Internet of Things - Applications and Challenges in Technology and Standardization Notes

5.4 Medical and healthcare industry

IoT solutions in healthcare include providing a platform for monitoring 'medical parameters and drug delivery' using smart phones with RFID sensors (Radio-frequency identification sensors). This will allow the monitoring of diseases, unofficial diagnoses and emergency medical assistance. Perhaps the most important thing that MIoT can provide is the storing of patient data for analysis and for decision making when performing an operation. This is the most interesting for us since there's huge potential for security threats in data storing private information.

5.5 Independent living

There's also the use of MIoT for the elderly and assisted living patient. Using 'things' to 'watch' patients also involves storing data. Just a thought, but if an elderly or assisted living patient were relying on 'notifications in anomaly situations' an adversary has the potential to physically harm the patient.

5.6 Pharmaceutical industry

MIoT solutions can be used in the form of a smart label for drugs to track how many are being administered and their statuses. These drugs could be discarded if the necessary conditions of its effectiveness were altered while being transported. Again, an adversary has the potential to disable alerts on such solutions. Some of these solutions would be aimed at stopping the fraud of drugs, and that gives way for the potential of an adversary to thwart these efforts to ensure the continuation of a fake drug income by creating bogus tests or results. Patients could also be reminded to take their drugs on schedule.

Proposed Embedded Security Framework for Internet of Things (IoT) Notes

Introduction

IoT will further connect the world. This will make human life easier through response to our needs. It combines the network with physical devices. It works by having every day objects talk to one another. Because it's still pretty new, there's a lot of market potential.

In order to enable the progression of IoT, we must address its security issues.

Previous security issues haven't been properly protected against and since it's injected after the product has been made it can obstruct the design's original purpose. Designers focus more on functionality and businesses focus more on short term profit.

Virtual Shopping Scenario for IoT

IoT basically eliminates the need for human interaction through a series of data transmissions. These transmissions can be open to hackers who want to steal your sensitive information. The resource constraints of these devices also allow for greater security risks.

section*Attacks on IoT Systems These attacks include physical attacks like micro-probing and reverse engineering, side channel attacks like timing, power, fault, and electromagnetic analysis, environmental attacks, cryptoanalysis attacks like ciphertext, known plaintext, chosen plaintext, and man in the middle attacks, software attacks like a virust, trojan horse, logic bomb, worm or DoS attack, and network attacks like monitor and eavsedropping, traffic analysis, camouflage, DoS attacks, node subcersion or malfunction, capture or outage, message corruption or false node, and replication or routing attacks.

1. phsyical attacks - these attacks mess with hardware. they are difficult to perform as they usually involve exprensive material. 'some examples are de-packaging of chip, layout reconstruction, micro-probing, and particle beam techniques.

2. side channel attacks - these attacks involve recording and analyzing the radiation or power of the encryption device to narrow down possible encryption patterns for key obstruction.

3. cryptoanalysis attacks - these attacks aim to find the encryption key of a system

4. software attacks - these are the main source of vulnerabilities and can include buffer overflow attacks or malicious code injection.

5. network attacks - these are unique to wireless systems and include active or passive adversaries such as eavesdropping or DoS attacks respectively.

Security Requirement for IoT

The major security concerns for IoT include user identification, tamper resistance, secure software execution, secure content, secure network access, secure data communications, identity management, and secure storage.

(a) user identification - refers to validating users

(b) tamper resistance - refers to maintaining security even when in the hands of an adversary

(c) secure execution environment - refers to well managed code security wise during runtime

(d) secure content - Digital Rights Management (DRM) 'protects the rights of the digital content used in the system'

(e) secure network access - refers to providing connections only if the device is authorized

(f) secure data communications - refers to the authentication of peers that communicate, providing confidentiality and integrity to the system, and protecting identity

(g) identity management - user rights and privileges along with their authentication

(h) secure storage - 'involves confidentiality and integrity of sensitive information stored in the system'

Issues and Challenges

Some issues and challenges include

(a) security can be resource consuming. this is especially true of low power embedded systems.

(b) cryptography is expensive. we would have to make light weight and optimized security algorithms.

(c) kind of the same as above, security is expensive

(d) there's no one solution/ the solutions are tailored to each individual product so it's had to have a catch all

(e) IoT has security threats in both software and hardware

(f) there is a need for standard interoperable security protocols

Related Work

Existing solutions are divided into the following:

(a) software only approach - makes use of embedded General Purpose Processors (GPP). It is cheap and flexible but can consume too much power. It provides many solutions (which are?)

(b) hardware only approach - makes use of Application Specific Integrated Circuits (ASICs) for cryptographic algorithms in hardware. This is great for energy but isn't very flexible or cost efficient.

