Virtual xfrm interfaces

Steffen Klassert

secunet Security Networks AG

Dresden

Linux IPsec Workshop, Dresden, March 26, 2018

Disadvantages of IPsec VTI interfaces

- VTI interfaces are L3 tunnels with configurable endpoints.
  - The tunnel endpoints are already determined by the SA.
- Separate interfaces for IPv4 and IPv6 tunnels needed.
- Only one VTI with wildcard tunnel endpoints.
  - Problematic if you need more than one (e.g. for namespaces).
- VTI is configured with GRE keys and routing marks.
  - Neither GRE keys nor routing marks were designated to configure a 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 tunnels with configurable endpoints.
  - The tunnel endpoints are already determined by the SA.
- Separate interfaces for IPv4 and IPv6 tunnels needed.
- Only one VTI with wildcard tunnel endpoints.
  - Problematic if you need more than one (e.g. for namespaces).
- VTI is configured with GRE keys and routing marks.
  - Neither GRE keys nor routing marks were designated to configure a 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 tunnels with configurable endpoints.
  - The tunnel endpoints are already determined by the SA.
- Separate interfaces for IPv4 and IPv6 tunnels needed.
- Only one VTI with wildcard tunnel endpoints.
  - Problematic if you need more than one (e.g. for namespaces).
- VTI is configured with GRE keys and routing marks.
  - Neither GRE keys nor routing marks were designated to configure a 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 tunnels with configurable endpoints.
  - The tunnel endpoints are already determined by the SA.
- Separate interfaces for IPv4 and IPv6 tunnels needed.
- Only one VTI with wildcard tunnel endpoints.
  - Problematic if you need more than one (e.g. for namespaces).
- VTI is configured with GRE keys and routing marks.
  - Neither GRE keys nor routing marks were designated to configure a 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 tunnels with configurable endpoints.
  - The tunnel endpoints are already determined by the SA.
- Separate interfaces for IPv4 and IPv6 tunnels needed.
- Only one VTI with wildcard tunnel endpoints.
  - Problematic if you need more than one (e.g. for namespaces).
- VTI is configured with GRE keys and routing marks.
  - Neither GRE keys nor routing marks were designated to configure a 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 tunnels with configurable endpoints.
  - The tunnel endpoints are already determined by the SA.
- Separate interfaces for IPv4 and IPv6 tunnels needed.
- Only one VTI with wildcard tunnel endpoints.
  - Problematic if you need more than one (e.g. for namespaces).
- VTI is configured with GRE keys and routing marks.
  - Neither GRE keys nor routing marks were designated to configure a 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 tunnels with configurable endpoints.
  - The tunnel endpoints are already determined by the SA.
- Separate interfaces for IPv4 and IPv6 tunnels needed.
- Only one VTI with wildcard tunnel endpoints.
  - Problematic if you need more than one (e.g. for namespaces).
- VTI is configured with GRE keys and routing marks.
  - Neither GRE keys nor routing marks were designated to configure a 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 tunnels with configurable endpoints.
  - The tunnel endpoints are already determined by the SA.
- Separate interfaces for IPv4 and IPv6 tunnels needed.
- Only one VTI with wildcard tunnel endpoints.
  - Problematic if you need more than one (e.g. for namespaces).
- VTI is configured with GRE keys and routing marks.
  - Neither GRE keys nor routing marks were designated to configure a 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 tunnels with configurable endpoints.
  - The tunnel endpoints are already determined by the SA.
- Separate interfaces for IPv4 and IPv6 tunnels needed.
- Only one VTI with wildcard tunnel endpoints.
  - Problematic if you need more than one (e.g. for namespaces).
- VTI is configured with GRE keys and routing marks.
  - Neither GRE keys nor routing marks were designated to configure a 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 tunnels with configurable endpoints.
  - The tunnel endpoints are already determined by the SA.
- Separate interfaces for IPv4 and IPv6 tunnels needed.
- Only one VTI with wildcard tunnel endpoints.
  - Problematic if you need more than one (e.g. for namespaces).
- VTI is configured with GRE keys and routing marks.
  - Neither GRE keys nor routing marks were designated to configure a 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).
- Interfaces should be configured with an interface ID that must match a (new) policy/SA lookup key.
- Should be possible to tunnel IPv4 and IPv6 through the same interface.
- Should be possible to use IPsec hardware offloads of the underlying interface.
- 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).
- Interfaces should be configured with an interface ID that must match a (new) policy/SA lookup key.
- Should be possible to tunnel IPv4 and IPv6 through the same interface.
- Should be possible to use IPsec hardware offloads of the underlying interface.
- 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).
- Interfaces should be configured with an interface ID that must match a (new) policy/SA lookup key.
- Should be possible to tunnel IPv4 and IPv6 through the same interface.
- Should be possible to use IPsec hardware offloads of the underlying interface.
- 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).
- Interfaces should be configured with an interface ID that must match a (new) policy/SA lookup key.
- Should be possible to tunnel IPv4 and IPv6 through the same interface.
- Should be possible to use IPsec hardware offloads of the underlying interface.
- 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).
- Interfaces should be configured with an interface ID that must match a (new) policy/SA lookup key.
- Should be possible to tunnel IPv4 and IPv6 through the same interface.
- Should be possible to use IPsec hardware offloads of the underlying interface.
- 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).
- Interfaces should be configured with an interface ID that must match a (new) policy/SA lookup key.
- Should be possible to tunnel IPv4 and IPv6 through the same interface.
- Should be possible to use IPsec hardware offloads of the underlying interface.
- 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).
- Interfaces should be configured with an interface ID that must match a (new) policy/SA lookup key.
- Should be possible to tunnel IPv4 and IPv6 through the same interface.
- Should be possible to use IPsec hardware offloads of the underlying interface.
- 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).
- Interfaces should be configured with an interface ID that must match a (new) policy/SA lookup key.
- Should be possible to tunnel IPv4 and IPv6 through the same interface.
- Should be possible to use IPsec hardware offloads of the underlying interface.
- Anything else?

