IPsec Workshop

Steffen Klassert

secunet Security Networks AG

Dresden

Netdev 2.2 Seoul, November 8, 2017

IPsec status update (what happened since netdev 1.2)

IPsec offload status update for Intel Niantic, Shannon Nelson

The intel implementation of the IPsec offload support, Don Skidmore

Describe trailer removal, Describe ESN, show performance numbers, Boris Pismenny

Open discussion

Avoid frame copy in skb_cow_data

- **Netdev 1.2 Problem:** Most of the ESP data frames are linearized with skb_cow_data.
- **Netdev 1.2 Solved TX:** Use separate src and dst buffers for crypto operations, RFC code exists.
- **Netdev 1.2 Solved RX:** Linearize only if the buffer is not writable, RFC code exists.
- **Feb 2017:** Proposed solution was merged.
- **Netdev 1.2 New Problem:** Some crypto algorithm implementations linearize if a nonlinear buffer is passed in (gcm-aesni).
- **Nov 2017 Problem:** Some crypto algorithm implementations linearize if a nonlinear buffer is passed in (gcm-aesni).

Avoid frame copy in skb_cow_data

- **Netdev 1.2 Problem:** Most of the ESP data frames are linearized with skb_cow_data.
- **Netdev 1.2 Solved TX:** Use separate src and dst buffers for crypto operations, RFC code exists.
- **Netdev 1.2 Solved RX:** Linearize only if the buffer is not writable, RFC code exists.
- **Feb 2017:** Proposed solution was merged.
- **Netdev 1.2 New Problem:** Some crypto algorithm implementations linearize if a nonlinear buffer is passed in (gcm-aesni).
- **Nov 2017 Problem:** Some crypto algorithm implementations linearize if a nonlinear buffer is passed in (gcm-aesni).

Avoid frame copy in skb_cow_data

- **Netdev 1.2 Problem:** Most of the ESP data frames are linearized with skb_cow_data.
- **Netdev 1.2 Solved TX:** Use separate src and dst buffers for crypto operations, RFC code exists.
- **Netdev 1.2 Solved RX:** Linearize only if the buffer is not writable, RFC code exists.
- **Feb 2017:** Proposed solution was merged.
- **Netdev 1.2 New Problem:** Some crypto algorithm implementations linearize if a nonlinear buffer is passed in (gcm-aesni).
- **Nov 2017 Problem:** Some crypto algorithm implementations linearize if a nonlinear buffer is passed in (gcm-aesni).

Avoid frame copy in skb_cow_data

- **Netdev 1.2 Problem:** Most of the ESP data frames are linearized with skb_cow_data.
- **Netdev 1.2 Solved TX:** Use separate src and dst buffers for crypto operations, RFC code exists.
- **Netdev 1.2 Solved RX:** Linearize only if the buffer is not writable, RFC code exists.
- **Feb 2017:** Proposed solution was merged.
- **Netdev 1.2 New Problem:** Some crypto algorithm implementations linearize if a nonlinear buffer is passed in (gcm-aesni).
- **Nov 2017 Problem:** Some crypto algorithm implementations linearize if a nonlinear buffer is passed in (gcm-aesni).

Avoid frame copy in skb_cow_data

- **Netdev 1.2 Problem:** Most of the ESP data frames are linearized with skb_cow_data.
- **Netdev 1.2 Solved TX:** Use separate src and dst buffers for crypto operations, RFC code exists.
- **Netdev 1.2 Solved RX:** Linearize only if the buffer is not writable, RFC code exists.
- **Feb 2017:** Proposed solution was merged.
- **Netdev 1.2 New Problem:** Some crypto algorithm implementations linearize if a nonlinear buffer is passed in (gcm-aesni).
- **Nov 2017 Problem:** Some crypto algorithm implementations linearize if a nonlinear buffer is passed in (gcm-aesni).

