strongSwan - Feature #2845

Support virtual XFRM interfaces added in 4.19

30.11.2018 18:03 - Stijn Tintel

Status:	Closed	**Start date:**	30.11.2018
Priority:	Normal	**Due date:**	
Assignee:	Tobias Brunner	**Estimated time:**	0.00 hour
Category:	libcharon		
Target version:	5.8.0		
Resolution:	Fixed		

Description

See https://www.mail-archive.com/netdev@vger.kernel.org/msg239295.html
Would be nice if this can be supported in strongSwan.

Associated revisions

Revision c334cd9f - 04.04.2019 10:41 - Tobias Brunner

Merge branch 'xfrmi'

This adds support for XFRM interfaces, which replace VTI devices and are
available with 4.19+ Linux kernels.

IPsec SAs and policies are associated with such interfaces via interface
IDs that can be configured on the CHILD_SA-level (dynamic IDs may
optionally be allocated for each instance and even direction) or on the
IKE_SA-level (again, dynamic IDs may be optionally allocated per IKE_SA).
IDs on an IKE_SA are inherited by all CHILD_SAs created under it, unless
the child configuration overrides them.

The effect the interface ID has on policies is similar to that of marks,
i.e. they won't match packets unless they are routed via interface with
matching interface ID. So it's possible to negotiate e.g. 0.0.0.0/0 as
traffic selector on both sides and then control the affected traffic via
routes/firewall.

It's possible to use separate interfaces for in- and outbound traffic (or
only use an interface in one direction and regular policies in the other).

Since iproute2 does not yet support XFRM interfaces, a small utility is
provided that allows creating and listing XFRM interfaces.

Interfaces may be created dynamically via updown/vici scripts or
statically (before or after establishing the SAs). Routes must be added
manually as needed (the daemon will not install any routes for outbound
policies with an interface ID).

When moving XFRM interfaces to other network namespaces they retain access
to the SAs and policies created in the original namespace, which allows
providing IPsec tunnels for processes in other network namespaces without
giving them access to the IPsec keys or IKE credentials.

Fixes #2845.

History

#1 - 11.12.2018 20:06 - Roman Moschenski

Stijn Tintel wrote:

> See https://www.mail-archive.com/netdev@vger.kernel.org/msg239295.html
> Would be nice if this can be supported in strongSwan.

Would be great to have a strongswan support for this new kernel feature. If I understand that correctly, we have to extend **add_policy_internal** function (https://github.com/strongswan/strongswan/blob/master/src/libcharon/plugins/kernel_netlink/kernel_netlink_ipsec.c#L2669) of the kernel_netlink plugin by passing a XFRMA_IF_ID beside of XFRMA_TMPL. That would add an ID of our virtual interface to the corresponding XFRM policy in the kernel:

https://github.com/torvalds/linux/blob/master/net/xfrm/xfrm_user.c#L1620
This feature could be a compile time flag to stay backwards compatible with older kernels or with kernels compiled with CONFIG_XFRM_INTERFACE=n

What do you think Tobias?

#2 - 13.12.2018 09:03 - Martin Willi

I have some plans to add XFRM interfaces support to strongSwan. No code and no schedule yet, though.

I guess the technical details about installing policies tied to interfaces are rather clear.
From a configuration perspective, we probably want to configure a (shared?) XFRM interface per CHILD_SA.

The question is what are the use cases and who is responsible for creating these interfaces (externally or strongSwan):

- One of my use cases is certainly layer 3 master devices, to bind specific tunnels to certain VRF domains
- Another nice use case could be passing XFRM interfaces into network namespaces, so a host can provide secured links to its containers
- Most users probably just want IPsec interfaces for simplified firewalling

#3 - 22.01.2019 07:47 - Markus Sattler

I created some basic patch to allow to set the XFRMA_IF_ID via config.

the PR can be found here:
https://github.com/strongswan/strongswan/pull/122

#4 - 12.02.2019 18:29 - Tobias Brunner

- Status changed from New to Feedback

I pushed some patches to the *2845-xfrmi* branch. They include a small utility that allows creating XFRM interfaces on newer kernels without having to patch iproute2.

I also did some tests, with the following findings:

- We definitely want to set the interface ID on the SAs as well as on the policies because that...
  - ...allows duplicate policies (the functionality in the kernel is similar to that of marks, two policies can use the same selector as long as their interface ID is different)
  - ...lets inbound traffic pass. Without matching inbound policy (including the interface ID) traffic is otherwise dropped...
    - ...unless *net.<ipv4|6>.conf.<xfrm-if>.disable_policy=1* is set to disable the inbound policy check on the XFRM interface (but that seems needlessly open)
  - ...prevents that the policies affect *all* matching traffic. Only with an ID are they specific to traffic routed to the associated XFRM interface (again that's similar to marks, where policies without them affect all packets and with only those that were previously marked)
- It's possible to use different interfaces and IDs for in- and outbound traffic
  - That's why I added two configuration options and the ability to allocate unique interface IDs per CHILD_SA and per direction (similar to the in-/outbound marks)
- It doesn't matter if a matching XFRM interface already exists when SAs are installed with a particular interface ID
  - That's why CHILD_SA-specific interfaces can easily be created in updown/vici scripts as long as the potentially unique interface ID is known (patches for this are included in the branch)
- Routing traffic that doesn't match a policy to an XFRM interface has the following effect:
  - outbound: *dropped* counter of the interface increases, no errors in /proc/net/xfrm_stat
  - inbound: *error/dropped* counter does not increase, however, XfrmInNoPols in /proc/net/xfrm_stat does, unless the policy check is disabled (see above)
- The physical interface that's associated with an XFRM interface doesn't seem to matter much (at least in our testing environment, where eth1 could be used on sun/moon even though the ESP packets are then exchanged over eth0)
  - **Edit**: If an outbound interface is configured in the child config, this is enforced (i.e. traffic is dropped if the associated physical interface does not match that of the policy)
- The same XFRM interface can be shared by multiple SAs/policies (e.g. for all CHILD_SAs created by multiple roadwarriors)
- Using trap policies in combination with XFRM interfaces is possible
- IPComp can be used

