Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

IPsec performance BoF

Steffen Klassert

secunet Security Networks AG

Dresden

Netdev 1.1 Seville, February 11, 2016

Table of contents

Avoid frame copy in skb_cow_data

- **Problem:** Most of the data frames are linearized in skb_cow_data.
- **RX:** Easy to solve, we can know if the buffer is writable.
- **Solution:** Only linearize if the buffer is not writable.
- **TX:** Need to expand the tail of the buffer.
- **Solution:** Add a page fragment with the tailbits to the buffer.
- Works ony if the stack generates buffers with $nr\_frags < MAX\_SKB\_FRAGS$ fragments.

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Avoid frame copy in skb_cow_data

- **Problem:** Most of the data frames are linearized in skb_cow_data.
- **RX:** Easy to solve, we can know if the buffer is writable.
- **Solution:** Only linearize if the buffer is not writable.
- **TX:** Need to expand the tail of the buffer.
- **Solution:** Add a page fragment with the tailbits to the buffer.
- Works ony if the stack generates buffers with $nr\_frags < MAX\_SKB\_FRAGS$ fragments.

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Avoid frame copy in skb_cow_data

- **Problem:** Most of the data frames are linearized in skb_cow_data.
- **RX:** Easy to solve, we can know if the buffer is writable.
- **Solution:** Only linearize if the buffer is not writable.
- **TX:** Need to expand the tail of the buffer.
- **Solution:** Add a page fragment with the tailbits to the buffer.
- Works ony if the stack generates buffers with $nr\_frags < MAX\_SKB\_FRAGS$ fragments.

Avoid frame copy in skb_cow_data

- **Problem:** Most of the data frames are linearized in skb_cow_data.
- **RX:** Easy to solve, we can know if the buffer is writable.
- **Solution:** Only linearize if the buffer is not writable.
- **TX:** Need to expand the tail of the buffer.
- **Solution:** Add a page fragment with the tailbits to the buffer.
- Works ony if the stack generates buffers with $nr\_frags < MAX\_SKB\_FRAGS$ fragments.

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Avoid frame copy in skb_cow_data

- **Problem:** Most of the data frames are linearized in skb_cow_data.
- **RX:** Easy to solve, we can know if the buffer is writable.
- **Solution:** Only linearize if the buffer is not writable.
- **TX:** Need to expand the tail of the buffer.
- **Solution:** Add a page fragment with the tailbits to the buffer.
- Works ony if the stack generates buffers with $nr\_frags < MAX\_SKB\_FRAGS$ fragments.

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Avoid frame copy in skb_cow_data

- **Problem:** Most of the data frames are linearized in skb_cow_data.
- **RX:** Easy to solve, we can know if the buffer is writable.
- **Solution:** Only linearize if the buffer is not writable.
- **TX:** Need to expand the tail of the buffer.
- **Solution:** Add a page fragment with the tailbits to the buffer.
- Works ony if the stack generates buffers with $nr\_frags < MAX\_SKB\_FRAGS$ fragments.

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Avoid frame copy in skb_cow_data

- **Problem:** Most of the data frames are linearized in skb_cow_data.
- **RX:** Easy to solve, we can know if the buffer is writable.
- **Solution:** Only linearize if the buffer is not writable.
- **TX:** Need to expand the tail of the buffer.
- **Solution:** Add a page fragment with the tailbits to the buffer.
- Works ony if the stack generates buffers with $nr\_frags < MAX\_SKB\_FRAGS$ fragments.

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Avoid frame copy in skb_cow_data (continued)

- **Question:** Can we instrument the stack to generate buffers with at most $MAX\_SKB\_FRAGS - 1$ fragments?
- **Local send:** Should be possible because TCP tries to use high order pages (32K), so we have not more than 3-4 page fragments per buffer.
- **Problem:** The crypto layer always assume to have order null pages in the scatterlists.
- **Question:** Is there a reason for that or can we change the crypto layer to handle high order pages?