(c) hybrid approach - combo of the two previous. This is the best approach and does the best with efficiency, flexibility, and cost, however it involves having a deeper understanding of the big picture and inner workings of the system.

previous solutions are split into 'basic security functions and countermeasures against security attacks.' see **Table 1** for more details...

The authors argue that there needs to be a embedded security framework to shift security focuses from function centric to the design of the entire system.

Building Blocks

Reiteration of security being a part of the design, not an afterthought as mentioned in intro to security. Building blocks include:

(a) cryptographic algorithms - the foundation of security. Due to the constraints of embedded systems there needs to be lightweight, efficient, and easy to deploy cryptography scheme 'that provides high levels of security while minimizing memory, execution speed requirements and power requirements.' ECC may play a big part in accomplishing this.

(b) secure storage - due to the necessity of keys for cryptography, the secure storing of these keys is critical. It also has to be consistent and refrain from losing any memory. Possible solutions would involve utilizing on-chip ROM, OTP tech, and off-chip flash mem.

(c) secure boot - this is to ensure that a system can be brought to start in a trusted state.

(d) secure JTAG - a debugging interface for chips used during development and manufacturing as well as helping debug throughout a software's life cycle. It is also potentially exploitable (be sure to research more about this later)

(e) secure execution environment (SEE) - refers to processor that can execute files securely. Needs a secure kernel?

Proposed Embedded Security Framework

The authors argue that this would entail the following:

(a) an environmental factor - how the environment that encloses the system finds and reacts to threats.

(b) security objectives - determining which data to protect and how which attacks your system will protect itself against.

(c) requirements - determine what you need

The authors make their proposed security from the start mindset quite clear by providing a diagram that basically lays out the entire lifecycle of a piece of software and adds embedded security to it with arrows to every step of the lifecycle.

There is also, as mentioned before, a trade off between performance, cost, and security which almost always contradict one another. The authors beleive that the following would be key features of the security framework:

(a) lightweight cryptography

(b) physical security

(c) standard security protocols

(d) secure operating systems

(e) secure storage

Security and Privacy for Implantable Medical Devices

Introduction

Some of the devices we're talking about here are pacemakers, implantable cardiac defibrillators, drug delivery systems, and neurostimulators. Over 25 million US citizens are reliant on IMD's. Currently- this was written in 2008- there is defense against accidents like ID numbers and redundancy but there hasn't been any defense against intentional harm such as replay attacks. Security can differ between different IMD devices.

Safety and utility goals

Traditional IMD design goals include safety and utility.This to the author also encompasses reliability and treatment efficiency.

- data access - provides access to the right people
- data accuracy - as it sounds
- device identification - broadcasting its identity to those who need to know of its existence.
- configurability - being able to change the appropriate settings on the device
- updatable software - since removing implanted devices can be dangerous, the safest way to recall defective devices is through a software update
- multidevice coordination - at the time of writing this wasn't as commonplace as it is now
- auditable - being able to track the devices operational history
- resource efficient - with more communication comes more power consumption

Security and privacy goals

The authors will be looking at how confidentiality, integrity, and availability of computer security apply to these implantable devices. They will be focusing on the security goals for the device itself, not on the security goals of storing the patient's data on a back end server.

- authorization - This has several broad categories

- personal authorization - making sure that each role has its corresponding rights. most obvious example is patients vs. doctors. reminds me of computer privileges.
- role-based authorization - actually this seems pretty much the same as the last point
- IMD selection - ensuring that when a device talks to another device that it is only talking to the devices it wants to talk to, not broadcasting its information to the world. Something quite interesting that the authors bring up here is how a device might be configured to ignore authorization rules if there is an emergency situation in which intervention even by an anonymous source is better than no intervention at all.
- availability - they say it plain and simple right off the bat: 'an adversary should not be able to mount a successful denial of service attack against an IMD.' Adversaries shouldn't be able to drain battery, overflow data storage, or stop communication traffic.
- device software and settings - there should be bounds on the settings so that adversaries and patients themselves can't do anything stupid like set the device in dev mode. A great example the authors give is not being able to increase the amount of morphine expelled from a device. Physicians shouldn't have too much power, and should be restricted from altering audit logs or debug mode. These devices should only accept authorized firmware updates.
- device-existence privacy - the fact that a patient has an IMD should be unknown. This is an interesting one because I hadn't thought of this before. You would be making it quite a bit more difficult for an adversary if you made it relatively unclear if a patient has an implanted device or not.
- device type privacy - even if the device is known, it should further ensure that the type of the device is not know. The authors give some interesting examples, such as if there was a device to treat a social stigma, a terminal illness, or if the device is very expensive. So even if it was known that you were using an IMD, it wouldn't be giving away much about why.
- specific device ID privacy - 'an adversary should not be able to wirelessly track individual IMDs.'
- measurement and log privacy - basically, the adversary shouldn't be able to find out any information from the device.
- bearer privacy - even if the adversary is able to track a device, they souldn't be able to find out who's actually using the device. On a side note, I think it's interesting that the author's chose all of these angles to look at security. It becomes obvious that they expect at least one of these security goals to fail, and still have that failure

be relatively ok since the other security goals prevent the adversary from doing too much harm. This is a smart way to look at it, as often the goals of computer security fail.

