MACsec Replay Protection vs. Out of Order frames

Brian Weis

11/5/2014

Purpose of Replay Protection

- While applications should be designed and implemented to detect duplicate state, this cannot be guaranteed.
  - It is good hygiene for a cryptographic system to not allow itself to be used as a vector to confound application state, or to be mis-used as a DoS mechanism
  - If replays cannot be completely detected, the number of factors allowing them should be minimized

Traditional Replay Protection

- The traditional goal of Replay Protection is to ensure that only one copy of any sent frame is allowed past a receiver (absolute replay protection).
  - The usual method is for the sender to include a monotonically-increasing sequence number
- Receivers maintain a record of which sequence numbers have been validated as authentic, and duplicates of counter values are discarded.
  - The storage of replay state is limited, using a bit-map ("window")
  - The window can be of any size without losing effectiveness
- Any frame older the the bottom of the window is rejected as being a duplicate whether or not it had been seen before.
  - This is not a form of delay protection, simply an artifact of limited storage.

MACsec replay protection

- MACsec replay protection differs slightly from traditional methods
  - The window does not record Packet Numbers (PNs) that have been validated as authentic, and so duplicates within the window cannot be detected and discarded
  - A large window does not have any more storage cost than a small window. But the larger the window the less effective is the replay protection.
- This trade-off is friendly to hardware implementations of MACsec, where recording previously seen PNs in a bit-map would be costly.
- The only means to prevent all discards is to omit the window altogether (i.e., enforce Strict replay protection) where only PNs larger than the most recently PN validated as authentic is accepted.

Problem: Out of order frames

- Whether frames can be expected to be delivered in order depends upon the network connecting the sender and receiver(s)
  - Between a pair of bridges, highly likely
  - Across an Ethernet Virtual Circuit (EVC), it's uncertain (depends upon characteristics of the provider network)
- And if at any point Quality of Service (QoS) is applied to the frames, out of order packets are likely to be received out of order.
- The same is true if P802.1Qbu Frame Preemption is applied to a stream of frames.

One Solution: Changing the Enforcement Check

- Annex Q of P802.1Qbu proposes a solution for the Frame Preemption use case, which is to move the replay protection enforcement check forward

Q.7.2 Preliminary check only

In the anticipated preemption use case scenario, no further bridges are interposed between the MACsec transmitter and receiver and the initial fragments of each frame are received in PN order (Q.6). Therefore the preliminary replay check, just before the fifo in Figure Q-1, can be used. While this will not enforce strict replay protection at all times, the receive fifo is bound to empty frequently since it is not possible to arrange for the applied load to match the service rate exactly for extended periods without risking overrun. In terms of changes to .1AE all that is required is to remove or turn off the 'if (replayProtect) && (rv.pn < sa->lowestPN))' check that occurs after the receive fifo.

Considerations: Changing the Enforcement Check

- It is useful to consider the attack model of an all-powerful attacker
  - Scenarios can be constructed that allow replays even when a Strict setting is enabled.
- MACsec receive-side logic is changed

Another Solution: Individual MACsec SA for each P802.1Qbu traffic class

- Intuition: P802.1Qbu says "The Port is supported by two instances of the MAC service".
- Using the traffic class as input to the choice of transmit SA will allow receivers to match frames from each traffic class to a distinct set of replay state
  - No change to MACsec
  - As MAC implementations add support for P802.1Qbu, they can also plan to support more MACsec SAs
  - The method can be used in other use cases where there are traffic classes (even when there are more than 2 traffic classes)

MKA needs some updates

- MKPDU needs to declare when a sender is to use multiple SAs
- Two SAs are created from the distributed SAK, presumably each having a unique SCI
  - New SAKs could be computed from the distributed SAK
  - SAKs are fate-sharing
- Not sure if the live lists reflect the additional SCI values