Avoid frame copy in skb_cow_data

- **Netdev 1.2 Problem:** Most of the ESP data frames are linearized with skb_cow_data.
- **Netdev 1.2 Solved TX:** Use separate src and dst buffers for crypto operations, RFC code exists.
- **Netdev 1.2 Solved RX:** Linearize only if the buffer is not writable, RFC code exists.
- **Feb 2017:** Proposed solution was merged.
- **Netdev 1.2 New Problem:** Some crypto algorithm implementations linearize if a nonlinear buffer is passed in (gcm-aesni).
- **Nov 2017 Problem:** Some crypto algorithm implementations linearize if a nonlinear buffer is passed in (gcm-aesni).

Adding a GRO codepath for IPsec.

- **Netdev 1.2:** Add GRO handlers for the IPsec protocols, RFC code exists.
- **Feb 2017:** xfrm_input got support to use gro_cells
- **Apr 2017:** Decapsulation/decryption at the GRO Layer (L2) was merged.
- **Solved:** IPsec GRO works now.

Adding a GRO codepath for IPsec.

- **Netdev 1.2:** Add GRO handlers for the IPsec protocols, RFC code exists.
- **Feb 2017:** xfrm_input got support to use gro_cells
- **Apr 2017:** Decapsulation/decryption at the GRO Layer (L2) was merged.
- **Solved:** IPsec GRO works now.

Adding a GRO codepath for IPsec.

- **Netdev 1.2:** Add GRO handlers for the IPsec protocols, RFC code exists.
- **Feb 2017:** xfrm_input got support to use gro_cells
- **Apr 2017:** Decapsulation/decryption at the GRO Layer (L2) was merged.
- **Solved:** IPsec GRO works now.

Adding a GRO codepath for IPsec.

- **Netdev 1.2:** Add GRO handlers for the IPsec protocols, RFC code exists.
- **Feb 2017:** xfrm_input got support to use gro_cells
- **Apr 2017:** Decapsulation/decryption at the GRO Layer (L2) was merged.
- **Solved:** IPsec GRO works now.

Adding a GRO codepath for IPsec.

- **Netdev 1.2:** Add GRO handlers for the IPsec protocols, RFC code exists.
- **Feb 2017:** xfrm_input got support to use gro_cells
- **Apr 2017:** Decapsulation/decryption at the GRO Layer (L2) was merged.
- **Solved:** IPsec GRO works now.

Adding a GSO codepath for IPsec.

- **Netdev 1.2:** Move the existing xfrm GSO handling from xfrm to the generic GSO layer (L2).
- **Netdev 1.2:** Works if we can offload the crypto operations to a NIC, RFC code exists.
- **Apr 2017:** Code was merged
- **Sloved:** GSO works for hardware offload.
- **Netdev 1.2:** Does not work on software crypto.
- **Netdev 1.2 Problem:** We can't handle asynchronous crypto operations in the GSO layer.
- **Netdev 1.2 Deferred:** Still no solution for software crypto.
- **Nov 2017:** Solution exists, still in RFC state.

Adding a GSO codepath for IPsec.

- **Netdev 1.2:** Move the existing xfrm GSO handling from xfrm to the generic GSO layer (L2).
- **Netdev 1.2:** Works if we can offload the crypto operations to a NIC, RFC code exists.
- **Apr 2017:** Code was merged
- **Sloved:** GSO works for hardware offload.
- **Netdev 1.2:** Does not work on software crypto.
- **Netdev 1.2 Problem:** We can't handle asynchronous crypto operations in the GSO layer.
- **Netdev 1.2 Deferred:** Still no solution for software crypto.
- **Nov 2017:** Solution exists, still in RFC state.

Adding a GSO codepath for IPsec.

- **Netdev 1.2:** Move the existing xfrm GSO handling from xfrm to the generic GSO layer (L2).
- **Netdev 1.2:** Works if we can offload the crypto operations to a NIC, RFC code exists.
- **Apr 2017:** Code was merged
- **Sloved:** GSO works for hardware offload.
- **Netdev 1.2:** Does not work on software crypto.
- **Netdev 1.2 Problem:** We can't handle asynchronous crypto operations in the GSO layer.
- **Netdev 1.2 Deferred:** Still no solution for software crypto.
- **Nov 2017:** Solution exists, still in RFC state.

Adding a GSO codepath for IPsec.

