CSA cloud security alliance

Mobile Working Group

Peer Reviewed Document

Security Guidance for Early Adopters of the Internet of Things (IoT)

April 2015

Acknowledgements

Initiative Members

Brian Russell, Leidos (Initiative Lead)
Cesare Garlati (Mobile WG co-chair)
David Lingenfelter (Mobile WG co-chair)

K S Abhiraj
Gene Anderson
Megan Bell
Girish Bhat
Kyle Boyce
Poonlarb Chatchawalkhosit (CSA Thailand Chapter)
Michael Cook
Robert de Monts
Tom Donahoe
Chris Drake
Michele Drgon
Masaaki Futagi
Aaron Guzman
Nader Henein
James Hunter
Gregory Johnson
Alberto Manfredi
Arlene Mordeno
Valmiki Mukherjee
Mats Naslund
Javier Nieto
Chinmoy Rajpal
Tim Owen
Aniket Rastogi
Guido Sanchidrian
Eiji Sasahara
Jarrod Stenberg
Shankar Subramaniyan
Thriveni T K
Srinivas Tatipamula
Drew Van Duren
John Yeoh

Editors

Girish Bhat
Michele Drgon
Larry Hughes
Gregory Johnson
Arlene Mordeno
Jean Pawluk
Brian Russell
Shankar Subramaniyan
Thriveni T K
Srinivas Tatipamula
John Yeoh

Table of Contents

Executive Overview

This document is a product of the CSA Mobile Working Group — IoT Initiative. The document was created using inputs from a number of security and mobility experts representing diverse industries. We have tried to incorporate references and information from existing guidance in the field whenever possible in order to avoid duplication and promote alignment with the work of other industry bodies.

The guidance in this document has been created in a manner that allows for usefulness across industries. This was achieved by examining architectures across multiple industries and selecting security controls that would support each industry.

1. Introduction

This document provides guidance for the secure implementation of Internet of Things (IoT)-based systems. We borrow terminology from ITU-T Y.2060 to define various aspects of the IoT. Specifically ITU-T Y.2060 defines the **IoT** as a "global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies." ITU-T Y.2060 also provides the following definitions:

- **Device**: …"a piece of equipment with the mandatory capabilities of communication and the optional capabilities of sensing, actuation, data capture, data storage and data processing."
- **Thing**: …"an object of the physical world (physical things) or the information world (virtual things), which is capable of being identified and integrated into communication networks."

We have provided the guidance in this document to aid implementers of the IoT in deploying and using IoT in a secure manner. Traditional enterprise security solutions do not sufficiently address the security needs of the IoT as the IoT introduces new challenges including:

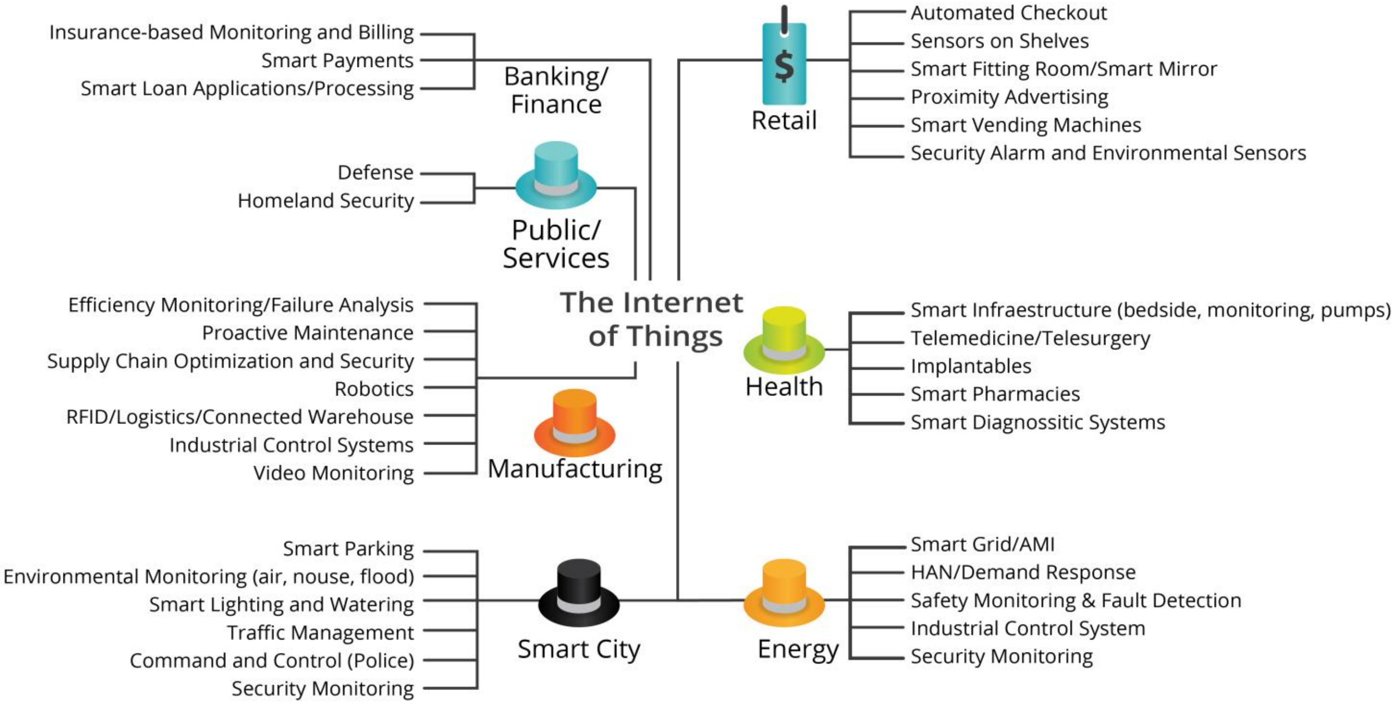
- Increased privacy concerns that are often confusing
- Platform security limitations that make basic security controls challenging
- Ubiquitous mobility that makes tracking and asset management a challenge
- Mass quantities that make routine update and maintenance operations a challenge
- Cloud-based operations that make perimeter security less effective

2. Purpose

The marketplace is seeing the beginning of widespread adoption of IoT within the consumer sector. Wearables, smart home appliances, lighting and other smart devices are becoming mainstream. The popularity of smart consumer devices is anticipated to continue to grow at a frenzied pace well into the future.

The adoption of IoT within the business and public sector has lagged behind the consumer market, however the 2015 Verizon IoT Report predicts that the number of Business-to-Business (B2B) IoT connections will increase 28% year-over-year between 2011 and 2020. Industries such as manufacturing, energy, transportation and retail are already adopting IoT initiatives. In their 2015 Industrial Internet of Things Positioning Paper, Accenture predicts that by 2030 the "Industrial" IoT will be worth $7.1 trillion in the United States alone and will support enhancements in efficiency, safety, productivity, and service provisioning.

Municipalities around the world are also adopting the IoT, working towards becoming smart cities that rely on data captured from thousands of diverse sensors spread across a geographic region. In the healthcare sector, we are also starting to see what the IoT will look like with manufacturers embedding network connectivity and intelligence within devices like patient bedside equipment. We can see the beginnings of interconnections between personal and business IoT capabilities, where smart wearables will soon be able to collect information and transmit that information to healthcare providers through the cloud. The transportation sector is another exciting area where the concept of IoT connected vehicles is sprouting and the infrastructure to support these vehicles is gaining traction. Furthermore, experiments with driverless cars will yield a future where the ability to collect and analyze sensor data from IoT-based roadside equipment (RSE) will become even more important. In the energy sector, integrated and interconnected systems (eg. modern substation integrated systems, smart grid systems) are trending to increase the level of power system automation and remote accessibility to deliver information to a wide range of users in near-real-time and also to control the number of tasks involved to streamline operations and performance.


The Internet of Things
Banking/ Finance
Insurance-based Monitoring and Billing
Smart Payments
Smart Loan Applications/Processing
Public/ Services
Defense
Homeland Security
Manufacturing
Efficiency Monitoring/Failure Analysis
Proactive Maintenance
Supply Chain Optimization and Security
Robotics
RFID/Logistics/Connected Warehouse
Industrial Control Systems
Video Monitoring
Smart City
Smart Parking
Environmental Monitoring (air, nouse, flood)
Smart Lighting and Watering
Traffic Management
Command and Control (Police)
Security Monitoring
Retail
$
Automated Checkout
Sensors on Shelves
Smart Fitting Room/Smart Mirror
Proximity Advertising
Smart Vending Machines
Security Alarm and Environmental Sensors
Health
Smart Infraestructure (bedside, monitoring, pumps)
Telemedicine/Telesurgery
Implantables
Smart Pharmacies
Smart Diagnossitic Systems
Energy
Smart Grid/AMI
HAN/Demand Response
Safety Monitoring & Fault Detection
Industrial Control System
Security Monitoring

As each industry begins to implement capabilities of the IoT to meet unique needs and requirements, it will be important to understand that each unique implementation should be evaluated for security weaknesses. Although this document provides a generic set of security controls for the IoT, there will always be some level of customization required given the context of each distinct IoT implementation.

Some interesting cybersecurity data points to consider as the IoT readies for mass adoption:

- 80 percent of Amazon's top 25 best-selling SOHO wireless router models have security vulnerabilities*[SOHO]
- 30 percent of IT professionals and 46 percent of employees do not change the default administrator password on their wireless routers*[SOHO]
- The average age of the code base on those ubiquitous low-end routers is 4-5 years*[DG]
- With Linux.Darlloz malicious code, 38% on infections are IoT devices like routers, set-top boxes, cameras and printers*[SYM]

3. IoT Threats to Individuals and Organizations

The IoT introduces large quantities of new devices that will be deployed or embedded throughout an organization or even within a system. Data captured from these devices can then be analyzed and acted upon. In some cases, the deployed devices are capable of performing some tasks. These described edge devices will become ubiquitous and allow for massive data collection activities. The analysis of this data will allow previously unseen linkages to be made which may cause concern for the privacy of individuals or groups of people. In some cases, individuals may not even be aware that they are being tracked or recorded given the ability for next generation microchips to be embedded in virtually any platform. In all cases, assuring the security of each component within an IoT system is important to keep malicious actors from taking advantage of the power of the IoT in an unauthorized manner.

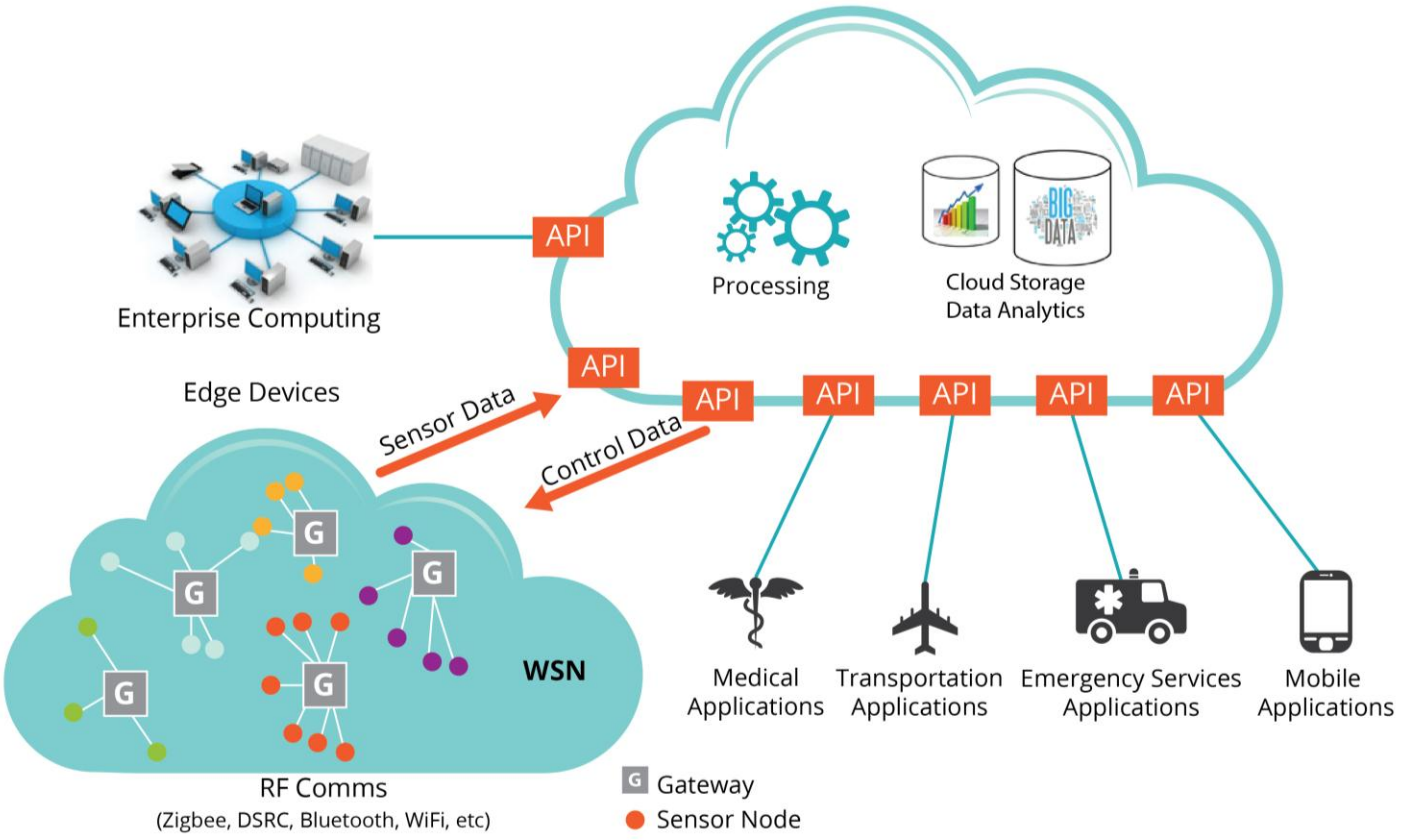
Some examples of new threats and attack vectors that malicious actors could take advantage of are:

- **Control systems, vehicles, and even the human body can be accessed and manipulated causing injury or worse** through unauthorized access to physical sensing, actuation and control systems (including vehicle, SCADA, implantable and non-implanted medical devices, manufacturing plants and other cyber-physical implementations of the IoT)
- **Health care providers can improperly diagnose and treat patients** based on modified health information or manipulated sensor data
- **Intruders can gain physical access to homes or commercial businesses** through attacks against electronic, remote controlled door lock mechanisms.
- **Loss of vehicle control** can be caused by denial-of-service against internal bus communications
- **Safety-critical information such as warnings of a broken gas line can go unnoticed** through DDoS of IoT sensor information
- **Critical infrastructure damage can occur** through override of safety critical features or power supply /temperature regulation
- **Malicious parties can steal identities and money** based on leakage of sensitive information including Personal Health Information (PHI)
- **Unanticipated leakage of personal or sensitive information can occur** by aggregating data from many different systems and sensors, or the merging of personal data that has been collected under differing consumer privacy preferences and expectations

- **Unauthorized tracking of people's locations** can occur through usage pattern tracking based on asset usage time and duration
- **Unauthorized tracking of people's behaviors and activities** can occur through examination of location-based sensing data that exposes patterns and allows analysis of activities, often collected without explicit notice to the individual
- **Unlawful surveillance** through persistent remote monitoring capabilities offered by small-scale IoT devices
- **Inappropriate profiles and categorizations of individuals can be created** through examination of network and geographic tracking and IoT metadata
- **Manipulation of financial transactions** through unauthorized POS and mPOS access
- **Monetary loss** arising from the inability to provide service
- **Vandalism, theft or destruction of IoT assets** that are deployed in remote locations and lack physical security controls
- **Ability to gain unauthorized access to IoT edge devices** to manipulate data by taking advantage of the challenges related to updating software and firmware of embedded devices (e.g., embedded in cars, houses, medical devices
- **Ability to gain unauthorized access to the Enterprise network** by compromising IoT edge devices and taking advantage of trust relationships
- **Ability to create botnets** by compromising large quantities of IoT edge devices
- **Ability to impersonate IoT devices** by gaining access to keying material held in devices that rely upon software-based trust stores
- **Unknown fielding of compromised devices** based on security issues within the IoT supply chain

The IoT relies upon edge components that collect data or perform some action. These components may take the form of standalone devices, for example smart sensors or smart meters, or be embedded in larger systems, such electronic control units (ECUs) of connected vehicles. These edge components collect, store or process data. They are networked, either together or through some gateway typically using Radio Frequency (RF) communications. This allows for communication with a backend service, oftentimes, hosted within the cloud. Data analytics systems can make sense of data and in some cases instruct the components to perform some action. There will be a number of applications that make use of data collected from IoT edge components, or the resultant analysis derived from edge components.

