst 0.0.0.0/0
        socket out priority 0 ptype main
src ::/0 dst ::/0
        socket in priority 0 ptype main
src ::/0 dst ::/0
        socket out priority 0 ptype main
src ::/0 dst ::/0
        socket in priority 0 ptype main
src ::/0 dst ::/0
        socket out priority 0 ptype main
```

ip xfrm state show

```
src 172.31.1.5 dst <publicIPFritzbox>
```

```
        proto esp spi 0xc5f26cb4 reqid 2 mode tunnel
        replay-window 32 flag af-unspec
        auth-trunc hmac(sha1) 0x262.... 96
        enc cbc(aes) 0x2d9e7....
        encap type espinudp sport 4500 dport 4500 addr 0.0.0.0
src <publicIPFritzbox> dst 172.31.1.5
        proto esp spi 0xc239f583 reqid 2 mode tunnel
        replay-window 32 flag af-unspec
        auth-trunc hmac(sha1) 0xe.... 96
        enc cbc(aes) 0x38....
        encap type espinudp sport 4500 dport 4500 addr 0.0.0.0
```

Is this a bug in 5.2.0? Is there a way I can further debug this issue?

Thanks in advance!

Associated revisions

Revision 84337ac8 - 12.09.2014 13:55 - Tobias Brunner

ikev1: Properly handle different proposal numbering schemes

While the examples in RFC 2408 show proposal numbers starting at 1 and increasing by one for each subsequent proposal this is not mandatory. Actually, IKEv1 proposals may start at any number, the only requirement is that the proposal numbers increase monotonically they don't have to do so consecutively.

Most implementations follow the examples and start numbering at 1 (charon, racoon, Shrew, Cisco, Windows XP, FRITZ!Box) but pluto was one of the implementations that started with 0 and there might be others out there.

The previous assumption that implementations always start numbering proposals at 0 caused problems with clients that start numbering with 1 and whose first proposal consists of multiple protocols (e.g. ESP+IPComp).

Fixes #661.

Revision 2f7fef56 - 12.09.2014 13:56 - Tobias Brunner

ikev1: Skip unusable IPComp proposals

Fixes #661.

Revision 8a6b01dd - 12.09.2014 13:56 - Tobias Brunner

ikev1: Don't cache last block of INFORMATIONAL messages as IV

We don't expect a response with the same MID, but apparently some devices (e.g. FRITZ!Box) do that for DPDs, while still treating the response as a new exchange. By storing the last message block as IV we can't decrypt the first block of such a response.

Fixes #661.

History

#1 - 28.07.2014 11:21 - Tobias Brunner

- Category set to interoperability

- Status changed from New to Feedback

- Assignee set to Tobias Brunner

> Is this a bug in 5.2.0? Is there a way I can further debug this issue?

Hard to tell. Is the output of the various status commands you posted different when one or the other is initiator? The log of the two scenarios might help.

#2 - 28.07.2014 12:17 - Marcel Müller

Tobias Brunner wrote:

> Is the output of the various status commands you posted different when one or the other is initiator? The log of the two scenarios might help.

Looks absolutely identical to me. I'll install 5.2.0 again and get the status outputs in detail. Should I configure any detailed logging or is the default /var/log/syslog enough? Do you need /var/log/auth.log as well?

Thanks!

#3 - 28.07.2014 12:24 - Tobias Brunner

> Should I configure any detailed logging or is the default /var/log/syslog enough? Do you need /var/log/auth.log as well?

You could increase the default to 3 but set the level for the *asn*, *enc* and *job* subsystems to 1. And no, auth.log is not required.

To create separate log files for each run it might be easier to log to a specific file (again see LoggerConfiguration).

#4 - 28.07.2014 14:52 - Tobias Brunner

Thanks for the logs and status outputs. As far as the strongSwan side is concerned there is apparently no difference between the two scenarios. So I guess it's the FRITZ!Box that behaves differently when it is initiating the connection. The question is how and what might trigger it. Do you have logs and/or status output of the FRITZ!Box?

#5 - 28.07.2014 15:24 - Marcel Müller

Tobias Brunner wrote:

> Do you have logs and/or status output of the FRITZ!Box?

Sent the logs!
Please note that there is an issue after which the Fritzbox refuses to accept new incoming connection after closing an IPSec session from strongSwan (ipsec down 9001). At this point avmike logs “new_neighbour_template failed”. After adding/removing the VPN (and therefore restarting avmike) everything starts working fine again for incoming connection. I think (!) that this is an issue only AVM fix (?! Is strongSwan doing something “special” on a close?) but is independent of this issue.

#6 - 28.07.2014 23:26 - Marcel Müller

Noticed something in the logs of the Fritzbox. When the fritzbox connects to strongSwan 5.2.0 it says that the remote peer does support NAT-T RFC 3947. It does not say that when using 4.6.2:

5.2.0

```
2014-07-28 12:32:19 avmike:mainmode Zentrale: selected lifetime: 3600 sec(no notify)
  2014-07-28 12:32:19 avmike:Zentrale remote peer supported XAUTH
  2014-07-28 12:32:19 avmike:Zentrale remote peer supported DPD
  2014-07-28 12:32:19 avmike:Zentrale remote peer supported NAT-T RFC 3947
2014-07-28 12:32:20 avmike:mainmode Zentrale: add SA 12
2014-07-28 12:32:20 avmike:Zentrale: switching to NAT-T (Initiator)
2014-07-28 12:32:20 avmike:Zentrale: Warning: source changed from <publicIPLancom>:500 to <publicIPLancom>:450
0
2014-07-28 12:32:20 avmike:Zentrale: Phase 1 ready
2014-07-28 12:32:20 avmike:Zentrale: local is behind a nat
2014-07-28 12:32:20 avmike:Zentrale: remote is behind a nat
2014-07-28 12:32:20 avmike:Zentrale: start waiting connections
2014-07-28 12:32:20 avmike:Zentrale: Phase 2 starting (start waiting)
2014-07-28 12:32:20 avmike:Zentrale: Phase 2 ready
2014-07-28 12:32:20 avmike:< cb_sa_created(name=Zentrale,id=12,...,flags=0x00032101)
2014-07-28 12:32:20 avmike:Zentrale: start waiting connections
2014-07-28 12:32:20 avmike:Zentrale: NO waiting connections
2014-07-28 12:32:30 avmike:>>>4500 nat-t-keepalive[<publicIPLancom>:4500]
2014-07-28 12:32:38 avmike:Zentrale
 fehlerhafte Paketlaenge: Hdr-length > read-Data
2014-07-28 12:32:38 avmike:FreeIPsecSA: spi=c5fb5855        protocol=3 iotype=2
2014-07-28 12:32:38 avmike:< cb_sa_deleted(name=Zentrale,id=12,what=2)
2014-07-28 12:32:38 avmike:Zentrale
 fehlerhafte Paketlaenge: Hdr-length > read-Data
2014-07-28 12:32:38 avmike:mainmode Zentrale: del SA 12
2014-07-28 12:32:38 avmike:wolke_del_neighbour_sa_by_remote: no SAs available, set canceld = TRUE
2014-07-28 12:32:38 avmike:< cb_sa_deleted(name=Zentrale,id=12,what=3)
```

