strongSwan - Issue #2446

Traffic loss during IKE reauth despite make-before-break enabled

12.10.2017 11:22 - Emeric Poupon

Status:	Feedback		
Priority:	Normal		
Assignee:			
Category:			
Affected version:	5.6.0	**Resolution:**	

Description

Some significant effort has been made to prevent traffic loss during the CHILD SA rekeying (#1291) and IKE SA rekeying (see make-before-break).

However, we still observe some traffic loss during the IKE SA re-authentication.
This is a problem since we end up with more and more simultaneous IKE SA/CHILD SA over time, which leads to further problems.

According to Tobias, there is still something to do, see https://lists.strongswan.org/pipermail/dev/2017-April/001762.html

History

#1 - 12.10.2017 18:15 - Tobias Brunner

- Status changed from New to Feedback

> Some significant effort has been made to prevent traffic loss during ... IKE SA rekeying (see make-before-break).

IKE_SA rekeying and reauthentication are completely different things when using IKEv2 (where that option applies), see ExpiryRekey . With IKE_SA rekeying you don't have any traffic loss.

> However, we still observe some traffic loss during the IKE SA re-authentication.

Yes, that's expected.

> we end up with more and more simultaneous IKE SA/CHILD SA over time, which leads to further problems.

That's likely a different problem (like mixing reauthentication with trap policies).

> According to Tobias, there is still something to do, see https://lists.strongswan.org/pipermail/dev/2017-April/001762.html

As explained there, it might be difficult to change this.

#2 - 13.10.2017 09:38 - Emeric Poupon

Tobias Brunner wrote:

> > Some significant effort has been made to prevent traffic loss during ... IKE SA rekeying (see make-before-break).
>
> IKE_SA rekeying and reauthentication are completely different things when using IKEv2 (where that option applies), see ExpiryRekey . With IKE_SA rekeying you don't have any traffic loss.

Sorry, I meant IKE SA reauth (I am going to edit the issue to correct that)

> > However, we still observe some traffic loss during the IKE SA re-authentication.
>
> Yes, that's expected.

I understand, but I consider this to be significant enough to open an issue.

> we end up with more and more simultaneous IKE SA/CHILD SA over time, which leads to further problems.

That's likely a different problem (like mixing reauthentication with trap policies).

Indeed losing traffic has bad consequences when auto=route is used. As it is a common use case, I think it deserves an issue.

> According to Tobias, there is still something to do, see https://lists.strongswan.org/pipermail/dev/2017-April/001762.html

As explained there, it might be difficult to change this.

Do you suggest some workaround to avoid the extra IKE SA when using auto=route?

#3 - 13.10.2017 10:09 - Tobias Brunner

> Some significant effort has been made to prevent traffic loss during ... IKE SA rekeying (see make-before-break).

IKE_SA rekeying and reauthentication are completely different things when using IKEv2 (where that option applies), see ExpiryRekey . With IKE_SA rekeying you don't have any traffic loss.

Sorry, I meant IKE SA reauth (I am going to edit the issue to correct that)

That would not really correct that sentence as there have been no efforts at all to prevent traffic loss during IKE_SA reauthentication. What's wrong is the *make-before-break* part as that does not apply to rekeying, only reauthentication, and there has also been made no effort to prevent traffic loss during IKE_SA rekeying, as there never was any.

> However, we still observe some traffic loss during the IKE SA re-authentication.

Yes, that's expected.

I understand, but I consider this to be significant enough to open an issue.

I'm not sure you actually understand the underlying issue.

> we end up with more and more simultaneous IKE SA/CHILD SA over time, which leads to further problems.

That's likely a different problem (like mixing reauthentication with trap policies).

Indeed losing traffic has bad consequences when auto=route is used. As it is a common use case, I think it deserves an issue.

How could losing traffic have bad consequences when *auto=route* is used? This is not the issue I was referring to. I was referring to using *auto=route* with break-before-make reauthentication, where there is a short time without CHILD_SA installed in the kernel when the kernel might trigger an acquire due to traffic matching the trap policy, which in turn triggers another CHILD (and perhaps IKE) SA besides the ones that are already being re-created due to the reauthentication. But that should not happen with make-before-break reauthentication as there should always be a CHILD_SA installed. The traffic loss that happens with the latter is not related to *auto=route* and has other reasons (see my email).

> According to Tobias, there is still something to do, see https://lists.strongswan.org/pipermail/dev/2017-April/001762.html

As explained there, it might be difficult to change this.

Do you suggest some workaround to avoid the extra IKE SA when using auto=route?

