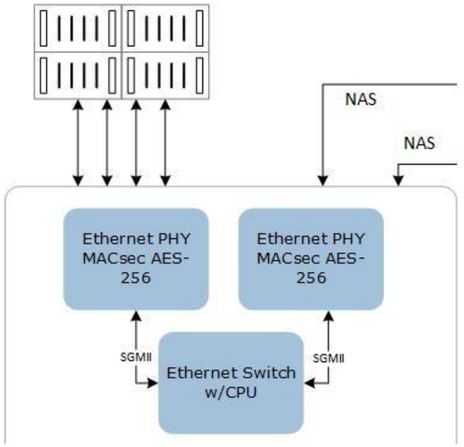
MACsec security violation in Rx handling for offloaded devices

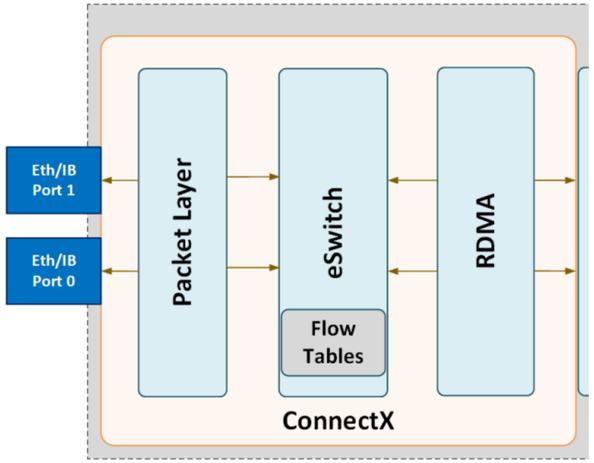
Rahul Rameshbabu

Background

How MACsec device offload support was first implemented



Microsemi MSCC PHY



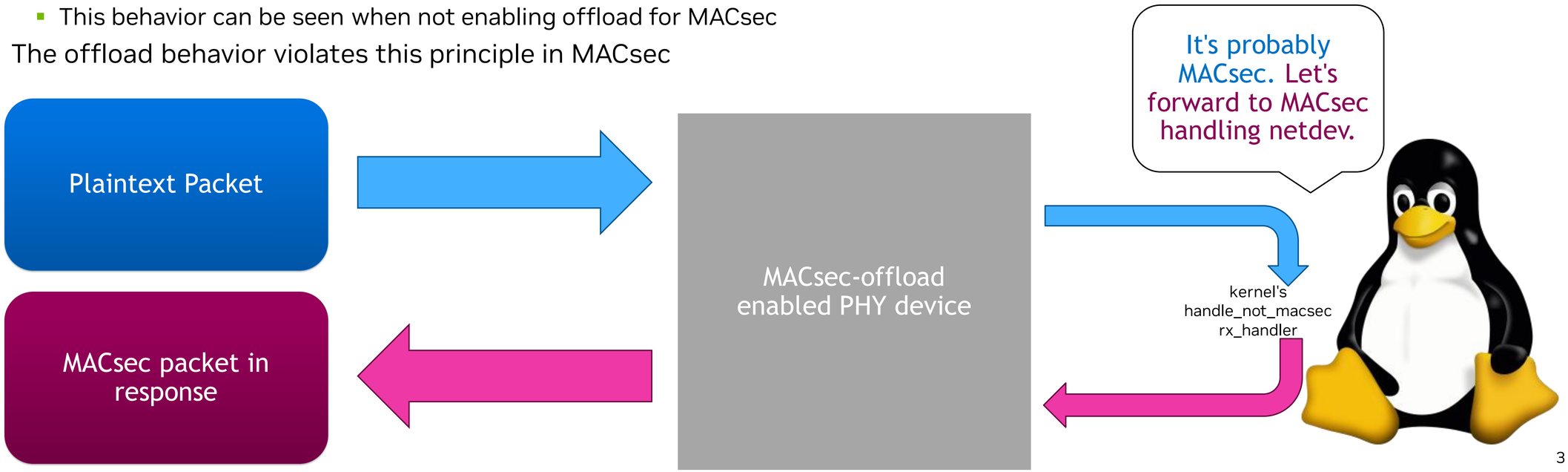
Flow steering lets helps
us know whether a packet went through security engine using the completion event

- MACsec is a L2 layer security protocol with concepts like those in IPsec and TLS
  - For more details: https://confluence.nvidia.com/display/SW/Security+Offload+Handover
- Support for device offload with MACsec was first drafted using Microsemi 1 Gig ethernet phys and then later expanded upon using Marvell Atlantic NICs
  - https://lore.kernel.org/netdev/20200113223148.746096-1-antoine.tenart@bootlin.com/
    - Initial work involving Microsemi
  - https://lore.kernel.org/netdev/20200325125246.987-1-irusskikh@marvell.com/
    - Expansion of work for supporting multicast traffic by Marvell
- mlx5 is late to the game
  - However, we support being able to check whether our completion events indicate the packet went through HW decryption
- Offload work for MACsec uses some relatively naive header matching to guess whether the packet was MACsec offload or not

