White Paper

Securing IoT Nodes

Author
Andrew Bickley
Arrow Electronics EMEA
Technology Marketing Director, IoT

October 12, 2017

It is not news that the Internet of things has vast potential for value creation and is being heralded as the next phase in the hyper-digitization of our world.

IoT systems typically constitute centralized cloud systems connected to field devices which are connected to sensors. These connected systems find application in an array of fields ranging from industrial sensor networks, medical equipment, smart homes and buildings and even critical infrastructure like process control systems or SCADA devices. While setting up a simple IoT system is now trivial, building robust systems that are secure and tamper-proof requires considering many factors. With cyber-attacks escalating, the security of Internet of things systems is a growing concern as the impact of breaches could be financially drastic and the reputation damage everlasting. Security concerns related to IoT are cited as key inhibitors for widespread proliferation of IoT devices particularly in infrastructure applications.

Security in IoT systems has many aspects and breaches can occur along any single point of weakness. IoT security warrants a multi-modal approach that addresses legacy, current and emerging security challenges while contending with low-cost sensors and end-points, cloud networks and mobile users all at once. Recent hacking attacks have exploited the weakest link in the IoT architecture. For IoT, the “thing” is often the weakest spot and the IoT system is only as secure as its weakest endpoint. Software used to crawl the internet to identify end node vulnerabilities is easily available online and can exploit an IoT network. A survey conducted by 451 Research points to majority of IoT security concerns being are around the endpoints (Figure 1)

Contents

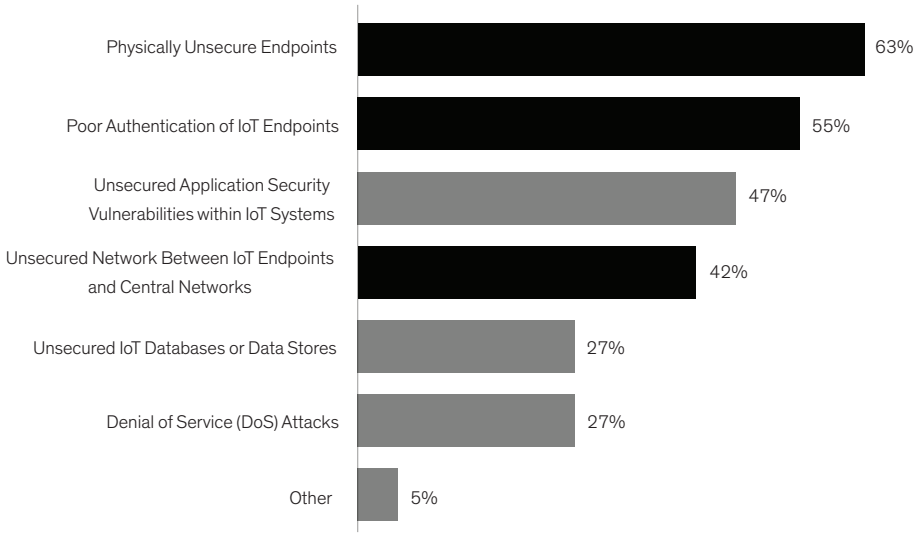

Figure 1: Survey of "Greatest IoT Security Concerns" (Source: 451 Research)

End-nodes are thus a unique area of security concern in IoT systems. The security of end nodes cannot be optional. Designs require careful consideration and special techniques. In this article, we focus on these aspects.

End Nodes & Compliance Classes

In devising a strategy to secure IoT nodes, it is important to keep in mind the capabilities of the end-node and the security compliance level required. End nodes can be classified into a few categories based on the any given node's specific capabilities related to memory, communication protocol support and operating system support. The degree of support has direct implications for the security measures that must be employed. Further, security sophistication will also vary based on the end device. Table 1 delineates the end node classifications.