Try to find out why that extra IKE_SA is created (check the logs). But one reason could be creating SAs from both ends (e.g. by using *auto=route* on both). That currently can't really be avoided depending on timing as the uniqueness check is prone to races. That would get you two IKE_SAs and duplicate CHILD_SAs (in general duplicate SAs should not be a problem anymore, though). If there are more, again, check the logs to see why.

#4 - 13.10.2017 10:32 - Emeric Poupon

Let's try another way and start from the beginning:
- strongSwan 5.5.3 + FreeBSD 10.3
- both peers are using 'auto=route' + 'make-before-break'.
- there is a single connection configured between both peers.
- uniqueids is set to 'no'
- IKE SA are reauthenticated, CHILD SA are rekeyed
- we send traffic at a rate of 500kfps, for each direction, matching the negotiated TS.

The problems we observe is that an extra acquire is received sometimes from the kernel during the IKE SA reauth, this leads to an extra IKE SA to be negotiated.
If we set a very short ike lifetime (i.e. 60s), we can end up with more than a hundred IKE SA in a few hours.

During the tests, the tool we use to send/receive traffic shows that some traffic is lost.

Here is the situation, what can we do now?

#5 - 13.10.2017 10:41 - Tobias Brunner

> The problems we observe is that an extra acquire is received sometimes from the kernel during the IKE SA reauth, this leads to an extra IKE SA to be negotiated.

Why is that? Did you investigate via logs etc.?

> If we set a very short ike lifetime (i.e. 60s), we can end up with more than a hundred IKE SA in a few hours.

Did you disable *charon.reuse_ikesa*?

> During the tests, the tool we use to send/receive traffic shows that some traffic is lost.

As explained, with reauthentication (no matter which method) it's difficult to avoid traffic loss.

> Here is the situation, what can we do now?

Use IKE_SA rekeying. What's the reason you use reauthentication?

#6 - 13.10.2017 11:19 - Emeric Poupon

Tobias Brunner wrote:

> > The problems we observe is that an extra acquire is received sometimes from the kernel during the IKE SA reauth, this leads to an extra IKE SA to be negotiated.
>
> Why is that? Did you investigate via logs etc.?

We just noticed that the acquire counter from the kernel is increasing, but yes we have to investigate charon's logs to confirm this.
We will provide more details on this asap.

> > If we set a very short ike lifetime (i.e. 60s), we can end up with more than a hundred IKE SA in a few hours.
>
> Did you disable *charon.reuse_ikesa*?

No.

> > Here is the situation, what can we do now?
>
> Use IKE_SA rekeying. What's the reason you use reauthentication?

We use reauthentication because we want to make sure to revalidate the certificates that are being used (updated CRL, certificate expiry). Does that make sense?

#7 - 13.10.2017 12:14 - Tobias Brunner

> > If we set a very short ike lifetime (i.e. 60s), we can end up with more than a hundred IKE SA in a few hours.
>
> Did you disable *charon.reuse_ikesa*?
>
> No.

Hm, then it's strange that there would be additional IKE_SAs as the existing ones should get reused (you'd still end up with additional CHILD_SAs if an acquire is triggered while doing reauthentication). Maybe some other race, the log might give you a clue.

> > Here is the situation, what can we do now?
>
> Use IKE_SA rekeying. What's the reason you use reauthentication?
>
> We use reauthentication because we want to make sure to revalidate the certificates that are being used (updated CRL, certificate expiry). Does that make sense?

Unfortunately, yes, it's currently the only way to do this. But I'd really like to avoid using reauthentication for it as it has just so many drawbacks. There is already a method on the IKE_SA to recheck certificates used by the peer but that's only called on initiators during make-before-break reauthentication (to avoid a problem with overlapping reauthentication if the CRLs or OCSP are fetched via the IPsec tunnel, see 5.4.0 and the diagram in #989-7).
This method could be used for this purpose, but the question is when to trigger it. One option would be to manually trigger a recheck for all SAs (e.g. after replacing a statically configured CRL), or trigger it automatically once a CRL (or OCSP response, or even the certificate) is scheduled to expire (this information is currently not available outside the revocation plugin, though). Another option would be to do it e.g. whenever an IKE_SA rekeying is triggered (would spread out rechecks if lots of SAs use the same CA/CRL), or via a separate timeout.

#8 - 13.10.2017 17:05 - Emeric Poupon

Tobias Brunner wrote:

> > If we set a very short ike lifetime (i.e. 60s), we can end up with more than a hundred IKE SA in a few hours.
>
> Did you disable *charon.reuse_ikesa*?
>
> No.
>
> Hm, then it's strange that there would be additional IKE_SAs as the existing ones should get reused (you'd still end up with additional CHILD_SAs if an acquire is triggered while doing reauthentication). Maybe some other race, the log might give you a clue.

