strongSwan - Bug #2430

traffic selector not included in IKE_AUTH during re-authentication

19.09.2017 14:16 - c c

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Tobias Brunner	**Estimated time:**	0.00 hour
Category:	libcharon		
Target version:	5.6.1		
Affected version:	5.5.3	**Resolution:**	Fixed

Description

It is observed that for an established IPsec tunnel, during re-authentication, strongswan does not include traffic selector payload in IKE_AUTH.
Thus peer(also strongSwan) complained and re-authentication failed, as follows:
It is OK without reauth.

```
Sep  5 13:52:21 15[IKE] scheduling reauthentication in 440s
Sep  5 13:52:21 15[IKE] maximum IKE_SA lifetime 450s
Sep  5 13:52:21 15[IKE] sending end entity cert "C=CN, ST=Some-State, L=CC, O=Comp, CN=PC001"
Sep  5 13:52:21 15[IKE] traffic selectors (null)=== (null) inacceptable
Sep  5 13:52:21 15[IKE] failed to establish CHILD_SA, keeping IKE_SA
Sep  5 13:52:21 15[ENC] generating IKE_AUTH response 1 [ IDr CERT AUTH N(AUTH_LFT) N(TS_UNACCEPT) ]
```

Associated revisions

Revision 26bda4e9 - 02.11.2017 09:48 - Tobias Brunner

ikev2: Abort make-before-break reauth if we don't find children to recreate

We do something similar in reestablish() for break-before-make reauth.
If we don't abort we'd be sending an IKE_AUTH without any TS payloads.

References #2430.

History

#1 - 19.09.2017 14:33 - Tobias Brunner

- Status changed from New to Feedback

Please post the complete logs, configs etc. of both ends.

#2 - 25.09.2017 15:15 - c c

- File log4strongswan.zip added

Please see the attachment for the configs and logs, this is basically what happened:
1. A(reauth=no) initiated negotiation with B(reauth=yes), IKE SA and child SA established successfully.
2. A continuously initiated negotiation with B, probably due to reauth and lifetime configuration.
3. Second attempt of negotiation failed on B, when trying to install same policy with different reqid.
B returned error TS_UNACCEPT to A
A:auto = route, B:auto = start
4. A then changed traffic selector to null, B again returned error TS_UNACCEPT.
5. There were some repetitions of step 3 and 4, at quick pace.

So it seems to me there are following problems:
1. why A initiated negotiation repeatedly and at very quick pace, it seemed definitely relevant to reauth.
2. On B, the policy installing failure seemed unnecessary, why must use a different reqid for each new child SA.
3. A changed ts to null after receiving TS_UNACCEPT, this is also strange.

#3 - 26.09.2017 11:39 - Tobias Brunner

- Tracker changed from Issue to Bug

- Target version set to 5.6.1

> 1. A(reauth=no) initiated negotiation with B(reauth=yes), IKE SA and child SA established successfully.

As documented on ExpiryRekey*reauth=no* has no effect on the client if the peer requests a reauthentication using AUTH_LIFETIME notifies (which happens if it has *reauth=yes* set).

> 2. A continuously initiated negotiation with B, probably due to reauth and lifetime configuration.

Yes, refer to the link above. The local *margintime* has an effect on this. The server here has a very low *ikelifetime* and the initiator's *margintime* is larger than that, so a reauthentication is scheduled immediately.

> 3. Second attempt of negotiation failed on B, when trying to install same policy with different reqid.

That's because you use an old release (5.1.3). You should update to a more recent one where that's not a problem anymore (see release notes for 5.3.0 and you'll fine lots of closed issues with that error message). You also seem to have make-before-break reauthentication configured on the client, but as documented on ExpiryRekey this is not compatible with versions before 5.3.0 (even if you disable it there is no guarantee the responder's old release will be able to install the policies, see e.g. #431).

> 4. A then changed traffic selector to null, B again returned error TS_UNACCEPT.

That's because it has no CHILD_SAs to recreate (the one it tried to established failed in the previous attempt), so there won't be a child-create task (i.e. no TS payloads will be added to the IKE_AUTH request). When using make-before-break reauthentication there does not seem to be a check whether there are any CHILD_SAs (or child-create tasks) available, which is the case if reestablish() is called during break-before-make reauthentication. I pushed a fix for this to the *2430-mbb-reauth-no-children* branch.

#4 - 26.09.2017 17:59 - c c

Thanks for the info, regarding this:

> The server here has a very low ikelifetime and the initiator's margintime is larger than that, so a reauthentication is scheduled immediately.

If client's margintime(over_time) is larger than server ikelifetime, the reauthentication happens repeatedly without any delay. This is definitely undesired. Should there be an improvement for this?
Since with AUTH_LIFETIME, the lifetime field somewhat overrides the client configuration. Maybe its better to use it directly as ikelifetime or margintime for reauthentication, instead of comparing with client margintime.

#5 - 02.11.2017 09:52 - Tobias Brunner

I pushed the fix for the reauth without CHILD_SAs to master.

> Maybe its better to use it directly as ikelifetime or margintime for reauthentication, instead of comparing with client margintime.

As documented, it is used as *ikelifetime* but the local *margintime* is still considered. So if the lifetimes on both ends are significantly different that could cause problems (but it must be really significant because in practical scenarios *ikelifetime* should be large enough so that any reasonable *margintime* should work fine).

#6 - 16.11.2017 10:08 - Tobias Brunner

- Category set to libcharon

- Status changed from Feedback to Closed

- Assignee set to Tobias Brunner

- Resolution set to Fixed

Files

log4strongswan.zip	13.4 KB	25.09.2017	c c