strongSwan - Issue #3364
strongSwan Android APP no keep-alive / MOBIKE are sent during sleep
06.03.2020 10:45 - Beat Zahnd

Status: Feedback
Priority: Normal
Assignee: 
Category: android
Affected version: 5.8.2
Resolution: 

Description
I migrated from IPsec/L2TP to IKEv2 with certificates. Using the Android stronSwan App the connection works well as long as the mobile phone is not put to sleep. The server is libreswan 3.27 on Debian.

After a few minutes on sleep no more traffic is possible. If for about 2 minutes no traffic is exchange the cellular network NAT changes its port. In this tcpdump the port changes from 21977 to 21978. But the server continues to send to 21977. There was no MOBIKE ticket sent.

20:43:26.048376 IP 84.75.x.x.4500 > 178.197.x.x.21977: UDP-encap: ESP(spi=0xbe045de8,seq=0x178), length 88
20:43:26.080322 IP 84.75.x.x.4500 > 178.197.x.x.21977: UDP-encap: ESP(spi=0xbe045de8,seq=0x179), length 88
20:43:26.164652 IP 178.197.x.x.21977 > 84.75.x.x.4500: UDP-encap: ESP(spi=0x85bb58ce,seq=0x178), length 88
20:43:26.166635 IP 178.197.x.x.21977 > 84.75.x.x.4500: UDP-encap: ESP(spi=0x85bb58ce,seq=0x179), length 88
20:43:26.168593 IP 178.197.x.x.21977 > 84.75.x.x.4500: UDP-encap: ESP(spi=0x85bb58ce,seq=0x17a), length 88
20:43:26.214647 IP 178.197.x.x.21977 > 84.75.x.x.4500: UDP-encap: ESP(spi=0x85bb58ce,seq=0x17b), length 100
20:43:26.216616 IP 178.197.x.x.21977 > 84.75.x.x.4500: UDP-encap: ESP(spi=0x85bb58ce,seq=0x17c), length 100
20:43:26.218614 IP 178.197.x.x.21977 > 84.75.x.x.4500: UDP-encap: ESP(spi=0x85bb58ce,seq=0x17d), length 100
20:43:26.220576 IP 178.197.x.x.21977 > 84.75.x.x.4500: UDP-encap: ESP(spi=0x85bb58ce,seq=0x17e), length 100
20:43:26.222600 IP 178.197.x.x.21977 > 84.75.x.x.4500: UDP-encap: ESP(spi=0x85bb58ce,seq=0x17f), length 100
20:44:45.725612 IP 178.197.x.x.21977 > 84.75.x.x.4500: UDP-encap: ESP(spi=0x85bb58ce,seq=0x180), length 88
20:44:45.725670 IP 178.197.x.x.21977 > 84.75.x.x.4500: UDP-encap: ESP(spi=0x85bb58ce,seq=0x181), length 88
20:44:45.725681 IP 178.197.x.x.21977 > 84.75.x.x.4500: UDP-encap: ESP(spi=0x85bb58ce,seq=0x182), length 88
20:44:45.725849 IP 84.75.x.x.4500 > 178.197.x.x.21977: UDP-encap: ESP(spi=0xbe045de8,seq=0x17a), length 88
20:44:45.726333 IP 178.197.x.x.21977 > 84.75.x.x.4500: UDP-encap: ESP(spi=0x85bb58ce,seq=0x183), length 88
20:47:52.022713 IP 178.197.x.x.21978 > 84.75.x.x.4500: UDP-encap: ESP(spi=0x85bb58ce,seq=0x184), length 88
20:47:52.022804 IP 178.197.x.x.21978 > 84.75.x.x.4500: UDP-encap: ESP(spi=0x85bb58ce,seq=0x185), length 88
20:47:52.022836 IP 178.197.x.x.21978 > 84.75.x.x.4500: UDP-encap: ESP(spi=0x85bb58ce,seq=0x186), length 88
20:47:52.022858 IP 178.197.x.x.21978 > 84.75.x.x.4500: UDP-encap: ESP(spi=0x85bb58ce,seq=0x187), length 88
20:47:52.022879 IP 178.197.x.x.21978 > 84.75.x.x.4500: UDP-encap: ESP(spi=0x85bb58ce,seq=0x188), length 88
20:47:52.022892 IP 178.197.x.x.21978 > 84.75.x.x.4500: UDP-encap: ESP(spi=0x85bb58ce,seq=0x189), length 88
20:47:52.022909 IP 178.197.x.x.21978 > 84.75.x.x.4500: UDP-encap: ESP(spi=0x85bb58ce,seq=0x18a), length 88
Previously with IPsec L2TP there was nat-keep-alive with about 1 min interval. This is not seen now using the strongswan App.

