RDMA Support for SELinux

Draft for SC'15

August, 2015

Mellanox TECHNOLOGIES

Connect. Accelerate. Outperform.™

SELinux Network Security

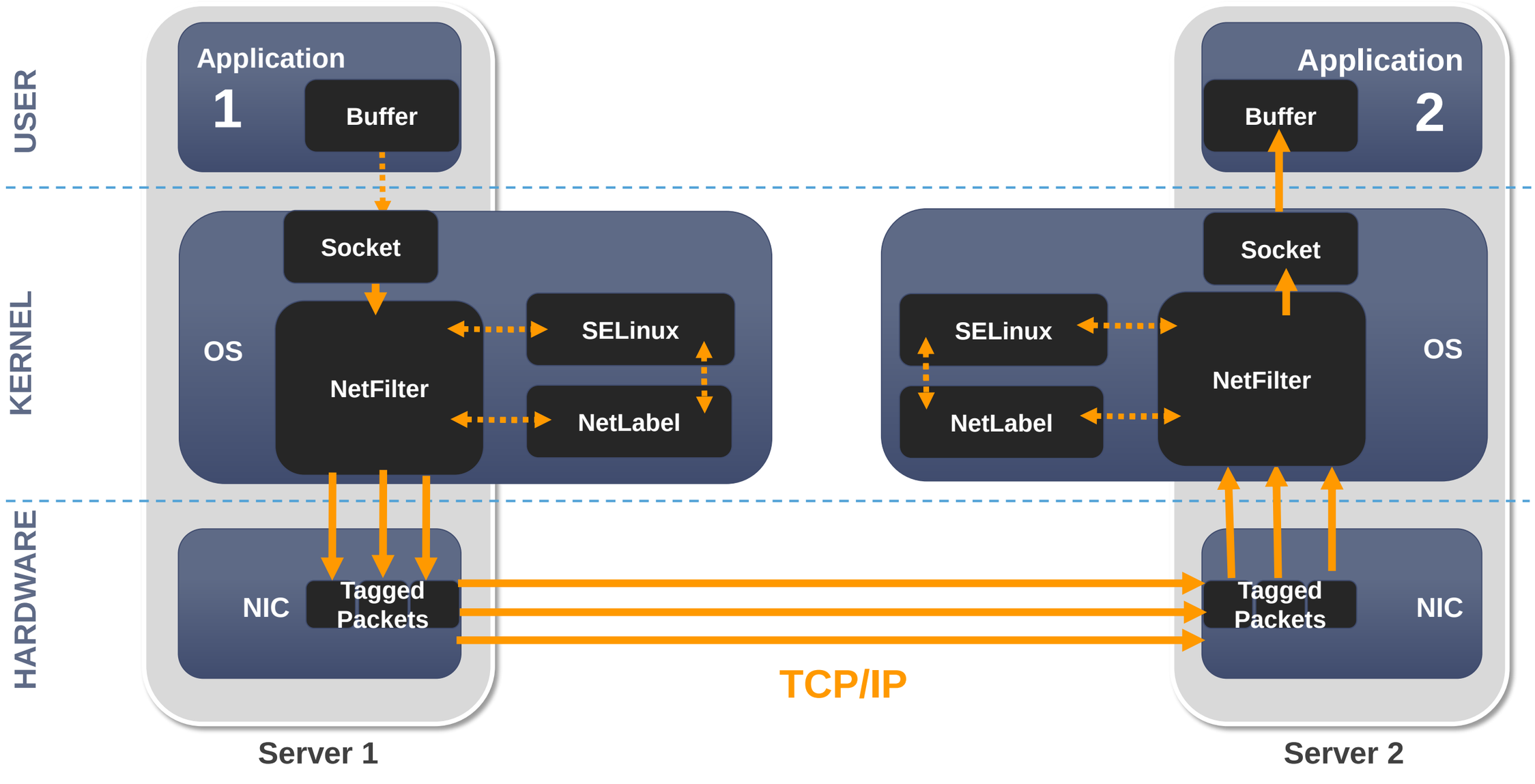
- Network object labeling
  - Interfaces
    - E.g., eth2
    - Used in the past for packet tagging
      - Today packets are tagged by network traffic labeling
  - Nodes
    - Label IPv4/6 addresses and network masks
  - Ports
    - Label TCP/UDP port numbers
  - Sockets
    - Usually inherit the security descriptor of the creating process