- **Netdev 1.2:** Move the existing xfrm GSO handling from xfrm to the generic GSO layer (L2).
- **Netdev 1.2:** Works if we can offload the crypto operations to a NIC, RFC code exists.
- **Apr 2017:** Code was merged
- **Sloved:** GSO works for hardware offload.
- **Netdev 1.2:** Does not work on software crypto.
- **Netdev 1.2 Problem:** We can't handle asynchronous crypto operations in the GSO layer.
- **Netdev 1.2 Deferred:** Still no solution for software crypto.
- **Nov 2017:** Solution exists, still in RFC state.

Adding a GSO codepath for IPsec.

- **Netdev 1.2:** Move the existing xfrm GSO handling from xfrm to the generic GSO layer (L2).
- **Netdev 1.2:** Works if we can offload the crypto operations to a NIC, RFC code exists.
- **Apr 2017:** Code was merged
- **Sloved:** GSO works for hardware offload.
- **Netdev 1.2:** Does not work on software crypto.
- **Netdev 1.2 Problem:** We can't handle asynchronous crypto operations in the GSO layer.
- **Netdev 1.2 Deferred:** Still no solution for software crypto.
- **Nov 2017:** Solution exists, still in RFC state.

Adding a GSO codepath for IPsec.

- **Netdev 1.2:** Move the existing xfrm GSO handling from xfrm to the generic GSO layer (L2).
- **Netdev 1.2:** Works if we can offload the crypto operations to a NIC, RFC code exists.
- **Apr 2017:** Code was merged
- **Sloved:** GSO works for hardware offload.
- **Netdev 1.2:** Does not work on software crypto.
- **Netdev 1.2 Problem:** We can't handle asynchronous crypto operations in the GSO layer.
- **Netdev 1.2 Deferred:** Still no solution for software crypto.
- **Nov 2017:** Solution exists, still in RFC state.

Adding a GSO codepath for IPsec.

- **Netdev 1.2:** Move the existing xfrm GSO handling from xfrm to the generic GSO layer (L2).
- **Netdev 1.2:** Works if we can offload the crypto operations to a NIC, RFC code exists.
- **Apr 2017:** Code was merged
- **Sloved:** GSO works for hardware offload.
- **Netdev 1.2:** Does not work on software crypto.
- **Netdev 1.2 Problem:** We can't handle asynchronous crypto operations in the GSO layer.
- **Netdev 1.2 Deferred:** Still no solution for software crypto.
- **Nov 2017:** Solution exists, still in RFC state.

Adding a GSO codepath for IPsec.

- **Netdev 1.2:** Move the existing xfrm GSO handling from xfrm to the generic GSO layer (L2).
- **Netdev 1.2:** Works if we can offload the crypto operations to a NIC, RFC code exists.
- **Apr 2017:** Code was merged
- **Sloved:** GSO works for hardware offload.
- **Netdev 1.2:** Does not work on software crypto.
- **Netdev 1.2 Problem:** We can't handle asynchronous crypto operations in the GSO layer.
- **Netdev 1.2 Deferred:** Still no solution for software crypto.
- **Nov 2017:** Solution exists, still in RFC state.

Adding a GSO codepath for IPsec.

- **Netdev 1.2:** Move the existing xfrm GSO handling from xfrm to the generic GSO layer (L2).
- **Netdev 1.2:** Works if we can offload the crypto operations to a NIC, RFC code exists.
- **Apr 2017:** Code was merged
- **Sloved:** GSO works for hardware offload.
- **Netdev 1.2:** Does not work on software crypto.
- **Netdev 1.2 Problem:** We can't handle asynchronous crypto operations in the GSO layer.
- **Netdev 1.2 Deferred:** Still no solution for software crypto.
- **Nov 2017:** Solution exists, still in RFC state.

Adding IPsec HW offload support

- **Netdev 1.2:** An IPsec HW offload API was created, RFC code exists.
- **Apr 2017:** The IPsec HW offload API was merged.
- **Solved:** IPsec hardware offload is now fully implemented (API + GRO/GSO).
- **Netdev 1.2:** The Mellanox mlx5 driver uses the API, RFC code exists.
- **Jun 2017:** The Mellanox mlx5 IPsec offload was merged.
- **New:** Intel works on IPsec offload for their hardware → Don Skidmore.
- **New:** Oracle works on the the ixgbe driver to get IPsec offload for the Niantic 10G NICs → Shannon Nelson.