Avoid frame copy in skb_cow_data (continued)

- **Question:** Can we instrument the stack to generate buffers with at most $MAX\_SKB\_FRAGS - 1$ fragments?
- **Local send:** Should be possible because TCP tries to use high order pages (32K), so we have not more than 3-4 page fragments per buffer.
- **Problem:** The crypto layer always assume to have order null pages in the scatterlists.
- **Question:** Is there a reason for that or can we change the crypto layer to handle high order pages?

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Avoid frame copy in skb_cow_data (continued)

- **Question:** Can we instrument the stack to generate buffers with at most $MAX\_SKB\_FRAGS - 1$ fragments?
- **Local send:** Should be possible because TCP tries to use high order pages (32K), so we have not more than 3-4 page fragments per buffer.
- **Problem:** The crypto layer always assume to have order null pages in the scatterlists.
- **Question:** Is there a reason for that or can we change the crypto layer to handle high order pages?

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Avoid frame copy in skb_cow_data (continued)

- **Question:** Can we instrument the stack to generate buffers with at most $MAX\_SKB\_FRAGS - 1$ fragments?
- **Local send:** Should be possible because TCP tries to use high order pages (32K), so we have not more than 3-4 page fragments per buffer.
- **Problem:** The crypto layer always assume to have order null pages in the scatterlists.
- **Question:** Is there a reason for that or can we change the crypto layer to handle high order pages?

Avoid frame copy in skb_cow_data (continued)

- **Forwarding:** Since October 2013 (commit "net: gro: allow to build full sized skb") GRO can build buffers with frag_list.
- **Problem:** We can't add a page fragment with the IPsec tailbits to the buffer.
- **General forwarding problem:** Such buffers can't be offloaded to hardware, we need to linearize and segment them in the stack.
- **Question:** Can we find a consensus to build fair buffers for local receive and forwarding?

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Avoid frame copy in skb_cow_data (continued)

- **Forwarding:** Since October 2013 (commit "net: gro: allow to build full sized skb") GRO can build buffers with frag_list.
- **Problem:** We can't add a page fragment with the IPsec tailbits to the buffer.
- **General forwarding problem:** Such buffers can't be offloaded to hardware, we need to linearize and segment them in the stack.
- **Question:** Can we find a consensus to build fair buffers for local receive and forwarding?

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Avoid frame copy in skb_cow_data (continued)

- **Forwarding:** Since October 2013 (commit "net: gro: allow to build full sized skb") GRO can build buffers with frag_list.
- **Problem:** We can't add a page fragment with the IPsec tailbits to the buffer.
- **General forwarding problem:** Such buffers can't be offloaded to hardware, we need to linearize and segment them in the stack.
- **Question:** Can we find a consensus to build fair buffers for local receive and forwarding?

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Avoid frame copy in skb_cow_data (continued)

- **Forwarding:** Since October 2013 (commit "net: gro: allow to build full sized skb") GRO can build buffers with frag_list.
- **Problem:** We can't add a page fragment with the IPsec tailbits to the buffer.
- **General forwarding problem:** Such buffers can't be offloaded to hardware, we need to linearize and segment them in the stack.
- **Question:** Can we find a consensus to build fair buffers for local receive and forwarding?

Adding a software GRO codepath for IPsec.

- **GRO:** Add GRO handlers for the IPsec protocols, RFC code exists.
- **Problem:** The stack does not see IPsec packets anymore, could scare users.
- **Question:** Should this be cofigurable aside from enable/disable GRO?

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Adding a software GRO codepath for IPsec.

- **GRO:** Add GRO handlers for the IPsec protocols, RFC code exists.
- **Problem:** The stack does not see IPsec packets anymore, could scare users.
- **Question:** Should this be cofigurable aside from enable/disable GRO?

Adding a software GRO codepath for IPsec.

- **GRO:** Add GRO handlers for the IPsec protocols, RFC code exists.
- **Problem:** The stack does not see IPsec packets anymore, could scare users.
- **Question:** Should this be cofigurable aside from enable/disable GRO?

