# strongSwan - Bug #2051

## Missing DH transform in first IKE proposal transform set

13.07.2016 11:04 - Paul Wouters

| | | | |
|---|---|---|---|
| **Status:** | Closed | **Start date:** | 13.07.2016 |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | Tobias Brunner | **Estimated time:** | 0.00 hour |
| **Category:** | libcharon | | |
| **Target version:** | 5.5.1 | | |
| **Affected version:** | 5.4.0 | **Resolution:** | Fixed |

**Description**

It seems strongswan 5.4.0 can send an IKE proposal transform set that does not specify a DH transform.

the following configuration triggers such a malformed packet:

```
[...]
    authby=secret
    keyexchange=ikev2
    auto=add
    ike=aes128-sha1
    esp=aes128-sha1
    fragmentation=yes
    type=transport
```

This is parsed by libreswan:

```
| *****parse IKEv2 Transform Substructure Payload:
|    last transform: v2_TRANSFORM_NON_LAST (0x3)
|    length: 12 (0xc)
|    IKEv2 transform type: TRANS_TYPE_ENCR (0x1)
|    IKEv2 transform ID: AES_CBC (0xc)
| ******parse IKEv2 Attribute Substructure Payload:
|    af+type: IKEv2_KEY_LENGTH (0x800e)
|    length/value: 128 (0x80)
| remote proposal 1 transform 0 (ENCR=AES_CBC_128) matches local proposal 1 transform 0
| remote proposal 1 transform 0 (ENCR=AES_CBC_128) matches local proposal 2 transform 0
| remote proposal 1 transform 0 (ENCR=AES_CBC_128) matches local proposal 3 transform 0
| *****parse IKEv2 Transform Substructure Payload:
|    last transform: v2_TRANSFORM_NON_LAST (0x3)
|    length: 8 (0x8)
|    IKEv2 transform type: TRANS_TYPE_INTEG (0x3)
|    IKEv2 transform ID: AUTH_HMAC_SHA1_96 (0x2)
| remote proposal 1 transform 1 (INTEG=HMAC_SHA1_96) matches local proposal 1 transform 0
| remote proposal 1 transform 1 (INTEG=HMAC_SHA1_96) matches local proposal 2 transform 0
| remote proposal 1 transform 1 (INTEG=HMAC_SHA1_96) matches local proposal 3 transform 0
| *****parse IKEv2 Transform Substructure Payload:
|    last transform: v2_TRANSFORM_LAST (0x0)
|    length: 8 (0x8)
|    IKEv2 transform type: TRANS_TYPE_PRF (0x2)
|    IKEv2 transform ID: PRF_HMAC_SHA1 (0x2)
| remote proposal 1 transform 2 (PRF=HMAC_SHA1) matches local proposal 1 transform 0
| remote proposal 1 transform 2 (PRF=HMAC_SHA1) matches local proposal 2 transform 0
| remote proposal 1 transform 2 (PRF=HMAC_SHA1) matches local proposal 3 transform 0
| Seeing if local proposal 1 matched
```

Note the lack of DH transform.

I suspect there is code re-use for PFS in CREATE_CHILD_SA that is allowed to skip PFS by specifying no DH that is accidentally re-used in the Initial Exchanges where this is obviously not allowed.

There might be an additional bug if strongswan itself accepts this kind of malformed proposal.

## Associated revisions

**Revision fcca7d29 - 05.10.2016 14:26 - Tobias Brunner**

ikev2: Respond with NO_PROPOSAL_CHOSEN if proposal without DH group was selected

Fixes #2051.

**Revision 9b191d59 - 05.10.2016 14:26 - Tobias Brunner**

proposal: Make DH groups mandatory in IKE proposals parsed from strings

References #2051.

## History

**#1 - 13.07.2016 15:23 - Tobias Brunner**

*- Description updated*

*- Category set to libcharon*

*- Status changed from New to Feedback*

*- Priority changed from High to Normal*

> It seems strongswan 5.4.0 can send an IKE proposal transform set that does not specify a DH transform.

It does, if it's explicitly configured to do so.

> I suspect there is code re-use for PFS in CREATE_CHILD_SA that is allowed to skip PFS by specifying no DH that is accidentally re-used in the Initial Exchanges where this is obviously not allowed.