```
2014-07-28 12:32:38 avmike:FreeIPsecSA: spi=fe7fe494        protocol=3 iotype=1
```

4.6.2

```
2014-07-28 23:15:54 avmike:Zentrale: Warning: source changed from <publicIPLancom>:500 to <publicIPLancom>:606
58
2014-07-28 23:15:54 avmike:mainmode Zentrale: selected lifetime: 3600 sec(no notify)
  2014-07-28 23:15:54 avmike:Zentrale remote peer supported XAUTH
  2014-07-28 23:15:54 avmike:Zentrale remote peer supported DPD
2014-07-28 23:15:54 avmike:Zentrale: Warning: source changed from <publicIPLancom>:500 to <publicIPLancom>:606
58
2014-07-28 23:15:55 avmike:mainmode Zentrale: add SA 1
2014-07-28 23:15:55 avmike:Zentrale: Warning: source changed from <publicIPLancom>:500 to <publicIPLancom>:606
58
2014-07-28 23:15:55 avmike:Zentrale: Phase 1 ready
2014-07-28 23:15:55 avmike:Zentrale: start waiting connections
2014-07-28 23:15:55 avmike:Zentrale: Phase 2 starting (start waiting)
2014-07-28 23:15:56 avmike:Zentrale: Phase 2 ready
2014-07-28 23:15:56 avmike:< cb_sa_created(name=Zentrale,id=1,...,flags=0x00002101)
2014-07-28 23:15:56 avmike:Zentrale: start waiting connections
2014-07-28 23:15:56 avmike:Zentrale: NO waiting connections
```

Maybe this is the problem here?! Looks like "switching to NAT-T (Initiator)" is the probleme here...

#7 - 29.07.2014 14:19 - Marcel Müller

I tried to remove sending the NAT-T vendor payload to test if this is the problem, but the fritzbox is nonetheless initiatiing NAT-T... There must be a difference between 4.6.2 and 5.2.0 advertising NAT-T compatibility...

```
2014-07-29 14:07:39 avmike:mainmode Zentrale: selected lifetime: 3600 sec(no notify)
2014-07-29 14:07:39 avmike:Zentrale remote peer supported XAUTH
2014-07-29 14:07:39 avmike:Zentrale remote peer supported DPD
2014-07-29 14:07:40 avmike:mainmode Zentrale: add SA 1
2014-07-29 14:07:40 avmike:Zentrale: switching to NAT-T (Initiator)
2014-07-29 14:07:40 avmike:Zentrale: Warning: source changed from <publicIPLancom>:500 to <publicIPLancom>:450
0
2014-07-29 14:07:40 avmike:Zentrale: Phase 1 ready
2014-07-29 14:07:40 avmike:Zentrale: local is behind a nat
2014-07-29 14:07:40 avmike:Zentrale: remote is behind a nat
2014-07-29 14:07:40 avmike:Zentrale: start waiting connections
2014-07-29 14:07:40 avmike:Zentrale: Phase 2 starting (start waiting)
2014-07-29 14:07:40 avmike:Zentrale: Phase 2 ready
2014-07-29 14:07:40 avmike:< cb_sa_created(name=Zentrale,id=1,...,flags=0x00032101)
2014-07-29 14:07:40 avmike:Zentrale: start waiting connections
2014-07-29 14:07:40 avmike:Zentrale: NO waiting connections
2014-07-29 14:07:50 avmike:>>>4500 nat-t-keepalive[<publicIPLancom>:4500]
```

#8 - 30.07.2014 17:44 - Tobias Brunner

Do you have the nat_traversal option enabled in 4.6.2? If so, it's odd that pluto would not send the NAT-T vendor IDs. If not, it means pluto will not use NAT traversal (which would work I guess if the Lancom router does some kind of IPsec passthrough). You could try enabling it to see if the behavior is similar to 5.2.0 with NAT-T (you can't disable NAT-T in charon).

What's a bit odd with 5.2.0 is that charon and avmike both report:

```
Jul 28 12:36:07 08[IKE] <9001|1> remote host is behind NAT
```

```
2014-07-28 11:05:58 avmike:Zentrale: local is behind a nat
```

Does your ISP NAT the FRITZ!Box? Otherwise, a NAT should not be detected on that side.

Just by comparing the logs, this might be an issue:

```
2014-07-28 12:32:38 avmike:Zentrale fehlerhafte Paketlaenge: Hdr-length > read-Data
```

Or do you get similar messages with 4.6.2 (or when strongSwan is the initiator)? I also wonder what packet this refers to, and what header (UDP/IKE even ESP?).

#9 - 30.07.2014 18:10 - Marcel Müller

I have set nat_traversal=yes in 4.6.2 (updated to 4.6.4 yesterday, behaves identical to 4.6.2). But it's working with nat_traversal=no as well. Although even with nat_traversal=yes Fritzbox still does not detect NAT-T as supported:

```
2014-07-30 17:59:59 avmike:mainmode Zentrale: selected lifetime: 3600 sec(no notify)
2014-07-30 17:59:59 avmike:Zentrale remote peer supported XAUTH
2014-07-30 17:59:59 avmike:Zentrale remote peer supported DPD
2014-07-30 18:00:00 avmike:mainmode Zentrale: add SA 1
2014-07-30 18:00:00 avmike:Zentrale: Phase 1 ready
2014-07-30 18:00:00 avmike:Zentrale: start waiting connections
2014-07-30 18:00:00 avmike:Zentrale: Phase 2 starting (start waiting)
2014-07-30 18:00:00 avmike:Zentrale: Phase 2 ready
2014-07-30 18:00:00 avmike:< cb_sa_created(name=Zentrale,id=1,...,flags=0x00002101)
2014-07-30 18:00:00 avmike:Zentrale: start waiting connections
2014-07-30 18:00:00 avmike:Zentrale: NO waiting connections
```

No, the Fritzbox itself is not nat'ed by the ISP (Telekom btw).

Regarding "Fehlerhafte Paketlaenge": Looks like this happens when the connection is closed by doing "ipsec stop". Tried to get some debug logs from avmike but haven't had any success, sorry.

Would it help to have a log of 4.6.4 when the Fritzbox is initiator of the connection?
Is there an quick-and-dirty way to disable NAT-T in charon in the source? Just so we can test if this is the culprit here?

BTW: Tried different versions yesterday, 4.6.4 works, 5.0.0 does not. So this is not somethings introduced in 5.2.0 or 5.1.0.

#10 - 30.07.2014 18:52 - Tobias Brunner

- File disable-isakmp-natt-charon.patch added

> No, the Fritzbox itself is not nat'ed by the ISP (Telekom btw).

I wonder what goes wrong then. Perhaps we or the FB calculates the hashes for the NAT-D payloads incorrectly (I'm pretty sure we got that right, though). Anyway, it's not really an issue as the other peer is behind a NAT anyway, but still strange.

> Regarding "Fehlerhafte Paketlaenge": Looks like this happens when the connection is closed by doing "ipsec stop".

That should be the same as ipsec down <name> i.e. DELETE notifications are sent.

> Would it help to have a log of 4.6.4 when the Fritzbox is initiator of the connection?

Maybe. But the pluto log is quite different.

> Is there an quick-and-dirty way to disable NAT-T in charon in the source? Just so we can test if this is the culprit here?

Try the attached patch.

> BTW: Tried different versions yesterday, 4.6.4 works, 5.0.0 does not. So this is not somethings introduced in 5.2.0 or 5.1.0.

Right, it is probably due to a difference in the new IKEv1 implementation introduced with 5.0.0. This was created from scratch and is not related at all to 4.x.

#11 - 30.07.2014 19:08 - Marcel Müller

Applied the patch, Fritzbox does not use NAT-T now:

```
2014-07-30 19:03:30 avmike:mainmode Zentrale: selected lifetime: 3600 sec(no notify)
2014-07-30 19:03:30 avmike:Zentrale remote peer supported XAUTH
2014-07-30 19:03:30 avmike:Zentrale remote peer supported DPD
2014-07-30 19:03:30 avmike:mainmode Zentrale: add SA 1
2014-07-30 19:03:30 avmike:Zentrale: Phase 1 ready
2014-07-30 19:03:30 avmike:Zentrale: start waiting connections
2014-07-30 19:03:30 avmike:Zentrale: Phase 2 starting (start waiting)
2014-07-30 19:03:31 avmike:Zentrale: Phase 2 ready
2014-07-30 19:03:31 avmike:< cb_sa_created(name=Zentrale,id=1,...,flags=0x00002101)
2014-07-30 19:03:31 avmike:Zentrale: start waiting connections
2014-07-30 19:03:31 avmike:Zentrale: NO waiting connections
```

But still no working connection.

I'll get the pluto logs and send them to you.

#12 - 31.07.2014 16:01 - Tobias Brunner

Thanks. What's odd in the pluto logs is that it only receives two vendor IDs (XAuth, DPD), the NAT-T vendor IDs are not received. Did you disable NAT-T on the FRITZ!Box in this test?

I also had another look at the charon log, there we receive three more vendor IDs:

```
Jul 28 12:32:19 06[IKE] <1> received XAuth vendor ID
Jul 28 12:32:19 06[IKE] <1> received DPD vendor ID
Jul 28 12:32:19 06[IKE] <1> received NAT-T (RFC 3947) vendor ID
Jul 28 12:32:19 06[IKE] <1> received draft-ietf-ipsec-nat-t-ike-03 vendor ID
Jul 28 12:32:19 06[ENC] <1> received unknown vendor ID: a2:22:6f:c3:64:50:0f:56:34:ff:77:db:3b:74:f4:1b
```

The NAT detection payloads the FRITZ!Box sends there are odd too. It sends the hash for the internal IP (10.90.1.1) twice, but apparently does not send one for the external IP (it sends two others, though, not sure for which IPs).

Another difference is that IPComp is negotiated with pluto. Not sure if pluto always accepted it if the other peer proposes it, but charon does definitely not, you'd have to enable IPComp explicitly with *compress=yes* to use it.

Looking again at the charon logs I see the following first two proposals sent by the FRITZ!Box:

```
received proposals: ESP:AES_CBC_256/HMAC_SHA1_96/MODP_1024/NO_EXT_SEQ, IPCOMP:, ESP:AES_CBC_256/HMAC_SHA1_
96/MODP_1024/NO_EXT_SEQ, ...
```

So they are the same except for the additional IPComp protocol in the first (not sure why the transforms, e.g. DEFLATE, are not listed). Perhaps there is a confusion between the two. I don't think, for instance, that charon sets the proposal and transform numbers to the ones it received (this is a SHOULD in RFC 2408), which might cause problems for the FRITZ!Box when mapping the returned proposal to the ones it sent.

Could you try setting *compress=yes*, this should cause charon to choose the first proposal. Alternatively, *esp=aes192-sha1-modp1024!* might work, so charon would ignore the first two and select the third.

#13 - 01.08.2014 11:03 - Marcel Müller

Ah - this looks good! Setting only compress = yes did not help, but setting esp=aes192-sha1-modp1024! looks good now! Connection is working regardless of who is the initator. Thank you very much!
Only thing I noticed while using this, strongSwan reestablishes the connection each 2-5 mins. (setting dpdaction=restart btw. Looks like FB does not like DPD?)

```
Security Associations (1 up, 0 connecting):
        9001[4]: ESTABLISHED 2 minutes ago, 172.31.1.5[<publicIPLancom>]...<publicFritz>[<dynDNSFritz>]
        9001{1}:  INSTALLED, TUNNEL, ESP SPIs: c9b8f699_i d4be7ab9_o, IPCOMP CPIs: 2c56_i 9374_o
        9001{1}:   172.31.0.0/16 === 10.90.1.0/24
```

Never had anything longer than 5 mins there ("ESTABLISHED 5 minutes ago"). Looking in charon.log I see a lot of this

```
invalid HASH_V1 payload length, decryption failed?
```

I'll get the complete log.

#14 - 01.08.2014 13:11 - Marcel Müller

Mhm, after a while the connection stopped working again... Looks like this permanent reestablishing of the connection is not good.
Edit:
Is it correct, to have "REKEYING, TUNNEL" in ipsec status?