The Internet of Things

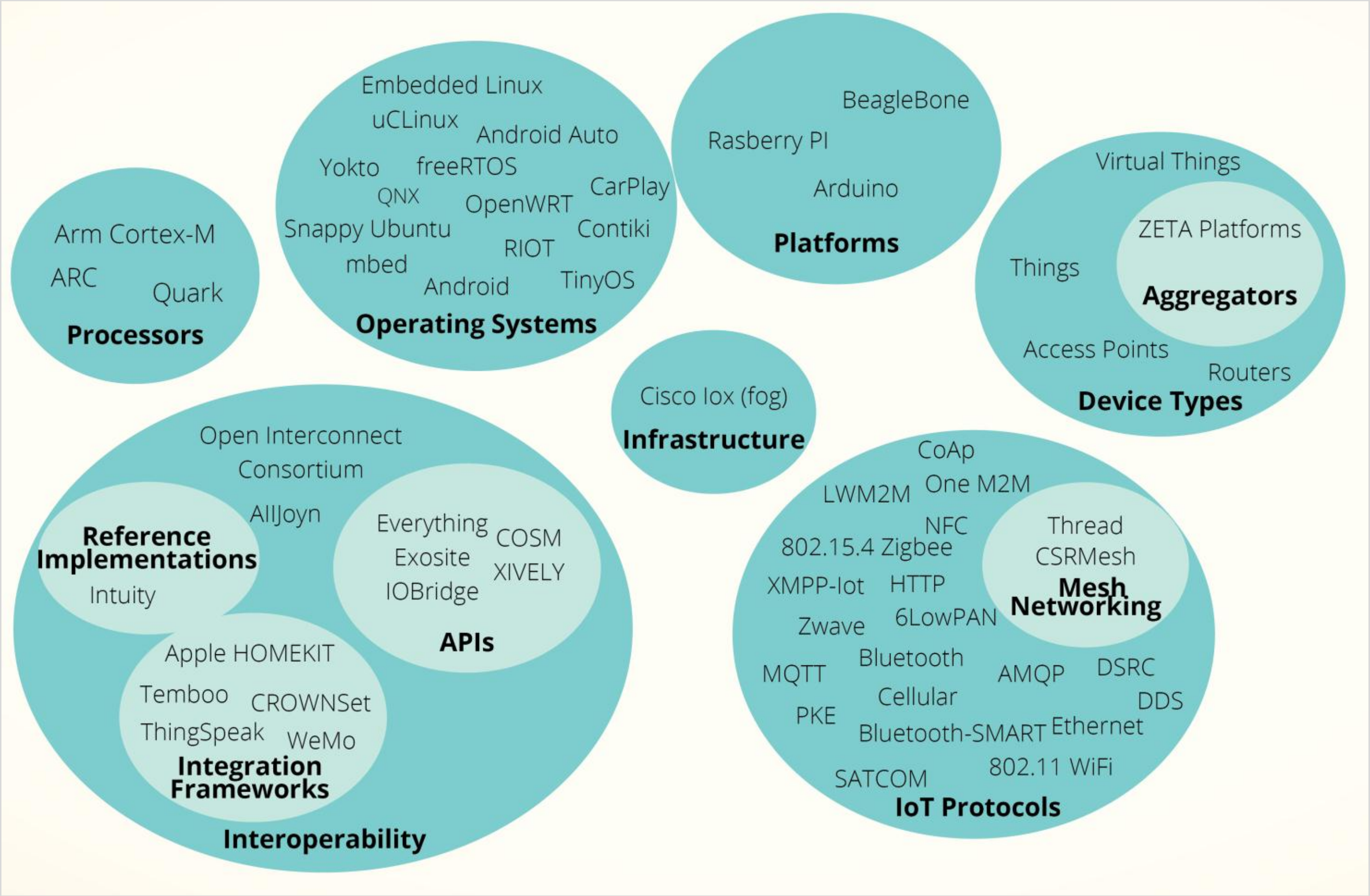


Because the generation and analysis of data is so essential to the IoT, consideration must be given to protecting data throughout its lifecycle. Managing information at this level is complex because data will flow across many administrative boundaries with different policies and intents. Individuals will surely have different privacy goals than corporate entities, which in turn will have different goals than government or other organizations. Oftentimes, data is processed or stored on edge devices that have highly limited capabilities and are vulnerable to sophisticated attacks.

Privacy implications must also be considered to include developing an understanding of potential privacy issues when many different sources aggregate to a single point. Privacy controls are required at various points across the IoT ecosystem, particularly at point of user consent to data capture, transfer of data between IoT partners and at the points within the system that the data is stored and used.

Given the various technological and physical components that truly make up an IoT ecosystem, it is good to consider the IoT as a system-of-systems. The architecting of these systems that provide business value to organizations will often be a complex undertaking, as enterprise architects work to design integrated solutions that include edge devices, applications, transports, protocols, and analytics capabilities that make up a fully functioning IoT system. This complexity introduces challenges to keeping the IoT secure, and ensuring that a particular instance of the IoT cannot be used as a jumping off point to attack other enterprise information technology (IT) systems.

The IoT Ecosystem



International Data Corporation (IDC) estimates that 90% of organizations that implement the IoT will suffer an IoT-based breach of backend IT systems by the year 2017*[IDC]. This is an interesting data point, given current concerns related to the lack of security engineering and secure development best practices employed by many IoT developers today. For early adopters, enterprise mitigations that take into account the potential vulnerabilities exposed by insecure IoT platforms are needed.

This guidance from the Cloud Security Alliance (CSA) discusses some of the challenges associated with the adoption of the IoT and concludes with a set of recommendations that can be followed by early business adopters of the IoT to meet the following goals:

- Maintain the confidentiality and integrity of both business and personal data collected within the IoT through the provisioning of encryption, authentication and integrity protections throughout the IoT infrastructure
- Understand and address stakeholder privacy concerns prior to the implementation of the IoT capabilities by performing a privacy impact assessment
- Safeguard the infrastructure from attacks that target the IoT as a vector into an organization's assets, through the use of IoT device life cycle controls and a layered security approach
- Initiate a global approach to combat security threats by sharing threat information with security vendors, industry peers and Cloud Security Alliance

4. Challenges to Secure IoT Deployments

There are many challenges to deploying a secure IoT implementation, and many of the existing security technologies on the market will play a role in mitigating IoT risks within an enterprise. However, the IoT also introduces new challenges to security engineering. Many of these would benefit from targeted research or industry collaboration to to determine the optimal long-term approaches to resolution. This table provides the CSA's view of the top challenges facing early adopters of the IoT with a mapping to the recommended CSA IoT security control detailed later in the document.

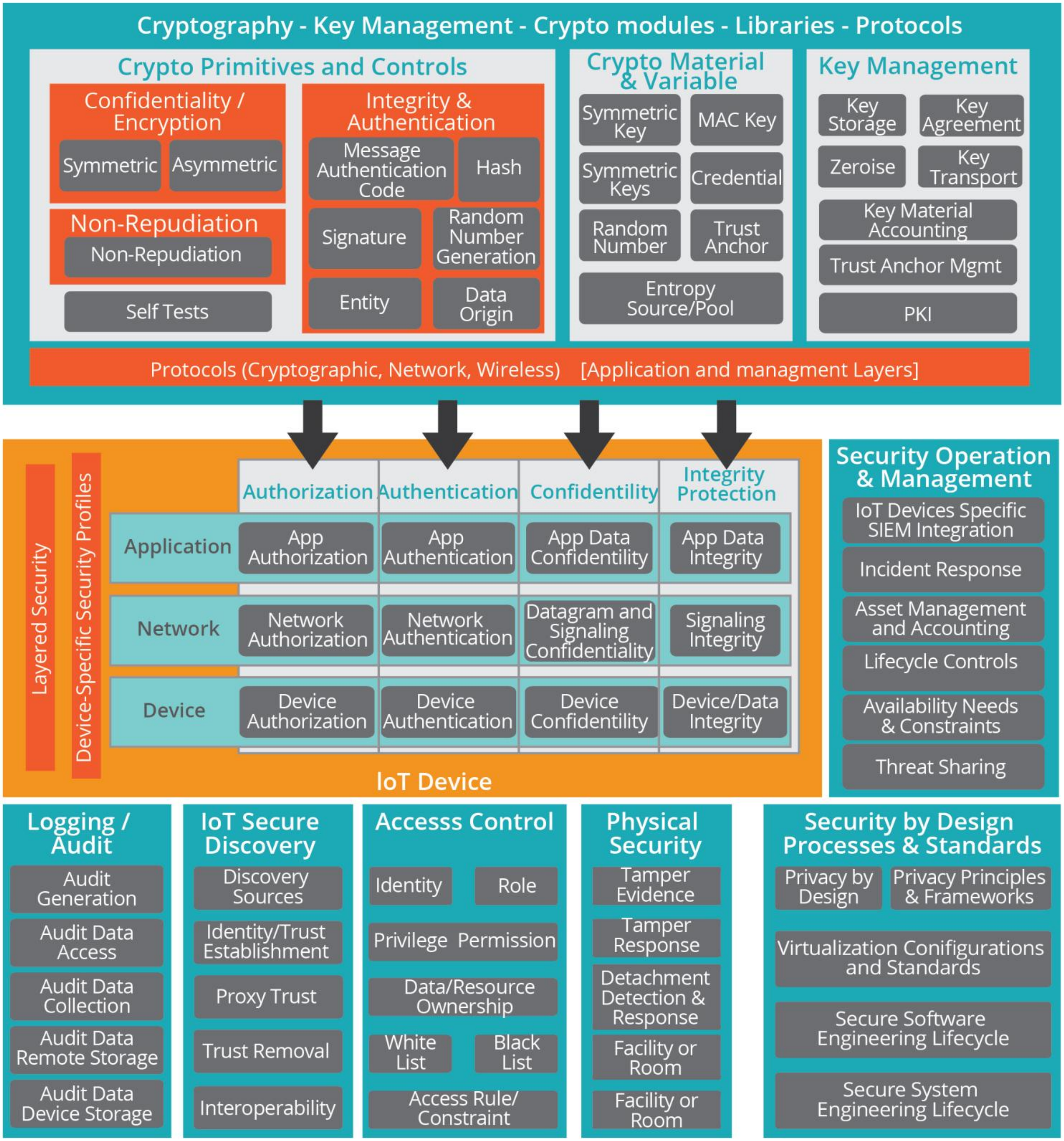
Key Challenge	Challenge Discussion	Mapping to Recommended CSA IoT Security Control
Many IoT Systems are poorly designed and implemented, using diverse protocols and technologies that create complex configurations.	The IoT encompasses edge devices, messaging and transport protocols, Application Programming Interfaces (APIs), data analytics, storage, software, and various other technology concepts. Edge devices themselves are complex, consisting of multiple layers of technology and requiring an understanding of hardware, firmware, software and a plethora of protocols. All of this can be applied to myriad use cases across many industries. Before being able to secure a system, it is important to first understand the functional and technological details of the system to be secured. This will require security engineers to work closely with the developers of the IoT capability to introduce security requirements early in the design process. Using a methodical systems security engineering approach for each IoT implementation within an enterprise is recommended. Taking a systems security engineering approach to IoT implementations allows designers to identify areas of complexity that can be simplified. As an example, limiting implementations to the use of as few protocols and touch points as possible.	#2: Apply a Secure Systems Engineering approach to architecting and deploying a new IoT system.
Lack of mature IoT technologies and business processes	Standards supporting the IoT have not yet been fully developed, leaving the market open to competing platforms, protocols, and interfaces. This lack of standards drives increased complexity which can introduce vulnerabilities and provides attackers with a way to infiltrate the enterprise.	#2: Apply a Secure Systems Engineering approach to architecting and deploying a new IoT system.
Limited guidance for lifecycle maintenance and management of IoT devices	Guidance on the secure configuration of the limited capability operating systems that underlie many IoT edge devices is limited or nonexistent. Performing firmware, software and patch updates for IoT devices will require a new approach with considerations given to identifying update provisioning obligations and responsibilities throughout the supply chain. Organization's procuring IoT assets should also clearly understand and agree on the vendor's model for licensing to ensure that they are able to continue receiving patches and software updates throughout the course of the IoT asset's life. If IoT devices fall behind on required security updates, they will be much easier for attackers to exploit. In this regard, organization's should consider the likelihood that IoT devices will eventually become unsupported as phase-out dates come into play from each vendor. Keeping track of IoT devices and the software and firmware on each device is also an issue. The amount of IoT devices alone introduces a challenge to effectively managing them.	#5: Define Life-cycle controls for IoT devices

Key Challenge	Challenge Discussion	Mapping to Recommended CSA IoT Security Control
The IoT introduces unique physical security concerns	Many IoT edge devices will be deployed in exposed environments, allowing attackers to more easily acquire them for further lab analysis. This is concerning because most IoT edge devices are limited in capability, requiring software-based solutions for the protection of sensitive material such as cryptographic keys. Attackers with sufficient resources can reverse engineer these edge devices. Ideally, the use of tamper-resistant protections would be implemented however this may not always be feasible. The fact that many IoT applications desire very low-cost devices causes a conflict with devices' ability to withstand attacks and tampering.	#3: Implement layered security protections to defend IoT assets
IoT privacy concerns are complex and not always readily evident.	Some privacy concerns are not readily identifiable and some concerns are not solvable by simply enforcing confidentiality protections, identity or location to transactions.	#1: Analyze privacy impacts to stakeholders and adopt a Privacy-by-Design approach to IoT development and deployment
Limited best practices available for IoT developers	Many IoT developers are not yet familiar with secure development best practices. The rush to create new IoT-based capabilities will likely result in limited focus on the security of the new functionality being created.	#2: Apply a Secure Systems Engineering approach to architecting and deploying a new IoT Systems.
There is a lack of standards for authentication and authorization of IoT edge devices	Requirement for low-power and wearable devices bring a wealth of new, simpler wireless protocols, which often meshes together and do not implement mature and secure encryption and authentication; these protocols can be attacked "on the fly" and without physical contact Some IoT devices have no authentication capabilities while others have limited support. Very few have capabilities that support multi-factor authentication. It is also not clear how useful multi-factor authentication for IoT edge devices will be in general. One of the primary benefits of traditional 2-factor authentication is that one of the "factors" is "out-of-band" relative to the other. But, in IoT devices, both of the credential (e.g., keys) may need to be stored in the same device, losing the out-of-band benefit. Although some standards or commercial options are available (e.g., certificate authentication, commercial or semi-commercial identity providers such as Google, there is a lack of ability to create device-specific profiles and authorization options and the privacy implications of using these services providers has not been fully explored. .	#6: Define and implement an authentication/ authorization framework for the Organization's IoT Deployments
There are no best practices for IoT-based incident response activities.	Organizations must be able to plan for the compromise of IoT devices, keys and certificates. This includes performing forensic analysis on compromised systems and devices.	#5: Define Life-cycle controls for IoT devices

Key Challenge	Challenge Discussion	Mapping to Recommended CSA IoT Security Control
Audit and Logging standards are not defined for IoT components	Monitoring IoT edge devices for security events poses unique difficulties. Many of these edge devices will be single-purpose sensors that may not be capable of tracking all interactions with the device. Other devices may be limited in their ability to instantiate an RF connection for the purpose of sending audit logs, based on battery constraints. Obtaining near real-time situational awareness of the security posture of IoT devices will be difficult. Another challenge is aggregating log data from many widespread IoT segments into a single event management system, and then actually being able to derive some intelligence from the activities within each of these segments.	#7: Define and implement a Logging/audit framework for the Organization's IoT ecosystem
Restricted interfaces available to interact IoT devices with security devices and applications. No focus yet on identifying methods for achieving situational awareness of the security posture of an organization's IoT assets.	Integrating IoT devices into an organization's existing security system would provide situational awareness of the overarching security posture of the organization. Unfortunately, there are typically no interfaces made available to connect with existing SIEM systems, and options are typically limited for connecting with Identity and Access Management systems and other security systems. Given that this is the case, it is likely that intermediary products will soon rise to support brokering between IoT device pools and an organization's security infrastructure.	#3: Implement layered security protections to defend IoT assets #6: Define and implement an authentication/ authorization framework for the Organization's IoT Deployments #6: Define and implement a Logging/audit framework for the Organization's IoT ecosystem
Security standards for platform configurations involving virtualized IoT platforms supporting multi-tenancy is immature.	This involves use cases where the "cloud" stretches all the way out to the device (e.g., two businesses being hosted as tenants on the same physical IoT platform). This results in the need for lightweight, yet secure virtualization /isolation solutions.	#3: Implement layered security protections to defend IoT assets

5. Recommended Security Controls

The following security controls are recommended for organizations implementing IoT capabilities. These controls have been tailored to IoT-specific characteristics to allow early adopters of the IoT to mitigate many of the risks associated with this new technology.