Class 0	Class 1	Class 2	Class 3
> Very low-cost and constrained devices like low-power sensors. > Due to minimal memory and processing capabilities, they directly do not have IP-based communication and use gateways via low foot-print protocols like ZigBee and BLE. > Devices don't support RTOS	> Theses devices have limited memory and processing capabilities, communicate via protocols such as CoAP (Constrained Application Protocol). > Examples include blood glucose meter or a thermo-stat based on 8/16-bit MCUs. > They could communicate with other devices without the help of gateways. > RTOS could be implemented in these devices.	> These devices support communication protocols like HTTP and are less constrained. > They run on 32-bit MCUs or MPUs. > Examples include IP cameras, smart meters or high-end medical devices. > They could operate on an RTOS or Embedded Linux	> These are high-end nodes or gateways that can run protocols and applications with no modifications. > They can operate on an RTOS, Embedded Linux or full OSes

Table 1: IoT End-Node Device Classes

Understanding the true impact of a security breach – financial, safety, reputation etc. is essential to design a security scheme. To guide developers, iotsecurityfoundation.org has created security Compliance Classes based on implications of data/device breach. They are graded based on impact of a data breach on human privacy, business operations, infrastructure, and human safety. The integrity of the device, availability of the device and confidentiality of the data form the bedrock objectives for the compliance framework.

Compliance Class	Description	Security Objective		
		Integrity	Availability	Confidentiality
Class 0	Compromise to the data generated or level of control provided is likely to result in little discernible impact on an individual or organisation.	Basic	Basic	Basic
Class 1	Compromise to the data generated or level of control provided is likely to result in only limited impact on an individual or organisation	Medium	Medium	Basic
Class 2	In addition to class 1, the device resists attacks on availability that would have significant impact an individual or organisation, or impact many individuals, for example by limiting operations of an infrastructure to which it is connected	Medium	High	Medium
Class 3	In addition to class 2, the device is designed to protect sensitive data including sensitive personal data	Medium	High	Medium
Class 4	In addition to class 3, where the data generated or level of control provided or if a security breach occurs have the potential to affect critical infrastructure or cause personal injury.	High	High	High

Table 2: End-Node Security Compliance Classes

Detailed descriptions of the meaning of basic, medium and high-security objectives are here. The combination of the end-node device categories and compliance classes creates a robust structure for framing security related issues and provides guidance for developers.

End-Point IoT Security Considerations

The above definitions and terminology are the basis for considering many aspects that are involved in implementing appropriate security in IoT end-nodes.

1. **Protection Scope** – Developers and designers must understand the type and nature of data being captured by the end-points. Further, knowing if any potential breach will compromise privacy/confidentiality and break regulatory requirements is important. Since the entire promise of IoT resides in making even mundane devices like a light bulb 'smart', careful attention needs to be paid to devices that were never before considered a security threat. A 'light bulb' or 'door lock' may have no intrinsic data but data related to their usage – on/off times etc. can be used for malicious purposes. Device developers must assess the degree of protection that can be traded off with implementation costs. Class 0 type devices rarely require Class 4 level security. Understanding which class of security objectives would be required is crucial.
2. **Security breach impact:** Scenario planning is essential to assessing and modeling the true impact of a successful attack on the IoT system. Understanding what level of information will be compromised and the degree to which an intruder can infiltrate/control the system is important to devise effective counter measures. Class 3 devices that require Class 4 security deserve particular attention for scenario analysis and modes of security failure. Implications of breach on financials, stakeholder/public safety, and lost opportunity costs should be included in assessing impact. Simultaneously, un-intended consequences like granting access to non-IoT systems (ex. Enterprise ERP, MRP etc) should be also considered in assessing the risk.

3. **Security levels:** Implementing security measures for IoT devices can be a challenge. Creating access controls, authentication methods, encryption, etc. on constrained IoT devices without compromising functionality, user experience and adding cost is a tough task. Again, it is important to weigh the benefits of adding security with the cost of the breach. The level of security implemented will depend on the device class and could be through hardware and software techniques. Balancing flexibility, future-proofing, and robustness with cost effectiveness is crucial in choosing hardware or software based mechanisms for security implementation.

