# IPsec Crypto Offload

# Table of contents

> ### ⓘ Note
>
> This feature is supported on crypto-enabled products of BlueField-2 DPUs, and ConnectX-6 Dx and ConnectX-7 adapters (but not of ConnectX-6 or ConnectX-6 Lx).
>
> Newer/future crypto-enabled DPU and adapter product generations should also support the feature, unless explicitly stated in their documentation.

> ### ⓘ Note
>
> For NVIDIA BlueField-2 DPUs and ConnectX-6 Dx adapters Only: If your target application will utilize bandwidth of100Gb/s or higher, where a substantial part of the bandwidth will be allocated for IPsec traffic, please refer to the NVIDIA BlueField-2 DPUs Product Release Notes or NVIDIA ConnectX-6 Dx Adapters Product Release Notes document to learn about a potential bandwidth limitation. To access the relevant product release notes, please contact your NVIDIA sales representative.

## Overview and Configuration

IPsec crypto offload feature, also known as IPsec inline offload or IPsec aware offload feature enables the user to offload IPsec crypto encryption and decryption operations to the hardware.

Note that the hardware implementation only supports AES-GCM encryption scheme.

To enable the feature, support in both kernel and adapter firmware is required.

- For support in the kernel, make sure the following flags are set as follows.

```
CONFIG_XFRM_OFFLOAD=y
CONFIG_INET_ESP_OFFLOAD=m
CONFIG_INET6_ESP_OFFLOAD=m
```

**Note**: These flags are enabled by default in RedHat 8 and Ubuntu 18.04.

- For support in the firmware, make sure the below string is found in the dmesg.

```
mlx5e: IPSec ESP acceleration enabled
```

# Configuring Security Associations for IPsec Offloads

To program the inline offload security associations (SA), add the option "offload dev <netdev interface> dir out/in" in the "ip xfrm state" command for transmitting and receiving SA.

Transmit inline offload SA xfrm command example:

```
sudo ip xfrm state add src 192.168.1.64/24 dst 192.168.1.65/24 proto esp spi 0x46dc6204
reqid 0x46dc6204 mode transport aead
'rfc4106(gcm(aes))' 0x60bd6c3eafba371a46411830fd56c53af93883261ed1fb26767820ff493f43ba35b0dcc
offload dev p4p1 dir out sel src 192.168.1.64 dst 192.168.1.65
```

Receive inline offload SA xfrm command example:

```
sudo ip xfrm state add src 192.168.1.65/24 dst 192.168.1.64/24 proto esp spi 0xaea0846c
reqid 0xaea0846c mode transport aead
'rfc4106(gcm(aes))' 0x81d5c3167c912c1dd50dab0cb4b6d815b6ace8844304db362215a258cd19deda8f89c
offload dev p4p1 dir in sel src 192.168.1.65 dst 192.168.1.64
```

# Setting xfrm Policies Example

First server:

```
+ sudo ip xfrm state add src 192.168.1.64/24 dst 192.168.1.65/24 proto esp spi 0x28f39549
reqid 0x28f39549 mode transport aead
'rfc4106(gcm(aes))' 0x492e8ffe718a95a00c1893ea61afc64997f4732848ccfe6ea07db483175cb18de9ae411
offload dev enp4s0 dir out sel src 192.168.1.64 dst 192.168.1.65
+ sudo ip xfrm state add src 192.168.1.65/24 dst 192.168.1.64/24 proto esp spi 0x622a73b4
reqid 0x622a73b4 mode transport aead
'rfc4106(gcm(aes))' 0x093bfee2212802d626716815f862da31bcc7d9c44cfe3ab8049e7604b2feb1254869d2
offload dev enp4s0 dir in sel src 192.168.1.65 dst 192.168.1.64
+ sudo ip xfrm policy add src 192.168.1.64 dst 192.168.1.65 dir out tmpl src 192.168.1.64/24
dst 192.168.1.65/24 proto esp reqid 0x28f39549 mode transport
+ sudo ip xfrm policy add src 192.168.1.65 dst 192.168.1.64 dir in tmpl src 192.168.1.65/24
dst 192.168.1.64/24 proto esp reqid 0x622a73b4 mode transport
+ sudo ip xfrm policy add src 192.168.1.65 dst 192.168.1.64 dir fwd tmpl src
192.168.1.65/24 dst 192.168.1.64/24 proto esp reqid 0x622a73b4 mode transport
```

Second server:

```
+ ssh -A -t root@l-csi-0921d /bin/bash
+ set -e
+ '[' 0 == 1 ']'
+ sudo ip xfrm state add src 192.168.1.64/24 dst 192.168.1.65/24 proto esp spi 0x28f39549
reqid 0x28f39549 mode transport aead
'rfc4106(gcm(aes))' 0x492e8ffe718a95a00c1893ea61afc64997f4732848ccfe6ea07db483175cb18de9ae411
offload dev enp4s0 dir in sel src 192.168.1.64 dst 192.168.1.65
+ sudo ip xfrm state add src 192.168.1.65/24 dst 192.168.1.64/24 proto esp spi 0x622a73b4
reqid 0x622a73b4 mode transport aead
'rfc4106(gcm(aes))' 0x093bfee2212802d626716815f862da31bcc7d9c44cfe3ab8049e7604b2feb1254869d2
offload dev enp4s0 dir out sel src 192.168.1.65 dst 192.168.1.64
+ sudo ip xfrm policy add src 192.168.1.65 dst 192.168.1.64 dir out tmpl src 192.168.1.65/24
dst 192.168.1.64/24 proto esp reqid 0x622a73b4 mode transport
+ sudo ip xfrm policy add src 192.168.1.64 dst 192.168.1.65 dir in tmpl src 192.168.1.64/24
dst 192.168.1.65/24 proto esp reqid 0x28f39549 mode transport
+ sudo ip xfrm policy add src 192.168.1.64 dst 192.168.1.65 dir fwd tmpl src
192.168.1.64/24 dst 192.168.1.65/24 proto esp reqid 0x28f39549 mode transport
```

```
+ echo 'IPSec tunnel configured successfully'
```