  - data integrity - no adversary should be able to change the data of patients or their events.

Classes of adversaries

- passive adversaries - those that eavesdrop to the signals of the devices
- active adversaries - those that can initiate malicious communications with the device
- coordinated adversaries - like a tag team between someone who's close with a patient and someone who's a device programmer
- insiders - healthcare professionals, software developers, hardware engineers, and patients.

Equipment of adversaries

- standard equipment - using commercial equipment, perhaps stealing a device programmer from a clinic.
- custom equipment - personally build equipment that doesn't necessarily follow the laws of transmitter power, etc.

Tensions - Security versus accessibility

There are some tensions between security and privacy goals and traditional goals. The more secure the device is, the more difficult it could be for an operation, especially one in which you are not operated on by professionals you are used to seeing.

Security versus device resources

Strong security methods like pki can be exhaustive of resources. If a device uses cryptographic functions it could decrease its performance and shorten its lifespan. Something else I didn't know before: using cryptographic functions in a device could actually amplify the negative effect of a denial of service attack since the there can be repeated attempts at authentication which require greater use of resources or exhaust memory.

security versus usability

This is seen in traditional applications as well. The more secure you make something the harder it can be to use. User interaction quality may go down as a result of over-complication. This is especially a concern for medical users during an emergency situation. Another example is distance at which the device can communicate. This is shortened as the level of security increases.

mHealth App Development
HIPPA Requirements

What is mHealth

mHealth is mobile health. It is used to refer to consumer health apps by health care professionals. It includes wearables, but doesn't have to.

What are mHealth Applications?

Any software that runs on smartphones or tablets and manages or tracks personal health. This can include physical activity or biometrics such as health rate and prescription regimens. Some examples are RunKeeper and Couch to 5k. Sleep monitoring apps are also considered mHealth, as are heart rate monitors.

HIPPA Compliance and mHealth Applications

HIPPA stands for Health Insurance Portability and Accountability Act. This is a standard for sensitive patient data. It was created in 1996 and now it applies to a lot of technology. App developers have to be concerned with this if their app accesses 'medical records, billing information, health insurance information, and any individually identifiable health information.' This is considered private health information, or PHI. PHI is not 'calories burned, steps taken, or distance covered.' That means things like Nike Fuelband do not fall under accessing PHI, whereas things like Apple's health app do. If an app does track or access PHI then the app has to comply with HIPPA.

Developing HIPPA Compliant mHealth Applications

Apparently using True Vault 'will ensure that you meet the technical and physical safegaurds required by the HIPPA Security Rule.' Will have to look more into this late

Privacy and Security in Mobile Health (mHealth) Research

Introduction

New advancements in technology allow scientists to collect info from patients using real time wearables. These can use sensors and can provide a patient's 'biology, psychology (attitudes, cognitions, and emotions), behavior and daily environment.' This can give new insights into diseases and preventative measures for diseases and optimize a patient's outcomes and possibilities of recovery due to mHealth devices' ability to provide a constant stream of data. Even though mHealth has great potential it has progressed slowly due to concerns for privacy and security. Great quote: 'Because most mobile devices (including phones and sensors) are carried by the person and collecting data throughout the day, researchers are now able to begin thinking about big data at the level of the individual (Estrin 2014).' Followed by another great quote: 'Fusion of streaming biological, physiological, social, behavioral, environmental, and locational data can now dwarf the traditional genetics and electronic health records-based datasets of so-called big data.' People of who recently could not participate in research now can due to the accessibility of smartphones. When dealing with research participants, observers and health care professionals must be aware of the interconnected web of privacy, security, and confidentiality concerns that mHealth apps present. The National Committee for Vital and Health Statistics describes the differences between privacy, security and confidentiality with the following: 'Health information privacy is an individual's right to control the acquisition, uses, or disclosures of his or her identifiable health data. Confidentiality, which is closely related, refers to the obligations of those who receive information to respect the privacy interests of those to whom the data relate. Security is altogether different. It refers to physical, technological, or administrative safeguards or tools used to protect identifiable health data from unwarranted access or disclosure (Cohn 2006).' This article chooses to point out security issues regarding the specific mHealth applications of alcohol, drug use, and mental health.