10:22:13.362197 IP 178.197.235.175.65141 > 84.75.x.x.4500: isakmp-nat-keep-alive
10:23:14.333471 IP 178.197.235.175.65141 > 84.75.x.x.4500: isakmp-nat-keep-alive
10:24:15.802514 IP 178.197.235.175.65141 > 84.75.x.x.4500: isakmp-nat-keep-alive

NAT-T keep-alive time has been set to 15. But this does not help.

MOBIKE works. If i switch on WiFi or off then MOBIKE cookies are sent and the connection works again, as long as the mobile phone is not on sleep.

The mobile is a Sony with Android 8
Mar 3 22:52:59 12[IKE] authentication of 'CN=bz' (myself) with RSA signature successful
Mar 3 22:52:59 12[IKE] sending end entity cert "CN=bz"
Mar 3 22:52:59 12[ENC] generating IKE_AUTH request 1 [ IDi CERT N(INIT_CONTACT) CERTREQ AUTH CPRQ (ADDR ADDR6 DNS DNS6) N(ESP_TFC_PAD_N) SA TSi TSr N(MOBIKE_SUP) N(NO_ADD_ADDR) N(EAP_ONLY) N(MSG_ID_SYN_SUP) ]
Mar 3 22:52:59 12[ENC] generating IKE_AUTH request 1 [ EF(1/2) ]
Mar 3 22:52:59 12[ENC] generating IKE_AUTH request 1 [ EF(2/2) ]
Mar 3 22:52:59 16[NET] sending packet: from 10.125.64.234[41633] to 84.75.x.x[4500] (1368 bytes)
Mar 3 22:52:59 16[NET] sending packet: from 10.125.64.234[41633] to 84.75.x.x[4500] (719 bytes)
Mar 3 22:52:59 16[ENC] received fragment #1 of 4, waiting for complete IKE message
Mar 3 22:52:59 16[NET] received packet: from 84.75.x.x[4500] to 10.125.64.234[41633] (535 bytes)
Mar 3 22:52:59 16[ENC] received fragment #2 of 4, waiting for complete IKE message
Mar 3 22:52:59 16[NET] received packet: from 84.75.x.x[4500] to 10.125.64.234[41633] (535 bytes)
Mar 3 22:52:59 16[ENC] received fragment #3 of 4, waiting for complete IKE message
Mar 3 22:52:59 16[NET] received packet: from 84.75.x.x[4500] to 10.125.64.234[41633] (366 bytes)
Mar 3 22:52:59 16[ENC] received fragment #4 of 4, reassembled fragmented IKE message (1784 bytes)
Mar 3 22:52:59 16[ENC] parsed IKE_AUTH response 1 [ IDr CERT AUTH CPRP ADDR DNS SA TSi TSr ]
Mar 3 22:52:59 15[IKE] received end entity cert "CN=hackersgarden.dyn.ch"
Mar 3 22:52:59 15[CFG] using trusted ca certificate "C=CH, ST=ZH, O=hackersGarden"
Mar 3 22:52:59 15[CFG] checking certificate status of "CN=bz"
Mar 3 22:52:59 15[CFG] certificate status is not available
Mar 3 22:52:59 15[CFG] reached self-signed root ca with a path length of 0
Mar 3 22:52:59 15[CFG] using trusted certificate "CN=bz"
Mar 3 22:52:59 15[IKE] signature validation failed, looking for another key
Mar 3 22:52:59 15[CFG] using trusted certificate "CN=hackersgarden.dyn.ch"
Mar 3 22:52:59 15[CFG] using trusted certificate "C=CH, ST=ZH, O=hackersGarden"
Mar 3 22:52:59 15[CFG] checking certificate status of "CN=hackersgarden.dyn.ch"
Mar 3 22:52:59 15[CFG] certificate status is not available
Mar 3 22:52:59 15[IKE] reached self-signed root ca with a path length of 0
Mar 3 22:52:59 15[IKE] IKE_SA android[7] established between 10.125.64.234(CN=bz)...84.75.x.x[hackersgarden.dyn.ch]
Mar 3 22:52:59 15[IKE] scheduling rekeying in 35506s
Mar 3 22:52:59 15[IKE] maximum IKE_SA lifetime 36106s
Mar 3 22:52:59 15[IKE] CHILD_SA android[7] established with SPIs dd2dd5ff_i 1acc2969_o and TS 192.168.1.100/32 === 0.0.0.0/0
Mar 3 22:52:59 15[DMN] successfully created TUN device