- Network traffic labeling
  - Internal labeling
    - Tag traffic according to local OS policies
    - The de-facto standard is SECMARK
      - Extends standard iptables/netfilter to mark packets with security descriptors
  - Labeled networking
    - Labels on the wire
    - Each local system interprets the label to enforce is MAC policies
    - Supported schemes
      - Labled IPSec
      - CIPSO, NetLabel

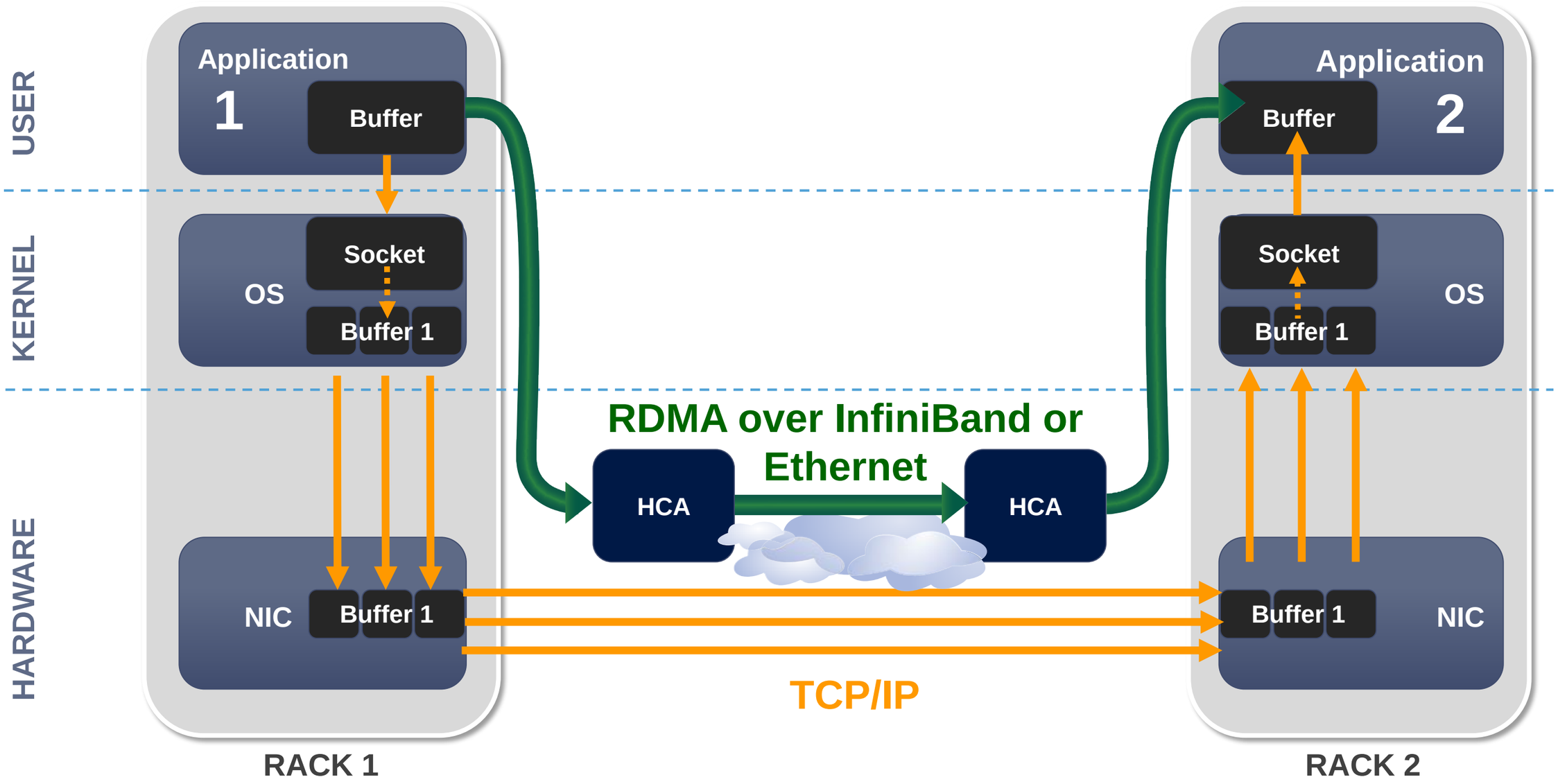
- SELinux network policies define
  - What a process can do with network objects
    - For example: allow ‘ftp_t’ (the FTP process) to bind a socket to ‘ftp_data_port’ (TCP port 20)
  - What traffic a process can send/receive

- Policy enforcement
  - Object policies are enforced during system calls
  - Traffic policies are enforced per-packet