Control	Description
1	Analyze privacy impacts to stakeholders and adopt a Privacy-by-Design approach to IoT development and deployment
2	Apply a Secure Systems Engineering approach to architecting and deploying a new IoT System.
3	Implement layered security protections to defend IoT assets
4	Implement data protection best-practices to protect sensitive information
5	Define lifecycle controls for IoT devices
6	Define and implement an authentication/authorization framework for the organization's IoT Deployments
7	Define and implement a logging/audit framework for the organization's IoT ecosystem

5.1. Analyze privacy impacts to stakeholders and adopt a Privacy-by-Design approach to IoT development and deployment

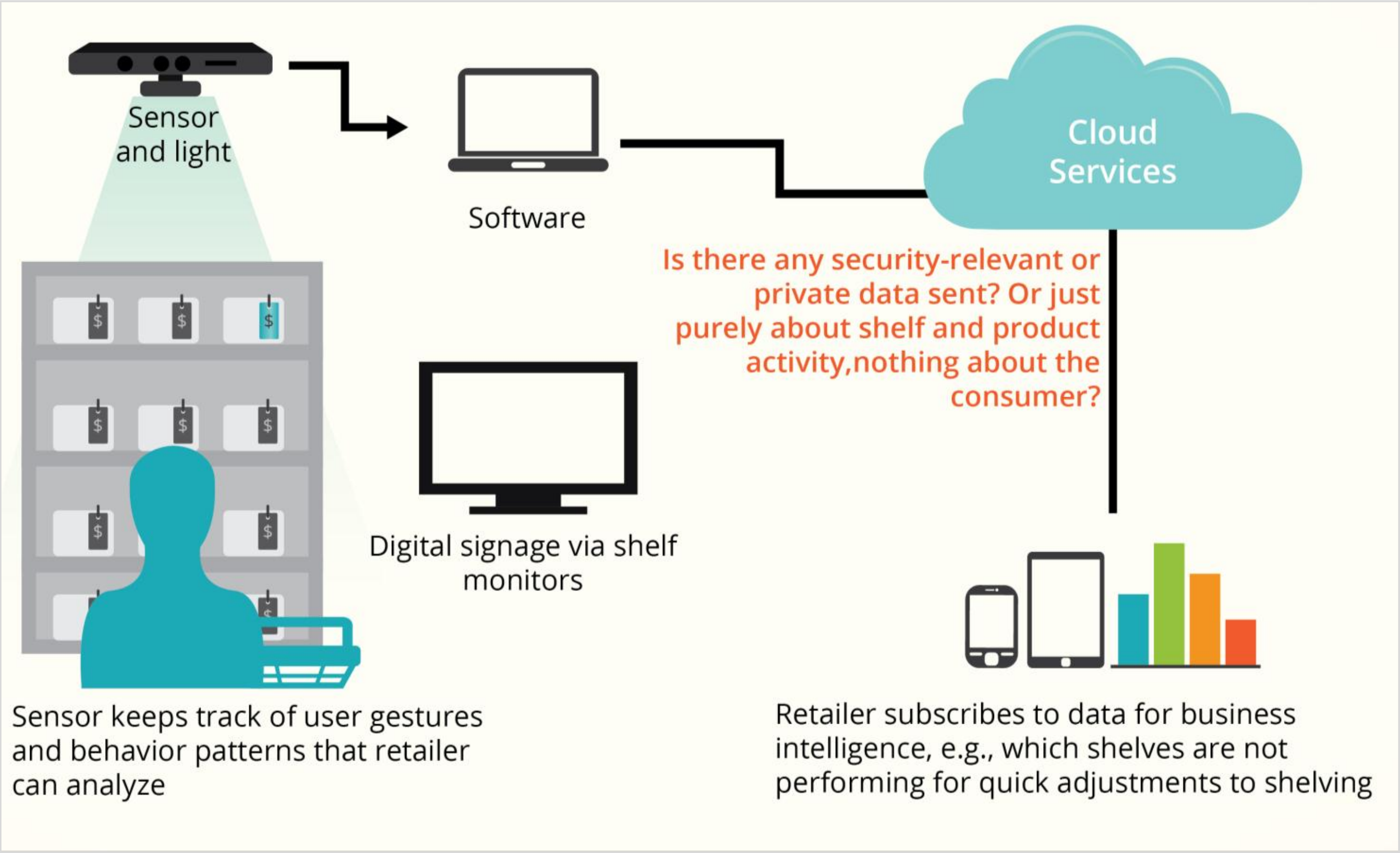
The IoT provides organizations with powerful tools for collecting and analyzing data. This data comes in many forms, and in many cases with the IoT, there is residual data that is either collected or can be assembled through careful analysis. As organizations begin to adopt the IoT we will see the placement of sensors, video cameras, and other hardware aimed at collecting information. These IoT components will be deployed pervasively in public spaces as well as private homes, and in some cases even worn by individuals. Many IoT components will include the use of Global Positioning System (GPS) trackers that can provide location-tracking of individuals or those individuals' assets (e.g., cars/telephones). Another aspect of the IoT is that many IoT systems will overlap in regards to the types of data that is collected. As such, the potential to expose sensitive information in aggregate is raised, even if the two collection systems are operated by entirely different entities. In these instances, enterprising marketers or malicious attackers can make use of this aggregate data to meet their objectives, without the knowledge of the individuals being tracked.

One of the unique challenges related to privacy in the IoT is that there will soon be an ability to overwhelm society with data collection devices and sensors. These devices will sometimes be used maliciously and other times may inadvertently capture information about individuals that have not consented to being tracked. From a system-owner perspective it will be important to understand what actions are allowable on the data that is collected inadvertently from individuals.

IoT sensors will also be used in ways that enhance a customer experience however. In these instances the customer will be provided notification that they are interacting with some IoT system. A prime example of this can be found in the retail industry. A number of examples of IoT deployments in the retail industry are provided below. These examples provide a good foundation for understanding the questions to ask during designs of IoT systems, to ensure that stakeholder privacy is considered at all times.

As can be seen from the figure below, much data is likely to be collected by an IoT system that can be associated with individual consumers.

Sensors on Shelves

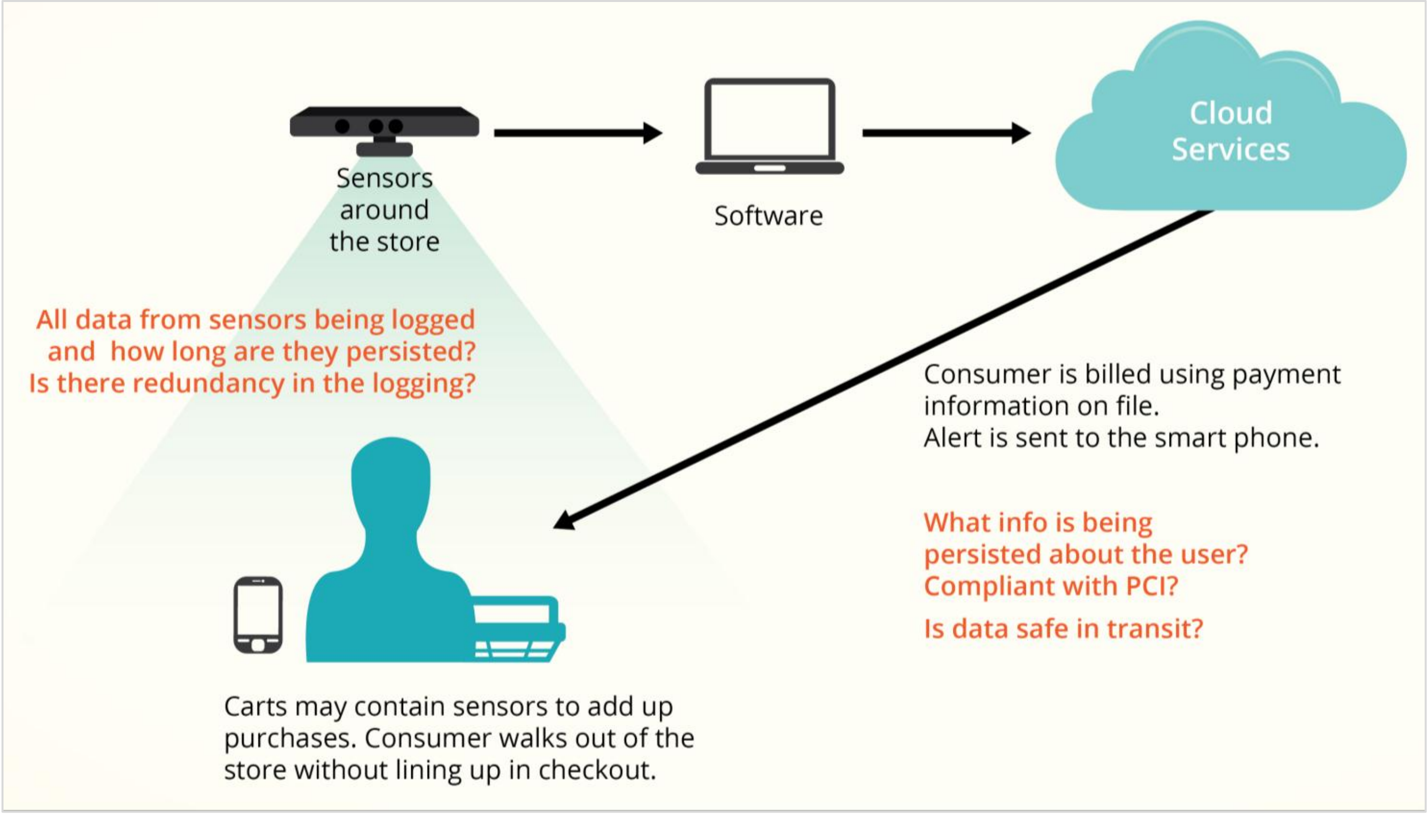


Is there any security-relevant or private data sent? Or just purely about shelf and product activity,nothing about the consumer?

Sensor keeps track of user gestures and behavior patterns that retailer can analyze

Retailer subscribes to data for business intelligence, e.g., which shelves are not performing for quick adjustments to shelving

Automated Checkout



All data from sensors being logged and how long are they persisted? Is there redundancy in the logging?

Consumer is billed using payment information on file.
Alert is sent to the smart phone.

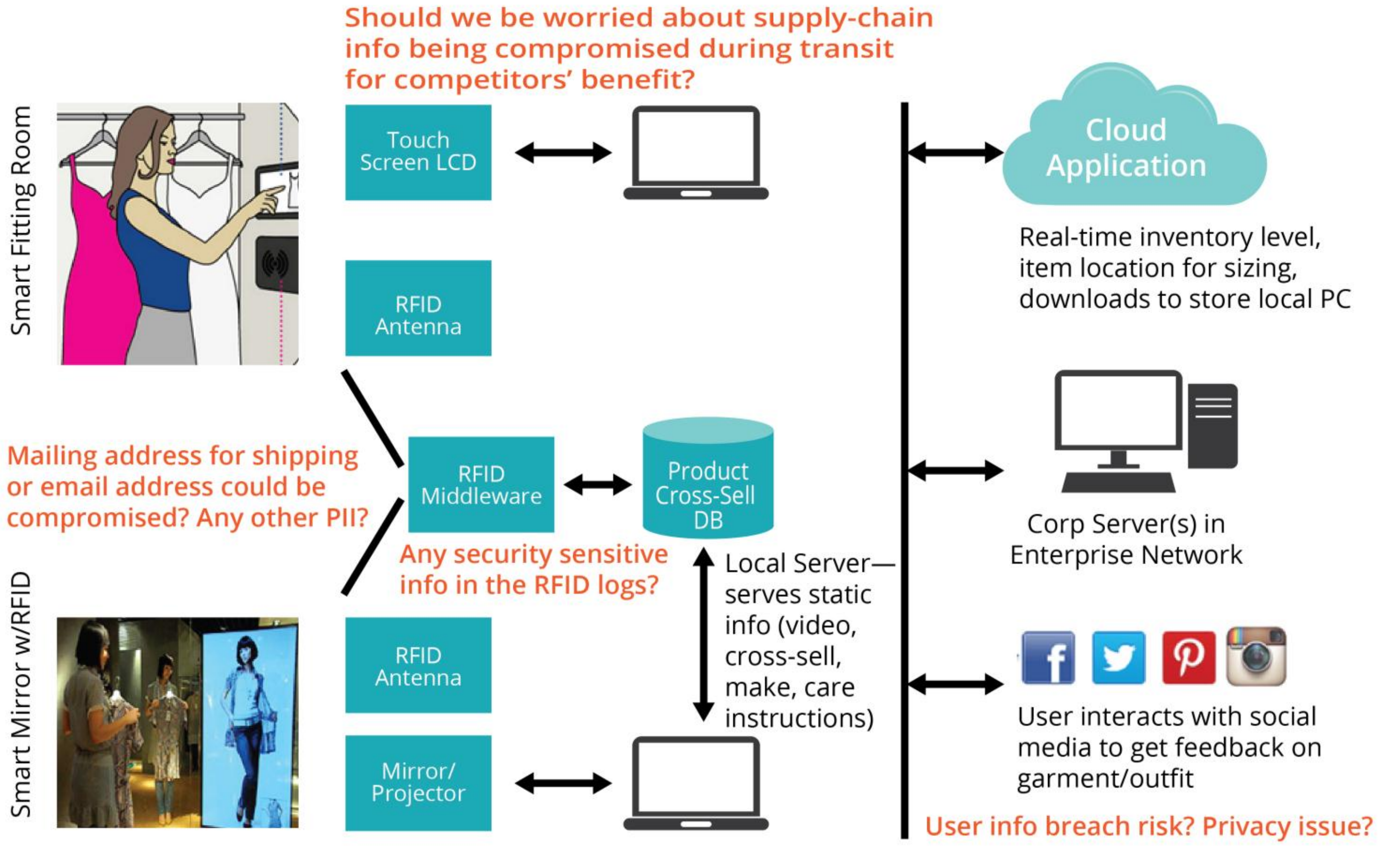
What info is being persisted about the user? Compliant with PCI?

Is data safe in transit?

Carts may contain sensors to add up purchases. Consumer walks out of the store without lining up in checkout.

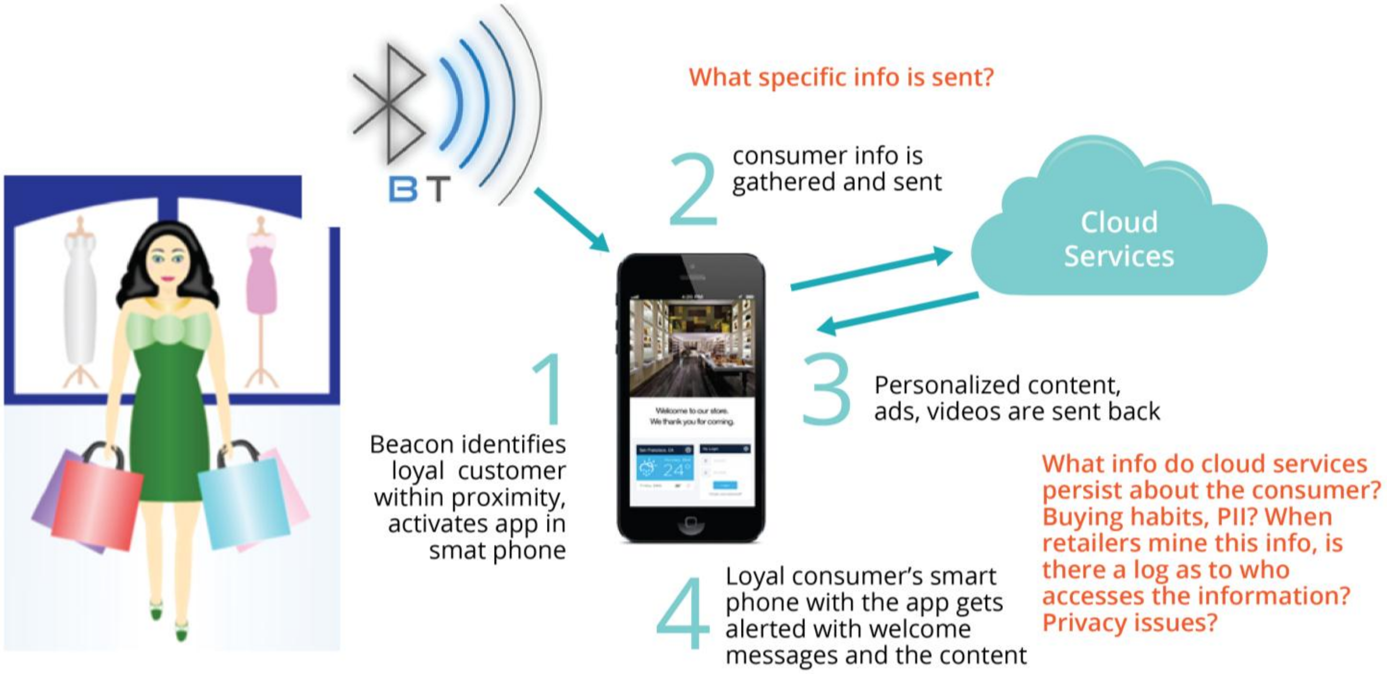
It must be considered exactly what data persists about each user, and the impact that it stands to have on compliance and privacy regulations. The same applies to compliance with industry standards such as PCI, which mandates that PII be encrypted both at rest and in transit.

