strongSwan - Bug #3637

Android VPN client: server-side DNS server setting gets lost after a while

23.11.2020 18:22 - Davor Josipovic

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Tobias Brunner	**Estimated time:**	0.00 hour
Category:	android		
Target version:	5.9.2		
Affected version:	5.9.1	**Resolution:**	Fixed

Description

The Android VPN client (2.3.1) works great for me. After connection, I can access my local network by DNS name.

There is this one issue though. After a few minutes of idle time, the connection counter on the android lock-screen will reset. Looking into logs of Android client, this reset seems to go hand-in-hand with ESP CHILD_SA DELETE and CREATE. After the reset, I am no longer able to access my resources by DNS name.

I assume that at this point the DNS server from the local network is no longer the preferred one? I don’t think this behavior is intended?

Setting the DNS server explicitly in the profile fixes the issue.

Associated revisions

Revision cd10ae2f - 04.02.2021 16:52 - Tobias Brunner

android: Explicitly apply DNS servers to the TUN device

If the peer deletes the CHILD_SA, we recreate it due to the close action. However, if we create a new TUN device, we do so with a new VpnService.Builder object and on that the DNS servers were never applied. The latter happened only on the fly in the attribute handler when an IKE_SA was established. Now we do this explicitly when creating the TUN device, like the virtual IPs and routes. While we could avoid the recreation of the TUN device if the CHILD_SA is recreated, there is the theoretical possibility that the remote traffic selectors change. This way we also avoid adding stuff to the builder in different places.

Fixes #3637.

History

#1 - 24.11.2020 11:30 - Tobias Brunner

- Category set to android

- Status changed from New to Feedback

> After a few minutes of idle time, the connection counter on the android lock-screen will reset.

What "connection counter"?

> Looking into logs of Android client, this reset seems to go hand-in-hand with ESP CHILD_SA DELETE and CREATE. After the reset, I am no longer able to access my resources by DNS name.

Why is there a delete? Is only the CHILD_SA recreated? Please post the logs.

> I assume that at this point the DNS server from the local network is no longer the preferred one? I don’t think this behavior is intended?

Sounds weird, but without knowing what exactly is happening, I can't say more.

> Setting the DNS server explicitly in the profile fixes the issue.

Hm, interesting.

#2 - 24.11.2020 13:55 - Davor Josipovic

- File snip.png added

Tobias Brunner wrote:

> After a few minutes of idle time, the connection counter on the android lock-screen will reset.

What "connection counter"?

See attached image.

> Looking into logs of Android client, this reset seems to go hand-in-hand with ESP CHILD_SA DELETE and CREATE. After the reset, I am no longer able to access my resources by DNS name.

Why is there a delete? Is only the CHILD_SA recreated? Please post the logs.

```
Nov 24 12:58:40 13[IKE] sending keep alive to xxx.xxx.xxx.xxx[4500]
Nov 24 13:01:00 06[IKE] sending keep alive to xxx.xxx.xxx.xxx[4500]
Nov 24 13:01:45 08[IKE] sending keep alive to xxx.xxx.xxx.xxx[4500]
Nov 24 13:02:16 10[NET] received packet: from xxx.xxx.xxx.xxx[4500] to yyy.yyy.yyy.yyy[zzzzz] (76 bytes)
Nov 24 13:02:16 10[ENC] parsed INFORMATIONAL request 0 [ D ]
Nov 24 13:02:16 10[IKE] received DELETE for ESP CHILD_SA with SPI cdbeff1a
Nov 24 13:02:16 10[IKE] closing CHILD_SA android{3} with SPIs 1ec0d519_i (9946 bytes) cdbeff1a_o (5804 bytes)
and TS 192.168.1.60/32 === 10.0.2.2/32 10.0.2.15/32 192.168.1.0/24
Nov 24 13:02:16 10[IKE] sending DELETE for ESP CHILD_SA with SPI 1ec0d519
Nov 24 13:02:16 10[IKE] CHILD_SA closed
Nov 24 13:02:16 10[IKE] establishing CHILD_SA android{4} reqid 2
Nov 24 13:02:16 10[ENC] generating CREATE_CHILD_SA request 6 [ N(ESP_TFC_PAD_N) SA No KE TSi TSr ]
Nov 24 13:02:16 10[NET] sending packet: from yyy.yyy.yyy.yyy[zzzzz] to xxx.xxx.xxx.xxx[4500] (684 bytes)
Nov 24 13:02:16 10[ENC] generating INFORMATIONAL response 0 [ D ]
Nov 24 13:02:16 10[NET] sending packet: from yyy.yyy.yyy.yyy[zzzzz] to xxx.xxx.xxx.xxx[4500] (76 bytes)
Nov 24 13:02:16 11[NET] received packet: from xxx.xxx.xxx.xxx[4500] to yyy.yyy.yyy.yyy[zzzzz] (236 bytes)
Nov 24 13:02:16 11[ENC] parsed CREATE_CHILD_SA response 6 [ SA No TSi TSr ]
Nov 24 13:02:16 11[CFG] selected proposal: ESP:AES_CBC_256/HMAC_SHA1_96/NO_EXT_SEQ
Nov 24 13:02:16 11[IKE] ignoring KE exchange, agreed on a non-PFS proposal
Nov 24 13:02:16 11[IKE] CHILD_SA android{4} established with SPIs d4541004_i cb82bf61_o and TS 192.168.1.60/32
 === 10.0.2.2/32 10.0.2.15/32 192.168.1.0/24
Nov 24 13:02:16 11[DMN] setting up TUN device for CHILD_SA android{4}
Nov 24 13:02:16 11[DMN] successfully created TUN device
```