Any ideas why the keep-alive and MOBIKE does not work on the cellular network?

History
#1 - 06.03.2020 11:53 - Tobias Brunner
- Status changed from New to Feedback

Using the Android strongSwan App the connection works well as long as the mobile phone is not put to sleep.

What do you mean with "put to sleep" exactly?

There was no MOBIKE ticket sent.

MOBIKE will only cause an exchange if the client's IP changes. (Although, there is a patch somewhere that triggers a DPD, with NAT-T payloads to detect NAT mapping changes, even if the IP stays the same after addresses have changed - whether that would be the case here and it would help, is another question, though.)
Previously with IPsec L2TP there was nat-keep-alive with about 1 min interval. This is not seen now using the strongswan App.

You can't compare the two. The former runs as Android system service, the latter is a simple app. If Android suspends it, no NAT keepalives can be sent (and obviously no inbound traffic from the server could be processed either). The keepalives are also sent from the C part of the app, which Android might not know much about (if that even would be a consideration).

NAT-T keep-alive time has been set to 15. But this does not help.

So the app is already suspended after 15 seconds? Or is the whole device kind of off?

The mobile is a Sony with Android 8

Could be a factor if their system image does force suspension of any app/process after a short while, no idea.

Any ideas why the keep-alive and MOBIKE does not work on the cellular network?

The app does not keep any explicit wake locks, in case that would make a difference (Android can ignore them). However, it is registered as foreground service and it has a VpnService instance open. I wonder what would happen if the device had some traffic to send that required the VPN connection.

By the way, the log is not really helpful as it only shows the initial connection setup.

#2 - 06.03.2020 14:04 - Beat Zahnd

Tobias Brunner wrote:

Using the Android strongSwan App the connection works well as long as the mobile phone is not put to sleep.

What do you mean with "put to sleep" exactly?

Either by just letting it unused until the screen is switched off and the phone is locked or by pressing the power button. But not switched off.

MOBIKE will only cause an exchange if the client's IP changes. (Although, there is a patch somewhere that triggers a DPD, with NAT-T payloads to detect NAT mapping changes, even if the IP stays the same after addresses have changed - whether that would be the case here and it would help, is another question, though.)

Sure, after IP change it is anyway needed. But it would be also needed if the cellular link is lost for more than the NAT timeout off my cellular provider. During this time no communication is possible and therefore NAT keep-alives are useless. Such interruptions could occur often and there is no way to force cellular providers to increase the NAT timeout...

Previously with IPsec L2TP there was nat-keep-alive with about 1 min interval. This is not seen now using the strongswan App.