Smart Fitting Room/Smart Mirror


Should we be worried about supply-chain info being compromised during transit for competitors' benefit?
Smart Fitting Room
Touch Screen LCD
RFID Antenna
Cloud Application
Real-time inventory level, item location for sizing, downloads to store local PC
Mailing address for shipping or email address could be compromised? Any other PII?
RFID Middleware
Product Cross-Sell DB
Corp Server(s) in Enterprise Network
Any security sensitive info in the RFID logs?
Local Server—serves static info (video, cross-sell, make, care instructions)
Smart Mirror w/RFID
RFID Antenna
Mirror/ Projector
User interacts with social media to get feedback on garment/outfit
User info breach risk? Privacy issue?

In addition to verifying that all sensitive information is protected sufficiently, it is also important to consider risks related to the supply chain. If components that make up your IoT system are compromised in the supply chain, the risk of exposure of sensitive information is high.

Proximity Advertising


BT
What specific info is sent?
2 consumer info is gathered and sent
Cloud Services
1 Beacon identifies loyal customer within proximity, activates app in smat phone
3 Personalized content, ads, videos are sent back
What info do cloud services persist about the consumer? Buying habits, PII? When retailers mine this info, is there a log as to who accesses the information? Privacy issues?
4 Loyal consumer's smart phone with the app gets alerted with welcome messages and the content

Another consideration is related to who has access to stored privacy data. This data will likely be provided to third parties and access to any sensitive information should be logged for auditing purposes and checked for compliance against policies.

Given the complexity of the IoT privacy landscape, it is important for any organization offering IoT-based capabilities to expend appropriate resources to ensure the safeguarding of stakeholder sensitive information. When architecting an IoT system, following Privacy-by-Design principles will allow for the integration of appropriate privacy safeguards within the system. These principles can be followed while designing the implementation of the various components that make up an IoT System for any particular organization. The European Union (EU) Article 29 Data Protection Working Party released guidance in September 2014 stating that all IoT stakeholders should adopt these principles to implementations within any region of the world. The following sections provide an IoT-specific view of these principles that organizations can use to bolster their privacy programs to support IoT deployments.

5.1.1. Privacy-by-Design Principles

Users of IoT systems should be made aware of all of the data collected from or about them, and should be given the opportunity to opt out of data collection practices at a granular level. Recognizing the concerns that many of the IoT devices may not have proper user interface, companies should find suitable methods to provide the choice and notice to consumers.

5.1.1.1. Proactive not Reactive; Preventive not Remedial

Within the context of an IoT System, it is important to consider the potential privacy ramifications to all stakeholders prior to putting the system into an operational state. At the beginning, analysis will focus on data types collected to understand which are sensitive and what regulations apply to each data type. Next, more in-depth analysis should be undertaken to understand the indirect privacy ramifications of the various IoT component operations. As an example, when dealing with applications that track connected vehicles, it would be important to understand whether the tracking would expose driving patterns that, although anonymized, could be traced back to an individual or group when combined with data collected by other systems. Another case in point regards to the collection of data by smart meters that is fed to the utility companies for analysis. If access to this data is not tightly controlled, attackers can deduce when a person is at home exposing opportunity for physical attacks. Looking at privacy of data-in-aggregate vs. privacy of the data collected by a single system will allow for the identification of potentially serious privacy concerns prior to them being exposed or taken advantage of by unscrupulous persons.

5.1.1.2. Privacy as the default

In January of 2014, the Chairman of the Federal Trade Commission (FTC) noted that IoT stakeholders have a responsibility to "make security a part of their product development process, to collect the minimum amount of data necessary, and to notify consumers of unexpected use of their data and provide simplified choices regarding this use."* Organizations that deploy IoT capabilities should take note of this, and ensure that they have built in privacy controls into their systems, on top of the device or application-specific privacy controls provided by any IoT vendor.

5.1.2. Privacy Embedded into Design

Organizations implementing IoT functionality will be faced with first understanding the true privacy concerns of their stakeholders. As such, conducting an analysis to determine the data elements that an IoT system will process is critical. This should ideally be conducted in conjunction with the recommended threat analysis, and early on in the design of the IOT System.

Once a thorough understanding of the potential indirect effects of data collection has been gained, the appropriate safeguards can be designed into the IoT System from the beginning, versus after a privacy concern has been raised or exploited. Also, companies should reevaluate their personal data breach notification program to cover the aspects related to IoT.

5.1.3. Full Functionality — Positive Sum, not Zero-Sum

There is typically a balance between the objectives of functionality and security that must be maintained to ensure that any particular system works correctly, meets business objectives, and is still secure. The same can be said of privacy. In the case of the IoT, it is critically important that trade-offs between functionality, security and privacy be made early on in the design process in order to ensure that all objectives are met equally. Identifying a privacy issue well into the operational life of an IoT system will make the process of retrofitting privacy controls challenging. Adhere to these principles of Privacy-by-Design to identify and implement those trade-offs when the cost of doing so is relatively minor during design of the IoT system.

5.1.4. End-to-End Security — Lifecycle Protection

Within the IoT, data collected will have a long lifespan. It is important to consider the full lifespan of the data collected, both within the collecting organization and within any third parties to which it is provided. Stakeholders should be made aware of when data is provided to third parties, the controls used to secure it, and how and when the data is disposed of.

Lifecycle protection also applies to second-order data (information about people that is inferred or determined based on primary data) as well. For instance, if a sensor in your car collects how far, where, how fast, and other attributes of your driving habits, then someone can infer various things about you, for example, your shopping or working habits, or who you socialize or interact with. The owner of the data (e.g., the car company) may erase your primary data upon sale of your vehicle, but in fact keep all the inferred information (social connection, shopping habits, etc.).

5.1.5. Visibility and Transparency

Stakeholders should be able to easily identify the data collected from them for any particular IoT system, as well as the planned or potential uses for that data. Stakeholders should also be allowed to opt in to data collection, at both a coarse and granular level. As an example, if an application tracks their driving patterns (e.g., for insurance purposes), the user should be able to explicitly authorize the use of their data for that purpose (coarse). The user should also be able to explicitly authorize individual data elements if so desired, for example the storage of driving patterns or history obtained through GPS.

5.1.6. Respect for User Privacy

Maintaining the privacy of stakeholder information will eventually become a discriminating factor for companies in the era of the IoT. With so many opportunities to mishandle user privacy, the organizations that take the necessary steps to safeguard sensitive information will be viewed far more favorably than the ones that do not. Given this, it is important to instill a culture of privacy awareness within the organization. This could include appointing one or more privacy advocates to evaluate the privacy impacts of any new IoT system being implemented. These people would ideally be given the authority to mandate changes to IoT system designs in the event that privacy concerns are identified.

User privacy is also concerning from an indirect perspective. In the case of some IoT devices, for example smart glasses, the user has consented to privacy clauses, but the observed party most likely has not. Further research must be conducted to understand the impacts and regulations required around these type of scenarios.

5.1.7. Privacy Impact Assessment

The EU WP29 guidance also points to a recommended framework for conducting a Privacy Impact Assessment (PIA)*[EU].

If it is found that a device collects, processes or stores Privacy Protected Information (PPI), more stringent controls will be required. These controls should be a mix of policy-based and technical. For example:

- Provisioning of the device may require more administrative approvals
- A review by Internal Audit or Compliance should be conducted to determine if it is viable to have PPI data on IoT devices
- Data stored on the device should be encrypted using sufficiently strong cryptographic algorithms
- Data transmitted from/to the device should be encrypted using sufficiently strong cryptographic algorithms
- Access to the device, both physical and logical, should be restricted to authorized personnel

There are various recommendations on privacy requirements that should be considered based on region, including:

- North America
  - Internet of Things, Privacy and Security in a Connected World, Federal Trade Commission (FTC) Staff Report
- Europe
  - Privacy Recommendations for the IoT, WP29 of the EU (European data protection advisory body)

5.2. Apply a Secure Systems Engineering approach to architecting and deploying a new IoT System.

Although some IoT functionality may simply consist of sensors feeding data into an analytics engine, it is likely that most IoT value-added capabilities will be the result of a number of diverse components working together with data traversing many networks. As these systems are architected, it is important to define and inject security requirements into the designs to account for the implementation of security functionality prior to deployment. A standard practice for performing this activity, which can be adopted from the design of more traditional systems, is threat modeling.

5.2.1. Threat Modeling

A reference for threat modeling can be found in Adam Shostack's book "Threat Modeling: Designing for Security." Microsoft also defines a well-thought-out threat modeling approach using multiple steps to determine the severity of threats introduced by a new system. The threat Modeling approach (based on Microsoft SDL):

5.2.1.1. Step 1: Identify Assets

This is for cataloguing the various components of the IoT System that will be deployed. Consider not only the IoT devices but also the data stores and applications that the devices communicate and the users that interact with the system.

5.2.1.2. Step 2: Create a System/ Architecture Overview

This step provides a solid foundation for understanding not only the expected functionality of the IoT System, but also how an attacker could misuse the system. Begin with the process of documenting expected functionality and then spend time to consider and document misuse cases for the system. It is also important to create an architectural diagram that details the new IoT System and how the system interfaces with other enterprise computing resources and security systems. This diagram can also serve as the starting point for identifying trust boundaries, authentication and authorization mechanisms as well as logging conops.

The creation of system architecture is aided through use case analysis. The following example use cases from the healthcare sector can provide insight into security considerations for IoT implementations.

1. A person wears some type of monitor that reports through the cloud to his/her physician
   a. Under extreme circumstances, would first responders be automatically dispatched?
   b. Would a new pharmacy prescription be automatically generated (by some rule), or alternatively would the prescription information be routed to several pharmacies that would compete for the purchase?
   c. Would an appointment be auto-scheduled?
   d. Would health records be updated?
   e. If medical response is dispatched is data transferred to an ambulance?

2. An implanted device receives a command
   a. Does the device use PKI? If so, can the device confirm revocation status of the sender?
   b. Can the device validate the message?
   c. Can the device create a secure link or session with the sender?
   d. Can the device request confirmation?

3. A physician establishes a communication session with a smart home/home monitor
   a. Is the communication channel secured with PKI?
   b. Are PII and medical data transferred securely?
   c. Does the physician issue commands to devices? If so, is there integrity checking and nonrepudiation through logging?

4. A hospital transfers a patient’s record or diagnosis to a computer or PDA
   a. Can the patient interact with hospital services, such as scheduling another appointment?
   b. Can the patient confirm the authenticity of the message?
   c. Can the patient effectively remove the message?

5. A patient’s blood donation is handled by an online analyzer
   a. Is the tracking number for the donor protected locally or centrally?
   b. Will the patient be notified directly of any finding?
   c. If the patient has an STD which agencies will be notified?
   d. What are the trust mechanisms?
   e. Will the blood packet be handled by a robot?
   f. Will the patient’s pharmacy or doctor be messaged on any particular finding?
   g. Will a maintenance center be messaged about the state of the analyzer?

7. In an emergency, multiple first responders are dispatched
   a. Is medical data transferred securely to the correct ambulance?
   b. Can responders communicate patient data securely? Is it through point-to-point or central routing?
   c. Is security, trust and privacy managed by multiple trust chains?

8. A pharmaceutical company issues an alert regarding drug infusion pumps
   a. Is the pharmaceutical company's message trusted by pharmacies?
   b. Does the alert impact a patient's dispensing device?
   c. Does a doctor issue controls to the dispensing device?
   d. Does the infusion pump have closed loop communications to the controller/monitor?

9. A doctor performs telesurgery using a robot
   a. Is the communication channel trusted and secure?
   b. Is the robot's distinguished name trusted with the console?
   c. Does the communication depend on DNS?
   d. What is the strength of the algorithms and key lengths use by the IP VPN?
   e. What is the trust chain and CRL management for the entire topology?
   f. Are backup communications channels trusted at the same level as the primary?
   g. Are pharmaceutical providers and records keeping updated in real time?

10. A government agency issues a health alert that affects implanted devices
    a. In what order are stakeholders notified? (doctors, pharmacies, manufacturers, system administrators, etc.)
    b. Is the message authenticated and verified?
    c. If a device is recalled, what databases need to be updated?
    d. Is the inventory managed to ensure that all devices are properly administered?

11. An implanted or wearable device needs updating
    a. Is the update remotely managed?
    b. Is there two-way trust between the device and the central server?
    c. Is the channel secure and trusted?
    d. Is the inventory managed to ensure that all devices are properly managed?
    e. Are stakeholders notified if procedures or instructions change?
    f. Is the pharmacy notified if drugs are involved?

12. The controlling physician for a specific device is replaced by another physician
    a. Are credentials managed centrally or locally?
    b. Is there a two-way trust between the physician and the device?
    c. Can the device be updated remotely to assign a new trust?

13. A manufacturer alters its instructions for a remotely controlled medical device
    a. Is configuration management properly maintained, so that stakeholders know the version of devices/instructions?
    b. Are medical universities included as part of the stakeholders?
    c. Is there an authoritative database for configuration management?

14. In a connected vehicle environment an ambulance/first responder vehicle coordinates patient records with a medical provider
    a. Are the communications protected with PKI?
    b. Is there two-way trust between the ambulance and the medical provider?
    c. Are patient records purged after the patient has been dispatched?
    d. Is on-board equipment remotely managed?
15. A patient with an implanted device dials 911
    a. Is the patient data made available to the dispatcher?
    b. Can the dispatcher route data to a remote provider or doctor?
    c. Is a two-way trust relationship established?
    d. Are patient records automatically updated?
    e. Can information be securely communicated with an ambulance?

16. A private cloud is deployed in South America to serve remote medical communities
    a. Is infrastructure auditable to verify that security standards are met?
    b. Does the system support remotely connected devices?
    c. Is there two-way trust with the remote clients?
    d. How are stakeholder identities authenticated?

17. Nanobiomedical devices are remotely deployed
    a. Are two-way trust relationships established with the central facility?
    b. Is each component in the topology trusted?
    c. Are recovered modules properly protected from sensitive medical information? (physical security)
    d. Is the inventory tracked securely?

Once the logical architecture view is complete, in is important to identify and examine the specific technologies that will make up the IoT System. This includes understanding and documenting lower level details regarding the IoT devices, such as the processor type and operating system. This will provide information needed to understand the specific types of vulnerabilities that may eventually be exposed and define processes for how and how often patches and firmware updates should be applied. Understanding and documenting the protocols that are used by each IoT device will also allow for updates to the architecture, especially if gaps are found in the encryption applied to the data transmitted throughout the system and the organization.