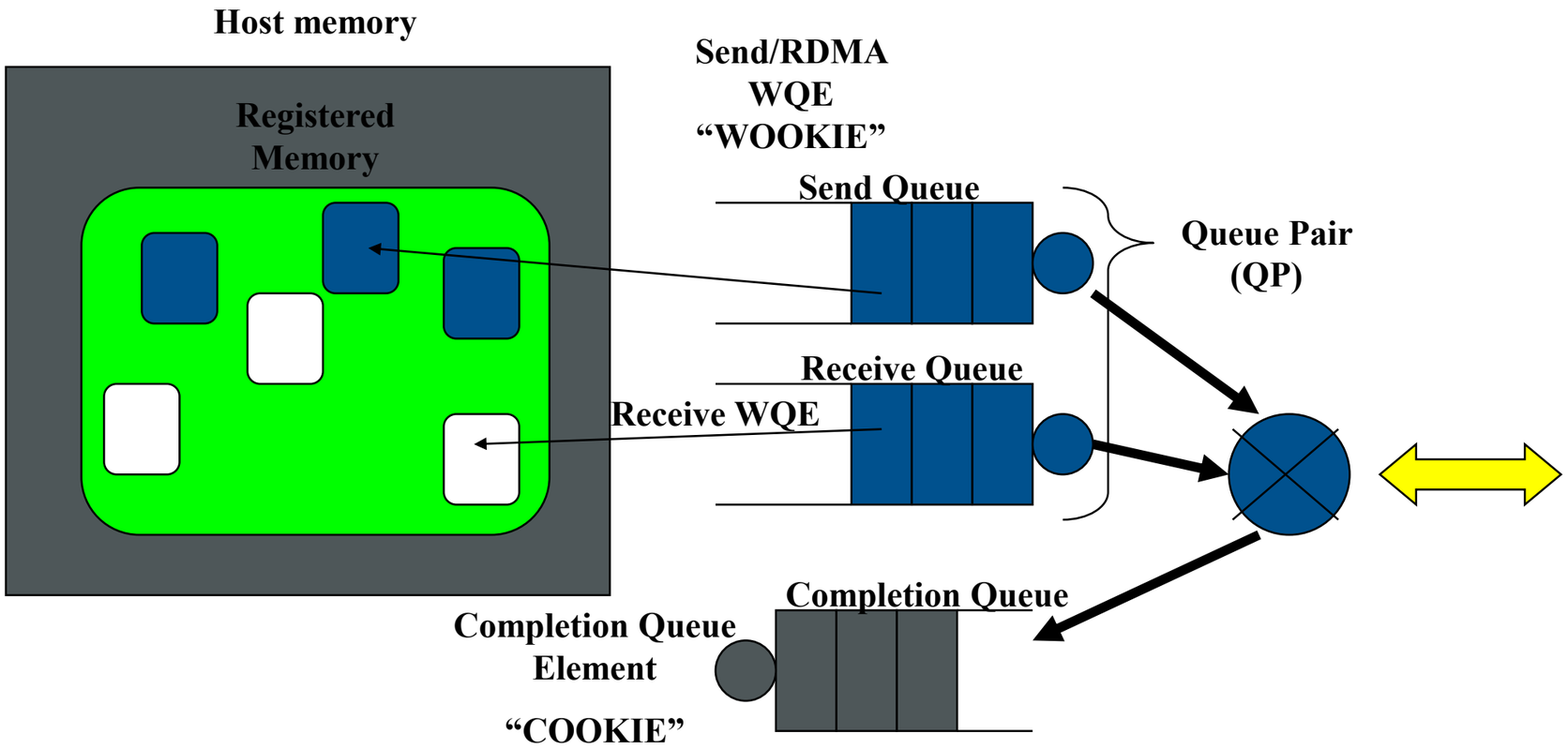
MLS Networking in Ethernet



RDMA – Bypasses the Kernel



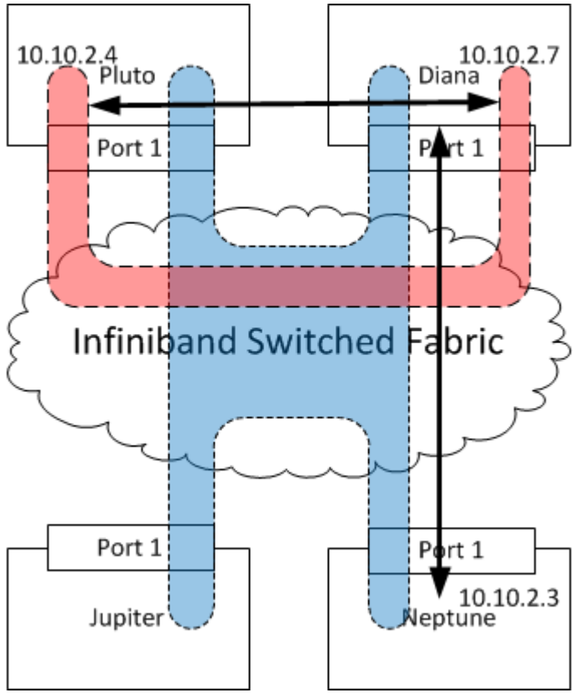
The RDMA Model



- WQEs are pointers to entire Memory Regions!

