strongSwan - Issue #3490

Selecting incorrect auth mode for IKEv1

22.06.2020 21:03 - fbh dev

Status:	Feedback		
Priority:	Normal		
Assignee:			
Category:	ikev1		
Affected version:	5.8.4	**Resolution:**	

Description

We are noticing customers with multiple proposals are having issues bring their tunnels up. This is the same issue as #3329.

I have a proposed patch, and would appreciate a review when possible. My main concern is around race conditions between the cert pre/post tasks and the main_mode one.

The patch is attached for review.

History

#1 - 22.06.2020 21:05 - fbh dev

We are using 5.7.1, though I think this is consistent with 5.8.x. I will review the code to see if this has been addressed already.

#2 - 22.06.2020 21:10 - fbh dev

This hasn't been fixed yet in the master branch, please review attached patch when you get the chance and let me know if it addresses the root cause. Main concern is around the cert pre/post tasks.

#3 - 22.06.2020 21:13 - fbh dev

This is when customers use cert based auth in the first transform but psk in a later one, and the PSK one is the one that is accepted. Strongswan is always using the first auth mode from the list of transforms.

#4 - 23.06.2020 10:02 - Tobias Brunner

- Category set to ikev1

- Status changed from New to Feedback

> Main concern is around the cert pre/post tasks.

Yeah, that won't work. In isakmp_cert_pre_t there is not yet a proposal stored on the IKE_SA, it's only selected afterwards in the mode-specific tasks.

> This is when customers use cert based auth in the first transform but psk in a later one, and the PSK one is the one that is accepted.

How does that even make sense? Just stop using such weird configs, or even better use IKEv2.

#5 - 23.06.2020 18:47 - fbh dev

> Yeah, that won't work. In isakmp_cert_pre_t there is not yet a proposal stored on the IKE_SA, it's only selected afterwards in the mode-specific tasks.

Any other suggestions here, such as potentially parsing something from the message?

> How does that even make sense? Just stop using such weird configs, or even better use IKEv2.

I'll advise this. I'm not sure why customers do this, but with thousands of them, you're bound to run into a few with an odd setup.

#6 - 24.06.2020 13:29 - Tobias Brunner

> Any other suggestions here, such as potentially parsing something from the message?

I guess sa_payload_t could e.g. return an enumerator with auth methods, so the task could check if any auth method requires certificates (not sure if everything the task does would make sense if a proposal without certificates is later selected). But there are more serious issues. As discussed in #3329, the authentication method is not part of the proposal selection (only the algorithms are considered), and, as mentioned, only one auth method can be configured in strongSwan. Plus where proposals are selected, the configured auth method is actually not available at all, as identities are required to select the peer config for that.

> I'm not sure why customers do this, but with thousands of them, you're bound to run into a few with an odd setup.

Fixing those few configs (which really don't make much sense) seems a lot easier than patching this.

#7 - 30.06.2020 23:29 - fbh dev

Ok, I'll take a shot at fixing it. The common scenario that we run into is that customers setup a router as a transit hub for their virtual networks. Their on-prem network is connected to the transit router using RSA and to the virtual networks via PSK. That's why this issue is recurring.

#8 - 01.07.2020 05:32 - fbh dev

- File temp2.patch added

Ok, here's my second attempt.

#9 - 01.07.2020 05:36 - fbh dev

- File temp3.patch added

3rd attempt.

#10 - 01.07.2020 07:04 - fbh dev

I'm contemplating my current approach. Rather than returning a list of auth_methods, which might complicate the logic as that will allow it to proceed further, maybe we can just check if both payloads (PLV1_CERTREQ and PLV1_CERTIFICATE) exist in the message, as part of use_certs. I guess all the information should be contained in a single IKE message, so that might be sufficient.

#11 - 01.07.2020 07:06 - fbh dev

fbh dev wrote:

> I'm contemplating my current approach. Rather than returning a list of auth_methods, which might complicate the logic as that will allow it to proceed further, maybe we can just check if both payloads (PLV1_CERTREQ and PLV1_CERTIFICATE) exist in the message, as part of use_certs. I guess all the information should be contained in a single IKE message, so that might be sufficient.

I'll have to validate that in our customer's case, there aren't any PLV1_CERTREQ or PLV1_CERTIFICATE payloads, when they are using dual auth modes.

#12 - 01.07.2020 11:13 - Tobias Brunner

> The common scenario that we run into is that customers setup a router as a transit hub for their virtual networks. Their on-prem network is connected to the transit router using RSA and to the virtual networks via PSK. That's why this issue is recurring.

I don't get it? They use the same config for both ends? Why would mixing that make sense (hopefully it's not possible to connect to their internal network via PSK due to this).

> maybe we can just check if both payloads (PLV1_CERTREQ and PLV1_CERTIFICATE) exist in the message, as part of use_certs. I guess all the information should be contained in a single IKE message, so that might be sufficient.

This is IKEv1, nothing is contained in a single IKE message (unless you are considering Aggressive Mode). The authentication method is negotiated during the first exchange (however, that's not made "public"), then follows the key exchange (with optional CERTREQ in the response) and only then are certificates exchanged (plus CERTREQ in the request).

#13 - 01.07.2020 11:58 - fbh dev

Tobias Brunner wrote:

> This is IKEv1, nothing is contained in a single IKE message (unless you are considering Aggressive Mode). The authentication method is negotiated during the first exchange (however, that's not made "public"), then follows the key exchange (with optional CERTREQ in the response) and only then are certificates exchanged (plus CERTREQ in the request).

Good call out. I guess then the patch is moving in the right direction.

#14 - 01.07.2020 12:10 - fbh dev

Tobias Brunner wrote:

> (not sure if everything the task does would make sense if a proposal without certificates is later selected).

FWIW, this seems to be the current behavior since the first auth_mode is using certs, so use_certs is currently returning true already.

I'm between 2 options. Keep the patch as is or making process_certs and process_certreqs return a boolean and bail out if the necessary header isn't found. This would happen in the CR_KE/CR_AUTH state.

One source of confusion is that I'm not quite understanding the difference in terms of code flow between the build/process methods in the cert pre/post tasks. The good thing is that it seems the build_certs methods do nothing if the VPN on our side is configured to use PSK, so they seem idempotent.

#15 - 02.07.2020 18:00 - fbh dev

any thoughts here?

#16 - 02.07.2020 18:01 - Tobias Brunner

> any thoughts here?

Yes, fix the configs.

#17 - 02.07.2020 18:13 - fbh dev

Tobias Brunner wrote:

> Yes, fix the configs.

that's not feasible

#18 - 04.07.2020 05:03 - fbh dev

- File temp4.patch added

attempt 4. forgive my stubbornness.

#19 - 21.07.2020 21:26 - fbh dev

- File 004_ikev1_auth_method_proposal_match.patch added

Here's what I've ended up with, which is confirmed to work on my end.
Can we PLEASE get it reviewed and hopefully merged up-stream?

Files

004_ikev1_auth_mode_match.patch	5.72 KB	22.06.2020	fbh dev
temp2.patch	9.28 KB	01.07.2020	fbh dev
temp3.patch	9.48 KB	01.07.2020	fbh dev
temp4.patch	12.9 KB	04.07.2020	fbh dev
004_ikev1_auth_method_proposal_match.patch	19.2 KB	21.07.2020	fbh dev