4. **Commissioning & Updating:** Securely commissioning and decommissioning end-nodes is an important part of any IoT implementation. Processes and measures must be in place for adding and removing IoT end-points from the network. Authenticating legitimate devices/users and validating network changes without human interaction is crucial. Mechanisms like verification of certificates and one-time use session keys are required for securely onboarding devices. Similarly, adding capability through remote device firmware upgrades is also a very important aspect of the security scheme. Software updates to add new features, fix security holes and keep the environment current should be done securely and only authorised software should be allowed to be installed in devices.

5. **Security protocols and processes:** Many aspects of security are not related to devices and networks. Often human factors lead to security breaches even in impregnable security schemes. To mitigate the human factors, companies deploying or managing IoT networks/systems need robust security protocols in place. Procedures to identify, react, respond and address security breaches proactively should be pre-defined and a core response team should be identified for exceptional circumstances.

The above, when considered at design and implementation time of IoT systems, can go a long way in ensuring that the right end-node security scheme is implemented while optimising for cost effective and security requirements.

Modes for Implementing IoT End-Node Security

Several mechanisms are available to robustly address the considerations outlined in the prior sections. Practitioners have many choices for implementing security measures in all aspects of the IoT system. The below list compiles a few of these mechanisms.

1. **Tamper detection** allows a device to sense any active attempt to compromise the device integrity or the data associated with the device. Attacks could be physical (probing), measurement of heat/electromagnetic radiations, or attempts to reverse engineer. Since electronic circuits emit heat and electromagnetic signatures, it is possible for an attacker to deduce information about data being processed without knowledge of the actual structure of the circuitry itself. Hardware-based tamper detection is robust and allows for defensive actions when attempts to read data or physically break into the device are detected. Tamper detect features also prevents reverse engineering by storing and processing device private keys in a secure environment.

2. **Secure Data Storage:** Embedded devices often store user data, passwords and other sensitive data. Using encryption ensures this data is safe from hackers. Further, keys used for encryption should be stored in a secure location as data can be decrypted if an attacker reads out the keys. Many hardware approaches are available to afford secure and protected memory for storing encryption keys.

3. **Securing Data transmission:** In typical embedded system architectures, devices and systems are connected across heterogeneous networks employing various standard and proprietary protocols. Since communications can be vulnerable to eavesdropping and falsification, secure transmission is paramount. Encryption keeps the message secret so only the authorized receiver can see it. Transport Layer Security (TLS) protocols encrypt communications and provide secure data transfer over the network. TLS ensures that trust is established between the server and the client before data transfer, preventing anyone from listening to and understanding the content.

4. **Authentication** is the process of identifying users, devices (end nodes, computers, machines) in the network and applications that run on these devices. Passwords, usernames or biometric recognition (facial, fingerprints, voice, etc.) are the primary modes of authentication in most enterprise systems. These modes establish a trust relationship with the system and allow the appropriate access. IoT systems, however, require methods that do not involve human interaction and often involve mechanisms like verification of certificates and one-time use session keys for AES encryption. Random number generators for one-time session key and a two-step authentication process usually results in highest security. Authentication can be implemented using both hardware and software-based approaches.

5. **Secure boot** blocks unauthorized booting of computing devices and prevents compromised devices from exchanging data. The secure boot process implements a chain of trust. Starting with an implicitly trusted component, every other component is authenticated before being executed. A secure boot scheme adds cryptographic checks to each stage of the boot process. This process checks the integrity of all of the software images that are executed and protects against unauthorized or maliciously modified software. Cryptographic protocols such as AES, RSA2048 or ECC521 are typically applied in IoT systems. A unique signature is generated and is saved in the device along with the device binary. Using the device's public key, the signature and the authenticity/integrity of the code can be verified on power-up, enabling secure booting.