Adding IPsec HW offload support

- **Netdev 1.2:** An IPsec HW offload API was created, RFC code exists.
- **Apr 2017:** The IPsec HW offload API was merged.
- **Solved:** IPsec hardware offload is now fully implemented (API + GRO/GSO).
- **Netdev 1.2:** The Mellanox mlx5 driver uses the API, RFC code exists.
- **Jun 2017:** The Mellanox mlx5 IPsec offload was merged.
- **New:** Intel works on IPsec offload for their hardware → Don Skidmore.
- **New:** Oracle works on the the ixgbe driver to get IPsec offload for the Niantic 10G NICs → Shannon Nelson.

Adding IPsec HW offload support

- **Netdev 1.2:** An IPsec HW offload API was created, RFC code exists.
- **Apr 2017:** The IPsec HW offload API was merged.
- **Solved:** IPsec hardware offload is now fully implemented (API + GRO/GSO).
- **Netdev 1.2:** The Mellanox mlx5 driver uses the API, RFC code exists.
- **Jun 2017:** The Mellanox mlx5 IPsec offload was merged.
- **New:** Intel works on IPsec offload for their hardware → Don Skidmore.
- **New:** Oracle works on the the ixgbe driver to get IPsec offload for the Niantic 10G NICs → Shannon Nelson.

Adding IPsec HW offload support

- **Netdev 1.2:** An IPsec HW offload API was created, RFC code exists.
- **Apr 2017:** The IPsec HW offload API was merged.
- **Solved:** IPsec hardware offload is now fully implemented (API + GRO/GSO).
- **Netdev 1.2:** The Mellanox mlx5 driver uses the API, RFC code exists.
- **Jun 2017:** The Mellanox mlx5 IPsec offload was merged.
- **New:** Intel works on IPsec offload for their hardware → Don Skidmore.
- **New:** Oracle works on the the ixgbe driver to get IPsec offload for the Niantic 10G NICs → Shannon Nelson.

Adding IPsec HW offload support

- **Netdev 1.2:** An IPsec HW offload API was created, RFC code exists.
- **Apr 2017:** The IPsec HW offload API was merged.
- **Solved:** IPsec hardware offload is now fully implemented (API + GRO/GSO).
- **Netdev 1.2:** The Mellanox mlx5 driver uses the API, RFC code exists.
- **Jun 2017:** The Mellanox mlx5 IPsec offload was merged.
- **New:** Intel works on IPsec offload for their hardware → Don Skidmore.
- **New:** Oracle works on the the ixgbe driver to get IPsec offload for the Niantic 10G NICs → Shannon Nelson.

Adding IPsec HW offload support

- **Netdev 1.2:** An IPsec HW offload API was created, RFC code exists.
- **Apr 2017:** The IPsec HW offload API was merged.
- **Solved:** IPsec hardware offload is now fully implemented (API + GRO/GSO).
- **Netdev 1.2:** The Mellanox mlx5 driver uses the API, RFC code exists.
- **Jun 2017:** The Mellanox mlx5 IPsec offload was merged.
- **New:** Intel works on IPsec offload for their hardware → Don Skidmore.
- **New:** Oracle works on the the ixgbe driver to get IPsec offload for the Niantic 10G NICs → Shannon Nelson.

Adding IPsec HW offload support

- **Netdev 1.2:** An IPsec HW offload API was created, RFC code exists.
- **Apr 2017:** The IPsec HW offload API was merged.
- **Solved:** IPsec hardware offload is now fully implemented (API + GRO/GSO).
- **Netdev 1.2:** The Mellanox mlx5 driver uses the API, RFC code exists.
- **Jun 2017:** The Mellanox mlx5 IPsec offload was merged.
- **New:** Intel works on IPsec offload for their hardware → Don Skidmore.
- **New:** Oracle works on the the ixgbe driver to get IPsec offload for the Niantic 10G NICs → Shannon Nelson.