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Adding a software GRO codepath for IPsec.

- **GRO:** Add GRO handlers for the IPsec protocols, RFC code exists.
- **Problem:** The stack does not see IPsec packets anymore, could scare users.
- **Question:** Should this be cofigurable aside from enable/disable GRO?

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Adding a software GSO codepath for IPsec.

- **GSO:** Move the existing xfrm GSO handling from xfrm to the generic GSO layer (L2).
- **GSO:** Do just the tunnel/transport mode encapsulation with a dummy ESP header at the xfrm layer.
- **GSO:** Add full ESP header informations and encryption to the segments in the GSO layer.

Adding a software GSO codepath for IPsec.

- **GSO:** Move the existing xfrm GSO handling from xfrm to the generic GSO layer (L2).
- **GSO:** Do just the tunnel/transport mode encapsulation with a dummy ESP header at the xfrm layer.
- **GSO:** Add full ESP header informations and encryption to the segments in the GSO layer.

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Adding a software GSO codepath for IPsec.

- **GSO:** Move the existing xfrm GSO handling from xfrm to the generic GSO layer (L2).
- **GSO:** Do just the tunnel/transport mode encapsulation with a dummy ESP header at the xfrm layer.
- **GSO:** Add full ESP header informations and encryption to the segments in the GSO layer.

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Adding a software GSO codepath for IPsec.

- **GSO:** Move the existing xfrm GSO handling from xfrm to the generic GSO layer (L2).
- **GSO:** Do just the tunnel/transport mode encapsulation with a dummy ESP header at the xfrm layer.
- **GSO:** Add full ESP header informations and encryption to the segments in the GSO layer.

Adding a software GSO codepath for IPsec (continued).

- **Question (1):** How to handle asynchronous crypto operations in the GSO layer?
- **Question (2):** What to do if the NIC driver returns *NETDEV_TX_BUSY* after asynchronous crypto operation?
    - **Possible solution (Q1):** Add a callback for each GSO handler?
    - **Possible solution (Q1/Q2):** Use a 'crypto qdisc'?
    - **Possible solution (Q1):** Handle the encapsulation at the GSO layer and do the crypto operations later?
    - **Possible solution (Q2):** Enqueue packet again to the qdisc?
    - **Possible solution (Q2):** Enqueue to a separate queue and process it with NET_TX_SOFTIRQ?
    - Other ideas???

Adding a software GSO codepath for IPsec (continued).

- **Question (1):** How to handle asynchronous crypto operations in the GSO layer?
- **Question (2):** What to do if the NIC driver returns *NETDEV_TX_BUSY* after asynchronous crypto operation?
  - **Possible solution (Q1):** Add a callback for each GSO handler?
  - **Possible solution (Q1/Q2):** Use a 'crypto qdisc'?
  - **Possible solution (Q1):** Handle the encapsulation at the GSO layer and do the crypto operations later?
  - **Possible solution (Q2):** Enqueue packet again to the qdisc?
  - **Possible solution (Q2):** Enqueue to a separate queue and process it with NET_TX_SOFTIRQ?
  - Other ideas???

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Adding a software GSO codepath for IPsec (continued).

- **Question (1):** How to handle asynchronous crypto operations in the GSO layer?
- **Question (2):** What to do if the NIC driver returns *NETDEV_TX_BUSY* after asynchronous crypto operation?
  - **Possible solution (Q1):** Add a callback for each GSO handler?
  - **Possible solution (Q1/Q2):** Use a 'crypto qdisc'?
  - **Possible solution (Q1):** Handle the encapsulation at the GSO layer and do the crypto operations later?
  - **Possible solution (Q2):** Enqueue packet again to the qdisc?
  - **Possible solution (Q2):** Enqueue to a separate queue and process it with NET_TX_SOFTIRQ?
  - Other ideas???

Adding a software GSO codepath for IPsec (continued).