So after this happens at 13:02:16, local DNS resolution no longer works.

#3 - 24.11.2020 17:06 - Tobias Brunner

- Tracker changed from Issue to Bug

- Target version set to 5.9.2

> Looking into logs of Android client, this reset seems to go hand-in-hand with ESP CHILD_SA DELETE and CREATE. After the reset, I am no longer able to access my resources by DNS name.

> Why is there a delete? Is only the CHILD_SA recreated? Please post the logs.

So the server just deletes the CHILD_SA? Why? Do you have the server logs?

> So after this happens at 13:02:16, local DNS resolution no longer works.

I see. The problem is that a new TUN device is created (using a new VpnService.Builder instance) when the new CHILD_SA is established, but that's obviously not a complete IKE_SA, where DNS servers are actually negotiated. The virtual IPs and routes are explicitly added to the builder whenever a TUN device is created, the DNS servers are not, they are only added on the fly when the configuration attributes are assigned during IKE_AUTH. Recreating only the CHILD_SA is kinda rare, which is probably why nobody noticed so far.

I guess there are several options to avoid this. One would be to change how builders (or rather the data added to them) are managed, so they are more IKE_SA-centric and CHILD_SA events don't affect them other than to add routes for negotiated traffic selectors until a new IKE_SA is created. This would probably require quite some refactoring and would only really be a benefit if we'd have to handle multiple CHILD_SAs per IKE_SA (or even multiple IKE_SAs), which is currently not the case. Or we could try to apply the DNS servers to the builder similar to the virtual IPs, i.e. whenever a TUN device is created (they are already stored on the IKE_SA, so that might not be that much work). Or we could try to avoid creating a new TUN device in the first place when only a CHILD_SA is recreated (however, at least in theory, the remote traffic selectors might be different when the CHILD_SA is recreated, not sure if that's a use case though, i.e. deleting the CHILD_SA in order to assign different remote traffic selectors when the client recreates it). While the latter is relatively simple to implement (and it's unlikely that the traffic selectors change) it might still make more sense to unify the handling of these attributes and install the DNS servers explicitly. So I've implement the latter in the *3637-android-dns* branch.

I could create a new beta version of the app if you want to try it out. Although it would probably be better to avoid the deletion of the CHILD_SA in the first place.

#4 - 24.11.2020 19:20 - Davor Josipovic

Tobias Brunner wrote:

> > Looking into logs of Android client, this reset seems to go hand-in-hand with ESP CHILD_SA DELETE and CREATE. After the reset, I am no longer able to access my resources by DNS name.
>
> Why is there a delete? Is only the CHILD_SA recreated? Please post the logs.

So the server just deletes the CHILD_SA? Why? Do you have the server logs?

This is server-side:

```
Nov 24 13:02:16 XXXXXXXX charon-systemd[YYY]: deleting CHILD_SA after 120 seconds of inactivity
Nov 24 13:02:16 XXXXXXXX charon-systemd[YYY]: closing CHILD_SA net{123} with SPIs cdbeff1a_i (5804 bytes) 1ec0
d519_o (9946 bytes) and TS 10.0.2.2/32 10.0.2.15/32 192.168.1.0/24 === 192.168.1.60/32
Nov 24 13:02:16 XXXXXXXX charon-systemd[YYY]: sending DELETE for ESP CHILD_SA with SPI cdbeff1ay
```

> So after this happens at 13:02:16, local DNS resolution no longer works.