IPsec flowcache removal

- **Netdev 1.2:** Discussion about the flowcache removal, RFC code exists (Florian Westphal).
- **Jul 2017:** The flowcache removal was merged.
- **Solved:** The DoS problem with the flowcache went away.
- **New problem:** The flowcache provided a fast lookup for policies and SAs, lookups might be slow now.
- **Deferred:** No solution so far.

IPsec flowcache removal

- **Netdev 1.2:** Discussion about the flowcache removal, RFC code exists (Florian Westphal).
- **Jul 2017:** The flowcache removal was merged.
- **Solved:** The DoS problem with the flowcache went away.
- **New problem:** The flowcache provided a fast lookup for policies and SAs, lookups might be slow now.
- **Deferred:** No solution so far.

IPsec flowcache removal

- **Netdev 1.2:** Discussion about the flowcache removal, RFC code exists (Florian Westphal).
- **Jul 2017:** The flowcache removal was merged.
- **Solved:** The DoS problem with the flowcache went away.
- **New problem:** The flowcache provided a fast lookup for policies and SAs, lookups might be slow now.
- **Deferred:** No solution so far.

IPsec flowcache removal

- **Netdev 1.2:** Discussion about the flowcache removal, RFC code exists (Florian Westphal).
- **Jul 2017:** The flowcache removal was merged.
- **Solved:** The DoS problem with the flowcache went away.
- **New problem:** The flowcache provided a fast lookup for policies and SAs, lookups might be slow now.
- **Deferred:** No solution so far.

IPsec flowcache removal

- **Netdev 1.2:** Discussion about the flowcache removal, RFC code exists (Florian Westphal).
- **Jul 2017:** The flowcache removal was merged.
- **Solved:** The DoS problem with the flowcache went away.
- **New problem:** The flowcache provided a fast lookup for policies and SAs, lookups might be slow now.
- **Deferred:** No solution so far.

IPsec flowcache removal

- **Netdev 1.2:** Discussion about the flowcache removal, RFC code exists (Florian Westphal).
- **Jul 2017:** The flowcache removal was merged.
- **Solved:** The DoS problem with the flowcache went away.
- **New problem:** The flowcache provided a fast lookup for policies and SAs, lookups might be slow now.
- **Deferred:** No solution so far.

IPsec offload status update for Intel Niantic, Shannon Nelson

The intel implementation of the IPsec offload support, Don Skidmore

Describe trailer removal, Describe ESN, show performance numbers, Boris Pismenny

Open discussion

Redesigning the IPsec VTI interfaces

Disadvantages of IPsec VTI interfaces

- VTI interfaces are L3 tunnel interfaces with configurable tunnel endpoints.
  - The tunnel endpoints are already determined by the SA.
  - Configuring tunnel endpoints at a VTI does not make much sense.
- Only one VTI with wildcard tunnel endpoints can be configured.
  - Problematic if you need more than one (e.g. for namespaces).
- VTI is configured with a combination of GRE keys and routing marks.
  - Neither GRE keys nor routing marks were designated to configure a VTI.
  - Routing by mark does not work well with VTI.
- VTI works just with tunnel mode SAs.
  - Not an interface to route transport or beet mode.

Disadvantages of IPsec VTI interfaces

- VTI interfaces are L3 tunnel interfaces with configurable tunnel endpoints.
  - The tunnel endpoints are already determined by the SA.
  - Configuring tunnel endpoints at a VTI does not make much sense.
- Only one VTI with wildcard tunnel endpoints can be configured.
  - Problematic if you need more than one (e.g. for namespaces).
- VTI is configured with a combination of GRE keys and routing marks.
  - Neither GRE keys nor routing marks were designated to configure a VTI.
  - Routing by mark does not work well with VTI.
- VTI works just with tunnel mode SAs.
  - Not an interface to route transport or beet mode.

Disadvantages of IPsec VTI interfaces

- VTI interfaces are L3 tunnel interfaces with configurable tunnel endpoints.
  - The tunnel endpoints are already determined by the SA.
  - Configuring tunnel endpoints at a VTI does not make much sense.
- Only one VTI with wildcard tunnel endpoints can be configured.
  - Problematic if you need more than one (e.g. for namespaces).
- VTI is configured with a combination of GRE keys and routing marks.
  - Neither GRE keys nor routing marks were designated to configure a VTI.
  - Routing by mark does not work well with VTI.
- VTI works just with tunnel mode SAs.
  - Not an interface to route transport or beet mode.