> Another nice use case could be passing XFRM interfaces into network namespaces, so a host can provide secured links to its containers

I tried that. However, it currently only works partially (4.20 kernel). While it works fine for outbound traffic, it currently doesn't for inbound traffic. It seems the XFRM interface is stored in a list that's associated with the original netns and for inbound traffic the lookup uses the current netns (i.e. the one the interface was moved to), so no XFRM interface is found and the packet is dropped. So it looks like this requires a kernel patch.

> One of my use cases is certainly layer 3 master devices, to bind specific tunnels to certain VRF domains

How would such a scenario look like exactly?

#5 - 13.02.2019 14:02 - Tobias Brunner

- File xfrm-interface-cross-ns.patch added

Attached a hackish kernel patch for the cross network namespace issue. **Edit**: A better patch can be found below.

#6 - 13.02.2019 14:27 - Martin Willi

> I pushed some patches to the *2845-xfrmi* branch. [...]
> I also did some tests, with the following findings:

Thanks for your work on this, and for sharing your findings. Sounds promising so far.

> > One of my use cases is certainly layer 3 master devices, to bind specific tunnels to certain VRF domains

> How would such a scenario look like exactly?

The idea is to bind one or more tunnels to a VRF domain using layer 3 master devices. This would allow proper separation of traffic flows, for example when using a multi-tenancy capable IPsec gateway. So for each customer, one would have an xfrmi associated to a dedicated physical or VLAN internal interface, where traffic is forwarded to/from:

```
          +-------+  +-----------+  +--------+
external  | xfrmi |--| L3 master |--| VLAN-X | internal network
encrypted +-------+  +-----------+  +--------+ plain
                           |
                    +------------+
                    | associated |
                    |  routing   |
                    |   table    |
                    +------------+
```

As the customer tunnel configuration defines the xfrmi and indirectly the VRF domain, we can provide IPsec for multiple customers using a single public IP. Also, the different customers can even use overlapping subnets.

I guess this can be achieved also by namespacing the xfrmi and the associated internal interface, but VRFs might be somewhat simpler from an administration perspective.

#7 - 13.02.2019 14:44 - Roman Moschenski

Great work! Have I to use vici/swanctl.conf ? No way to add new parameters (if_id_in, if_id_out) in ipsec.conf ?

#8 - 13.02.2019 14:47 - Tobias Brunner

Thanks for the explanation Martin. I'd have to look into that VRF stuff at some point.

> Have I to use vici/swanctl.conf ? No way to add new parameters (if_id_in, if_id_out) in ipsec.conf ?

Yes and no (ipsec.conf is deprecated, no new features will be added).

#9 - 18.02.2019 10:24 - Tobias Brunner

- File xfrm-interface-cross-ns-v2.patch added

Better kernel patch for the cross namespace issue (also against 4.20, ~~for 5.0 the skb->sp check has to be replaced with secpath_exists()~~). **Edit:** The 5.0 kernel contains this patch. For 4.19 and 4.20 kernels it's still required to fix this issue (unless it gets backported).

#10 - 18.02.2019 12:53 - Roman Moschenski

Do you plan to include these new patches in 2845-xfrmi branch in to a new strongswan release?

#11 - 19.03.2019 11:23 - Eyal Birger

For a responder it would be nice if there were an option for the if_id to be allocated per IKE SA so that all child SAs of the same IKE SA could share the same if_id.

#12 - 26.03.2019 18:37 - Tobias Brunner

- Category set to libcharon

- Assignee set to Tobias Brunner

- Target version set to 5.8.0

- Resolution set to Fixed

> For a responder it would be nice if there were an option for the if_id to be allocated per IKE SA so that all child SAs of the same IKE SA could share the same if_id.

I've added this feature in the branch. If multiple CHILD_SAs share such an interface, the creation either has to be done via *ike-updown* vici event or with refcounting in a classic updown script (which is called for each CHILD_SA, actually for each combination of local and remote TS).

#13 - 04.04.2019 11:05 - Tobias Brunner

- Status changed from Feedback to Closed

Files

xfrm-interface-cross-ns.patch	1.18 KB	13.02.2019	Tobias Brunner
xfrm-interface-cross-ns-v2.patch	964 Bytes	18.02.2019	Tobias Brunner