strongSwan - Bug #824

kernel_netlink plugin decides on wrong interface for route

17.01.2015 10:57 - Jan Engelhardt

Status:	Closed	**Start date:**	17.01.2015
Priority:	Normal	**Due date:**	
Assignee:	Tobias Brunner	**Estimated time:**	0.00 hour
Category:	kernel-interface		
Target version:	5.5.0		
Affected version:	5.1.3	**Resolution:**	Fixed

Description

If an IP address is added to more than one interface, the logic in /strongswan-5.1.3/src/libhydra/plugins/kernel_netlink determines the incorrect interface to use for the route it will add to table 220.

Consider this setup:

```
# ip a
2: enp4s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen
 1000
    inet6 2a01:db8:42::60/128 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::5604:a6ff:fef1:7b28/64 scope link
       valid_lft forever preferred_lft forever
3: tapvbox0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default ql
en 500
    inet6 2a01:db8:42::60/64 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::64ba:7fff:fe2d:8b80/64 scope link
       valid_lft forever preferred_lft forever

# ip -6 r l
2a01:db8:42::60 dev enp4s0  proto kernel  metric 256
2a01:db8:42::/64 dev tapvbox0  proto kernel  metric 256
fe80::/64 dev enp4s0  proto kernel  metric 256
fe80::/64 dev tapvbox0  proto kernel  metric 256
default via fe80::1 dev enp4s0  metric 1024

#/etc/ipsec.conf
conn abc
        leftid=@x60
        rightid=@x0f
        left=2a01:db8:42::60
        right=2a01:db8:99::f
        leftsubnet=2a01:db8:42::/64
        rightsubnet=2a01:db8:99::/64
```

Now, when charon is about to configure the routes on this host,
it revelas, with sufficient debugging (knl 2) turned on,

```
11[KNL] getting a local address in traffic selector 2a01:db8:42::/64
11[KNL] using host 2a01:db8:42::60
11[KNL] using fe80::1 as nexthop to reach 2a01:db8:99::f
11[KNL] 2a01:db8:42::60 is on interface tapvbox0
                                       ^^^^^^^^

11[KNL] installing route: 2a01:db8:99::/64 via fe80::1
        src 2a01:db8:42::60 dev tapvbox0
                            ^^^^^^^^
```

Well yes, '60 is on tapvbox0. But it is not the interface I had in mind. In particular, because 2a01:db8:99::f is reachable through

enp4s0, not tapvbox0.

I feel calling the function which figures out "'60 is on interface tapvbox0" is wrong. Instead, the interface to use is reported in the response of the RTM_GETROUTE call with which the nexthop was determined earlier.

```
11[KNL] getting a local address in traffic selector 2a01:db8:42::/64
```

(I would also hope charon chooses the local address based upon the value I specified in left= in ipsec.conf rather than go look for which address in the system is within one of the TS subnets.)

Related issues:

Related to Bug #809: [KNL] unable to install source route for 213.a.b.41	**Closed**	**29.12.2014**
Related to Bug #1347: Route won't be created for passthrough with subnet othe...	**Closed**	**11.03.2016**

Associated revisions

Revision 96b1fab5 - 10.06.2016 18:15 - Tobias Brunner

Merge branch 'interface-for-routes'

Changes how the interface for routes installed with policies is
determined. In most cases we now use the interface over which we reach the
other peer, not the interface on which the local address (or the source IP) is
installed. However, that might be the same interface depending on the
configuration (i.e. in practice there will often not be a change).

Routes are not installed anymore for drop policies and for policies with
protocol/port selectors.

Fixes #809, #824, #1347.

History

#1 - 11.02.2015 16:30 - Tobias Brunner

- Tracker changed from Bug to Issue

- Status changed from New to Feedback

> I feel calling the function which figures out "'60 is on interface tapvbox0" is wrong. Instead, the interface to use is reported in the response of the RTM_GETROUTE call with which the nexthop was determined earlier.

The function get_interface_name, which looks up the name of an interface based on an IP, currently simply searches for the first interface it knows about that has this IP installed. So this is not related at all to the route lookup used to determine the next hop. In this case it would have been nice if get_nexthop() returned the interface name as well, but currently these functions/calls are completely unrelated, which hasn't been an issue so far. Might be something to consider in the future.

As a workaround you could maybe define *charon.interfaces_ignore=tapvbox0* in strongswan.conf, what way the other interface is searched first. If you don't need to send and receive IKE packets over tapvbox0 that should work fine.

> (I would also hope charon chooses the local address based upon the value I specified in left= in ipsec.conf rather than go look for which address in the system is within one of the TS subnets.)

left and the TS in *leftsubnet* are not really related. *left* is used as source address for IKE traffic and a route lookup is done to determine it (if *left* is configured that address is preferred, but if it is not found any other usable address to reach *right* is used). When determining the source address for the route that is installed with the IPsec SA, charon just uses the first address it finds that is contained in the TS (it prefers virtual IPs it installed itself over other addresses).

#2 - 17.02.2015 17:40 - Tobias Brunner

- Related to Bug #809: [KNL] unable to install source route for 213.a.b.41 added

#3 - 11.03.2016 19:31 - Tobias Brunner

- Related to Bug #1347: Route won't be created for passthrough with subnet other than /24 added

#4 - 10.06.2016 18:34 - Tobias Brunner

- Tracker changed from Issue to Bug

- Category changed from libhydra to kernel-interface

- Status changed from Feedback to Closed

- Assignee set to Tobias Brunner

- Target version set to 5.5.0

- Resolution set to Fixed

This is believed to be fixed. Please open a new ticket if you still find there is an issue.