6. **Secure firmware updates** can be challenging in IoT implementation. Protecting both the software itself and the system being updated is essential. Firmware over the air (FOTA) updates while efficient, create various security issues – a wrong/malicious firmware might be uploaded, the transmission of the new updates can fail, or the new firmware simply does not work as intended. Updating firmware while ensuring security, system stability, and transmission reliability requires authentication, version control, package integrity, complete and error-free transmission and operability check post update. All this has to be accomplished while limiting user interaction to the bare minimum to protect against human errors.

7. **Secure manufacturing** of IoT devices is essential to avoid counterfeits, protect the product ecosystem and ensure quality. At the manufacturing stage, secure firmware programming enables IoT device makers to reliably and securely program authorized firmware and also protect the firmware from being modified, pirated or installed on any cloned hardware. Using a hardware security module (HSM) during programming of production-level firmware to generate OEM product certificates, securely manage and store keys and provide tamper-detect alerts is essential for secure manufacturing.

8. **Secure decommissioning** of IoT end nodes and proper handling of associated assets (data) is an integral part of managing security. Solution providers must plan for device end-of-life and have processes to securely remove them from the network and ensure they do not introduce vulnerabilities that can be exploited. Data should be wiped clean and irretrievable. In certain instances, hard drives and data devices must be destroyed, shredded and properly recycled. Arrow offers Sustainability Technology Services to ensure devices are properly managed and a secure chain-of-custody is maintained throughout the entire lifecycle.

9. **Security policies and procedures** are essential to ensure that the human factors in security are adequately robust. To ensure the integrity, security, resilience, and quality of products and services as they move through the supply chain, managers should adhere to best practice based processes and operating procedures. In addition, the ability to track and manage issues throughout the entire lifecycle is essential. Policies and procedures should span key steps in the lifecycle management process including product design and development, manufacturing (OEMs), provisioning, third-party installation, device activation/reactivation/deactivation, device maintenance, device firmware updates and device recalls/retirement.

Summary

Security of IoT systems has been noted as a key inhibitor of the widespread proliferation of the internet of things. End node IoT devices are vulnerable to many threats. To understand and apply good end node security models, a standards-based classification of security levels and device types is needed. In addition, several factors including – protection scope, potential impact of a security breach, types of security levels needed, commissioning and upgrade models and overall security protocols and processes need careful consideration by practitioners (Figure 2). Once the right security needs, protocols and procedures are defined, designers and developers can unitize a variety tools and approaches to design and implement robust systems.

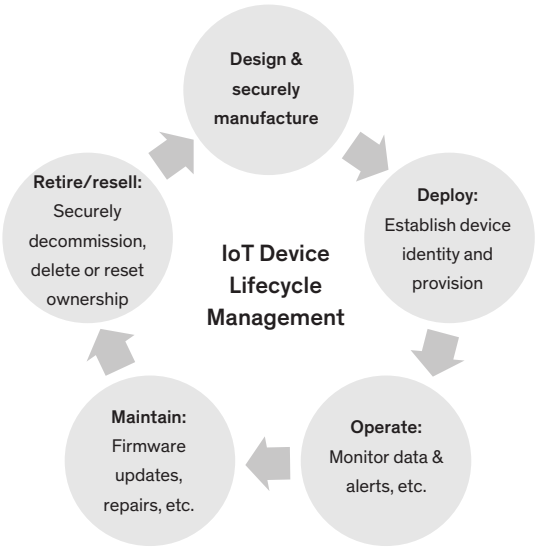

Figure 2: IoT Device Lifecycle Management

References

> IoT Security Compliance Framework
> Security Requirements for Embedded Devices – What is Really Needed?
> Embedded Hardware Security for IoT Applications
> Security Considerations Based on Classification of IoT Device Capabilities

Online
arrow.com/iot

Five Years Out

Arrow Central Europe GmbH
Internet of Things
Frankfurter Straße 211
63263 Neu-Isenburg
GERMANY

10_12/17