The Problem with MACsec offload today

RX handling is problematic and potentially insecure

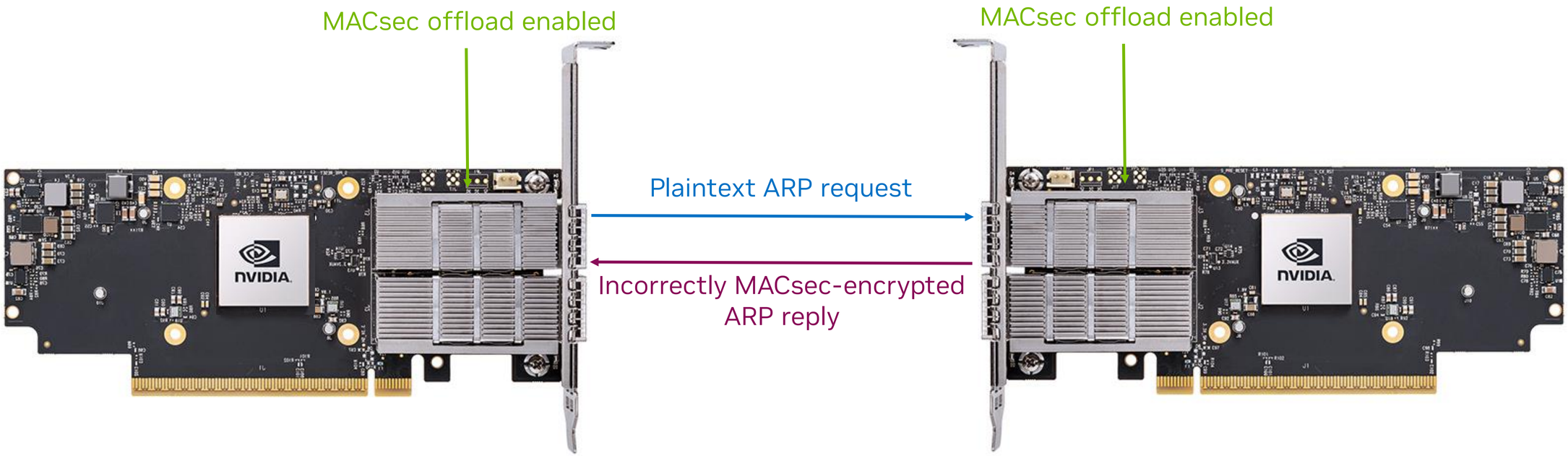
- The current design of the MACsec offload handling path tries to use "best guess" mechanisms for determining whether a packet associated with the currently handled skb in the datapath was processed via HW offload
- The best guess mechanism uses the following heuristic logic (in order of precedence)
  - Check if header destination MAC address matches MACsec netdev MAC address -> forward to MACsec port
  - Check if packet is multicast traffic -> forward to MACsec port
  - MACsec security channel was able to be looked up from skb offload context (mlx5 only) -> forward to MACsec port
- Problem: plaintext traffic can potentially solicit a MACsec encrypted response from the offload device
  - Core aspect of MACsec is that it identifies unauthorized LAN connections and excludes them from communication
    - This behavior can be seen when not enabling offload for MACsec
  - The offload behavior violates this principle in MACsec



PoC

Quick demonstration of the issue

- With the current MACsec offload implementation in the core stack, it is possible to demonstrate the issue even with mlx5 hardware which can annotate whether offloads occurred in CQEs
  - All it takes is configuring MACsec offload on two ports
  - On the one side, send plaintext multicast traffic (ex. ARP request)
  - On the other side, you will see that the MACsec TX counters are incrementing for each ARP reply sent



Proposed Solution

Open to Comments

- Modify the core stack to support a flow that bypasses the "best guess" heuristic logic in the MACsec Rx datapath handling
  - Any device that advertises metadata information indicating whether a packet has gone through HW offload for MACsec or not (mlx5) has a different handling path that bypasses the "best guess" heuristic
  - This eliminates the previous problems discussed for these devices
- I wonder if it makes sense to drop MACsec offload support for devices that do not report any hardware tracing of packets to annotate whether it has been processed by some offload engine or not
  - Right now, I just have a warning in my draft about the issue
  - These devices seem problematic to use with MACsec offload
  - Do we have to do anything with regards to this? Seems like a problem to just fix the issue for mlx5 and move on
- Current draft
  - http://l-gerrit.mtl.labs.mlnx:8080/q/topic:%22macsec-rx-improved-offload%22+(status:open%20OR%20status:merged)
  - Would like to get this through internal verification
- RFC on mailing list (does not depict the incorrect MACsec Rx handling as a security issue)
  - https://lore.kernel.org/netdev/20231116182900.46052-1-rrameshbabu@nvidia.com/