IEEE P802.1bx MKA Extensions Station Suspension Discussion

Brian Weis

1/29/2013

Discussion Strategy

- It's helpful to begin by reviewing the IEEE 802.1X-2010 semantics for a device that becomes non-responsive.
- This will provide a foundation for the Suspension P802.1bx discussion.

IEEE 802.1X-2010 Liveness Semantics

- When a live peer does not prove liveness within 6-8 seconds,
  - Non-responsive peer removed from Live Peer List
  - There is no mention of disabling MACsec ReceiveSAs or TransmitSAs. As long as controlledPortEnabled is TRUE then they appear to continue to be in use and enabled. (More on this later.)

IEEE 802.1X-2010 "Expiration" (1)

- When Live Peer List is empty, and there are no replacement Participants, the Logon Process is notified of an "expired" event. (Clause 12.2)

— MKA.deleted(): Called by MKA to notify the Logon Process that it has deleted the actor, either because its life has expired or because it has been replaced, as principal actor, by a new principal actor without a change of elected Key Server.

Table 9-3—MKA Participant timer values

Timer use	Timeout (parameter)	Timeout (seconds)
Per participant periodic transmission, initialized on each transmission, transmission on expiry (9.4).	MKA Hello Time or MKA Bounded Hello Time	2.0 0.5
Per peer lifetime, initialized when adding to or refreshing the Potential Peers List or Live Peers List, expiry cause removal from the list (9.4.3).	MKA Life Time	6.0
Participant lifetime, initialized when participant created or following receipt of an MKPDU, expiry causes participant to be deleted (9.14).		
Delay after last distributing an SAK, before the Key Server will distribute a fresh SAK following a change in the Live Peer List while the Potential Peer List is still not empty.		

IEEE 802.1X-2010 "Expiration" (2)

- Does expiration imply the presence of a policy lifetime (e.g., associated a CAK in a CAK cache)?
- But when EAP is used, a station's Login Process would expect the MKA instance to be deleted when there are no more live peers, and to be told so that it take certain policy actions (e.g., set controlledPortEnabled to FALSE, remove authorization state placed on the port)
  - See Figure 7-7, Figure 7-10, Figure 7-12
  - Given that endpoints can be behind hubs and other intermediate relays, MKA feedback can provide the only clue that an endpoint has disconnected.

The Controlled Port (CP) state machine (Figure 12-2) is responsible for asserting the controlledPortEnabled signal (IEEE Std 802.1AE-2006, 10.7.5) that the PAE uses to control the MAC Operational status of the Controlled Port. When controlledPortEnabled is false, the client of the Controlled Port can neither receive nor transmit. The CP also controls the portValid signal, setting it true when communication through the port is secured by MACsec (to the extent controlled by the SecY control variables macsecProtect, macsecValidateFrames, and macsecReplayProtect).

IEEE 802.1X-2010
Participant Deletion

- Actually, Participant deletion is a little more complicated than that. (Clause 9.14)

An MKA participant shall be deleted as a result of any of the following:

h) The CAK lifetime (if specified) has expired.
i) The CAK was derived from an EAP exchange, but has not resulted in the recognition of a Live Peer (9.4) with an acceptable MACsec Capability (Table 11-6) within a period MKA Life Time (Table 9-3).

An MKA participant may be deleted as a result of any of the following:

j) The last key server to distribute a key using that CAK is no longer in the participant's Live Peers List (9.4.3, 9.4.4) but is (as identified by its SCI) on the Live Peers List of another participant that is using the same Common Port.
k) The number of participants would otherwise exceed the number that can be supported by the system.

- Only conditions h) and j) are probably relevant to this discussion and j) is a special case.

IEEE 802.1X-2010 Semantics Discussion Questions

1. What are the operational and security considerations of deleting/not disabling a MACsec SA for a dead peer?
   - There's an attack where the "dead" device can have its memory slurped, keys removed, and a new device can resume the MACsec (but not MKA) connection. This seems incongruous with MACsec and MKA protection mechanisms and security guarantees. (Note: This is an impersonation attack, not a case of the trusted component causing the attack).
   - Note sure if an enabled ReceiveSA for a "dead" peer is guaranteed to be replaced/disabled on a SAK change. It would be wrong to continue accepting MACsec traffic from a device who is no longer a live MKA peer.
   - Note that when a PSK from a CAK Cache is used, the MKA Participant probably never "expires" (see next item), and so the controlledPortEnabled will always be TRUE.
   - CP sets controlledPortEnabled to FALSE, which would appear to cause the SecY to mark all MACsec SAs as not enabled. (See Figure 12-1)
2. What does the term "expire" mean? Does an empty Live Peers List always result in a Participant deletion, and is it always reported to the Logon Process? Is this an optional step depending on policy?

P802.1bx Station Suspension

- Objective:

  The project scope includes provision for “the ability to maintain secure communication while the operation of MACsec Key Agreement (MKA) is suspended”. In other words, an MKA station wishes for MACsec peers to continue MACsec connectivity with it even during a time when its MKA process “suspends” and cannot respond.

- Value:

  This is important to infrastructure Ethernet devices, which often support an In-Service Software Upgrade (ISSU) process that allows frame switching and processing to continue while the software upgrades

Suspension with IEEE 802.1X-2010

- For a pairwise session, controlledPortEnabled will become FALSE on the live peer, causing SAs to be deleted. *Suspension is not possible*
- In a multi-way session, the remaining peers will continue to use the existing SAK. *Live peers may or may not leave the dead peer ReceiveSA enabled, and if it is enabled they may change the SAK at any time.*

P802.1bx Station Suspension

- Differing mechanisms have been proposed as a solution.
- In Santa Cruz discussions we realized some philosophical questions. The first one to tackle is whether Suspension is a Liveness or Key Distribution?
  - Is suspension a liveness semantic (i.e., having to do with the state of the peers?)
  - Is suspension a key distribution semantic (i.e., knowing when to distribute a new key)?
  - (Or is it both?)

Liveness Semantic

Pros

- Live members (and even new members) could be aware of suspended members. Peer Lists (stored and transmitted) are the natural data structures in which to store this state.
- Future SAK distributions are the result of existing live list change semantics.
- ReceiveSAs for a dead peer can be disabled, while those for suspended peers remain enabled.
- Suspended member could return with their previous MI value, avoiding unnecessary SAK changes.

Cons

- Key Server does not control which peers are suspended and when. There may be problems if each peer keeps its own state.

Key Distribution Semantic

Pros

- Suspended peers would always return with a new MI value, so future SAK distributions are triggered using existing semantics.
- Live members (and even new members) could be aware of suspended members, although the state might be transmitted in a non-peer list parameter set.

Cons

- A dead peer and suspended peer would not treated the same way at the MACsec level, requiring ReceiveSAs of both to remain enabled.
- Unnecessary SAK changes may occur after a suspension (but the cost may be low).
- Because stations joining the CA during a suspension need to be aware of suspending members, either a new data structure needs to be created to hold suspended identities, or the peer lists also need to be modified to hold that state.

Conclusions

- <To be filled in after the discussion>