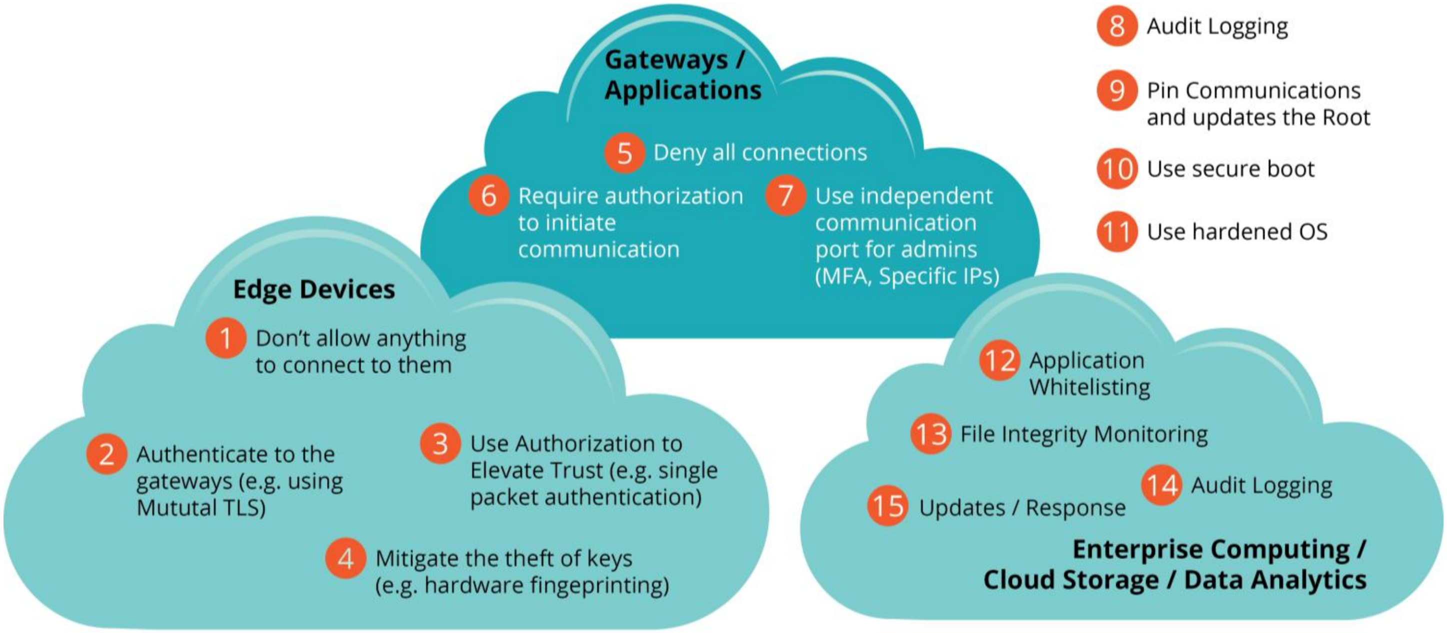
5.2.1.3. Step 3: Decompose the IoT System

At this stage, the focus is on understanding the life cycle of data as it flows through the system. Developing this understanding will allow for the identification of vulnerable or weak points within the security architecture that must be addressed. Identify and document the entry points for data within the system. In an IoT system, these entry points are typically sensors of some type. Trace the flow of data from the entry points and document the various components that interact with that data throughout the system. Identify high profile targets for attackers — these may be points within the system that aggregate or store data, or it may be high value sensors that require significant protections to maintain the overall integrity of the system. At the end of this activity a good understanding of the attack surface of the new IoT system will be had.

5.2.1.3.1. Step 3a: Define a Protective Architecture

Once you decompose the IoT system, shouldn't the next step be to design an architecture to protect the system? Why not give them a notional protective architecture? This could be where some of the elements of the SdP can be introduced. Based upon Junaid's comments I have included a notional diagram that we can adapt for the IoT environment. What do you think?

Elements of a Protection Architecture for IoT


Gateways / Applications
5 Deny all connections
6 Require authorization to initiate communication
7 Use independent communication port for admins (MFA, Specific IPs)
8 Audit Logging
9 Pin Communications and updates the Root
10 Use secure boot
11 Use hardened OS
Edge Devices
1 Don't allow anything to connect to them
2 Authenticate to the gateways (e.g. using Mututal TLS)
3 Use Authorization to Elevate Trust (e.g. single packet authentication)
4 Mitigate the theft of keys (e.g. hardware fingeprinting)
12 Application Whitelisting
13 File Integrity Monitoring
14 Audit Logging
15 Updates / Response
Enterprise Computing / Cloud Storage / Data Analytics

After giving them a notional architecture for protecting the IoT environment, then list the threats and provide detailed guidance as shown below.

5.2.1.4. Step 4: Identify and Document the Threats

The popular STRIDE* model can be applied to IoT System deployments. Use well known vulnerability repositories to better understand the environment, such as MITRE's Common Vulnerabilities and Exposures* database. Uncovering the unique threats to any particular IoT instantiation will be guided by these threat types:

Threat Type	IoT Description
Spoofing Identity	Examine the system for threats related to the spoofing of machine identity and the ability for an attacker to overcome automated trust relationships between devices. Carefully examine the authentication protocols employed to set up secure communications between various devices (M2M) and between devices and applications that make use of data provided by these devices. Examine the process for provisioning of identities to each IoT device and ensure that there are proper procedural controls in place to limit the ability to introduce a rogue device into the system.
Tampering with Data	Examine the path of data across the entire IoT system. Identify points in the system that provide an opportunity to tamper with the data at points of collection, processing, transport and storage. Carefully examine implementation of authorization mechanisms to ensure that data tampering is effectively dealt with.
Repudiation	Examine the IoT system design for nodes within the system that are critical data providers. These are likely sets of sensors that provide various data for analysis. In the case of the IoT, it is important to be able to trace back data to a source and ensure that it was indeed the expected source that provided that data. Examine the IoT system for weaknesses that would allow an attacker to inject a rogue node that would feed bad data into the system in an attempt to confuse upstream processes or take the system out of an operational state. Ensure that attackers are not able to abuse the intended functionality of IoT systems e.g. illegal operations are disabled or not allowed. State changes and time variations (e.g. disrupting message sequencing) should be taken into account.
Information Disclosure	Examine the path of data across the entire IoT system, including the backend processing systems. Ensure that any device that processes sensitive information has been identified and that proper encryption controls have been implemented to guard against disclosure of that information. Identify data storage nodes within the IoT system and ensure that data-at-rest encryption controls have been applied. Examine the IoT system for instances where IoT devices are vulnerable to being physically stolen and ensure that proper controls, such as key zeroization have been considered.
Denial of Service	Perform an activity that maps each IoT system to business goals, in an effort to ensure that appropriate Continuity of Operations (COOP) planning has occurred. Examine the throughput provided for each node in the system and ensure that it is sufficient to withstand relevant DoS attacks. Examine the messaging structures (e.g., data busses), data structures, improper use of variables and APIs used within applicable IoT components and determine if there are vulnerabilities that would allow a rogue node to drown out the transmissions of a legitimate node. Implementers of the IoT should also consider rate limiting APIs to mitigate DoS attacks.
Elevation of Privilege	Examine the administration capabilities provided by the various IoT devices that make up an IoT system. In some cases, there is only one level of authentication, which allows for configuration of device details. In other cases, distinct administrator accounts may be available. Identify instances where there are weaknesses in the ability to segregate administrative functions from user-level functions within IoT nodes. Identify weaknesses in the authentication methods employed by IoT nodes in order to design appropriate authentication controls into the system.
Bypassing Physical Security	Examine the physical protection mechanisms offered by each IoT device and plan mitigations where possible against any identified weaknesses. This is especially true for IoT deployments that are placed in public or remote areas.

Threat Type	IoT Description
Social Engineering Intrusions	Train staff to guard against social engineering attempts and regularly monitor assets for suspicious behavior.
Supply Chain Errors	Understand the various technological components that make up IoT devices and systems and keep track of vulnerabilities related to any of these technology layers.
Network Intrusions	Regularly monitor networks for suspicious behavior.

5.2.1.5. Step 5: Rate the Threats

Evaluating the likelihood and impact of each threat identified in the previous step allows for the allocation of appropriate levels of investment to mitigate each threat. Threats with a higher risk rating will likely command larger amounts of money to mitigate as they are likely the threats needing immediate mitigation. Any standard threat rating methodology can be used at this step, including the DREAD* approach from Microsoft.

5.2.2. Secure Development

IoT edge devices are combinations of hardware, operating systems, firmware and software. This means that as new devices are created vendors must be aware of the security vulnerabilities exposed at all layers of the technology stack. This includes things like hardening the underlying operating system (when applicable), and mitigating hardware-specific vulnerabilities in the platform. IoT devices at the edge also interface with many other devices and systems, creating in effect a system-of-systems. Some edge devices have very limited code built on top of those various frameworks, operating systems and platforms. More complex IoT edge devices do exist however, and these “things” require many of the same secure software development practices as traditional enterprise or mobile applications. This includes analyzing code for vulnerabilities, through the use of static and dynamic code analysis tools, as well as performing penetration tests against software to determine vulnerabilities that must be mitigated.

Organizations such as the Open Web Application Security Project (OWASP) are working to provide secure development guidance for IoT device makers. The OWASP IoT Top 10 identifies security issues that should be mitigated when developing IoT devices. These include items such as:

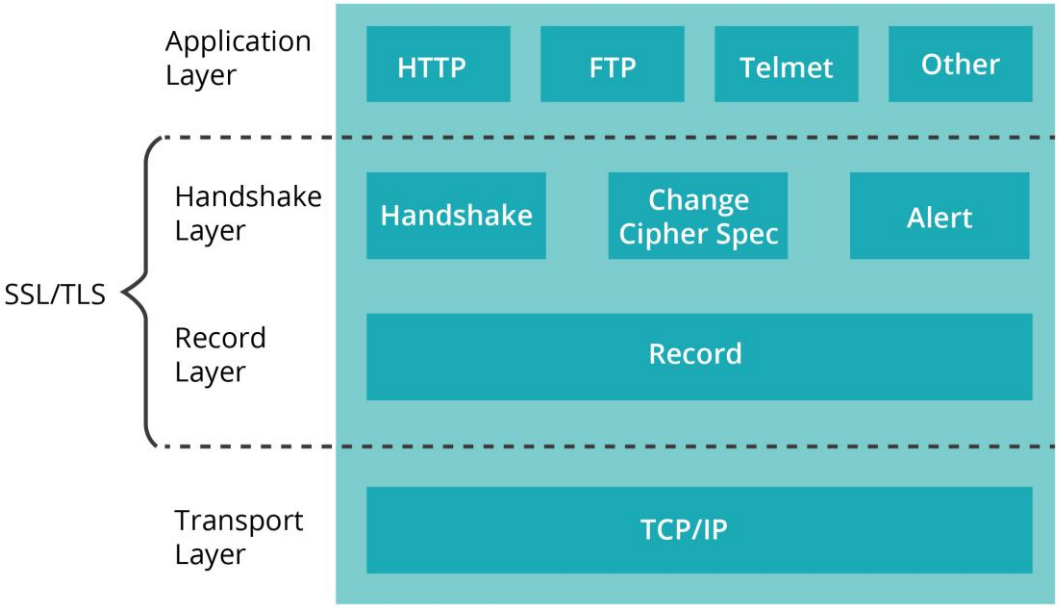
- Insecure cloud and mobile APIs,
- Lack of transport encryption, and
- Insufficient authentication and authorization

It will be important for IoT vendors to build up security engineering expertise quickly in order to ensure that they can design secure IoT solutions. Building up this expertise internally is another challenge as the cost to train staff that understand the security discipline and identify vulnerabilities is very high. Organizations such as Builditsecure.ly and I Am the Cavalry can provide guidance towards that goal, however another interesting approach for IoT developers is through the use of independent, crowd-sourced bug hunters. Sites such as BugCrowd allow developers to have independent security analysts perform code reviews and even in some cases review hardware implementations, then submit vulnerability findings.

Organizations working towards establishing secure development methodologies for the IoT should consider defining a framework for secure development. The framework should detail secure coding practices for embedded solutions, parameter manipulations and APIs. Leverage best practices for API security from REST and SOAP/XML to establish these secure coding practices for the IoT. Also detail the security risks associated with various protocols that may be used for communication and the access levels that should be provisioned to various users of IoT devices.

A key factor in understanding the proper security controls to apply to an IoT implementation is the set of data-link and transport protocols employed. Each of the various protocols available for IoT use support varying levels of security, and some can be matched together to provide an optimally secure configuration. As designers of an IoT System, organizations will be faced with understanding these protocols and how the use of one impacts the need to employ higher level protocols such as the Transport Layer Security (TLS) protocol for confidentiality and authentication. Appendix A provides a view into the various IoT protocols available for use.

SSL, TLS Protocol Layers



Another consideration is that the tools used to implement the communication and security protocols themselves must be scrutinized. For example, the Heartbleed bug that was exposed in April 2014 affected only the OpenSSL part of TLS. SSL is an encryption protocol used to encrypt traffic between a website and its user. Due to improper input validation in OpenSSL, the Heartbleed bug allowed hackers to extract highly personal information about web users from servers using OpenSSL. Other protocol implementations of TLS were not affected. (E.g., Microsoft, Mozilla and GnuTLS were not affected). Even today, many vulnerable OpenSSL devices still have not been patched: An analysis by John Matherly, the creator of the scanning tool Shodan, found that 300,000 machines remain unpatched as of December 2014. Many of them are “embedded devices” like webcams, printers, storage servers, routers and firewalls.

5.2.2.1. Secure Supply Chain

Different layers of the IoT technology stack often represent different aspects of the supply chain — hardware, firmware, operating systems, protocols, cloud providers. The interaction of all of these layers can oftentimes result in unique vulnerabilities only present in the single integrator's product. Manufacturers deploying devices should incorporate language in a service agreement that requires ODM's (Original Design Manufacturers) to remove debugging interfaces and non-essential applications before the manufacturer handoff. If vendors in the supply chain provide positive and negative tests for their contributions in the product, then each higher layer developer could use those tests to validate that the underlying foundation that is being used for their IoT device is secure in the context of the final application being developed. These tests could be containerized for easy download and executing through autonomous testing tools during iterative integration testing. As an example, if an IoT developer is relying on an external TLS implementation, that TLS library should come with a full suite of conformance tests that can be run autonomously.

Organizations procuring IoT assets should also be keenly aware of the supply chain's ability and plan to provide updated firmware, software and patches. Ideally, this will be worked into a license agreement with the IoT device vendor.

5.3. Implement layered security protections to defend IoT assets

The IoT converges an ecosystem's existing information technology (IT) and operational technology (OT) networks with millions of sensors, devices, and other smart objects. This convergence significantly expands security challenges, due to its increased breadth and depth over existing network connectivity.

IT and OT networks are managed with different priorities in mind, and each has distinct security needs. The priority of the IT network is to protect data confidentiality. The focus of the OT network is on physical security and secure access to ensure operational and employee safety.

With the convergence of these two environments, IoT security requires a new approach that combines physical and cybersecurity components. The result is improved employee safety and protection of the entire system from the outside, as well as the inside.

Organizations operating in the digital world today need layers of security so that an email message that gets through the firewall will get stopped by the mail server's antivirus; and if it makes it through that, then it should be stopped by the workstation's antivirus. If the hostile program actually secures a toehold on the workstation, it should be detected when it runs on the workstation because it's doing things that are suspicious or unexpected. Look for connections to sites on the Internet with known relationships to hostile activity, and block such sites by egress filtering on the firewall.

Attackers are leaving no stone unturned, prying into web applications, operating systems and even deeper in the hardware. They're taking advantage of conventional endpoints and mobile devices, slipping past and through network security, and even taking advantage of the human element operating the devices.

At the design phase, serious consideration must be given to Threat Modelling of the IoT architecture which must take into consideration the actors/roles, components being used, data entry and exit points at all layers mentioned below including the device layer. Various threat scenarios must be thought through with numerous misuse cases and then handed over to the development/build team to develop/build using security best practices followed by security testing.

From the IoT consumer perspective, thorough planning needs to be made before implementing an IoT initiative. Security needs to be carefully thought out at all these layers, as one or two secure layers are not enough to ensure a fully-secure implementation:

5.3.1. Network Layer

- Firewalls are designed to filter traffic based on type, port and destination. Firewalls have evolved by incorporating deeper analytics, such as IPS and traffic inspection services, enabling them to look deeper into packets and better detect malicious traffic. Such devices are one of the easiest starting points when implementing a layered defense.
- Frequently scan for open ports in firewalls and routers. Open ports are an invitation to hackers.
- Check if routers are vulnerable to misconfigured NAT-Port Mapping Protocol (NAT-PMP) services. NAT-PMP is a protocol that has no built-in authentication mechanism and trusts all hosts belonging to the router's local network, thereby allowing them to freely "punch" holes through the firewall. Misconfigured routers to NAT-PMP services are mentioned in OWASP's Top 10 Threats for Internet of Things*.
- Exercise Network Access Control (NAC) to unify endpoint security technology such as antivirus and host intrusion prevention. Antivirus products protect computers from malware, for example, by relying on comparisons to file signatures.
- Perform vulnerability assessments periodically and make sure that both user and system authentication to the network comply with your organization's security policies. This includes strong password policies, password management and periodic change of passwords.
- Disable guest and default passwords in network devices such as routers and gateways. This should be done immediately upon unpacking a new network device, before putting them to work in your network.
- It is good practice to document all MAC addresses for every device so that the router assigns IP addresses only to these devices. All unknown devices will be blocked from accessing the network.
- For wireless networks, use Wireless Protected Access 2 (WPA2) instead of Wireless Encryption Protocol (WEP). WPA2 requires using stronger wireless encryption. Always use a strong complex password to your wireless network.
- Also for wireless networks, use multiple Service Set Identifiers (SSID), rather than just one. This allows the network manager to assign different policies and functions for each SSID, thereby allowing the organization to assign devices into different SSIDs based on risk and criticality. Segmenting your wireless network this way ensures that if one device gets hacked, other devices will not be compromised as they are in a different segment.
- Use Private Pre-Shared Key (PPSK) to ensure that each sensor or device is securely connected to the Wi-Fi. Administrators can assign unique and revocable keys to each user and client on the network. These keys define what permissions should be assigned to a device connecting with that key. There are technology companies that provide this capability.
- Increasingly, IoT devices are storing their data in the cloud for analysis. It is important to secure this data appropriately through encryption and other means. (E.g. transmitting sensitive data such as healthcare data from a patient monitoring device to cloud storage.)