```
Security Associations (1 up, 0 connecting):
        9001[8]: ESTABLISHED 27 minutes ago, 172.31.1.5[<publicIPLancom>]...<publicIPFritzbox>[<dynDNSFritzbox
>]
        9001{1}:  REKEYING, TUNNEL, expires in 2 minutes
        9001{1}:   172.31.0.0/16 === 10.90.1.0/24
        9001{1}:  INSTALLED, TUNNEL, ESP SPIs: c1436d23_i c7ca0e73_o, IPCOMP CPIs: 0b53_i a76c_o
        9001{1}:   172.31.0.0/16 === 10.90.1.0/24
```

#15 - 02.08.2014 14:22 - Marcel Müller

After disabling DPD those permanent reconnects stopped. But strongSwan is still unable to reauthenticate after the keylife is up (keylife and

ikelifetime are fixed to 1h on the FB side).
I sent the charon log to you - maybe you have an idea?!

#16 - 04.08.2014 12:52 - Tobias Brunner

To not use IPComp just set *compress=no* (or don't configure it at all), the FB seems to always accept the IPComp proposal if it's acting as responder. If it happens if the FB is initiating, there might be something wrong with the proposal handling (how we treat IPComp in IKEv1 is kind of a hack). Would be great if you could send me a log of the FB initiating a connection with the level for the *enc* subsystem increased to 3.

The IKE SA reauthentication/rekeying fails because the CHILD_REKEYING state is not reset after the IPsec SA was rekeyed:

```
Aug  2 11:21:16 11[IKE] <9001|1> CHILD_SA 9001{1} established with SPIs cb85b2f6_i efa9751d_o and TS 172.3
1.0.0/16 === 10.90.1.0/24
...
Aug  2 12:06:55 11[KNL] received a XFRM_MSG_EXPIRE
Aug  2 12:06:55 11[KNL] creating rekey job for ESP CHILD_SA with SPI cb85b2f6 and reqid {1}
...
Aug  2 12:06:56 10[IKE] <9001|1> CHILD_SA 9001{1} established with SPIs c1d6e4b1_i 656fe7f4_o and TS 172.3
1.0.0/16 === 10.90.1.0/24
...
Aug  2 12:12:11 12[MGR] checkout IKE_SA
Aug  2 12:12:11 12[MGR] IKE_SA 9001[1] successfully checked out
Aug  2 12:12:11 12[IKE] <9001|1> unable to reauthenticate in CHILD_SA REKEYING state, delaying for 12s
```

The rekeyed IPsec SAs are currently not uninstalled (as seen in the status output you posted above), they just expire eventually (according to *lifetime*). That's why you also see the outbound SA later soft-expiring:

```
Aug  2 12:12:15 09[KNL] received a XFRM_MSG_EXPIRE
Aug  2 12:12:15 09[KNL] creating rekey job for ESP CHILD_SA with SPI efa9751d and reqid {1}
```

Because the CHILD_SA is kept in state CHILD_REKEYING no additional rekeying is triggered. But this also prevents the IKE_SA from getting rekeyed until the CHILD_SA finally expires. We might consider introducing a new state to handle this (e.g. CHILD_REKEYED), or actually delete the rekeyed CHILD_SA (but that might cause other problems as rekeying is not very well defined in IKEv1, for instance, there is no REKEY_SA notify like in IKEv2 that would tie the old and new SA together). You might want to consider changing the lifetimes a bit to avoid such collisions (e.g. increase *ikelifetime* if you don't really need reauthentication).

Anyway, the CHILD_SA does not expire in your case as the FB is reauthenticating the IKE_SA before that, and charon terminates the existing IKE_SA (with all CHILD_SAs) due to its uniqueness policy (*uniqueids=yes*):

```
Aug  2 12:18:17 16[IKE] <9001|1> deleting duplicate IKE_SA for peer '<identity of FB>' due to uniqueness p
olicy
```

We try to detect whether the new IKE_SA is an IKEv1 reauthentication or not, by comparing the IP and port in addition to the identity. Unfortunately, the port is not the same in this case. While the initial IKE_SA does not use NAT-T (I guess due to a patch preventing charon from sending the RFC 3947 NAT-T vendor ID) and runs on port 500, the new IKE_SA, initiated by the FB, does use NAT-T and runs on port 4500. I'd try to revert any NAT-T related modifications you applied so NAT-T gets used.

Anyway, the DELETE notifications that follow the uniqueness check could pose a problem for the FB that just wants to reauthenticate the IKE_SA, which for IKEv1 should not affect the CHILD_SAs. You could end up without any CHILD_SAs.

#17 - 06.08.2014 19:03 - Marcel Müller

Thank you very much for this detailed explanation! I created logs with enc=3 when the FB is initator (one log with *compress=yes* and one log with *compress=no*).
Additionally I got a log with enc=3 with DPD enabled. There are decryption/hash-errors in the log you might want to look at as well. Looks like DPD never works with FB (regardless of who is the initator) and the connection is restarted every few minutes (which is "correct" as I set *dpdaction* to *restart*).

I changed the *ikelifetime* to 160m and will see how this works. I also rebuild strongSwan 5.2.0 from source so any changes (NAT-T for example) are reverted - although I thought I already reverted every NAT-T changes... Looks like port 4500 is now used in both cases.

