strongSwan - Bug #543

charon doesn't work after an IP address change

08.03.2014 22:19 - Yves-Alexis Perez

Status:	Closed	**Start date:**	08.03.2014
Priority:	Normal	**Due date:**	
Assignee:	Tobias Brunner	**Estimated time:**	0.00 hour
Category:	libhydra		
Target version:	5.2.0		
Affected version:	5.1.2	**Resolution:**	Fixed

Description

Hi,

I'm using charon with a roadwarrior setup, on my laptop. So I frequently switch from one network to another, sometimes on LAN, sometimes on WLAN (no WWAN but that could also happen).

Right now, each time I change network, I need to do ipsec restart because charon doesn't support reconfiguring IP addresses on the fly, and thus is somehow broken after a switch.

Associated revisions

Revision 3bf98189 - 19.06.2014 14:16 - Tobias Brunner

kernel-netlink: Follow RFC 6724 when selecting IPv6 source addresses

Instead of using the first address we find on an interface we should
consider properties like an address' scope or whether it is temporary
or public.

Fixes #543.

History

#1 - 10.03.2014 07:03 - Andreas Steffen

- Tracker changed from Bug to Issue

- Status changed from New to Feedback

- Assignee set to Andreas Steffen

Are you using IKEv2 with MOBIKE which supports dynamic IP address or network device changes? The old IKEv1 protocol is not able to do that.

Andreas

#2 - 10.03.2014 07:55 - Yves-Alexis Perez

Andreas Steffen wrote:

> Are you using IKEv2 with MOBIKE which supports dynamic IP address or network device changes? The old IKEv1 protocol is not able to do that.

I'm using IKEv2, not sure about MOBIKE though.

#3 - 10.03.2014 07:58 - Yves-Alexis Perez

Yves-Alexis Perez wrote:

> Andreas Steffen wrote:
>
> > Are you using IKEv2 with MOBIKE which supports dynamic IP address or network device changes? The old IKEv1 protocol is not able to do that.
>
> I'm using IKEv2, not sure about MOBIKE though.

According to http://wiki.strongswan.org/projects/strongswan/wiki/MobIke it should be enabled by default, but it doesn't look like it's the case.

charon correctly listens on all interfaces:

```
udp        0      0 0.0.0.0:4500            0.0.0.0:*                           0          1329610     22243/c
haron
udp        0      0 0.0.0.0:500             0.0.0.0:*                           0          1329609     22243/c
haron
udp6       0      0 :::4500                 :::*                                0          1329608     22243/c
haron
udp6       0      0 :::500                  :::*                                0          1329607     22243/c
haron
```

but it still doesn't use the right IP address after network change.

#4 - 10.03.2014 11:35 - Tobias Brunner

Is there a switch between interfaces, or does just the IP address change (or both)? Is the previous address still configured on any interface? Is the previous route still there?

It would help if you could provide logs captured during an address change (preferably with log level 2 for the *knl* log group).

#5 - 10.03.2014 11:53 - Yves-Alexis Perez

Tobias Brunner wrote:

> Is there a switch between interfaces, or does just the IP address change (or both)? Is the previous address still configured on any interface? Is the previous route still there?
>
> It would help if you could provide logs captured during an address change (preferably with log level 2 for the *knl* log group).

So I just resumed the laptop on a different subnet: same interface, the IP address changes and the previous one is gone (network-manager + DHCP). I don't have level 2 logs this time, but I get:

At suspend:

```
Mar 10 08:19:39 scapa charon: 11[KNL] 192.168.0.17 disappeared from eth0
Mar 10 08:19:39 scapa charon: 10[KNL] 2a01:yyyy:yyyy:yyyy:yyyy:yyyy:yyyy:yyyy disappeared from eth0
Mar 10 08:19:39 scapa charon: 15[KNL] interface eth0 deactivated
Mar 10 08:19:39 scapa charon: 05[KNL] fe80::xxxx:xxxx:xxxx:xxxx disappeared from eth0
```

