strongSwan - Bug #532

SGW initiated IPSec rekey fails when CREATE_CHILD_SA request has MODP_NONE value

26.02.2014 15:01 - Vijay Bhaskar

Status:	Closed	**Start date:**	26.02.2014
Priority:	Normal	**Due date:**	
Assignee:	Tobias Brunner	**Estimated time:**	0.00 hour
Category:	configuration		
Target version:	5.1.3		
Affected version:	5.1.0	**Resolution:**	Fixed

Description

In the SGW initiated IPSec rekey, if CREATE_CHILD_SA request has MODP_NONE client is responding back with NO_PROPOSAL_CHOSEN reply. As per the IKEv2 chipher suite supported values from strongswan, modp NONE support is not available. Is there any option in strongswan to support this or set in ipsec.conf?

Associated revisions

Revision a213944d - 31.03.2014 14:32 - Tobias Brunner

proposal: Don't fail DH proposal matching if peer includes NONE

The DH transform is optional for ESP/AH proposals. The initiator can
include NONE (0) in its proposal to indicate that while it prefers to
do a DH exchange, the responder may still decide to not do so.

Fixes #532.

History

#1 - 28.02.2014 08:48 - Tobias Brunner

- Tracker changed from Bug to Issue

- Status changed from New to Feedback

- Priority changed from High to Normal

If you don't want to do a DH exchange when rekeying the IPsec SA just don't configure a DH group in the *esp* setting in ipsec.conf.

#2 - 28.02.2014 08:49 - Tobias Brunner

- Tracker changed from Issue to Bug

- Target version deleted (5.1.2)

#3 - 28.02.2014 08:49 - Tobias Brunner

- Tracker changed from Bug to Issue

- Assignee set to Tobias Brunner

#4 - 28.02.2014 11:26 - Vijay Bhaskar

Hi Tobias Brunner,

I see that SGW initiated CREATE_CHILD_SA request has DH group as "MODP_NONE" (0) in SA payload (transform) included in it and strongswan rejecting it with NO_PROPOSAL_CHOSEN. Is SGW is doing correctly in this case? Is it fine even if it include the DH transform (with value 0) in SA payload in CREATE_CHILD_SA request?

SA proposal in CREATE_CHILD_SA request by SGW is as below

proposal # 1 ===============

ENCRYPTION - 3des
INTEGRITY - SHA1
DH Group - 0 (MODP_NONE)

Proposal # 2 ==============
ENCRYPTION - 3des

INTEGRITY - MD5
DH Group - 0 (MODP_NONE)

Client esp parameter configuration is as below.

esp = 3des-sha1

#5 - 28.02.2014 13:48 - Tobias Brunner

> SA proposal in CREATE_CHILD_SA request by SGW is as below
>
> proposal # 1 ===============
>
> ENCRYPTION - 3des
> INTEGRITY - SHA1
> DH Group - 0 (MODP_NONE)
>
> Proposal # 2 ==============
> ENCRYPTION - 3des
> INTEGRITY - MD5
> DH Group - 0 (MODP_NONE)

How did you determine this? Is this from information from the log? Wireshark?

> Client esp parameter configuration is as below.
>
> esp = 3des-sha1

What did you configure in the SGW's config? Did you manually set *modpnull* there?

#6 - 28.02.2014 13:59 - Vijay Bhaskar

I have captured wireshark log and decoded it. Yes, SGW side we have an option to set DH group as NONE. It's third party SGW.

I have verified the behavior with strong swan server and in this case it is working fine(both sides configuration is "esp=3des-sha1"). Not seen any DH group (0) in SA proposal (PFS diabled)

#7 - 28.02.2014 15:49 - Tobias Brunner

- Tracker changed from Issue to Bug

> I have captured wireshark log and decoded it. Yes, SGW side we have an option to set DH group as NONE. It's third party SGW.

I see. I had a look at RFC 5996 and it seems we might treat the proposal a bit too strictly.

In section section 3.3.2 it says:

> If the initiator wishes to make use of the transform optional
> to the responder, it includes a transform substructure with
> Transform ID = 0 as one of the options.

So the SGW basically says it's up to the client to decide whether it wants to use DH or not. And therefore it should theoretically work with your client's config.

The reason it currently does not, is that strongSwan treats MODP_NONE kind of like it does other DH groups. So if the client does not include MODP_NONE in its proposal (which it doesn't, it contains no DH group) then there won't be a match. So to match the client's proposal, the SGW's proposal is currently expected to not contain a DH transform at all.

I pushed a fix (dcc88c4) for this to the *optional-proposals* branch of our repository.

#8 - 03.03.2014 08:42 - Vijay Bhaskar

Thank you Tobias Brunner. We will try with this fix.

#9 - 31.03.2014 14:34 - Tobias Brunner

- Status changed from Feedback to Closed

- Target version set to 5.1.3

- Resolution set to Fixed