You can't compare the two. The former runs as Android system service, the latter is a simple app. If Android suspends it, no NAT keepalives can be sent (and obviously no inbound traffic from the server could be processed either). The keepalives are also sent from the C part of the app, which Android might not know much about (if that even would be a consideration).

Not sure what I could do more. The nasty Sony Stamina Mode is disabled here and the strongSwan VPN Client app is excepted from power saving.

NAT-T keep-alive time has been set to 15. But this does not help.

So the app is already suspended after 15 seconds? Or is the whole device kind of off?

The mobile phone is in sleep as explained above. Other apps e.g. messengers, or mail are able to receive messages. If it is active (on) then keep alives are received on the server (again using the 45 sec default):

```
13:18:59.693162 IP 178.197.x.x.32061 > 84.75.x.x.4500: UDP-encap: ESP(spi=0x9853ad46,seq=0x1b01), length 88
13:18:59.740103 IP 178.197.x.x.32061 > 84.75.x.x.4500: UDP-encap: ESP(spi=0x9853ad46,seq=0x1b02), length 88
13:19:44.906099 IP 178.197.x.x.32061 > 84.75.x.x.4500: isakmp-nat-keep-alive
13:20:27.137180 IP 178.197.x.x.32061 > 84.75.x.x.4500: UDP-encap: ESP(spi=0x9853ad46,seq=0x1b02), length 100
13:21:12.982128 IP 178.197.x.x.32061 > 84.75.x.x.4500: isakmp-nat-keep-alive
```

The mobile is a Sony with Android 8
Could be a factor if their system image does force suspension of any app/process after a short while, no idea.

Only know about the Stamina Mode which is disabled and having the app excepted from battery saving.

Any ideas why the keep-alive and MOBIKE does not work on the cellular network?

The app does not keep any explicit wake locks, in case that would make a difference (Android can ignore them). However, it is registered as foreground service and it has a VpnService instance open. I wonder what would happen if the device had some traffic to send that required the VPN connection.

If I ping the mobile phone from the server to its VPN assigned IP then the connection persists even if pings are not answered after about 2 min. Seems to keep the NAT alive at least

By the way, the log is not really helpful as it only shows the initial connection setup.

In this case nothing more was logged even after the connection stalled for some time. Seems that MOBIKE was not enabled on the server. Here a log with MOBIKE enabled on the server and the connection stalled for 5 min. The last communication was at about 14:54. But there is no further logging as well.

Mar 6 13:49:37 00[DNN] -------------------------------
Mar 6 13:49:37 00[DNN] Starting IKE service (strongSwan 5.8.2dr1, Android 8.0.0 - 34.A.2.118/2018-09-01, F5 321 - Sony/F5321/Sony, Linux 3.10.84-perf-g51b663c723-04742-g5bcfac9e61, aarch64)
Mar 6 13:49:37 00[CFG] loaded user certificate 'CN=bz' and private key
Mar 6 13:49:37 00[CFG] loaded CA certificate 'C=CH, ST=ZH, O=hackersGarden'
Mar 6 13:49:37 09[JOB] spawning 16 worker threads
Mar 6 13:49:38 11[IKE] local host is behind NAT, sending keep alives
Mar 6 13:49:38 11[IKE] authentication of 'CN=bz' (myself) with RSA signature successful
Mar 6 13:49:38 11[IKE] sending end entity cert "CN=example.dyn.ch"
19.03.2020
05/07
The mobile phone is in sleep as explained above. Other apps e.g. messengers, or mail are able to receive messages.

Push notifications are based on a totally different mechanism (probably Firebase Cloud Messaging, which uses a single persistent connection for all messaging apps).

If I ping the mobile phone from the server to its VPN assigned IP then the connection persists even if pings are not answered after about 2 min. Seem to keep the NAT alive at least

Probably depends on the NAT implementation whether packets from the "outside" are considered when deciding whether to keep a mapping alive.