- **Question (1):** How to handle asynchronous crypto operations in the GSO layer?
- **Question (2):** What to do if the NIC driver returns *NETDEV_TX_BUSY* after asynchronous crypto operation?
  - **Possible solution (Q1):** Add a callback for each GSO handler?
  - **Possible solution (Q1/Q2):** Use a 'crypto qdisc'?
  - **Possible solution (Q1):** Handle the encapsulation at the GSO layer and do the crypto operations later?
  - **Possible solution (Q2):** Enqueue packet again to the qdisc?
  - **Possible solution (Q2):** Enqueue to a separate queue and process it with NET_TX_SOFTIRQ?
  - Other ideas???

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Adding a software GSO codepath for IPsec (continued).

- **Question (1):** How to handle asynchronous crypto operations in the GSO layer?
- **Question (2):** What to do if the NIC driver returns *NETDEV_TX_BUSY* after asynchronous crypto operation?
  - **Possible solution (Q1):** Add a callback for each GSO handler?
  - **Possible solution (Q1/Q2):** Use a 'crypto qdisc'?
  - **Possible solution (Q1):** Handle the encapsulation at the GSO layer and do the crypto operations later?
  - **Possible solution (Q2):** Enqueue packet again to the qdisc?
  - **Possible solution (Q2):** Enqueue to a separate queue and process it with NET_TX_SOFTIRQ?
  - Other ideas???

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Adding a software GSO codepath for IPsec (continued).

- **Question (1):** How to handle asynchronous crypto operations in the GSO layer?
- **Question (2):** What to do if the NIC driver returns *NETDEV_TX_BUSY* after asynchronous crypto operation?
  - **Possible solution (Q1):** Add a callback for each GSO handler?
  - **Possible solution (Q1/Q2):** Use a 'crypto qdisc'?
  - **Possible solution (Q1):** Handle the encapsulation at the GSO layer and do the crypto operations later?
  - **Possible solution (Q2):** Enqueue packet again to the qdisc?
  - **Possible solution (Q2):** Enqueue to a separate queue and process it with NET_TX_SOFTIRQ?
  - Other ideas???

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Adding a software GSO codepath for IPsec (continued).

- **Question (1):** How to handle asynchronous crypto operations in the GSO layer?
- **Question (2):** What to do if the NIC driver returns *NETDEV_TX_BUSY* after asynchronous crypto operation?
  - **Possible solution (Q1):** Add a callback for each GSO handler?
  - **Possible solution (Q1/Q2):** Use a 'crypto qdisc'?
  - **Possible solution (Q1):** Handle the encapsulation at the GSO layer and do the crypto operations later?
  - **Possible solution (Q2):** Enqueue packet again to the qdisc?
  - **Possible solution (Q2):** Enqueue to a separate queue and process it with NET_TX_SOFTIRQ?
  - Other ideas???

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Adding a software GSO codepath for IPsec (continued).

- **Question (1):** How to handle asynchronous crypto operations in the GSO layer?
- **Question (2):** What to do if the NIC driver returns *NETDEV_TX_BUSY* after asynchronous crypto operation?
  - **Possible solution (Q1):** Add a callback for each GSO handler?
  - **Possible solution (Q1/Q2):** Use a 'crypto qdisc'?
  - **Possible solution (Q1):** Handle the encapsulation at the GSO layer and do the crypto operations later?
  - **Possible solution (Q2):** Enqueue packet again to the qdisc?
  - **Possible solution (Q2):** Enqueue to a separate queue and process it with NET_TX_SOFTIRQ?
  - Other ideas???

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Some performance numbers

- Transport mode performance numbers (measured by Sowmini Varadhan).
- **Baseline:**
- 2.6 Gbps (ESP-NULL) 71% CPU utilization.
- 2.17 Gbps (AES-GCM-256) 83% CPU utilization.
- **Avoid frame copy + GSO/GRO:**
- 8 Gbps (ESP-NULL) 95% CPU utilization.
  (Bottleneck: segmentation, checksuming of the segments)
- 4.2 Gbps (AES-GCM-256) 100% CPU utilization
  (Bottleneck: crypto).