Manually configured proposals are just parsed literally, there are only a few fixes applied after parsing (e.g. PRFs are added if they are missing, integrity algorithms are removed if only AEAD algorithms are proposed - but e.g. mixing AEAD and classic algorithms in the same proposal is currently not prevented even though it's technically not correct).

When I recently added support for MODP_NONE (one of the fixes mentioned above removes this from IKE proposals) I also considered adding a check if an IKE proposal contains at least one DH group but didn't do so because the existing checks only modified proposals (i.e. they didn't reject them) and I thought it might potentially be useful for testing. And if the only proposal has no DH groups (i.e. if you configure *ike=aes128-sha1!*) the connection can't be initiated anyway due to the missing DH group. What happens without the ! is that the default proposal gets added and the first DH group from that is selected by the initiator (the proposals are not reordered or filtered, though, but since strongSwan, as responder, prefers its own proposals by default the order of the received proposals might not really matter).

Anyway, I pushed a commit to the *2051-ike-no-dh* branch that adds such a check. It only is applied to proposals parsed from strings, so invalid proposals may still be created programmatically (e.g. via *custom_proposal* hook of the *conftest* framework).

> There might be an additional bug if strongswan itself accepts this kind of malformed proposal.

If the responder is also configured with such a proposal it actually does select it (the proposals match after all). But then it can't use it due to the missing DH group. This currently results in an empty IKE_SA_INIT response, which I changed to a NO_PROPOSAL_CHOSEN notify in the aforementioned branch.

**#2 - 13.07.2016 16:41 - Paul Wouters**

Tobias Brunner wrote:

> > It seems strongswan 5.4.0 can send an IKE proposal transform set that does not specify a DH transform.
>
> It does, if it's explicitly configured to do so.

It was my understanding that strongswan appends items from the builtin default proposal list, so if I would use ike=3des or ike=3des=sha1, it would fill up the missing allowed entries (eg sha2, and the DH groups and aes/aes_gcm etc)

The confusion with "strict mode" here is why we actually removed the non-strict mode. If you configure anything that is the strict mode proposal set.

I thought my configuration meant "aes with sha1 prefered and whatever else is allowed per default added with less priority"

but regardless, even if I misunderstood wrong, I think it should never send out an bad proposal. Thanks for fixing that.

#### #3 - 13.07.2016 17:01 - Tobias Brunner

*- Target version set to 5.5.1*

> It seems strongswan 5.4.0 can send an IKE proposal transform set that does not specify a DH transform.

It does, if it's explicitly configured to do so.

> It was my understanding that strongswan appends items from the builtin default proposal list, so if I would use ike=3des or ike=3des=sha1, it would fill up the missing allowed entries (eg sha2, and the DH groups and aes/aes_gcm etc)

No, it does not modify the configured proposals. It just adds the default proposal, consisting of pretty much all loaded algorithms, as an additional proposal. That's actually documented in the man page or on the wiki.

> The confusion with "strict mode" here is why we actually removed the non-strict mode. If you configure anything that is the strict mode proposal set.

You mean Libreswan now defaults to *ike/esp=...!* ? I guess that makes the config a bit clearer. That's also the case with our swanctl interface (it provides the *default* keyword, to which *proposals* defaults, to explicitly add the default proposal if needed). I don't think we'll change that for the legacy stroke/ipsec.conf interface though.

> I thought my configuration meant "aes with sha1 prefered and whatever else is allowed per default added with less priority"

No, it just adds the configured algorithms as separate proposal at the front of the list of proposals.

#### #4 - 13.07.2016 17:50 - Paul Wouters

Tobias Brunner wrote:

It was already the default in openswan to be strict when specifying anything....

Note there are more options of possibly bad proposals. Eg using ike=aes128-null-modp1536 causes a KE payload that does not match the modp1536 group

#### #5 - 13.07.2016 18:29 - Tobias Brunner

> Eg using ike=aes128-null-modp1536 causes a KE payload that does not match the modp1536 group

As I mentioned there are currently practically no checks against manually configured proposals. But what do you mean exactly? With the above proposal the KE payload should be for the modp1536 group.

#### #6 - 05.10.2016 15:06 - Tobias Brunner

*- Status changed from Feedback to Closed*

*- Assignee set to Tobias Brunner*

*- Resolution set to Fixed*