Platform-based Design for Secure Medical Devices

By Eugene Kovalev

Closely advised by Paul Wortman - contains elements of his ongoing research
1. The current state of affairs
   a. Security in Medical Hardware
2. Internet of Things
   a. Its coming, whether we like it or not.
3. Existing Approaches
   a. Top-down design
   b. Bottom-up design
4. Platform-based Design
5. Advantages of PBD for medical devices.
Current State of Affairs

- Security in Medical Hardware
  - There is none...
- Notably this is not a disaster because most devices are not networked.
  - That’s rapidly changing
  - Pretty much no one is ready
  - Rush to market, minimal compliance to avoid liability, all that fun stuff
The Internet of Things

- It’s happening. Deal with it.
  - How do we deal with it?
- Design devices that are secure from the start
  - What design methodology can we use to make this process easy?
  - What design methodologies are out there?

**Four Categories of Networked Medical Devices**

1. **Consumer products for health monitoring:**
   - These devices -- such as FitBit, Nike FuelBand, or Withings -- generally communicate using Bluetooth to nearby personal mobile devices.

2. **Wearable, external medical devices:**
   - This category includes portable insulin pumps which often use proprietary wireless protocols to communicate.

3. **Internally embedded medical devices:**
   - Pacemakers and other medical devices are implanted in the patient but communicate wirelessly, either with proprietary wireless protocols or Bluetooth.

4. **Stationary medical devices:**
   - These devices, such as hospital-based chemotherapy dispensing stations or homecare cardio-monitoring for bed-ridden patients, often use more traditional wireless networks, such as WiFi networks in hospitals or patients' homes.
How do we design devices now?

- Definitions:
  a. Functional Space
  b. Architectural Space

- Top-down Design:
  a. Get the functional requirements first
  b. Try to meet as many as possible and amend others while moving into architectural space.

- Bottom-up Design:
  a. The opposite.
  b. See what hardware you have and then adapt.
Why do they suck?

- Bad examples of Top-down & Bottom-up design:
  - IoT Cars
  - Top-down design failures
    - That “secure” API
  - Bottom-up design failures
    - Car hardware has no chance against hackers
  - The system’s functions are blatantly unintegrated in the worst way possible
- Share lots of elements with Medical Devices
How do we fix this?

Platform-based Design (PBD)

- Considers both hardware and software requirements
  - Not nearly as obvious as it seems at first.
  - Not easy either.
- General Approach
- Examples
- How can it improve the current state of affairs?
1. Start with the functional requirements
2. Independently form the architectural requirements
3. Map one onto the other and reconcile as much as possible at the current design phase
   a. Sometimes experimentation and further development are needed for a more direct mapping.
4. Mapping becomes new functional requirement
5. Repeat recursively
Example

- Smart pill bottle
  - Functional Space:
    - Connect to medical provider
    - Secure communication
    - Small enough to fit in pocket
    - Carry pills
  - Architectural Space:
    - Pick CPU that can do crypto
    - Battery that lasts a whole day
    - Merge those two into a system module
  - Break functional requirements into smaller ones, build bigger system modules.
Advantages for Medical Devices

- Originates from the world of EE
- But starts with functional requirements.
  - Keeps its “eye on the prize” in both directions.
- Allows patient well-being and security to be concerns from the very start.
  - Doesn’t give up functional requirements for architectural and vice versa.
- Encourages upgradability because of high-level focus.
  - Better UI, Security, Overall Design.
Any Questions?