Partitions

- Partition – describes a set of endnodes within the fabric that may communicate
- Ports in different partitions are unaware of each other
  - Limited Membership
  - Full Membership
- Ports may be members of multiple partitions at once
- PKEY – partition label (a field in the LRH header)



IB Packet Format

Red fields are only present based on operation type and destination subnet

- LRH - local route header (required on all packets) – being addressed by LID
- GRH - Global route header (different subnets) - being addressed by GID
- BTH - base transport header (IB transport) – being addressed by QP number
- ExTH - Extended Transport Headers – Various headers of RDMA/Atomic/Ack
- MSG - Message payload
- ICRC - Invariant CRC (32 bit)
- VCRC - variant CRC (16 bit)

RDMA Traffic Labeling

- Applications initiate IO directly on HW-exposed endpoints (QPs)
  - HW generates all packets to satisfy transactions
  - The kernel is completely by-passed
  - Observation: arbitrary internal labeling or labeled networking is not an option

- RDMA traffic labeling must stem from HW architectural attributes
  - Network addresses (GIDs, LIDs) are not suitable
    - LIDs are not granular (single LID per HCA port)
    - GIDs are not always enforced (not carried by all traffic, UD QPs don't check GIDs)
  - Queue keys (Q_Keys) are not suitable
    - Apply only to UD QPs
    - Half of the key space is not enforced on transmission
  - Observation: partitions are the natural candidate
    - P_Key values are held in HCA partition tables
      - Populated by privileged network Subnet-Manager (SM)
    - P_Keys are carried on the wire of every data packet
      - The only exception is subnet datagram packets (SMPs), which are not accessible to applications
    - Every QPs is associated with a P_Key value
      - Determined by an index into the partition table
    - Partitions are strictly enforced at all times

Fundamental RDMA SELinux Support

- RDMA network security shall be based on partitioning
  - Host kernels control the association of P_Key values with security descriptors
  - SM configuration and P_Key assignment to HCAs shall ***not*** be part of the SELinux infrastructure
    - It is already a data-center wide privileged operation

- Object labeling
  - Each QP shall be associated with a security descriptor
    - Inherited by the creating process in the absence of a specific policy
  - Each RDMA_ID shall be associated with a security descriptor
    - Inherited by the creating process in the absence of a specific policy
  - P_Key value labeling
    - Associates a P_Key value with a security descriptor
    - System object descriptors are a good example (like network interfaces or nodes)
      - “system_u:object_r:rdma_partition_default_t”
      - “system_u:object_r:rdma_partition_topsecret_t”
  - Other objects shall not be labeled
    - Doesn’t make sense for other Verbs objects
    - RDMA devices and ports not required
    - GID addresses not suitable

Fundamental RDMA SELinux Support (cont.)

- Traffic labeling
  - Only network labeling shall be supported
  - P_Key values shall be the only network label

- Policies
  - The only policy rule is to allow a QP or RDMA_ID to be associated with a P_Key value
    - For example, “allow hpc_default_t rdma_partition_default_t : rdma_partition { modify }”, where
      - ‘hpc_default_t’ is the QP / RDMA_ID domain (type) inherited from the creating process
      - ‘rdma_partition_default_t’ is a partition security descriptor domain
      - ‘rdma_partition’ indicates that the subject is of partition type
      - ‘modify’ indicates that the QP is allowed to modify to reference the corresponding partition tag
  - More granular operations or rules are needed

- Enforcement
  - QP partitioning shall be enforced at all times
    - Upon QP creation, a violation shall result in an immediate error
    - During runtime
      - Any runtime violation due to policy changes or P_Key value changes shall transition the QP into ERROR state
  - RDMA-ID
    - All ingress/egress CM MADs shall be checked according to the partition policy
    - Any violation shall result in an immediate packet drop

Release Schedule

- TBD

Example Policy

- TBD

Thank You

Mellanox TECHNOLOGIES

Connect. Accelerate. Outperform.™