Ok I have had a look to the log. Since the throughput is really high, IKE messages are getting dropped and retransmitted.
But since the lifetimes are really short, the CHILD SA sometimes are deleted from the kernel before the IKE daemon can do it itself. This makes the logs quite complicated.
I restarted with a more conservative throughput and it seems to work fine so far. I will let you informed.

Now I clearly reproduce the behavior you described in the mail: the responder starts to send packets with the new SA but the initiator has not installed it yet.

> > Here is the situation, what can we do now?
>
> Use IKE_SA rekeying. What's the reason you use reauthentication?
>
> We use reauthentication because we want to make sure to revalidate the certificates that are being used (updated CRL, certificate expiry). Does that make sense?
>
> Unfortunately, yes, it's currently the only way to do this. But I'd really like to avoid using reauthentication for it as it has just so many drawbacks. There is already a method on the IKE_SA to recheck certificates used by the peer but that's only called on initiators during make-before-break reauthentication (to avoid a problem with overlapping reauthentication if the CRLs or OCSP are fetched via the IPsec tunnel, see 5.4.0 and the diagram in #989-7).
> This method could be used for this purpose, but the question is when to trigger it. One option would be to manually trigger a recheck for all SAs (e.g. after replacing a statically configured CRL), or trigger it automatically once a CRL (or OCSP response, or even the certificate) is scheduled to expire (this information is currently not available outside the revocation plugin, though). Another option would be to do it e.g. whenever an IKE_SA rekeying is triggered (would spread out rechecks if lots of SAs use the same CA/CRL), or via a separate timeout.

It would be great to avoid the problematic IKE SA reauth!
For us, a manual trigger would be fine, as we manually update the crl file.

#9 - 16.10.2017 12:02 - Tobias Brunner

> But since the lifetimes are really short, the CHILD SA sometimes are deleted from the kernel before the IKE daemon can do it itself. This makes the logs quite complicated.

It also explains why you get the acquires. As soon as the hard expire removes the installed SA in the kernel while the new SA that's established during the rekeying has not yet been installed the kernel will trigger an acquire.

Do you use such short lifetimes (or in particular short differences between soft and hard expires) only for testing, or is that your default configuration? If so, you should probably rethink that.

#10 - 16.10.2017 12:15 - Emeric Poupon

Tobias Brunner wrote:

> > But since the lifetimes are really short, the CHILD SA sometimes are deleted from the kernel before the IKE daemon can do it itself. This makes the logs quite complicated.
>
> It also explains why you get the acquires. As soon as the hard expire removes the installed SA in the kernel while the new SA that's established during the rekeying has not yet been installed the kernel will trigger an acquire.
>
> Do you use such short lifetimes (or in particular short differences between soft and hard expires) only for testing, or is that your default configuration? If so, you should probably rethink that.

No it is not the default configuration, it was just meant to trigger the problem faster.

We thought this traffic loss during IKE reauth was responsible for triggering additional acquire events, but hopefully it is just a problem related to our test.
Now to get rid of these packet drops we would be happy to just go for the ike rekey method, as soon as we have a way to revalidate crls/certs, as discussed earlier.

#11 - 17.11.2017 17:31 - Emeric Poupon

Hello,

Would you like that I fill a separate feature request about a way to revalidate the existing IKE SA when a CRL or a CA is being updated (and when a certificate is about to expire)?
By the way, do you have any plan to do this in the future?

#12 - 23.11.2017 09:25 - Emeric Poupon

Hello,

I am wondering if implementing https://tools.ietf.org/html/rfc6023 would be valid strategy to avoid this problem?
What do you think?

#13 - 27.11.2017 16:09 - Tobias Brunner

> Would you like that I fill a separate feature request about a way to revalidate the existing IKE SA when a CRL or a CA is being updated (and when a certificate is about to expire)?

Yeah, I guess that would make sense.

> By the way, do you have any plan to do this in the future?

Yes, but there is no ETA yet.

> I am wondering if implementing https://tools.ietf.org/html/rfc6023 would be valid strategy to avoid this problem?
> What do you think?

How so?

#14 - 27.11.2017 17:12 - Emeric Poupon

> I am wondering if implementing https://tools.ietf.org/html/rfc6023 would be valid strategy to avoid this problem?
> What do you think?

How so?

Well it may be easier to recreate the previously negotiated CHILD SA by reusing the mechanisms to differ the actual SA installation in the kernel?
Actually I guess it makes no sense since it would be very implementation specific?