strongSwan - Bug #2820

lan bypass: routes installed incorrectly in table 220 when quickly swapping IP addresses/subnets on two interfaces

06.11.2018 19:48 - Rob Cornall

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Tobias Brunner	**Estimated time:**	0.00 hour
Category:	kernel-interface		
Target version:	5.7.2		
Affected version:	5.7.1	**Resolution:**	Fixed

Description

Hi,

I'm having an issue with the bypass-lan plugin, where changing ip addresses on interfaces in quick succession is causing incorrect routes in table 220.

It is easy step to reproduce, just swap 2 interface IP addresses quickly.

For example with a setup I have the following interfaces:
-br0 using 192.168.1.0/24
-usb0 using 192.168.2.0/24

The routes are installed correctly:

```
$ ip ro
192.168.1.0/24 dev br0  src 192.168.1.0
192.168.2.0/24 dev usb0  src 192.168.2.0
$ ip ro show table 220
192.168.1.0/24 dev br0  src 192.168.1.0
192.168.2.0/24 dev usb0  src 192.168.2.0
$ ip ro show table 0
192.168.1.0/24 dev br0 table 220  src 192.168.1.0
192.168.2.0/24 dev usb0 table 220  src 192.168.2.0
192.168.1.0/24 dev br0  src 192.168.1.0
192.168.2.0/24 dev usb0  src 192.168.2.0
```

Then remove the addresses from both interfaces, and add new ones (swap the addresses) all in 1 command:

```
$ ip addr del dev br0 192.168.1.0/24; ip addr del dev usb0 192.168.2.0; ip addr add dev br0 192
.168.2.0/24; ip addr add dev usb0 192.168.1.0/24
```

Now the routes show as:

```
$ ip ro
192.168.1.0/24 dev usb0  src 192.168.1.0
192.168.2.0/24 dev br0  src 192.168.2.0
$ ip ro show table 220
192.168.1.0/24 dev br0  src 192.168.1.0
192.168.2.0/24 dev usb0  src 192.168.2.0
$ ip ro show table 0
192.168.1.0/24 dev br0 table 220  src 192.168.1.0
192.168.2.0/24 dev usb0 table 220  src 192.168.2.0
192.168.1.0/24 dev usb0  src 192.168.1.0
192.168.2.0/24 dev br0  src 192.168.2.0
```

Traffic destined for 1.0/24 will go out the wrong interface br0.

It seems like a race condition, because when running the commands 1 after another, or with sleeps, we see the correct routes:

```
$ ip addr del dev br0 192.168.1.0/24; sleep 1; ip addr del dev usb0 192.168.2.0; sleep 1
```

```
; ip addr add dev br0 192.168.2.0/24; sleep 1; ip addr add dev usb0 192.168.1.0/24
$
$ ip ro
192.168.1.0/24 dev usb0  src 192.168.1.0
192.168.2.0/24 dev br0  src 192.168.2.0
$ ip ro show table 220
192.168.1.0/24 dev usb0  src 192.168.1.0
192.168.2.0/24 dev br0  src 192.168.2.0
$ ip ro show table 0
192.168.1.0/24 dev usb0 table 220  src 192.168.1.0
192.168.2.0/24 dev br0 table 220  src 192.168.2.0
192.168.1.0/24 dev usb0  src 192.168.1.0
192.168.2.0/24 dev br0  src 192.168.2.0
```

I appreciate any help with this,
Rob

Associated revisions

Revision 2421b7dd - 22.11.2018 11:38 - Tobias Brunner

bypass-lan: Compare interface for unchanged policies

In case a subnet is moved from one interface to another the policies can
remain as is but the route has to change. This currently doesn't happen
automatically and there is no option to update the policy or route so
removing and reinstalling the policies is the only option.

Fixes #2820.

History

#1 - 07.11.2018 08:40 - Tobias Brunner

- Category changed from libcharon to kernel-interface

- Status changed from New to Feedback

Read the log (see HelpRequests for log settings).

#2 - 07.11.2018 21:24 - Rob Cornall

- File charon_debug_reproduced.log added

- File charon_debug_with_sleeps.log added

- File strongswan.conf added

- File bypass-lan.conf added

Hi, here i have attached charon_debug_reproduced.log for running the ip addr commands without sleeps.
And I also attached charon_debug_with_sleeps.log for running the same commands with 3 second sleeps in between.