At resume:

```
Mar 10 11:44:18 scapa charon: 10[KNL] fe80::xxxx:xxxx:xxxx:xxxx appeared on eth0
Mar 10 11:44:18 scapa charon: 07[KNL] 192.168.28.56 appeared on eth0
Mar 10 11:44:20 scapa charon: 08[KNL] 2001:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx appeared on eth0
```

so charon /did/ see the new IP address. But when trying to setup the tunnel, I get:

```
Mar 10 11:46:41 scapa charon: 07[NET] sending packet: from fe80::xxxx:xxxx:xxxx:xxxx[500] to <dst-ipv6>[500] (
360 bytes)
Mar 10 11:46:41 scapa charon: 03[NET] error writing to socket: Invalid argument
```

Note that it tries to use the link-local IPv6 address, which actually didn't change (it was removed and later readded to the interface). I tried to force IPv4 usage and it seems to work and correctly use 192.168.28.56.

So it might actually be a bad interaction between IPv6 link-local address or something like that.

I'll try to confirm it always work fine on IPv4 later.

#6 - 10.03.2014 15:20 - Tobias Brunner

How does your IPv6 routing table look like? (All tables, before/after the switch).

#7 - 10.03.2014 15:56 - Yves-Alexis Perez

Tobias Brunner wrote:

> How does your IPv6 routing table look like? (All tables, before/after the switch).

After the switch, but also after restarting charon (so a bit more like a “before” situation where everything works)

```
corsac@scapa: ip -6 route show
local ::1 dev lo  proto kernel  metric 256
2001:xxxx:xxxx:xxxx::/64 dev eth0  proto kernel  metric 256  expires 86281sec [the local /64]
2a01:yyyy:yyyy:yyyy::ff42 dev eth0  proto kernel  metric 256 [the right IP address, assigned by my remote char
on]
fe80::/64 dev eth0  proto kernel  metric 256  [link local]
default via fe80::zzzz:zzzz:zzzz:zzzz dev eth0  proto static  metric 1
default via fe80::zzzz:zzzz:zzzz:zzzz dev eth0  proto ra  metric 1024  expires 1681sec [the local gateway]
```

#8 - 10.03.2014 16:40 - Tobias Brunner

```
default via fe80::zzzz:zzzz:zzzz:zzzz dev eth0  proto static  metric 1
```

I suppose this is the problematic route. Since strongSwan uses your routing table to find an address to reach the other peer it will happily accept that default route and probably use the link-local address on *eth0* as source. But this means that when that lookup happens there is no better route to reach the other peer. This might be some kind of race condition if it takes a while for the real address to appear. You could try to increase ROAM_DELAY in source:src/libhydra/plugins/kernel_netlink/kernel_netlink_net.c, or perhaps we should ignore link-local addresses/routes by default (and add an option to use them in those rare? cases where that makes sense).

```
Mar 10 11:46:41 scapa charon: 03[NET] error writing to socket: Invalid argument
```

What's problematic with this is that strongSwan currently simply ignores such errors. Since the component that sends the messages is totally separated from the IKE_SA where the source address is set no new address lookup is triggered. Reporting such errors back might have helped in this case (using the existing roam job that triggers an address lookup and MOBIKE exchange might be an option to do so).

#9 - 10.03.2014 16:51 - Yves-Alexis Perez

Tobias Brunner wrote:

> [...]
>
> I suppose this is the problematic route. Since strongSwan uses your routing table to find an address to reach the other peer it will happily accept that default route and probably use the link-local address on *eth0* as source. But this means that when that lookup happens there is no better route to reach the other peer. This might be some kind of race condition if it takes a while for the real address to appear.

Actually, the global unicast address is already present, and should be used as source address as long as the destination address is also a global unicast one (RFC 6724 §5 rule 2).