5.3.2. Application Layer

The IoT does not require a completely new set of application security guidelines and best practices. The same set of guidelines at the application layer hold true for any traditional implementation.

- If your organization is writing your own applications, use appropriate authentication and authorization mechanisms. Scan for any passwords left in the clear in the application code (e.g. hardcoded telnet logins or passwords that were left behind during testing).
- If the organization is using any third party or open source libraries, then it is recommended to maintain an inventory of those libraries and keep them updated. Also, check the version and the corresponding vulnerabilities in those versions so that you can avoid using those vulnerable versions. This will ensure that security patches can be applied to the third party or open source libraries used.
- Check for any cross-site scripting (XSS) or Cross Site Request Forgery (CSRF) vulnerabilities. CSRF is a type of attack by a malicious web site, email, blog, instant message or program that causes a browser to perform an unwanted action on a trusted site. XSS enables attackers to inject client-side script into web pages viewed by other users, or may be used to bypass access controls. OWASP recommends scanning tools such as Zed Attack Proxy (ZAP) or Dynamic Application Security Testing (DAST).
- Ask for the security code review report from the vendor for any vulnerabilities that were uncovered during the development of the IoT platform and their corresponding remediation. This step will act as due diligence from a Static Application Security Testing (SAST) perspective. If the consumer is developing an application that will be hosted on top of the IoT platform, SAST must be performed on the application along with Dynamic Application Security Testing (DAST).
- Applications can also be managed and hosted, or provided as a service by another organization. Train users to change their default passwords for the service.
- Insecure cloud interfaces are cited in OWASP Top 10 for IoT. Make sure https is used and enforce lockouts when the maximum number of allowed authentication retries or idle time is reached.
- Use encryption for data at rest. Ensure privacy of data during transport by using strong encryption. Add salt or random data to hashed data to make it harder to hack.
- The encryption of data during transport must be able to take into consideration the resource constrained devices and hence must have a small footprint be lightweight instead of the traditional ones to avoid performance bottlenecks.
- Baseline “normal” behavior so that abnormal behavior can later be detected. The source of the traffic baseline can be firewalls, routers, switches, flow collectors and network taps. Because firewalls and routers pass traffic through them, they are an ideal place to start. What’s typically most interesting, from a security perspective, is the flow between an internal host and one on the Internet.
- One unique challenge of IoT device web applications are that they tend to use non-standard ports instead of the usual 80 or 443. Devices are made to listen on other ports. It is best to use a standard port scanner or shudder to discover what web services a particular device offers. Scan non-standard ports on IoT devices, since many do not use standard ones.

Beyond optional tamper mechanisms, physical IoT device interfaces may require additional protections. JTAG and unneeded serial and other manufacturer interfaces should be removed or tamper-covered before mass deployment. Private or secret keys should be stored in a “secure element” chip that runs in non-volatile memory and limits access to only authorized users.

5.3.3. Device Level

Some examples of devices are sensors, gateways that aggregate data, mobile devices, cameras, RFID readers, wearables and implantable devices. Depending on the industry, there may be devices that are not common across other industries, and therefore this list may lack specific guidelines for unique devices.

Ensure that the device's firmware gets upgrades, updates, and patches regularly.

- Take care regarding the sources of the update files and how they were transported. Make sure you scan the files or check for its integrity prior to installing them into your device. Check the "reputation" of a file, which can be done in a number of ways. Every computer file has a unique checksum—a relatively short mathematical value for the file. Another reputational characteristic of a file is how widely it has been used. Such assessments create a context for the file, indicating whether it is known to be good or bad or whether it is an unknown risk that should be monitored closely.
- Change the default pairing passwords for Bluetooth devices.
- Change the default password and implement a strong password policy.
- Harden the device by changing its default configuration, not just the password.
- Test before deploying devices. "Fuzzing" sends a device unexpected input data and tests how it reacts to detect possible defects.
- For mobile devices, fingerprint access is stronger. Implement lockouts based on idle time and maximum attempts to authenticate.
- Devices and sensors must be tested periodically to ensure proper functionality.
- Limit the data that is being collected or aggregated by a gateway to what is really necessary.
- In the medical field, wearables such as pacemakers and implantable devices are vulnerable to attacks that range from harmless eavesdropping to fatal hacking. Attackers can send unauthorized radio commands to reprogram the device, or send a denial of service attack to drain a device's battery. To protect these devices from fatal breaches, consider implementing the anti-jamming device to thwart an attacker from establishing an unauthorized wireless link between the device and a remote terminal. These devices are called "wearable shields". Test that authorized people such as doctors can still access the data but others cannot.

5.3.4. Physical Layer

Security at the physical layer has long been a mandatory practice in highly regulated industries, even before the IoT. For example, utility and energy companies are very strict about who gets access to lobbies and doors that lead to mission-critical machines and devices, as one breach could result to a disastrous blackout or hefty fines. Similarly, as devices and sensors are used in an Iot initiative, the same vulnerabilities apply. OWASP has identified poor physical security in the Top 10 IoT vulnerability.

- As with logical systems, a Physical Identity and Access management infrastructure governance should exist. Only authorized people should be allowed access to secure areas such as data centers, labs, and areas where devices are mission-critical. Badges should provide the least possible access.
- Physical keys should be provisioned as carefully as secure tokens.
- Monitoring cameras should be used to keep track of the devices deployed around an area. Cameras should be able to pan left and right to scan an area where devices and sensors are implemented.
- Document where devices are located. If possible, develop a graphical map showing where IoT assets are located within a building.

Given the wide variety of physical sites (indoors, outdoors), locations (site-secured vs. no human protections), enclosures and physical embedment options, physical security controls are a critical element to IoT security. Many devices are small and face significant price pressures. Regardless, tamper resistant enclosures, as well as tamper evidence and tamper response mechanisms should be considered based on level of exposure to different physical threat vectors. Tamper controls can be applied on the enclosure, a sub-section of the embedded device (e.g., daughter board with critical processor/memory components) as well as the cryptographic module (preferably hardware). If the device is immobile in its environment and affixed to a wall, pole or other mount, tamper removal indicators should be considered to alert operators of unauthorized device removal and theft.

5.3.5. Human Layer

The Human Layer may be the most difficult to secure and the greyest area when it comes to mitigating risks. There are so many things that could go wrong in this layer that it is a challenge to be prescriptive in a black and white manner. The following guidelines in this layer can be heeded lightly or seriously, depending on the organization's willingness to invest. The most important overall guideline here is to develop a "security-aware" culture and instill awareness, accountability and responsibility in order for the initiative to work:

- Designate a few leaders who are security evangelists. These people should have the personality and the drive to be on the forefront of keeping the IoT initiative going successfully with the least amount of security challenges.
- Constantly train the staff on things to avoid such as falling for too-good-to-be true offers or even unsolicited offers that look like legitimate business requests.
- Similar to downloading patches for devices, users should be trained to verify the reputation of any downloads from the internet.
- Train end-users how they can help in securing their mobile devices, such as the aforementioned practices of lockouts and strong passwords. Turn off Bluetooth in mobile devices when not in use. Immediately report lost mobile devices, and quarantine recovered mobile devices so that they can be scanned for evidence of tampering.
- Reward the staff when they find vulnerabilities.
- Make it easy for end-users, not just employees, to report vulnerabilities. Provide a user interface in a portal to report them. A reward system gives people initiative to report them.
- Document all IoT assets and the kinds of information they collect. Rank them by criticality so that more focus is put on the ones that are more mission-critical. This is taking the Risk-based approach, especially when not all IoT assets can be given the same amount of attention.
- Extend and enforce high standards in contracts with vendors and service providers.

Note that in all the layers above, the discussions have been from the IoT consumer perspective. However, IoT device manufacturers should also be aware of these layers so that they may be able to put themselves in the shoes of the consumer and strengthen their products. Device makers may only be focusing on the device layer itself, but they also need to take responsibility and help the consumer be aware of the other areas where security should be tightened.

In summary, prevention is the best way to prevent alarming incidents. The more these security guidelines at each layer are taken seriously, the more successful the IoT initiative will be.

5.4. Implement data protection best practices to protect sensitive information

Protection of data in its various states requires the application of encryption. There are numerous cryptographic primitives (encryption, integrity, authentication, etc.) available in a variety of cryptographic software/firmware libraries and hardware modules. The National Institute of Standards and Technology (NIST) provides good recommendations for algorithms, modes and key lengths to use for the protection of sensitive information. Algorithms and key sizes should be selected based on the minimum levels of protection established for the IoT cryptographic system.

Two of the primary considerations when choosing the cryptographic suite to use for protecting information, are security level and performance. Performance is especially true when dealing with constrained, typically embedded devices typical in IoT. Elliptic Curve Cryptography (ECC) provides strong algorithms, yet small asymmetric key sizes, for:

- Encryption key establishment (Elliptic Curve Diffie-Hellman — ECDH)
- Digital signatures (Elliptic Curve Digital Signature Algorithm — ECDSA) for message/data signing operations

When paired with a symmetric algorithm such as the Advanced Encryption Standard (AES), this cryptographic suite offers strong cryptographic protections suitable for disadvantaged devices.

Identifying the cryptographic algorithms and key sizes to support within an IoT device is only one aspect of the cryptographic puzzle. These algorithms must be able to operate within a trusted environment and keys must be stored within secure containers. Within larger systems, designers often employ Hardware Security Modules (HSM) for key storage and operations, however HSMs are often not viable for the IoT. Instead designers must explore other options, such as the Trusted Execution Environment (TEE) and Trusted Platform Module (TPM).

Cryptographic implementations used in IoT devices as well as management and data collection systems should undergo cryptographic algorithm validation testing, and possibly cryptographic module conformance testing through validation testing schemes such as NIST's Cryptographic Module Validation Program (CMVP) and Cryptographic Algorithm Validation Program (CAVP).

5.4.1. Data Identification, Classification, Security

Organizations adopting IoT capabilities need to first create an enterprise data security policy that includes approaches for protection of IoT data. This process begins with an explicit task of identifying data elements, associated classifications and other device or application attributes. The data model should not only catalog explicitly-defined information that the IoT devices transmits, receives or stores as part of an application, but also inherently physical world data, which superficially or in isolation, may not appear to be sensitive or private. Physical measurement values, device utilization metrics, etc. also need to be gathered in order to define a data security data protection policy pertaining not only to its applications but also the device's usage patterns. This is a strong pre-requisite to establishing Data Ownership policies and declarations (which pertain to the application data as well as the device's usage patterns), which are needed for the following:

- Data At Rest (DAR) Security
- Data In Transit (DIT) Security
- Data In Use (DIU) Security
- Data Loss Prevention (DLP)
- Data Integrity and Aggregation Policies

5.4.1.1. Data at Rest (DAR) Security

Depending on the complexity of the IoT device, many application-specific data elements may need to be encrypted when not actively used in executable processes. The device should encrypt these parameters using a DAR encryption key securely stored in a physically hardened, locked down cryptographic module resident in the device. In addition to sensitive application data, all secret and private keys, authentication, access control and other security configurations should be stored encrypted if possible. DAR security is designed to protect private information (e.g., medical data) in the event of device theft or loss.

5.4.1.2. Data in Transit (DIT) Security

Data-in-Transit refers to the sending or receiving of data (application, management commands, status, etc.) over a link or network. Whenever possible, DIT protections should include cryptographic confidentiality (encryption), integrity and authentication algorithms executed by a properly integrated cryptographic module. Well-validated network and/or application security protocols should be utilized to provide end-to-end DIT security whenever possible.

Unless symmetric keys are securely pre-placed in IoT devices, device control and data collection systems may need to establish one-time or limited-duration use keys to encrypt data to/from the devices. A fully ephemeral or static-ephemeral Diffie-Hellman exchange (using mutually recognized digital credentials) is useful for this purpose and will provide an encryption key with perfect forward secrecy.

5.4.1.3. Data In Use (DIU) Security

Protection of data in use on IoT edge devices requires a trusted environment for the execution of code. This includes both confidentiality and integrity of data. The Trusted Execution Environment (TEE) provides this capability for use on various processors. ARM-based IoT devices can additionally make use of technologies such as TrustZone for these operations. IoT devices based on other architectures, system-on-chip (SoC) and unique boards may have additional trusted execution logical and physical constructs available. Embedded microcontrollers should make use of security fuses to prevent external manipulation of Flash memory executables and critical data/configuration elements. In addition to micro-hardware security protections, where possible, the use of secured operating systems such as WindRiver are warranted. Many IoT device profiles are shrinking to small but powerful SoC units capable of running a variety of secured-boot Operating Systems featuring strict access controls, trusted execution environments, high security microkernels, kernel separation and other security features. Secure, formally modeled microkernels such as the National ICT Australia (NICTA) seL4 provide a strong foundation to IoT devices being built from the ground up.

5.4.1.4. Data Loss Prevention (DLP)

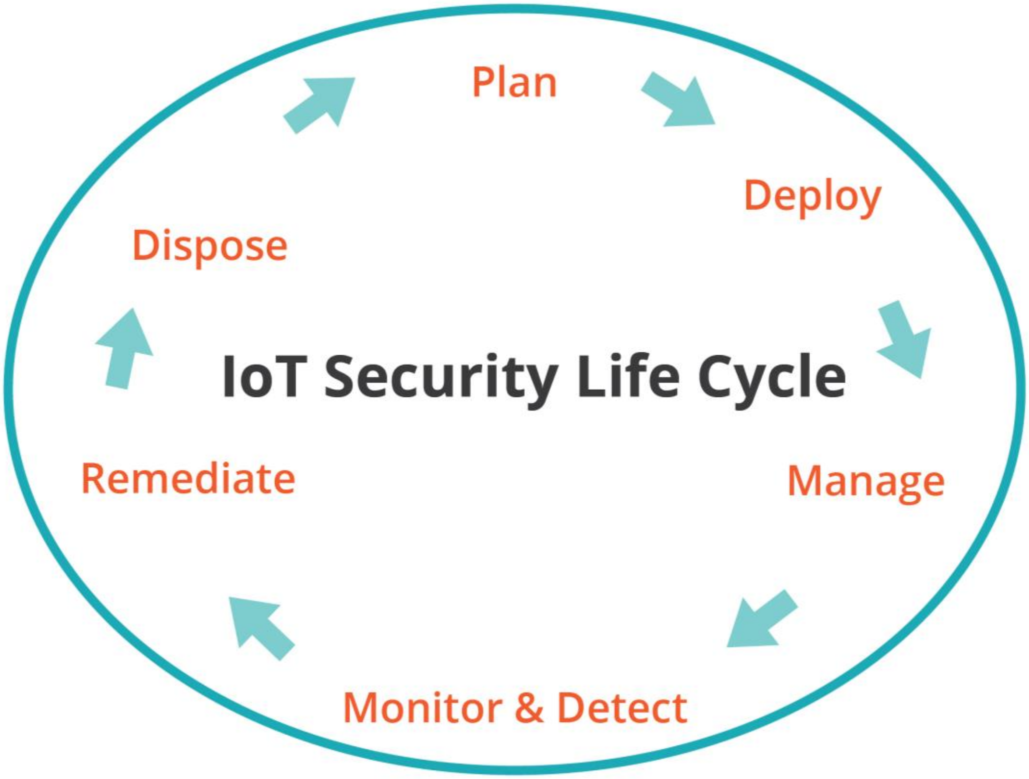
Data loss prevention is critical in the planning and execution of a well-designed IoT deployment. IoT devices used in medical, industrial control, household and other deployment paradigms are expected to collect and transfer vast amounts of information. DLP provides assurance that sensitive data is not distributed outside of the designated user base or network. DLP planning should be conducted early in the deployment, and periodically as new IoT devices are introduced into the corporate network. Data element tagging is a critical prerequisite to proper DLP and enables policy enforcement points, XML guards, one-way diodes and other devices to filter and regulate the onward transfer of sensitive data.