Disadvantages of IPsec VTI interfaces

- VTI interfaces are L3 tunnel interfaces with configurable tunnel endpoints.
  - The tunnel endpoints are already determined by the SA.
  - Configuring tunnel endpoints at a VTI does not make much sense.
- Only one VTI with wildcard tunnel endpoints can be configured.
  - Problematic if you need more than one (e.g. for namespaces).
- VTI is configured with a combination of GRE keys and routing marks.
  - Neither GRE keys nor routing marks were designated to configure a VTI.
  - Routing by mark does not work well with VTI.
- VTI works just with tunnel mode SAs.
  - Not an interface to route transport or beet mode.

Disadvantages of IPsec VTI interfaces

- VTI interfaces are L3 tunnel interfaces with configurable tunnel endpoints.
  - The tunnel endpoints are already determined by the SA.
  - Configuring tunnel endpoints at a VTI does not make much sense.
- Only one VTI with wildcard tunnel endpoints can be configured.
  - Problematic if you need more than one (e.g. for namespaces).
- VTI is configured with a combination of GRE keys and routing marks.
  - Neither GRE keys nor routing marks were designated to configure a VTI.
  - Routing by mark does not work well with VTI.
- VTI works just with tunnel mode SAs.
  - Not an interface to route transport or beet mode.

Disadvantages of IPsec VTI interfaces

- VTI interfaces are L3 tunnel interfaces with configurable tunnel endpoints.
  - The tunnel endpoints are already determined by the SA.
  - Configuring tunnel endpoints at a VTI does not make much sense.
- Only one VTI with wildcard tunnel endpoints can be configured.
  - Problematic if you need more than one (e.g. for namespaces).
- VTI is configured with a combination of GRE keys and routing marks.
  - Neither GRE keys nor routing marks were designated to configure a VTI.
  - Routing by mark does not work well with VTI.
- VTI works just with tunnel mode SAs.
  - Not an interface to route transport or beet mode.

Disadvantages of IPsec VTI interfaces

- VTI interfaces are L3 tunnel interfaces with configurable tunnel endpoints.
  - The tunnel endpoints are already determined by the SA.
  - Configuring tunnel endpoints at a VTI does not make much sense.
- Only one VTI with wildcard tunnel endpoints can be configured.
  - Problematic if you need more than one (e.g. for namespaces).
- VTI is configured with a combination of GRE keys and routing marks.
  - Neither GRE keys nor routing marks were designated to configure a VTI.
  - Routing by mark does not work well with VTI.
- VTI works just with tunnel mode SAs.
  - Not an interface to route transport or beet mode.

Disadvantages of IPsec VTI interfaces

- VTI interfaces are L3 tunnel interfaces with configurable tunnel endpoints.
  - The tunnel endpoints are already determined by the SA.
  - Configuring tunnel endpoints at a VTI does not make much sense.
- Only one VTI with wildcard tunnel endpoints can be configured.
  - Problematic if you need more than one (e.g. for namespaces).
- VTI is configured with a combination of GRE keys and routing marks.
  - Neither GRE keys nor routing marks were designated to configure a VTI.
  - Routing by mark does not work well with VTI.
- VTI works just with tunnel mode SAs.
  - Not an interface to route transport or beet mode.

Disadvantages of IPsec VTI interfaces

- VTI interfaces are L3 tunnel interfaces with configurable tunnel endpoints.
  - The tunnel endpoints are already determined by the SA.
  - Configuring tunnel endpoints at a VTI does not make much sense.
- Only one VTI with wildcard tunnel endpoints can be configured.
  - Problematic if you need more than one (e.g. for namespaces).
- VTI is configured with a combination of GRE keys and routing marks.
  - Neither GRE keys nor routing marks were designated to configure a VTI.
  - Routing by mark does not work well with VTI.
- VTI works just with tunnel mode SAs.
  - Not an interface to route transport or beet mode.