> You could try to increase ROAM_DELAY in source:src/libhydra/plugins/kernel_netlink/kernel_netlink_net.c,

I don't think that'd change anything, I already tried to wait dozens of minutes.

> or perhaps we should ignore link-local addresses/routes by default (and add an option to use them in those rare? cases where that makes sense).

I'm not sure if there's a reason why the address selection is done in charon and not in the kernel, but it should respect RFC 6724 if it does it itself.

#10 - 10.03.2014 18:04 - Tobias Brunner

- Tracker changed from Issue to Bug

- Category changed from charon to libhydra

- Assignee changed from Andreas Steffen to Tobias Brunner

- Target version set to 5.1.3

> I suppose this is the problematic route. Since strongSwan uses your routing table to find an address to reach the other peer it will happily accept that default route and probably use the link-local address on *eth0* as source. But this means that when that lookup happens there is no better route to reach the other peer. This might be some kind of race condition if it takes a while for the real address to appear.

> Actually, the global unicast address is already present, and should be used as source address as long as the destination address is also a global unicast one (RFC 6724 §5 rule 2).

I see.

> or perhaps we should ignore link-local addresses/routes by default (and add an option to use them in those rare? cases where that makes sense).

> I'm not sure if there's a reason why the address selection is done in charon and not in the kernel, but it should respect RFC 6724 if it does it itself.

The reason for doing this in charon is the source route we install to implement virtual IP addresses. These routes in table 220 have a higher priority to force the correct source address for outbound packets so that they match the installed IPsec policies. Therefore, we can't let the kernel select the source IP for IKE packets because if the VPN gateway's address is contained in *rightsubnet* (e.g. for *rightsubnet=::/0*) the kernel would use the virtual IP as source address, which wouldn't work.

But to select the source address for IPv6 we currently use about the same algorithm we use for IPv4, which probably does not fully apply. Basically we use the source address from the most closely matching route and if none is returned/configured we use one of the addresses on the route's outbound interface. If that's the case we currently use the first address (unless the previously used address is found). Instead we should probably follow the algorithm defined in RFC 6724 to order the addresses on the outbound interface.

#11 - 10.03.2014 18:07 - Yves-Alexis Perez

As an example, when the tunnel is down, here's what I get with ip:

```
root@scapa:~# ip route get 2a01:rrrr:rrrr:rrrr::r [remote IPsec gateway]
2a01:rrrr:rrrr:rrrr::r from :: via fe80::gggg:gggg:gggg:gggg dev eth0  src 2001:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:
xxxx  metric 0
[remote IPsec gateway]             [LL address of local gw]               [GU address of my laptop]
```

I'll check tomorrow that the same route entry appear when I wake up from sleep after switching network.

#12 - 10.03.2014 18:32 - Yves-Alexis Perez

Tobias Brunner wrote:

> The reason for doing this in charon is the source route we install to implement virtual IP addresses. These routes in table 220 have a higher priority to force the correct source address for outbound packets so that they match the installed IPsec policies. Therefore, we can't let the kernel select the source IP for IKE packets because if the VPN gateway's address is contained in *rightsubnet* (e.g. for *rightsubnet=::/0*) the kernel would use the virtual IP as source address, which wouldn't work.

Good point. I always thought that chicken-and-egg problem was handled the same kind of way than the encryption one (with the per-socket policies): something magic happening somewhere :)

#13 - 11.03.2014 09:31 - Yves-Alexis Perez

Yves-Alexis Perez wrote:

> As an example, when the tunnel is down, here's what I get with ip:
>
> [...]
>
> I'll check tomorrow that the same route entry appear when I wake up from sleep after switching network.

Confirmed.

#14 - 14.04.2014 12:49 - Tobias Brunner

- Target version changed from 5.1.3 to 5.2.0

#15 - 19.06.2014 14:20 - Tobias Brunner

- Status changed from Feedback to Closed

- Resolution set to Fixed

This should be fixed with the associated commit.