```
Anyway, the DELETE notifications that follow the uniqueness check could pose a problem for the FB that just wa
nts to reauthenticate the IKE_SA, which for IKEv1 should not affect the CHILD_SAs. You could end up without an
y CHILD_SAs.
```

Is this anything that needs to be changed in strongSwan or is this related to the CHILD_SA kept in state CHILD_REKEYING and might be fixed if you introduce a new CHILD_REKEYED state?

To sum this up, if NAT-T is used in both cases (and therefore the port is always 4500) and *ikelifetime* is increased to 160m - do you think I'll end up with a working (stable!) tunnel between FB and strongSwan? Or do I need to wait for changes related to rekeying? I'm just not able to judge how stable this is / will be...

#18 - 06.08.2014 21:04 - Marcel Müller

Ok, after increasing *ikelifetime* and making sure NAT-T is used I got this

```
Aug  6 19:52:13 06[IKE] <9001|2> detected rekeying of CHILD_SA 9001{1}
```

but still

```
       9001[3]: ESTABLISHED 13 minutes ago, 172.31.1.5[<publicIPLancom>]...<publicIPFritzbo>[<dynDNSFritzbox>]
        9001[3]: IKEv1 SPIs: ffdc4ee57d6b5839_i 314f5a6a4c87549a_r*, pre-shared key reauthentication in 2 hour
s
        9001[3]: IKE proposal: AES_CBC_256/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_1024
        9001{1}:  REKEYING, TUNNEL, expires in 33 minutes
        9001{1}:   172.31.0.0/16 === 10.90.1.0/24
        9001{1}:  INSTALLED, TUNNEL, ESP in UDP SPIs: c0f3bed7_i 00fd921c_o
        9001{1}:  AES_CBC_192/HMAC_SHA1_96, 0 bytes_i, 0 bytes_o (0 pkts, 16s ago), rekeying in 85 minutes
        9001{1}:   172.31.0.0/16 === 10.90.1.0/24
        9001{1}:  INSTALLED, TUNNEL, ESP in UDP SPIs: ca1fc6ca_i 5a721db2_o
        9001{1}:  AES_CBC_192/HMAC_SHA1_96, 139356 bytes_i (98 pkts, 1s ago), 139356 bytes_o (98 pkts, 1s ago)
, rekeying in 2 hours
        9001{1}:   172.31.0.0/16 === 10.90.1.0/24
```

I think that this looks ok now, does it? I'll decrease the *ikelifetime* to something between 1h and 2h to make sure the first CHILD_SA already expired before a new rekeying process is started and will then just ignore those CHILD_SAs in REKEYING-State, right?

#19 - 07.08.2014 11:52 - Marcel Müller

Regarding https://wiki.strongswan.org/issues/661#note-8 - detection of NAT on both sides: Looks like the FB is internally doing NAT to the AVM IKE-Daemon. The FB has an internal port-forwarding added to itself to forward port 500 and 4500. So I think detection of NAT on both sides would be ok.

#20 - 14.08.2014 14:51 - Marcel Müller

Looks like the tunnel is stable since last week.
Did you get the logs? Did you find anything regarding IPComp and DPD? Would love to use DPD to faster detect a problem.

Thanks in advance!

#21 - 15.08.2014 17:57 - Tobias Brunner

> Did you get the logs?

I have, thanks. Took me a while to find time to look through them.

> Did you find anything regarding IPComp and DPD? Would love to use DPD to faster detect a problem.

Regarding IPComp: The problem here is that strongSwan, for some inexplicable reason, assumes the proposal substructures are numbered starting with 0 (source:src/libcharon/encoding/payloads/sa_payload.c#L298). According to the examples in section 4.2.1 of RFC 2408 that's not the case. And thus, the first few proposal substructs sent by the FB are as follows:

```
#1 ESP AES-CBC-256, SHA-1, MODP-1024
#1 IPComp LZJH
#2 ESP AES-CBC-256, SHA-1, MODP-1024
#2 IPComp DEFLATE
#3 ESP AES-CBC-256, SHA-1, MODP-1024
#4 ESP AES-CBC-192, SHA-1, MODP-1024
#4 IPComp LZJH
...
```

Because the first proposal's number is 1 strongSwan does not skip it as is intended by the code linked above. Instead what happens is this:

```
struct_number = -1
cmp #1 == struct_number (-1)
insert #1 (ESP)
struct_number++ (0)
cmp #1 == struct_number (0)
insert #1 (IPComp)
struct_number++ (1)
// now the numbers are synced and the behavior is as intended
cmp #2 == struct_number (1)
```

```
insert 2 (ESP)
struct_number++ (2)
cmp #2 == struct_number (2)
remove #2 (ESP), ignore #2 (IPComp)
cmp #3 == struct_number (2)
insert #3 (ESP)
...
```

The first two proposals added to the list are not actually usable by strongSwan as they define a bundle (ESP+IPComp), but strongSwan treats them as separate proposals. Therefore we see this in the log:

```
received proposals: ESP:AES_CBC_256/HMAC_SHA1_96/MODP_1024/NO_EXT_SEQ, IPCOMP:, ESP:AES_CBC_256/HMAC_SHA1_96/M
ODP_1024/NO_EXT_SEQ, ...
```

(I didn't realize before that , IPCOMP:, actually denotes a separate proposal containing only an IPComp transform).

So if you configure *esp=aes256-sha1-modp1024* strongSwan will select the first proposal and send that back. The FB on the other hand will interpret that as the third proposal (the one without IPComp). Because the FB uses different SPIs for every proposal the two peers won't be able to communicate as strongSwan sends packets with #1's SPI and the FB expects them with #3's SPI.

Since all the proposals after the first one are treated correctly, setting e.g. *esp=aes192-sha1-modp1024* works around this issue.

Just setting *compress=yes* results in the following error, even though the FB clearly sends several proposals with IPComp:

```
expected IPComp proposal but peer did not send one, IPComp disabled
```

The problem here is that the code selects the first proposal with LZHJ but then later ignores it because the algorithm is not DEFLATE.

I pushed fixes for both issues to the *ikev1-proposal-fixes* branch.

Regarding DPDs: It's interesting that the first DPD exchange, initiated by the FB, is successful:

```
Aug  6 18:25:38 16[ENC] <9001|1> parsed INFORMATIONAL_V1 request 1923711021 [ HASH N(DPD) ]
...
Aug  6 18:25:38 16[ENC] <9001|1> generating INFORMATIONAL_V1 request 2768551452 [ HASH N(DPD_ACK) ]
```

The second is initiated by strongSwan and it fails:

```
Aug  6 18:26:08 05[ENC] <9001|1> generating INFORMATIONAL_V1 request 2822651511 [ HASH N(DPD) ]
...
Aug  6 18:26:08 14[ENC] <9001|1> encrypted => 64 bytes @ 0x8a3750
Aug  6 18:26:08 14[ENC] <9001|1>    0: 33 65 28 FB 4D 66 2D E6 6A C3 96 39 73 2F 8E 42  3e(.Mf-.j..9s/.B
Aug  6 18:26:08 14[ENC] <9001|1>   16: 12 9B 3F 65 2D DE 32 BC 9B 2D 77 50 D8 7F 03 57  ..?e-.2..-wP...W
Aug  6 18:26:08 14[ENC] <9001|1>   32: 89 10 01 04 82 32 13 39 F4 A3 0B AD 70 C7 AB 15  .....2.9....p...
Aug  6 18:26:08 14[ENC] <9001|1>   48: D6 5D BB BD 9B 66 DE 25 3B 05 81 C4 58 15 1C DF  .]...f.%;...X...
Aug  6 18:26:08 14[ENC] <9001|1> plain => 64 bytes @ 0x8a3750
Aug  6 18:26:08 14[ENC] <9001|1>    0: CD E9 FF CA BA 98 B9 A0 BF F7 F4 69 5A 8C 19 E0  ...........iZ...
Aug  6 18:26:08 14[ENC] <9001|1>   16: 9D C1 73 A8 7C C7 5A DD 00 00 00 20 00 00 00 01  ..s.|.Z.... ....
Aug  6 18:26:08 14[ENC] <9001|1>   32: 01 10 8D 29 8D D3 6C B5 1D 73 B6 06 37 7F 18 8B  ...)..l..s..7...
Aug  6 18:26:08 14[ENC] <9001|1>   48: C0 53 EA 54 0A 63 A0 AA 00 00 00 00 00 00 00 00  .S.T.c..........
Aug  6 18:26:08 14[ENC] <9001|1> invalid HASH_V1 payload length, decryption failed?
Aug  6 18:26:08 14[ENC] <9001|1> could not decrypt payloads
Aug  6 18:26:08 14[IKE] <9001|1> message parsing failed
Aug  6 18:26:08 14[IKE] <9001|1> ignore malformed INFORMATIONAL request
Aug  6 18:26:08 14[IKE] <9001|1> INFORMATIONAL_V1 response with message ID 2822651511 processing failed
```

The decryption however does not fail completely, otherwise it's highly unlikely we would get the proper eight zero padding bytes. Also, the DPD_ACK notify looks fine:

```
00 00 00 20  no next payload, length: 32 bytes
00 00 00 01  DOI (1, IPsec)
01 10 8D 29  protocol ID: 1 (IKE), SPI size: 16, notify type: 36137 (DPD_ACK)
8D D3 .. 06  initiator SPI
37 7F .. 54  responder SPI
0A 63 A0 AA  sequence number, same as in DPD request
```

But the hash payload does look invalid:

```
Aug  6 18:26:08 14[ENC] <9001|1>    0: CD E9 FF CA BA 98 B9 A0 BF F7 F4 69 5A 8C 19 E0  ...........iZ...
Aug  6 18:26:08 14[ENC] <9001|1>   16: 9D C1 73 A8 7C C7 5A DD                          ..s.|.Z.
```

The first byte denotes the next payload, which should be 0B (11) = NOTIFY, the next byte is reserved and should be zero, and the next two bytes are the length, which are obviously too high (should be 00 18 (24)).

One explanation for this could be that strongSwan uses the wrong IV to decrypt the message. That would render the first block of 16 bytes incorrect,

the rest would be fine (with CBC the first decrypted block is XORed with the IV all other blocks with the previous cipher text block).

I think that's exactly what happens here. For exchanges after phase 1 the first IV is derived from the message ID and the last cipher text block of the last phase 1 message. Subsequent messages of the same exchange (e.g. during Quick Mode) use the last block of the previous encrypted message as IV.

Now INFORMATIONAL exchanges are not really *exchanges* as they are unidirectional (i.e. consist of a single message sent to the peer). So strongSwan, for example, generates a unique message ID for each DPD message, whether it's a request or a response. But that's not what the FB does.

strongSwan, different MID for the response:

```
Aug  6 18:25:38 16[ENC] <9001|1> parsed INFORMATIONAL_V1 request 1923711021 [ HASH N(DPD) ]
...
Aug  6 18:25:38 16[ENC] <9001|1> generating INFORMATIONAL_V1 request 2768551452 [ HASH N(DPD_ACK) ]
```

FRITZ!Box, same MID for the response:

```
Aug  6 18:26:08 05[ENC] <9001|1> generating INFORMATIONAL_V1 request 2822651511 [ HASH N(DPD) ]
...
Aug  6 18:26:08 14[IKE] <9001|1> INFORMATIONAL_V1 response with message ID 2822651511 processing failed
```

This means that the FB derives the same IV for its response that strongSwan derived for its request. Hence, strongSwan can't decrypt the first block properly as, due to the same message ID in the header, it expects the IV to be the last block of the message it sent.

Unfortunately, strongSwan currently does not log the IV when encrypting IKEv1 messages. Otherwise, we could verify this by reverting the IV application and trying the original IV. I included a patch to log the IV in the aforementioned branch.

The FB is definitely wrong in reusing the message ID for the response, RFC 2409, section 5.7:

```
As noted the message ID in the ISAKMP header-- and used in the prf
computation-- is unique to this exchange and MUST NOT be the same as
the message ID of another phase 2 exchange which generated this
informational exchange.
```

As mentioned before, "exchange" here means sending just one message, the DPD response is a separate phase 2 exchange generated by the DPD request. So you might want to consider sending a bug report to AVM (reusing IVs for CBC is a bad idea).

As a fix we could perhaps avoid caching IVs for INFORMATIONAL exchanges, for which we don't expect multiple messages anyway (unlike for phase 1 and Quick Mode messages). I pushed a commit that does so to the same branch.

#22 - 18.08.2014 20:52 - Marcel Müller

Wow, thank you so much for this detailed response - absolutely makes sense!

I tested the branch you created using:

```
        ike=aes256-sha1-modp1024
        esp=aes256-sha1-modp1024!
        compress=yes
```

works great now and DPD is working flawlessly as well. So you were right: It was the IV messing up the decryption. I'll create a bug report at AVM, but I'm not sure if they'll change anything...

One thing I noticed, testing the following:

```
        ike=aes256-sha1-modp1024
       > esp=aes192-sha1-modp1024! <
        compress=yes
```

results in

```
Security Associations (1 up, 0 connecting):
        9001[1]: ESTABLISHED 18 seconds ago, 172.31.1.5[<...>]...<...>[<...>]
        9001[1]: IKEv1 SPIs: 1f30943a7cb2e6fd_i adc8c34092b3e551_r*, pre-shared key reauthentication in 85 min
utes
        9001[1]: IKE proposal: AES_CBC_256/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_1024

...