Disadvantages of IPsec VTI interfaces

- VTI interfaces are L3 tunnel interfaces with configurable tunnel endpoints.
  - The tunnel endpoints are already determined by the SA.
  - Configuring tunnel endpoints at a VTI does not make much sense.
- Only one VTI with wildcard tunnel endpoints can be configured.
  - Problematic if you need more than one (e.g. for namespaces).
- VTI is configured with a combination of GRE keys and routing marks.
  - Neither GRE keys nor routing marks were designated to configure a VTI.
  - Routing by mark does not work well with VTI.
- VTI works just with tunnel mode SAs.
  - Not an interface to route transport or beet mode.

Disadvantages of IPsec VTI interfaces

- VTI interfaces are L3 tunnel interfaces with configurable tunnel endpoints.
  - The tunnel endpoints are already determined by the SA.
  - Configuring tunnel endpoints at a VTI does not make much sense.
- Only one VTI with wildcard tunnel endpoints can be configured.
  - Problematic if you need more than one (e.g. for namespaces).
- VTI is configured with a combination of GRE keys and routing marks.
  - Neither GRE keys nor routing marks were designated to configure a VTI.
  - Routing by mark does not work well with VTI.
- VTI works just with tunnel mode SAs.
  - Not an interface to route transport or beet mode.

New design for XFRM interfaces

- Should be a virtual interface that ensures IPsec transformation.
- No limitation on xfrm_mode (tunnel, transport and beet).
- Should be possible to create multiple interfaces (e.g. to move to different namespaces).
- Should be possible to route IPv4 and IPv6 through the same interface.
- Interfaces should be configured with input/output mark/mask that must match input/output mark of the xfrm_state.
- Anything else?

New design for XFRM interfaces

- Should be a virtual interface that ensures IPsec transformation.
- No limitation on xfrm_mode (tunnel, transport and beet).
- Should be possible to create multiple interfaces (e.g. to move to different namespaces).
- Should be possible to route IPv4 and IPv6 through the same interface.
- Interfaces should be configured with input/output mark/mask that must match input/output mark of the xfrm_state.
- Anything else?

New design for XFRM interfaces

- Should be a virtual interface that ensures IPsec transformation.
- No limitation on xfrm_mode (tunnel, transport and beet).
- Should be possible to create multiple interfaces (e.g. to move to different namespaces).
- Should be possible to route IPv4 and IPv6 through the same interface.
- Interfaces should be configured with input/output mark/mask that must match input/output mark of the xfrm_state.
- Anything else?

New design for XFRM interfaces

- Should be a virtual interface that ensures IPsec transformation.
- No limitation on xfrm_mode (tunnel, transport and beet).
- Should be possible to create multiple interfaces (e.g. to move to different namespaces).
- Should be possible to route IPv4 and IPv6 through the same interface.
- Interfaces should be configured with input/output mark/mask that must match input/output mark of the xfrm_state.
- Anything else?

New design for XFRM interfaces

- Should be a virtual interface that ensures IPsec transformation.
- No limitation on xfrm_mode (tunnel, transport and beet).
- Should be possible to create multiple interfaces (e.g. to move to different namespaces).
- Should be possible to route IPv4 and IPv6 through the same interface.
- Interfaces should be configured with input/output mark/mask that must match input/output mark of the xfrm_state.
- Anything else?

New design for XFRM interfaces

- Should be a virtual interface that ensures IPsec transformation.
- No limitation on xfrm_mode (tunnel, transport and beet).
- Should be possible to create multiple interfaces (e.g. to move to different namespaces).
- Should be possible to route IPv4 and IPv6 through the same interface.
- Interfaces should be configured with input/output mark/mask that must match input/output mark of the xfrm_state.
- Anything else?

New design for XFRM interfaces

- Should be a virtual interface that ensures IPsec transformation.
- No limitation on xfrm_mode (tunnel, transport and beet).
- Should be possible to create multiple interfaces (e.g. to move to different namespaces).
- Should be possible to route IPv4 and IPv6 through the same interface.
- Interfaces should be configured with input/output mark/mask that must match input/output mark of the xfrm_state.
- Anything else?