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