> I see. The problem is that a new TUN device is created (using a new VpnService.Builder instance) when the new CHILD_SA is established, but that's obviously not a complete IKE_SA, where DNS servers are actually negotiated. The virtual IPs and routes are explicitly added to the builder whenever a TUN device is created, the DNS servers are not, they are only added on the fly when the configuration attributes are assigned during IKE_AUTH. Recreating only the CHILD_SA is kinda rare, which is probably why nobody noticed so far.
>
> I guess there are several options to avoid this. One would be to change how builders (or rather the data added to them) are managed, so they are more IKE_SA-centric and CHILD_SA events don't affect them other than to add routes for negotiated traffic selectors until a new IKE_SA is created. This would probably require quite some refactoring and would only really be a benefit if we'd have to handle multiple CHILD_SAs per IKE_SA (or even multiple IKE_SAs), which is currently not the case. Or we could try to apply the DNS servers to the builder similar to the virtual IPs, i.e. whenever a TUN device is created (they are already stored on the IKE_SA, so that might not be that much work). Or we could try to avoid creating a new TUN device in the first place when only a CHILD_SA is recreated (however, at least in theory, the remote traffic selectors might be different when the CHILD_SA is recreated, not sure if that's a use case though, i.e. deleting the CHILD_SA in order to assign different remote traffic selectors when the client recreates it). While the latter is relatively simple to implement (and it's unlikely that the traffic selectors change) it might still make more sense to unify the handling of these attributes and install the DNS servers explicitly. So I've implement the latter in the *3637-android-dns* branch.
>
> I could create a new beta version of the app if you want to try it out. Although it would probably be better to avoid the deletion of the CHILD_SA in the first place.

Thank you for the very detailed analysis! Quite instructive. I trust you and your team will make the right choice on what to do with this issue. (I can't help on that one.) And no need for a beta version, it is not a breaking bug for me. But since I like strongSwan so much, I had to take the time to report it.

So why the DELETE? Hmm... It's an old and minimal required config for Windows native VPN. Minimal in the sense that only strictly required settings are defined in swanctl.conf. And there you have it:

```
# Timeout before closing CHILD_SA after inactivity.
inactivity = 120s
```

I normally document everything, but unfortunately not this line. I also do not see in WindowsClients why it should have an inactivity set. Except for this (?) line:

> Another option is to set no rekey time, but only a hard lifetime to delete the CHILD_SA. The client will renegotiate the SA when required.

I should revisit all this once again when I have more time... And obviously thanks again for sorting it all out!

#5 - 25.11.2020 10:30 - Tobias Brunner

> And no need for a beta version, it is not a breaking bug for me. But since I like strongSwan so much, I had to take the time to report it.

OK, thanks for reporting it.

> So why the DELETE? Hmm... It's an old and minimal required config for Windows native VPN. Minimal in the sense that only strictly required settings are defined in swanctl.conf. And there you have it:
>
> [...]

I see, that really makes not much sense in roadwarrior scenarios. In particular, it may prevent traffic from the server back to the client (e.g. for notifications) as it will not automatically create a CHILD_SA if necessary, and for clients like our Android app that recreate the CHILD_SA right away there really is no benefit in deleting it for inactivity.

> I normally document everything, but unfortunately not this line. I also do not see in WindowsClients why it should have an inactivity set. Except for this (?) line:
>
> > Another option is to set no rekey time, but only a hard lifetime to delete the CHILD_SA. The client will renegotiate the SA when required.

Hm, maybe, but the inactivity timeout has nothing to do with the rekeying. To avoid issues on Windows clients I'd rather increase the lifetimes so that the clients rekey the SAs (the Android client will do so too after a few hours).

#6 - 04.02.2021 16:56 - Tobias Brunner

- Status changed from Feedback to Closed

- Assignee set to Tobias Brunner

- Resolution set to Fixed

The fix for the DNS servers is now in master and will be included in the next version of the app (not sure yet when that will be).

Files

snip.png	9.98 KB	24.11.2020	Davor Josipovic