2019-01-24; SDL -0207- RPC Message Protection Workshop Minutes

Note: Action items required from Steering Committee Representatives are in red

Attendees

- Ford (5 votes): Laura Towne, Stefan Bankowski, Zhimin Yang, Markos Rapitis, Hashavardan Patankar
- Toyota (5 points): Gaurav Pandey
- Mazda (3 votes): Marco Kok
- Luxoft (1 vote): Alex Kutsan, Maxim Ghiumiusliu
- Livio (Project Maintainer): Joey Grover, Jack Byrne, Jordyn Mackool, Brett Mcisaac, Nick Schwab

Detailed Proposal Overview (Zhimin Yang)

- Attached Appendix A
  - Proposal:https://github.com/smartdevicelink/sdl_evolution/blob/master/proposals/0207-rpc-message-protection.md

Review Issue Feedback (Project Maintainer)

- Attached Appendix B
  - Review issue: https://github.com/smartdevicelink/sdl_evolution/issues/634

Discussion:

- After debating between long term and short-term solutions, it was determined that based on member needs a short term solution had to be realized. The PM stated that moving forward if we have to do a short term, stop gap solution, the way in which the proposal is laid out is a more robust solution than other solutions/modifications mentioned like hard coding RPC requirements into Core's INI file. What needs to be defined is how the policy table gets that information to core. The policy table technical debt could will be minimized with better optimization of structures and organization, and the technical debt for mobile is not huge. It is important to make sure it is set up in a consist way.
- It was brought by the PM that a concern with the current proposal is how the policy table would organize the specific RPCs that need to be encrypted on a per app basis. The PM suggested coming up with a better way as to not extend the policy table exponential. A potential solution is to create a flag in function groups that they require encryption and then each app could have another flag on whether or not to enforce all the functional group encryption requirements. In the future, the goals is that an app is either fully encrypted or its not on an all or nothing approach at the transport level; if that's the case then what happens to those flags? Are they ignored?
  - Discussion outcome:
    - The flags would be ignored on new head units, but on older head units would still use them.
    - The benefit here is that the OEM can encrypt more things on the fly instead of hard coding into core.
- Right now, the OEM is able to pick the amount of and create their own functional groups (1- 16 or more)
  - A struct could be added that contains the functional groups that must be encrypted or a flag in the actual functional group that says "must be encrypted `true` or `false`.

        - For example: `remotecontrol` functional group would have an encryption `true` flag.
        - In the application entry in the policy table would have another flag that says if it requires RPC encryption or not.
            - There are two flags that allow you do this dynamically by RPC/functional group and by APP, but it is not listing out every RPC and app entry.
            - Example: Ford decides they want to start encrypting `vehicledata`, but there are apps on the road today that use `vehicledata` and they don't want them to break, so Ford would work with each app developer and slowly add that flag to the app entry itself (although it is already flagged to the functional group of `vehicledata`)
- It was noted that the force encryption flag per app level could be reused in the future for a long-term solution as well as the interim fix now.
- A question was probed: If RPCs can belong to multiple function groups, what happens when a specific RPC is in one functional group that required encryption and another that didn't?
    - In core it would default to if that functional group exists for that app then it would require encryption for all RPCs.
        - If an app has been assigned functional group A and B and both of them have a specific RPC, but A is encrypted and B is not; if the app has been granted permission for both, then it has to be encrypted. However if the app is only granted permission for group B, then group's A encryption requirement does not apply to that app.
            - Currently apps are granted functional groups that match the RPCs that they request. An OEM grants all functional groups or none.
            - This logic would need to be clearly stated (as stated in the current proposal)

    Next steps:

    1. Nick Schwab to provide feedback based on what the Policy table entry should be; post to review issue: https://github.com/smartdevicelink/sdl_evolution/issues/634
    2. Ford to revise the proposal to include the analysis for the Policy table entry.
    3. PM to write a draft of a new proposal #2 which is transport level that will have all RPCs encrypted.
        - Ford and PM to have an internal design section defining the proposal.
        - Once the proposal is confirmed by all parties, Ford will work internally to donate the code to SDL.

Official Members
Level 1 (Diamond): Ford, Toyota, Suzuki
Level 2 (Platinum): Subaru, Mazda
Level 3 (Gold): Xevo, Luxoft, Magellan, Elektrobit, Amazon, PSA, Pioneer, Denso, Clarion, KDDI, Dentsu, Abalta, Kawaski, Yamaha, iAuto, Panasonic, Line Corporation, JVCKenwood, Nissan, Nippon Seiki, Hakuhodo, NTT Docomo, NNG, Seven&i Holdings, Brison Inc., Isuzu Motors Ltd., Robert Bosch GmbH
Level 4 (Silver): Daihatsu, Intelematics, Mitsubishi Motor Corporation, Clarion Malaysia, UniMax Electronics, Garmin International, Micware Co. Ltd., Mitsubishi Electric Corporation, OE Works Mfg Inc.