Aug 18 20:34:13 16[CFG] <9001|1> selecting proposal:
Aug 18 20:34:13 16[CFG] <9001|1>   no acceptable ENCRYPTION_ALGORITHM found
```

```
Aug 18 20:34:13 16[CFG] <9001|1> received proposals: ESP:AES_CBC_256/HMAC_SHA1_96/MODP_1024/NO_EXT_SEQ
Aug 18 20:34:13 16[CFG] <9001|1> configured proposals: ESP:AES_CBC_192/HMAC_SHA1_96/MODP_1024/NO_EXT_SEQ
Aug 18 20:34:13 16[IKE] <9001|1> received 3600s lifetime, configured 5400s
Aug 18 20:34:13 16[IKE] <9001|1> no matching proposal found, sending NO_PROPOSAL_CHOSEN
```

(and obviously no working connection)

On the other hand, doing the same with compress=no results in

```
Security Associations (1 up, 0 connecting):
        9001[1]: ESTABLISHED 4 seconds ago, 172.31.1.5[<...>]...<...>[<...>]
        9001[1]: IKEv1 SPIs: cd9d1684bc8d8f46_i 0bc1b886139444be_r*, pre-shared key reauthentication in 86 min
utes
        9001[1]: IKE proposal: AES_CBC_256/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_1024
        9001{1}:  INSTALLED, TUNNEL, ESP in UDP SPIs: c2b418c9_i 81a19950_o
        9001{1}:  AES_CBC_192/HMAC_SHA1_96, 0 bytes_i, 0 bytes_o, rekeying in 84 minutes
        9001{1}:   172.31.0.0/16 === 10.90.1.0/24

...

Aug 18 20:36:34 06[CFG] <9001|1> selecting proposal:
Aug 18 20:36:34 06[CFG] <9001|1>   no acceptable ENCRYPTION_ALGORITHM found
Aug 18 20:36:34 06[CFG] <9001|1> selecting proposal:
Aug 18 20:36:34 06[CFG] <9001|1>   proposal matches
Aug 18 20:36:34 06[CFG] <9001|1> received proposals: ESP:AES_CBC_256/HMAC_SHA1_96/MODP_1024/NO_EXT_SEQ, ESP:AE
S_CBC_192/HMAC_SHA1_96/MODP_1024/NO_EXT_SEQ, ESP:AES_CBC/HMAC_SHA1_96/MODP_1024/NO_EXT_SEQ, ESP:3DES_CBC/HMAC_
SHA1_96/MODP_1024/NO_EXT_SEQ, ESP:DES_CBC/HMAC_SHA1_96/MODP_1024/NO_EXT_SEQ, ESP:AES_CBC_256/HMAC_MD5_96/MODP_
1024/NO_EXT_SEQ, ESP:AES_CBC_192/HMAC_MD5_96/MODP_1024/NO_EXT_SEQ, ESP:AES_CBC/HMAC_MD5_96/MODP_1024/NO_EXT_SE
Q, ESP:3DES_CBC/HMAC_MD5_96/MODP_1024/NO_EXT_SEQ, ESP:DES_CBC/HMAC_MD5_96/MODP_1024/NO_EXT_SEQ
Aug 18 20:36:34 06[CFG] <9001|1> configured proposals: ESP:AES_CBC_192/HMAC_SHA1_96/MODP_1024/NO_EXT_SEQ
Aug 18 20:36:34 06[CFG] <9001|1> selected proposal: ESP:AES_CBC_192/HMAC_SHA1_96/MODP_1024/NO_EXT_SEQ
```

based on our tests with the official 5.2.0 branch I would assume that *esp=aes192-sha1-modp1024!* and *compress=yes* should still work, shouldn't it?

Also AVMIKE still complaints about the wrong hdr-length / data-length when sending a DELETE notify from strongswan:

```
2014-08-18 21:46:24 avmike:Zentrale
 fehlerhafte Paketlaenge: Hdr-length > read-Data
2014-08-18 21:46:24 avmike:FreeIPsecSA: spi=c9b955e1        protocol=3 iotype=2
2014-08-18 21:46:24 avmike:< cb_sa_deleted(name=Zentrale,id=2,what=2)
2014-08-18 21:46:24 avmike:FreeIPsecSA: spi=ea71fcaa        protocol=3 iotype=1
2014-08-18 21:46:24 avmike:FreeIPsecSA: spi=5401        protocol=4 iotype=1
2014-08-18 21:46:24 avmike:Zentrale
 fehlerhafte Paketlaenge: Hdr-length > read-Data
2014-08-18 21:46:24 avmike:mainmode Zentrale: del SA 2
2014-08-18 21:46:24 avmike:wolke_del_neighbour_sa_by_remote: no SAs available, set canceld = TRUE
2014-08-18 21:46:24 avmike:< cb_sa_deleted(name=Zentrale,id=2,what=3)
2014-08-18 21:46:24 avmike:FreeIPsecSA: spi=ea67        protocol=4 iotype=2
```

This has nothing to do with the proposals obviously, but I thought I mention it (send the log as well).

I created the logs for both scenarios and will send them to you.
Thank you very much!

#23 - 19.08.2014 15:46 - Tobias Brunner

Thanks for the feedback.

> based on our tests with the official 5.2.0 branch I would assume that *esp=aes192-sha1-modp1024!* and *compress=yes* should still work, shouldn't it?

No, that's an issue with our current handling of IPComp in IKEv1. We just select the first proposal with IPComp and ignore the rest, so your local config must include that first proposal. The reason for this is that for IKEv2 IPComp is negotiated with a single notify, so there is just one CPI, while for IKEv1 each proposal may, and in case of the FB does, have a different CPI and compression algorithm. Unfortunately, our proposal objects currently can't carry this information so we just extract the first IPComp proposal that uses DEFLATE together with its CPI.

The reason why it "worked" with the original branch is that the selected first IPComp proposal was later dropped because it proposed LZJH (as mentioned, we only accept DEFLATE at the moment). So no IPComp proposal was actually selected and the complete proposal list was used to select a suitable proposal (and *aes192-sha1-modp1024* was part of that list). With b9c1d85d08 we now skip IPComp proposals we would later drop anyway. So now if *compress=yes* is set the aes256/DEFLATE proposal is selected, but that won't match your local configuration.

> Also AVMIKE still complaints about the wrong hdr-length / data-length when sending a DELETE notify from strongswan:

> [...]
> This has nothing to do with the proposals obviously, but I thought I mention it (send the log as well).

I'm not sure what the issue here is because the FB is apparently still able to delete the SAs. The SPIs of these SAs are stored in the last 4 or 16 bytes of the messages, so it must receive it completely. I also wonder which header's length field appears to be incorrect (IP/UDP/IKE/payloads?). Our logs don't show the final message with fixed length field in the IKE header (it is updated right after generating the rest of the message, which is the last part that is logged), but I guess it is correct otherwise our test cases would show similar errors. Without more detailed logs from the FB or some help from AVM it's hard to tell what exactly causes the FB to complain.

#24 - 20.08.2014 16:54 - Marcel Müller

Thanks for the explanation! With this in mind it's absolutly fine that strongSwan is just accepting aes256/DEFLATE. Just wanted to make sure there's nothing else broken.

Regarding AVM and DELETE notify:
It's really hard to get any support from avm regarding VPN as they "officially don't support VPN" (not even between their own boxes...). I'll try it again, but I think I'm out of luck here.
One thing that came to my mind regarding DELETE notify was the other way round (FB sends DELETE to strongSwan) and in this case the following happened:

```
Aug 20 16:14:29 05[NET] <9001|44> received packet: from <>[4500] to <>[4500] (76 bytes)
Aug 20 16:14:29 05[ENC] <9001|44> parsed INFORMATIONAL_V1 request 2263306012 [ HASH D ]
Aug 20 16:14:29 05[IKE] <9001|44> received DELETE for ESP CHILD_SA with SPI e61094b2
Aug 20 16:14:29 05[IKE] <9001|44> closing CHILD_SA 9001{1} with SPIs c2d8fcac_i (412083 bytes) e61094b2_o (835
19 bytes) and TS 172.31.0.0/16 === 10.90.1.0/24
Aug 20 16:14:29 10[NET] <9001|44> received packet: from <>[4500] to <>[4500] (76 bytes)
Aug 20 16:14:29 10[ENC] <9001|44> DELETE_V1 verification failed
Aug 20 16:14:29 10[ENC] <9001|44> could not decrypt payloads
Aug 20 16:14:29 10[IKE] <9001|44> message verification failed
Aug 20 16:14:29 10[IKE] <9001|44> ignore malformed INFORMATIONAL request
Aug 20 16:14:29 10[IKE] <9001|44> INFORMATIONAL_V1 request with message ID 1145358896 processing failed
Aug 20 16:14:29 13[NET] <9001|44> received packet: from <>[4500] to <>[4500] (76 bytes)
Aug 20 16:14:29 13[ENC] <9001|44> parsed INFORMATIONAL_V1 request 60305729 [ HASH D ]
Aug 20 16:14:29 13[IKE] <9001|44> received DELETE for ESP CHILD_SA with SPI c2d8fcac
Aug 20 16:14:29 13[IKE] <9001|44> CHILD_SA not found, ignored
Aug 20 16:14:29 03[NET] <9001|44> received packet: from <>[4500] to <>[4500] (76 bytes)
Aug 20 16:14:29 03[ENC] <9001|44> DELETE_V1 verification failed
Aug 20 16:14:29 03[ENC] <9001|44> could not decrypt payloads
Aug 20 16:14:29 03[IKE] <9001|44> message verification failed
Aug 20 16:14:29 03[IKE] <9001|44> ignore malformed INFORMATIONAL request
Aug 20 16:14:29 03[IKE] <9001|44> INFORMATIONAL_V1 request with message ID 2702274706 processing failed
Aug 20 16:14:29 15[NET] <9001|44> received packet: from <>[4500] to <>[4500] (92 bytes)
Aug 20 16:14:29 15[ENC] <9001|44> parsed INFORMATIONAL_V1 request 3157782179 [ HASH D ]
Aug 20 16:14:29 15[IKE] <9001|44> received DELETE for IKE_SA 9001[44]
Aug 20 16:14:29 15[IKE] <9001|44> deleting IKE_SA 9001[44] between 172.31.1.5[<>]...<>[<>]
```

so strongSwan is not happy with the DELETE from FRITZ!Box either... Maybe that can give us a clue what's wrong? I did a complete log of this with enc=3 and will send it to you.

#25 - 26.08.2014 13:35 - Tobias Brunner

> One thing that came to my mind regarding DELETE notify was the other way round (FB sends DELETE to strongSwan) and in this case the following happened:
>
> ```
> ...
> Aug 20 16:14:29 03[ENC] <9001|44> DELETE_V1 verification failed
> Aug 20 16:14:29 03[ENC] <9001|44> could not decrypt payloads
> Aug 20 16:14:29 03[IKE] <9001|44> message verification failed
> Aug 20 16:14:29 03[IKE] <9001|44> ignore malformed INFORMATIONAL request
> ...
> ```
>
> so strongSwan is not happy with the DELETE from FRITZ!Box either.

This happens when the FB deletes the IPComp SA with a separate DELETE notify, which is, of course, technically correct for IKEv1. We currently fail parsing the DELETE payload, though, if it contains a protocol other than IKE, ESP or AH. This is not really a problem as strongSwan deletes the IPComp SA together with the IPsec SA. But I suppose we should theoretically send a DELETE for the IPComp SA as well when we delete it. Not sure if that would resolve the errors seen on the FB.

#26 - 26.08.2014 16:23 - Marcel Müller

I see. But if the missing IPComp SA DELETE is the problem here, this should go away if I use *compress=no* and establish the tunnel without a IPComp, right?

#27 - 26.08.2014 16:59 - Tobias Brunner

> But if the missing IPComp SA DELETE is the problem here, this should go away if I use *compress=no* and establish the tunnel without a IPComp, right?

Correct. So if you still see those after disabling IPComp it must be something else.

#28 - 01.09.2014 19:18 - Marcel Müller

Just tested this - packet-length error still exists in avmike-Log. I'll see if I can get help from AVM, but i think this is a dead end...

Thank you very much for your help!

#29 - 07.07.2015 15:06 - Tobias Brunner

- Status changed from Feedback to Closed

- Resolution set to Fixed

Files

disable-isakmp-natt-charon.patch	3.28 KB	30.07.2014	Tobias Brunner