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Some performance numbers

- Transport mode performance numbers (measured by Sowmini Varadhan).
- **Baseline:**
- 2.6 Gbps (ESP-NULL) 71% CPU utilization.
- 2.17 Gbps (AES-GCM-256) 83% CPU utilization.
- **Avoid frame copy + GSO/GRO:**
- 8 Gbps (ESP-NULL) 95% CPU utilization. (Bottleneck: segmentation, checksuming of the segments)
- 4.2 Gbps (AES-GCM-256) 100% CPU utilization (Bottleneck: crypto).

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Some performance numbers

- Transport mode performance numbers (measured by Sowmini Varadhan).
- **Baseline:**
- 2.6 Gbps (ESP-NULL) 71% CPU utilization.
- 2.17 Gbps (AES-GCM-256) 83% CPU utilization.
- **Avoid frame copy + GSO/GRO:**
- 8 Gbps (ESP-NULL) 95% CPU utilization. (Bottleneck: segmentation, checksuming of the segments)
- 4.2 Gbps (AES-GCM-256) 100% CPU utilization (Bottleneck: crypto).

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Some performance numbers

- Transport mode performance numbers (measured by Sowmini Varadhan).
- **Baseline:**
- 2.6 Gbps (ESP-NULL) 71% CPU utilization.
- 2.17 Gbps (AES-GCM-256) 83% CPU utilization.
- **Avoid frame copy + GSO/GRO:**
- 8 Gbps (ESP-NULL) 95% CPU utilization.
  (Bottleneck: segmentation, checksuming of the segments)
- 4.2 Gbps (AES-GCM-256) 100% CPU utilization
  (Bottleneck: crypto).

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Some performance numbers (continued)

- **Next step:** Move segmentation and crypto operations away from the networking cpus.
- **Solution (1):** Separate into networking and crypto cpus with the parallel crypto template (pcrypt).
  (crypto bottleneck)
- **Solution (2):** Offload IPsec operations to the NIC.
  (crypto + segmentation bottleneck)

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Some performance numbers (continued)

- **Next step:** Move segmentation and crypto operations away from the networking cpus.
- **Solution (1):** Separate into networking and crypto cpus with the parallel crypto template (pcrypt).
  (crypto bottleneck)
- **Solution (2):** Offload IPsec operations to the NIC.
  (crypto + segmentation bottleneck)

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Some performance numbers (continued)

- **Next step:** Move segmentation and crypto operations away from the networking cpus.
- **Solution (1):** Separate into networking and crypto cpus with the parallel crypto template (pcrypt).
  (crypto bottleneck)
- **Solution (2):** Offload IPsec operations to the NIC.
  (crypto + segmentation bottleneck)

Adding IPsec HW offload support

- **HW offload:** Should use the same API as IPsec GSO would use (IPsec GSO considered as a software fallback).
- **Question:** How should the API for IPsec hardware offloads should look like?
- **Question:** What would the NIC driver need from the stack?

Adding IPsec HW offload support

- **HW offload:** Should use the same API as IPsec GSO would use (IPsec GSO considered as a software fallback).
- **Question:** How should the API for IPsec hardware offloads should look like?
- **Question:** What would the NIC driver need from the stack?

Adding IPsec HW offload support

- **HW offload:** Should use the same API as IPsec GSO would use (IPsec GSO considered as a software fallback).
- **Question:** How should the API for IPsec hardware offloads should look like?
- **Question:** What would the NIC driver need from the stack?

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Adding IPsec HW offload support

- **HW offload:** Should use the same API as IPsec GSO would use (IPsec GSO considered as a software fallback).
- **Question:** How should the API for IPsec hardware offloads should look like?
- **Question:** What would the NIC driver need from the stack?

Proceedings of NetDev 1.1: The Technical Conference on Linux Networking (February 10th-12th 2016. Seville, Spain)

Change xfrm_policy_lock from rwlock to rcu

Change xfrm_policy_lock from rwlock to rcu