Here a log with MOBIKE enabled on the server and the connection stalled for 5 min. The last communication was at about 14:54. But there is no further logging as well.

Was that with constant pinging? The app will only send keepalives (and log about them) if no other outbound traffic was sent. However, if the system suspends the app even if it recently was actively processing network traffic, I doubt we can do much about it.

Maybe wake locks could force the device from suspending the app, but besides being quite invasive they can apparently be ignored too by Android with its Doze/App Standby system. Not sure about Doze but App Standby should not be used for our app while a VPN connection is active as it has a foreground service active.

#4 - 06.03.2020 15:10 - Beat Zahnd

Then it seems the app is suspended and no keep alives are sent while the device is on sleep. This leads to NAT timeout and then the connection is lost forever. Just tried it with a different Android mobile phone with the same effect. Only a MOBIKE update would be able to bring it back again.

This renders the use of this kind of app based VPN almost unusable IMHO. In normal life a mobile phone is on sleep for most of the time. Or how is this intended to be handled in the strongswan app?

Or the VPN on the server shall accept UDP-encap ESP from all source ports even if they change almost randomly. But no idea how this could be feasible.

Ping: its just sending ICPM ping to the IP assigned to the child, 192.168.1.100 in the above case. Since this is sent inside the tunnel there is continous UDP traffic preventing NAT timeout.

#5 - 06.03.2020 15:19 - Tobias Brunner

Or how is this intended to be handled in the strongswan app?

No idea. We can't really change how Android suspends apps.

Or the VPN on the server shall accept UDP-encap ESP from all source ports even if they change almost randomly. But no idea how this could be feasible.
They should do that anyway (and notify the IKE daemon about the change, see RFC 7296, section 2.23). More problematic is sending traffic if the client didn't send anything for a while.

#6 - 07.03.2020 10:01 - Beat Zahnd

No idea. We can't really change how Android suspends apps.

Seems to be a tricky issue. I have no android experience. Just found the following: https://stackoverflow.com/questions/32774437/prevent-android-app-in-background-getting-suspended

In CharonVpnService.java:185 the return value is START_NOT_STICKY instead of the recommended START_STICKY. Not sure if this applies here.

They should do that anyway (and notify the IKE daemon about the change, see RFC 7296, section 2.23). More problematic is sending traffic if the client didn't send anything for a while.

Will check on the libreswan mailing list if this a configuration issue.

#7 - 09.03.2020 10:45 - Tobias Brunner

In CharonVpnService.java:185 the return value is START_NOT_STICKY instead of the recommended START_STICKY. Not sure if this applies here.

I don't think the process is actually killed here. That return value would only have an effect then. And our VpnService instance is already running as foreground service.

#8 - 13.03.2020 14:23 - Beat Zahnd

Still trying to find a solution...

Some statements from the libreswan mailing list:

- So that looks like the strongswan bug doing SHA1 for RFC7427 connections that RFC 8472 says should never use SHA1 and which libreswan didn't advertise

- I know iphone for example just ALWAYS sends a MOBIKE ADDRESS_UPDATE, even if its address did not change, so that the port change can be picked up by the server side. Strongswan should really do the same.

#9 - 13.03.2020 14:32 - Tobias Brunner

- So that looks like the strongswan bug doing SHA1 for RFC7427 connections that RFC 8472 says should never use SHA1 and which libreswan didn't advertise

A wrong statement (we never propose SHA-1 for that) and completely irrelevant.

- I know iphone for example just ALWAYS sends a MOBIKE ADDRESS_UPDATE, even if its address did not change, so that the port change can be picked up by the server side. Strongswan should really do the same.

That is a possibility (there is actually a patch somewhere that does that). However, it would only have any effect if on your device there is actually an event that would trigger this process in the first place and only once the screen is on again. While the screen is off, the device would probably still not be reachable.

By the way, I wasn't able to reproduce this issue with a Pixel 4 (Android 10) behind a NAT router. Even after hours of the screen being off the device was reachable from the server.