strongSwan - Bug #2358

IKEv1 redundant Child SA rekey is being deleted

11.06.2017 16:49 - Avinoam Meir

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Tobias Brunner	**Estimated time:**	0.00 hour
Category:	libcharon		
Target version:	5.6.0		
Affected version:	5.3.2	**Resolution:**	Fixed

Description

1. In this issue, has been written that for IKEv1 quick mode SA, the child SAs are not deleted in the rekey time because "the IKEv1 delete messages use unacknowledged INFORMATIONAL messages, we can't be sure that the peer actually receives that message."

But when a child sa identified as redundant (there is another newer child sa with the same peer and TS) it looks like the child sa deleted. look here https://github.com/strongswan/strongswan/blob/master/src/libcharon/sa/ikev1/task_manager_v1.c#L1805 -

This can cause a packet lose when the peer doesn't get the delete message and keep sending packet with the deleted SA.

Associated revisions

Revision 083208e8 - 26.06.2017 10:33 - Tobias Brunner

ikev1: Only delete redundant CHILD_SAs if configured

If we find a redundant CHILD_SA (the peer probably rekeyed the SA before
us) we might not want to delete the old SA because the peer might still
use it (same applies to old CHILD_SAs after rekeyings). So only delete
them if configured to do so.

Fixes #2358.

History

#1 - 19.06.2017 11:06 - Avinoam Meir

I proposed https://github.com/strongswan/strongswan/pull/74
I'll be glad to know what you think about the fix.

Thanks.

#2 - 20.06.2017 12:56 - Tobias Brunner

- Tracker changed from Issue to Bug

- Status changed from New to Feedback

- Target version set to 5.6.0

Obvious disclaimer: Use IKEv2!

> 1. In this issue

I assume you refer to #763.

> has been written that for IKEv1 quick mode SA, the child SAs are not deleted in the rekey time because "the IKEv1 delete messages use unacknowledged INFORMATIONAL messages, we can't be sure that the peer actually receives that message."

There is actually an option (*charon.delete_rekeyed*) since 5.4.0 (2f3c08d268) to delete rekeyed CHILD_SAs, which is particularly important if no time based limits are used (or with very long lifetimes and packet/byte limits) they might not get deleted otherwise (or not for a long time) as they are not used anymore.

> This can cause a packet lose when the peer doesn't get the delete message and keep sending packet with the deleted SA.

The check is intended to avoid triggering a rekey if the peer rekeyed the SA before us. So I guess if the peer is an implementation that continues using old SAs even after a rekeying (not sure how that makes sense, but...) we might want to avoid deleting them. But at least we should adhere to the setting mentioned above (i.e. I don't fully agree with your patch in the PR). Instead we should do something like I did in the *2358-ikev1-redundant-child* branch.

#3 - 25.06.2017 22:00 - Avinoam Meir

> Obvious disclaimer: Use IKEv2!

In this case we can use IKEv2 because the peer doesn't support it.

> So I guess if the peer is an implementation that continues using old SAs even after a rekeying (not sure how that makes sense, but...)

Yes, it's indeed a strange behavior of the peer device, but according to the protocol it is valid.

> Instead we should do something like I did in the 2358-ikev1-redundant-child branch.

Thanks for the change, it looks good. It's very hard to reproduce the issue, but I applied the patch and Strongswan runs well without issues.

#4 - 26.06.2017 10:35 - Tobias Brunner

- Status changed from Feedback to Closed

- Assignee set to Tobias Brunner

- Resolution set to Fixed

> > Instead we should do something like I did in the 2358-ikev1-redundant-child branch.
>
> Thanks for the change, it looks good. It's very hard to reproduce the issue, but I applied the patch and Strongswan runs well without issues.

Thanks for testing. Merged to master.