5.4.1.5. Aggregation Protection and Policies

The vast number of IoT devices lends itself to producing exceedingly large data sets useful in various data analytics systems. A critical step in managing IoT security is to ensure that the vast amount of data, in aggregate, does not violate user or system privacy rules. Aggregation policies need to be addressed in the privacy planning process such that appropriate controls are implemented. Do we consider PPI data and segregated PPI and non-aggregation or data scrubbing?

5.5. Define Life Cycle Security Controls for IoT devices

Life cycle controls for IoT edge devices require the management and monitoring of assets to ensure that they are authorized, and secure and regularly updated with the latest firmware, software and patches. In addition, organization's must have a documented method for securely disposing of IoT assets at the end of the life-cycle. Define a life-cycle management approach for IoT devices.



5.5.1. Plan

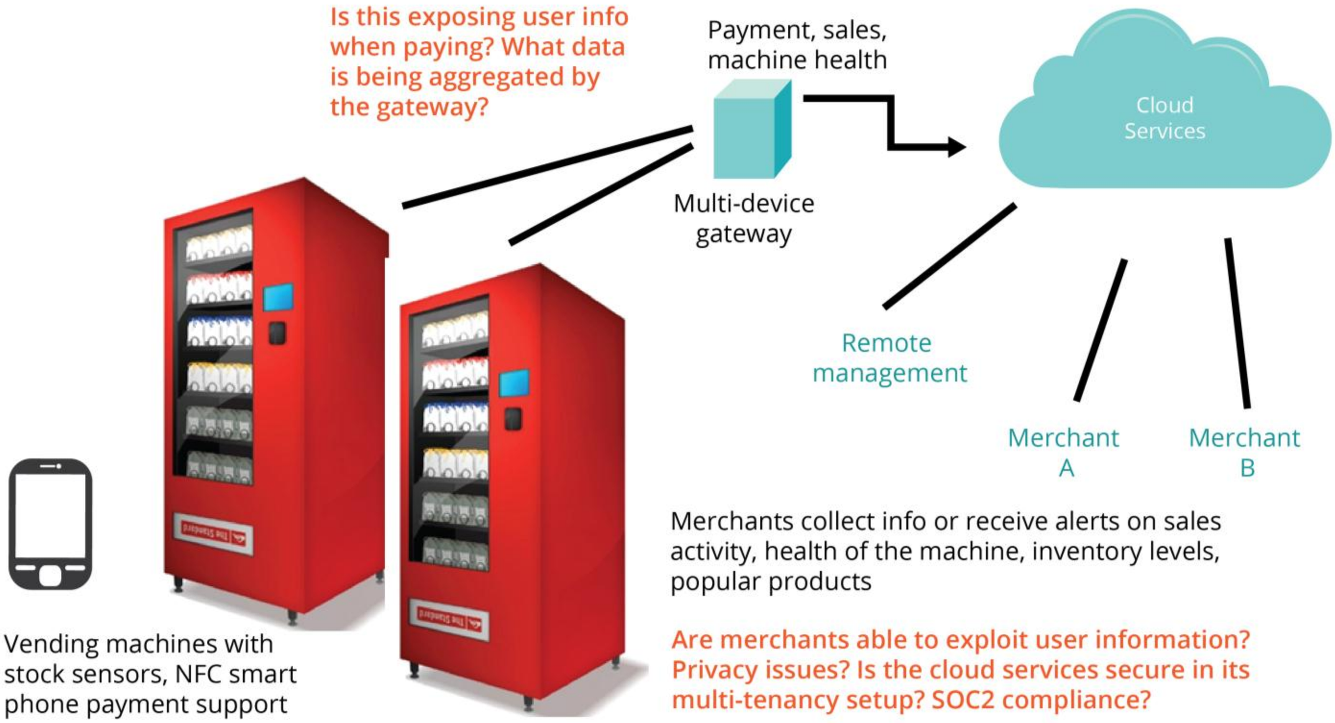
For each IoT deployment, consider the supporting infrastructure required for security management and monitoring. Identify appropriate interfaces to existing security equipment, updating network architectures to segment specific IoT enclaves.

1	Communications Planning	• Where will the device reside (corporate network, other) ○ some forced within the device; some forced by other policy enforcement points w/ in the network ○ what endpoints and what entities will be communicating; under what rule-sets • Publically Accessible IP address (if IPv4) • IPv6 baselining and transition plan ○ neighbor discovery ○ neighbor advertisements ○ support for extensions
2	Physical Security Planning	• Planning the deployment environment; where is the device; warehoused; what is the physical security (access controls)?
3	Logical Security Planning	• Security zone plan
4	Establish baseline for auditable behavior	• What audit capability does the device have? • Do you need to supplement with other audit capture devices that can oversee traffic to/from the devices? • What are the normal operating thresholds for the devices and what should trigger an alert (if outside of that threshold)
5	Establish an Authentication/ Authorization Plan	• Document the roles and services of each device type • Differentiate the security relevant roles • Establish an access control matrix for each device • Establish a plan for federation of devices if needed
6	Determine criticality of device(s) and/or information supported by device(s)	• Determine needed level of rigor in device registration for cryptographic material • Determine cipher suites required for protection of data and device functions
7	Develop deployment and bootstrap validation tests	• Validate the integration of the IoT devices into the security functionality provided by the infrastructure
8	Update Enterprise Architecture documentation	• IoT integration Patterns
9	Information Sharing Plan	• What data can be shared? • What data will be shared? • What are the privacy controls for that data?

10	Establish privacy requirements and controls	• Privacy of data when paired together (pill box labels example) • Can data be arbitrarily reduced without keeping the required protections
11	Establish a safety rqmts and mitigations	• A lot of devices do not support authentication/authorization • develop mitigations to risk as required • What ramifications of electronic abuse on safety of stakeholders

An IoT implementation for Smart Vending machines provides a good example for planning considerations.

Smart Vending Machines



Is this exposing user info when paying? What data is being aggregated by the gateway?

Vending machines with stock sensors, NFC smart phone payment support

Merchants collect info or receive alerts on sales activity, health of the machine, inventory levels, popular products

Are merchants able to exploit user information? Privacy issues? Is the cloud services secure in its multi-tenancy setup? SOC2 compliance?

5.5.2. Deploy

1. Secure configurations for operating systems of IoT edge devices
2. Establishment of device identity (accounts and certificates); Document and inventory devices (asset management)
3. Initial provisioning of key material and trust relationships
4. Operational Security Verification and Validation (V&V)
   a. Are you capturing the needed audit data?
   b. Are the accounts locked down sufficiently, etc.?
5. Negative testing (optional)
6. Deploy gateways as necessary

5.5.3. Manage

Management of IoT devices includes the management of the edge devices themselves, the software and firmware that is loaded onto those edge devices, licenses, and the application of routine patch updates to mitigate vulnerabilities in the devices. Management of IoT edge devices may result in a single point within the enterprise managing all assets, or in cases where there are multiple IoT devices embedded within a larger platform, the management point would likely be embedded within the platform itself, acting as a bridge between the downstream edge devices and an upstream management server. Cryptographic keys, certificates or pre-shared secrets (if used) must also be managed on each device.

5.5.3.1. Asset Management

There is a lot of diversity in IoT edge device types – ranging from low power sensors to ECUs within automobiles. Most IoT edge devices will require updates to one or more layers within the technology stack. Operating systems may require patching, firmware may require updating, and even purpose-built applications may require software updates. Keeping track of the firmware and software versions running on an IoT edge device is a critical aspect of asset management and will allow system administrators to quickly deploy needed updates to the right devices in minimal time. Be sure to define and follow a process to regularly check for updates to the firmware and software running on your IoT devices. Don't assume that the end-vendor will make you aware of updates to the underlying technology stack.

Making sure that these updates are legitimate and haven't been tampered with is just as important as with traditional computing technology. System Administrators should outline a process for validating the authenticity and integrity of all updates, and ensure that the end-to-end process for retrieving, storing and then updating IoT devices is secured.

There are standards that can likely be adapted to the IoT, for the efficient management of firmware updates. As an example, the Open Mobile Alliance's (OMA) Firmware Update Management Object (FUMO) and Software Component Management Object (SCOMO) can likely be gracefully adapted to support firmware and software updates to edge devices. There are vendors today that are already doing this, and the OMA Device Management (OMA DM) Working Group has created a gateway specification (GwMO) that supports the management of devices supporting Bluetooth and ZigBee protocols.

SCOMO also includes the ability to query information from devices, such as the inventory of software components on a device. This could support the ability to ensure that unauthorized applications are not installed on top of edge devices.

Of course, the IoT is not simply about the edge devices that collect and transmit data but also includes the transport links that move that data, the systems that process the data and the systems that make use of the data. Keep track of the software versions of these applications and systems as well and ensure that they are kept updated.

Because of the scale of the IoT, it is important for organizations to be able to manage the hardware / software inventory of their IoT devices. With the above in place next one that is important is license management — this is directly associated with the number of devices we have in the environment. Asset inventory also needs to capture the specific hardware / software versions of the devices in the environment.

Let’s take a case of a security bug in specific software / firmware version of the IoT device — without proper inventory we cannot assess "if the organization is exposed to the risk or if there is no impact to the organization". Another case i can think of if there is a firmware update to the product, we have to analyze if it is applicable then the product owner has to determine it...so mapping Owner to the Asset is very crucial to any asset inventory in the environment. Responsibility to manage the lifecycle of the IoT asset has to be with this owner. Asset Management as a tool / policy will help them in this cause.

What can happen in case Asset management is not done properly — it could lead to compliance issues (license / regulatory), Security (we need to know what is in the environment in order to keep it secure — no of devices / firmware versions / license status / warranty / updates).

5.5.3.2. Cryptographic Key and Certificate Management

IoT devices will most often make use of some combination of cryptographic keys, certificates or pre-shared secrets. When keys and certificates are used, care should be taken to consider the security measures invoked with their creation, distribution and general management. Considerations such as revocation, compromise recovery and initial registration of each device must be thought through and processes tailored based on the value of the information being protected.

Keys should not be made accessible to 3rd parties. The Enterprise should have complete control over key management and key life cycle. A Key and Certificate life cycle assures the security of the key material and the binding of the keys/certificates to users and devices. The lifecycle also defines the handling of certificates that are deemed compromised and the process for destroying key material when no longer needed. Other aspects of the lifecycle include processes for recovering keys when necessary as well as the process used to update the keys and certificates deployed throughout the enterprise on a regular basis.

Once a comprehensive lifecycle has been defined, opportunities to take advantage of secure automation capabilities provided by vendors can be examined. Automation of certificate provisioning and re-provisioning provide streamlined workflows, however it is important to evaluate threats related to automated handling of processes to guard against opening the door to new attack vectors.

The number of keys and certificates deployed across the IoT within an make tracking them a complex task. This may drive a need to increase the lifetime of device certificates which eases the administrative burden of certificate re-provisioning but can also increase the risk that the certificates can be compromised and misused without knowing about it. Although it is not always possible to maintain consolidated situational awareness of the state of all certificates and keys within an organization’s IoT inventory due to segmented networks and trust relationships, organizations should strive to do so when possible.

5.5.3.2.1. Limit Certificate (Key) Lifetimes and Enforce Key Rotation

The length of time that a particular key is used for is a critical factor in determining the risk associated with a particular key. A key that had an unlimited cryptoperiod for instance, would allow an attacker to spend countless hours attempting to perform brute force attacks as well as provide significant opportunities to harvest data in attempts to perform cryptanalysis. When choosing a cryptoperiod for keys, it is important to understand the environment that the keys will be stored and used within. Keys that are afforded rigorous security protections can often be provided with longer cryptoperiods vs keys that are resident in less-protected systems (e.g., smart meters without FIPS 140-2 approved crypto modules). Limiting the lifetime of a key also reduces the time that someone who has compromised the key has in order to make use of that key.

In the energy sector, NISTIR recommended provisioning key lifetimes between 3 and 6 years for utility-provisioned device certificates with a maximum lifetime of 10 years. The reasoning behind this recommendation is based on the need to limit the amount of material associated with a particular key that can be collected in support of cryptanalysis efforts. Updating the key on a regular basis ensures that the usefulness of an attackers attempts at cryptanalysis are limited, assuming sufficiently strong algorithms and key lengths are employed.

Certificates are cryptographically bound to the public/private key pair of a particular entity. As the key pair is updated, the certificate associated with the key pair must be updated as well. There are instances whereby a key that is still considered cryptographically sound can simply be bound to a new certificate, however organizations should consider planning to limit the lifetimes of each device certificate in order to ensure that they are updated on a regular basis and guard against attacks.

There will likely be no expiration date provisioned on some IoT manufacturer-provisioned certificates, which opens questions related to the level of trust that can be placed in those certificates and shows the need to have a plan in place should a manufacturer CA be compromised.

The recommended cryptoperiod for a particular key depends on the purpose and type of the key.
Consult NIST SP 800-57 Section 5.3.6 for specific recommendations by key type.

5.5.3.2.2. Define and enforce registration processes

The Registration and approval to issue a certificate to a relying party with a PKI certificate is an important aspect of ensuring the security of a PKI implementation. For devices that process PII or other sensitive information, care should be taken to employ a registration process that that is equivalent in security strength to the other controls applied to protect the data.

5.5.3.2.3. Define a compromise recovery plan

One of the most important aspects to any key management implementation is a clear understanding of what must be done in the case of a compromise to either an IoT edge device or more critically, to a CA. In the case of an edge device key or certificate compromise, a standard process of certificate revocation should be followed, however an inquiry into the compromise should also be conducted to understand how the key/certificate was compromised so remediation actions can be undertaken, limited the potential for the event to re-occur.

5.5.4. Monitor and Detect

- Automate security tasks such as vulnerability assessments and forms of penetration testing
- Develop dynamic, real time and continuous monitoring of devices to automate IoT threat intelligence

Security professionals are usually overloaded with securing many applications and devices at any given time due to the fact that for every security professional, there are 50-60 application code developers who may be developing code with potential vulnerabilities. Hence old methods of manual penetration testing and quarterly security reviews are no longer sufficient. There is a need to automate manual security testing using dynamic monitoring tools on a regular basis.

More involved penetration testing is still required to effectively evaluate IoT security posture on a regular basis. Considering the IoT, physical testing of devices with exposed JTAG and serial interfaces cannot be automated. A savvy user can take advantage of debugging hardware interfaces that are left after mass deployment has taken place. Because of this, perform full penetration testing activities on a quarterly basis.

The IoT also provides the opportunity for the use of big-data analytics along with the dynamic monitoring tools to predict real time threats. This will enable organizations to recover faster from security disasters. Health scans of each IoT device to check for liveliness and functional operational are also recommended.

Monitoring for security events within an IoT infrastructure should also be done, ideally on a 24/7 basis. Planning for the capture of security-relevant data and establishment of rules for identifying events or combinations of events-of-interest should be conducted early on in the engineering lifecycle. Consider having security analysts charged with near-real time monitoring of the security posture of your implementation.

Keeping track of the vulnerabilities associated with various software components of an IoT implementation, as well as the latest threats to IoT device types is also important. Consider assigning this duty to someone to keep track and report on the various threats to your specific IoT implementation.

5.5.5. Remediate

Update incident response plans to incorporate new IoT systems and define the procedures for handling compromise events. Establish call plans for security analysts to escalate events quickly and be prepared to fly out teams of incident responders to investigate and remediate issues.

5.5.5.1. Dispose

Due to the quantities involved with many IoT implementations, it is likely that many edge devices will be replaced on a regular basis. It is important to establish policies and procedures for the secure disposition of devices that have held sensitive information or key material that could provide access to sensitive information. Devices that have held sensitive information should be securely wiped to include removal of key material and certificates from each device.