Current implementation of the XFRM interfaces

- Stripped-down the VTI6 implementation to provide the basic interface.
- Created a new lookup key for policies and SAs, the xfrm interface id.
- It is possible to insert policies and SAs that differ only in the xfrm interface id.
  - The policy and SA lookups need some advanced testing!!!
- **Known problem:**
  Currently needs to be bound to a physical interface.
- **Known problem:**
  Policy wildcard src/dst addresses (0.0.0.0/0) $\rightarrow$ routing loop

Current implementation of the XFRM interfaces

- Stripped-down the VTI6 implementation to provide the basic interface.
- Created a new lookup key for policies and SAs, the xfrm interface id.
- It is possible to insert policies and SAs that differ only in the xfrm interface id.
  - The policy and SA lookups need some advanced testing!!!
- **Known problem:**
  Currently needs to be bound to a physical interface.
- **Known problem:**
  Policy wildcard src/dst addresses (0.0.0.0/0) → routing loop

Current implementation of the XFRM interfaces

- Stripped-down the VTI6 implementation to provide the basic interface.
- Created a new lookup key for policies and SAs, the xfrm interface id.
- It is possible to insert policies and SAs that differ only in the xfrm interface id.
  - The policy and SA lookups need some advanced testing!!!
- **Known problem:**
  Currently needs to be bound to a physical interface.
- **Known problem:**
  Policy wildcard src/dst addresses (0.0.0.0/0) → routing loop

Current implementation of the XFRM interfaces

- Stripped-down the VTI6 implementation to provide the basic interface.
- Created a new lookup key for policies and SAs, the xfrm interface id.
- It is possible to insert policies and SAs that differ only in the xfrm interface id.
  - The policy and SA lookups need some advanced testing!!!
- **Known problem:**
  Currently needs to be bound to a physical interface.
- **Known problem:**
  Policy wildcard src/dst addresses (0.0.0.0/0) → routing loop

Current implementation of the XFRM interfaces

- Stripped-down the VTI6 implementation to provide the basic interface.
- Created a new lookup key for policies and SAs, the xfrm interface id.
- It is possible to insert policies and SAs that differ only in the xfrm interface id.
  - The policy and SA lookups need some advanced testing!!!
- **Known problem:**
  Currently needs to be bound to a physical interface.
- **Known problem:**
  Policy wildcard src/dst addresses (0.0.0.0/0) → routing loop

Current implementation of the XFRM interfaces

- Stripped-down the VTI6 implementation to provide the basic interface.
- Created a new lookup key for policies and SAs, the xfrm interface id.
- It is possible to insert policies and SAs that differ only in the xfrm interface id.
  - The policy and SA lookups need some advanced testing!!!
- **Known problem:**
  Currently needs to be bound to a physical interface.
- **Known problem:**
  Policy wildcard src/dst addresses (0.0.0.0/0) → routing loop

Current implementation of the XFRM interfaces

- Stripped-down the VTI6 implementation to provide the basic interface.
- Created a new lookup key for policies and SAs, the xfrm interface id.
- It is possible to insert policies and SAs that differ only in the xfrm interface id.
  - The policy and SA lookups need some advanced testing!!!
- **Known problem:**
  Currently needs to be bound to a physical interface.
- **Known problem:**
  Policy wildcard src/dst addresses (0.0.0.0/0) $\rightarrow$ routing loop

Current implementation of the XFRM interfaces

Does it match all usecases? What is missing? Bugs?