Note that no tunnels are up, and no extra configuration has been made.

Thanks,
Rob

#3 - 08.11.2018 11:41 - Tobias Brunner

Thanks for the logs. It's pretty clear what happens. In case of the quick switch only one roam job is triggered (the *kernel-netlink* plugin delays creating roam jobs for 100 ms). However, at that point it appears to the *bypass-lan* plugin as if there haven't been any changes as the host is still connected to the exact same subnets. That's because it currently doesn't consider the interfaces when checking for changes. Interestingly, the initial version of the plugin actually cached the interface names. However, it didn't use them for comparison, only to set on the policies. But since that caused problems when uninstalling them I removed that with c2129d1cbe.

The problem with such a subnet switch (is that something common? or even naturally occurring?) is that there currently is no interface to update shunt policies (installing them overlapping also doesn't work). So the only option would be to remove and readd them, but that'd require quite some work in the plugin just to handle something that seems like a rare edge case to me.

Could you give some more information on this scenario/use case?

#4 - 08.11.2018 12:49 - Tobias Brunner

There is something else to consider. From an IPsec policy's perspective there really is no change if there is such a switch (it is not associated with any interface). The routes for shunts are also only necessary if there are IPsec tunnels that install conflicting routes (which might or might not be necessary, depending on whether virtual IPs are used and/or other routes match traffic). So maybe not installing any routes at all could be an option in your scenario (via *charon.install_routes*). The routes are also neither installed by the plugin, nor by the shunt-manager, but instead by the *kernel-netlink* plugin together with the OUT policy. So maybe a better approach would be to just automatically update the installed routes if the interface changes (but probably also requires quite some work).

#5 - 08.11.2018 23:21 - Rob Cornall

Thanks for the response,

Yes agreed it is an uncommon scenario :) . For our particular case we are setting up some different interfaces for some testing, but also in a real-world scenario it is *possible* for a user to decide to swap subnets on 2 interfaces at once, but very unlikely to happen - so not a major concern right now.

We want to use lan bypass for full-tunnel setups, and we also are supporting virtual IP for which I believe the install_routes option we probably want to keep.

#6 - 09.11.2018 07:47 - Tobias Brunner

- Tracker changed from Issue to Bug

- Subject changed from lan bypass: routes installed incorrectly in table 220 when quickly changing ip addresses to lan bypass: routes installed incorrectly in table 220 when quickly swapping IP addresses/subnets on two interfaces

> For our particular case we are setting up some different interfaces for some testing, but also in a real-world scenario it is *possible* for a user to decide to swap subnets on 2 interfaces at once, but very unlikely to happen - so not a major concern right now.

Sure, a user can theoretically do all kinds of weird stuff :) I guess if this happens manually, the plugin should have enough time to react to the changes.

> We want to use lan bypass for full-tunnel setups, and we also are supporting virtual IP for which I believe the install_routes option we probably want to keep.

Yes, with virtual IPs you require the routes to force that IP as source address.

As mentioned before, we could remove and reinstall the policies if the interface changes (not really nice, but it should work). I pushed a commit that does that to the *2820-bypass-lan-interface* branch (only compile tested). Let me know what you think.

#7 - 13.11.2018 22:57 - Rob Cornall

I have tried out the workaround and it seems to work fine - thanks for that.

I think we would prefer waiting for an upstream fix to use though, if you plan on pushing a fix in upstream?

#8 - 14.11.2018 12:14 - Tobias Brunner

- Assignee set to Tobias Brunner

- Target version set to 5.7.2

> I think we would prefer waiting for an upstream fix to use though, if you plan on pushing a fix in upstream?

I can line up that workaround for the next release.

#9 - 22.11.2018 11:41 - Tobias Brunner

- Status changed from Feedback to Closed

- Resolution set to Fixed

Files

charon_debug_reproduced.log	9.24 KB	07.11.2018	Rob Cornall
charon_debug_with_sleeps.log	12.1 KB	07.11.2018	Rob Cornall
strongswan.conf	917 Bytes	07.11.2018	Rob Cornall
bypass-lan.conf	497 Bytes	07.11.2018	Rob Cornall