5.6. Define and implement an authentication/authorization framework for the Organization’s IoT Deployments

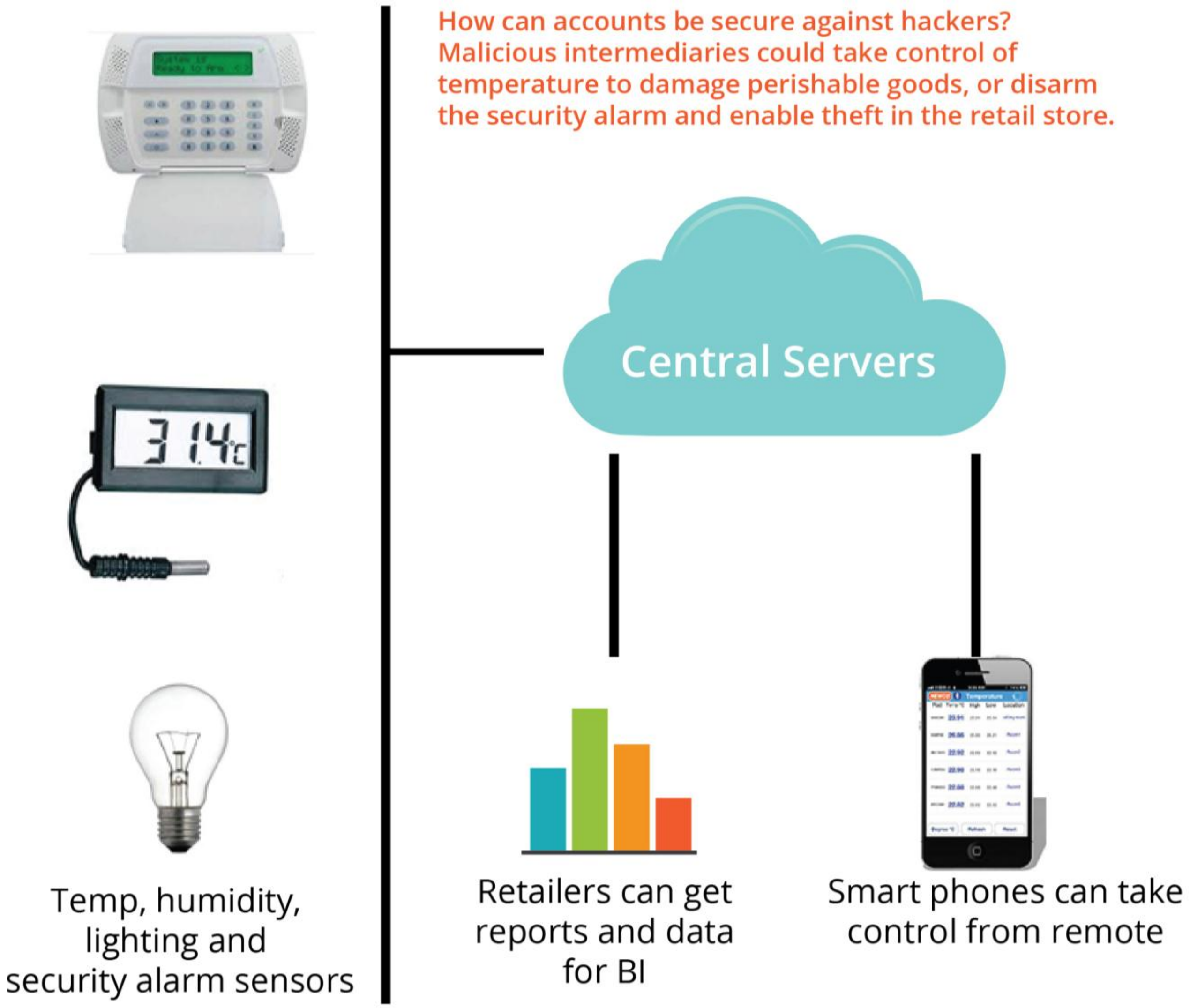
Scenarios for IoT authentication are numerous. IoT components may communicate with each other, requiring machine-to-machine (m2m) authentication. IoT components may communicate with cloud applications, mobile applications, web applications or even directly with people. One of the challenging aspects of authentication and authorization within the IoT is that many devices will operate under constrained conditions, meaning that the protocols employed may limit authentication options, or that the devices will not be capable of using certain authentication capabilities, for example certificate-based authentication.

There are a number of diverse use cases related to the authentication of IoT components within an enterprise. Most of these use cases can be abstracted to generalize the authentication requirements within the IoT to the following:

- An IoT device authenticates to another IoT device
- An IoT device authenticates to a gateway or controller device, or vice-versa
- A gateway/controller authenticates to some service (in the cloud), or vice-versa
- Various applications authenticate to some service (in the cloud)
- A user authenticates to an IoT device (e.g., doctor authenticates to medical implantable)
- An administrator authenticates to an IoT device (e.g., traffic management center authenticates to road-side equipment)

An example use case from the retail industry can show some of the questions to ask related to securing accounts from compromise. Figure x shows the use case of security alarms and environmental sensors collecting data from multiple edge components and sharing that data with various mobile devices. In this scenario, it is important to ensure that all accounts are locked down sufficiently to mitigate the threat of an attacker gaining access to sensitive systems that could result in physical damage.

Security Alarm & Environmental Sensors



How can accounts be secure against hackers? Malicious intermediaries could take control of temperature to damage perishable goods, or disarm the security alarm and enable theft in the retail store.

Central Servers
Temp, humidity, lighting and security alarm sensors
Retailers can get reports and data for BI
Smart phones can take control from remote

If we examine how IoT authentication is implemented today, we can see that there are a number of options available. These options typically include:

- Pre-shared key/shared secret
- Certificate-based authentication
- Token-based authentication

We have also seen instances in the past where companies have implemented weaker authentication schemes, such as the use of a hashed MAC for identification. These approaches may support ease-of-use, however they are not recommended for implementations where the security of the platform is required.

Shared secrets can be used although they can introduce significant management overhead burdens. If a shared-secret approach is used, be sure that the scheme is based on the use of a NIST specified Hashed Message Authentication Code (HMAC) algorithm that cryptographically binds a message's content and identity (the provisioned key). HMACs provide data origin authentication and message integrity verification functions. An example of an HMAC scheme is HMAC-SHA-256.

Certificate-based authentication can support protocols such as TLS and DTLS. One of the challenges with certificate-based authentication is related to the size of the X.509 certificate structure. An alternative structure does exist, which is optimized for machine-to-machine authentication transactions and memory-constrained devices. IEEE defined 1609.3 certificates for use with the Digital Short Range Communications (DSRC) used in vehicle-to-vehicle communications. The certificate structure is optimized for memory-constrained devices and is much smaller than a standard X.509 certificate. Developers should consider the potential to move to IEEE 1609.3 certificates when faced with constrained devices and environments.

The use of certificates (whether X.509 or 1609.3) introduces the likely need for a Public Key Infrastructure that centrally manages all of the certificates provisioned to devices. This includes critical functions such as trusted registration and compromise recovery.

Token-based authentication schemes such as OATH 2 and OpenID Connect Federated Authentication provide useful alternatives to shared secrets and certificates, and also allow for the introduction of comprehensive policy controls applied to IoT access requirements.

The authentication method chosen depends on the constraints of the device. Shared secret authentication is considered a less-desirable alternative to certificate-based authentication. With shared secrets, the overhead involved with managing the secrets becomes significant as the number of devices increases. Certificate-based authentication introduces concerns related to the processing of certificates and the asymmetric algorithms used for functions like key authenticated establishment.

For device-to-device transactions, whether it be from peer devices or from edge devices to gateways or propagators, it is often best to take advantage of the authentication capabilities built directly into the protocols that they support. As an example, the Constrained Application Protocol (CoAP) provides four modes of operation. Each mode has a threat level that requires some level of authentication.

- No Security — assumes security at another protocol layer
- preSharedKey — a symmetric key is shared across the group authorized to communicate
- rawPublicKey — a single asymmetric key is provisioned to each device implementing CoAP
- Certificate — each device implementing CoAP is provisioned with an X.509 certificate

In CoAPs, "No Security" mode assumes security is being applied at another protocol layer. The preSharedKey mode provides rudimentary authentication between devices, however is not recommended given the difficulty in keeping the key secure and the difficulty in managing the key. The preSharedKey mode relies upon the use of a single key within a device communication net. While this approach may be sufficient for small quantities of devices, it does not scale well. If the key is compromised, changing out keys on all participating devices becomes time consuming and difficult.

Better approaches to authentication using CoAP for device-to-device transactions are the use of the rawPublicKey and certificate modes. The rawPublicKey mode requires the provision of unique asymmetric keys to each device, eliminating the concern related to comprise of a single key requiring rekey of all devices. Certificate mode is similar to preSharedKey mode although adds the additional measure of trusting segments of devices (based on CA issuer being in the trust store) and strong support for revocation. You would typically use this mode when employing a Public Key Infrastructure (PKI) for your devices.

By examining other IoT protocols from an authentication perspective, we can observe the following:

Protocol	m2m Authentication Options	Analysis
MQTT	username/password	MQTT allows for sending a username and password, although recommends that the password be no longer than 12 characters. Username and password are sent in the clear, and as such it is critical that TLS be employed when using MQTT.
CoAP	preSharedKey rawPublicKey certificate	CoAP supports multiple authentication options for device-to-device communication. Pair with Datagram TLS (D-TLS) for higher level confidentiality services.
XMPP	aMultiple options available, depending on protocol	XMPP supports a variety of authentication patterns via the Simple Authentication and Security Layer (SASL – RFC4422). Mechanisms include one-way anonymous as well as mutual authentication with encrypted passwords, certificates and other means implemented through the SASL abstraction layer.
DDS	X.509 Certificates (PKI) using RSA and DSA algorithms. Tokens	The Object Management Groups Data Distribution Standard (DDS) Security Specification provides endpoint authentication and key establishment to perform subsequent message data origin authentication (i.e., HMAC). Both digital certificates and various identity / authorization token types are supported.
Thread	(Beta standard to be released)	A smart device wireless networking IPv6 protocol, Thread is anticipated to utilize and improve security options found in other wireless protocols.
Zigbee (802.15.4)	Pre-shared keys	Zigbee provides both network and application level authentication (and encryption) through the use of Master key (optional), Network (mandatory) and, optionally, Application Link keys [RBJ1] .
Bluetooth	Shared Key	Bluetooth provides authentication services through two different device pairing options, Standard and Simple Pairing. The Standard pairing method is automatic; the Simply pairing method includes a human-in-loop to verify (following a simple Diffie-Hellman exchange) that the two devices display the same hash of the established key. Bluetooth offers both one-way as well as mutual authentication options. Bluetooth secure simple pairing offers ‘Just works’, ‘Passkey entry’ and ‘Out of Box’ options for device-device authentication.

Protocol	m2m Authentication Options	Analysis
Bluetooth-LE	Unencrypted data authenticated using Connection Signature Resolving Key (CSRK) Device Identity/Privacy is via an Identity Resolving Key (IRK)	Bluetooth-LE introduces to the Bluetooth world a two-factor authentication system, the LE Secure Connections pairing model which combines – based on device capability – several of the available association models available. In addition, Elliptic-Curve Diffie Hellman is used for key exchange.
HTTP/REST	Basic Authentication (cleartext) (TLS methods) OAUTH2	HTTP/REST typically requires the support of the TLS protocol for authentication and confidentiality services. Although Basic Authentication (where credentials are passed in the clear) can be used under the cover of TLS, this is not a recommended practice. Instead attempt to stand up a token-based authentication approach such as OAUTH 2.

The process of selecting the optimal authentication mechanisms for your IoT deployment can be guided by answering the following set of questions:

1.	Does your implementation require machine-to-machine communications?	If so, examine the device communication protocols and determine if they natively support authentication.
2	Do your IoT devices support one of the communication protocols that provides authentication services?	If not, consider layering security to include higher level authentication services, such as TLS or DTLS.
3	Is your IoT device inventory constrained in memory or processing power?	If yes, consider working with vendors to support IEEE 1609.3 certificates
4	Who will manage your devices? Is remote management required?	First, plan out your implementation. Create an authentication and access control matrix and select the strongest method of authentication supported by each edge device.
5	Do your IoT devices expose network-based remote management functions such as SNMP or SSH?	Lock down each device prior to fielding to only support authorized management services. Establish a policy and procedures for network-based remote management of your devices.
6	Do your IoT devices implement RESTful interfaces?	Consider a token-based approach such as OAUTH 2 for authentication of edge devices.
7	Do the devices connect directly to services within the cloud?	Ensure that the authentication design includes API keys that support device and application authentication to the cloud service.

5.6.1. API discussion/API keys

API security is an important part of IoT security. IoT providers are releasing open APIs to products enabling many new varied uses and possibilities. This precedence given to APIs is driving ecosystems in the IoT space and it becomes imperative for IoT vendors to put security first in a vendor's local environment. Another concern is to prevent a vendor from abusing its API access on the cloud, where the IoT devices can talk to each other if given global access. We need to develop a security model that meets both local needs and cloud-connected global needs.

5.6.2. Identity and Access Management

In addition to IAM, implementing a privileged user management system tracks activities and interactions of administrators, administrative consoles, and applications. This is especially useful in large deployments of IoT and assists to define credential policies, force password rotations after every use, issue and removal of credentials, recording of activities (e.g. keystrokes for forensics).

For consumer-identities, especially in retail or transportation, there is a notion of centralized policy-based and consent-based systems that support consumer empowerment to consent to:

- Which of their attributes or information can be in the clear
- What information needs to be masked when being displayed
- What information can be used by third party analytics and marketing
- Other preferences

This is highly related to compliance, but marrying these preferences with the organization's security policies provides a stronger security framework.

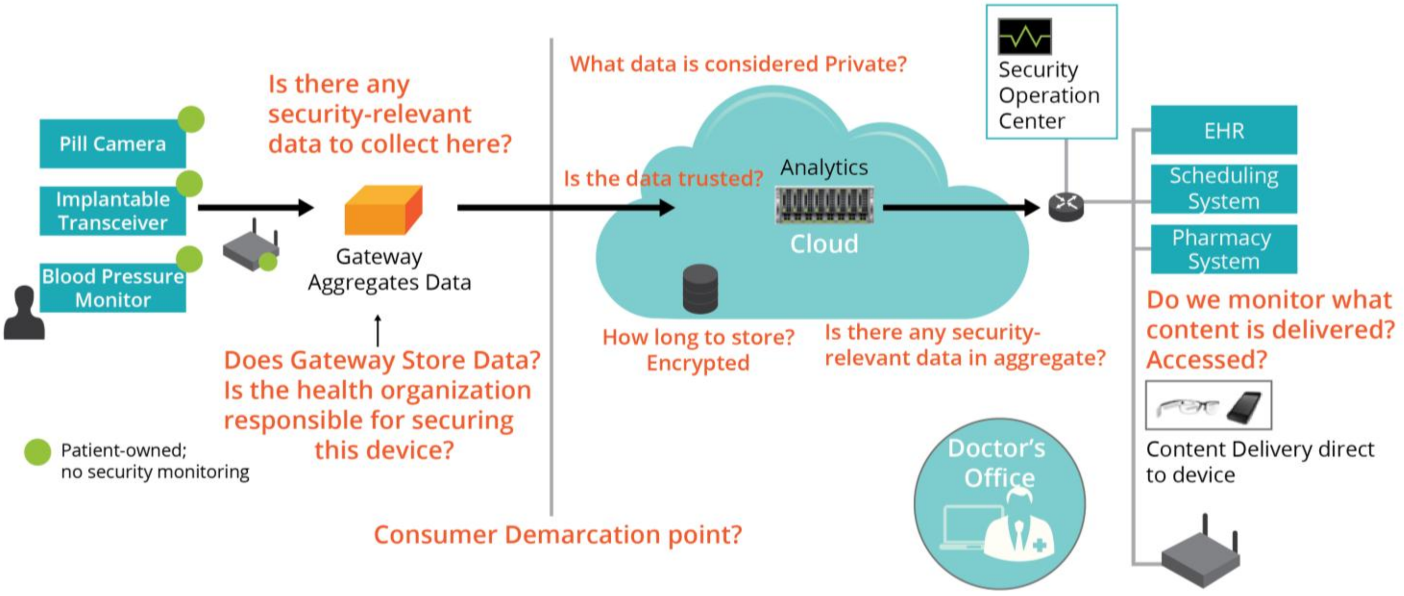
Having a centralized IAM that is policy-based across all services where IoT devices are involved will be easier to manage than having a distributed IAM for every IoT device and associated applications. Strengthening and hardening one system with uniform policies across the board is more consistent than provisioning a separate authN/AuthZ for every service.

5.7. Define a Logging and Audit Framework for the Organization's IoT Ecosystem

Today's IoT devices typically lack the ability to integrate with a SIEM and SIEM systems have yet to prove that they are sufficient for monitoring widely deployed and massive quantities of IoT devices. In many cases, the IoT devices do not support the harvesting of data through exposed Application Programming Interfaces (APIs), and in some cases IoT devices may belong to outside organizations but provide data that is critical to the successful operation of an IoT-based analytics system. Other constraints on defining a comprehensive logging architecture include the cost of transmitting data over Radio Frequency (RF). In many cases, this drains battery power at unacceptable levels.

Planning for an IoT audit/logging framework is aided by examining example use cases. The figure below provides an example of a remote patient monitoring use case.

Remote Patient Monitoring

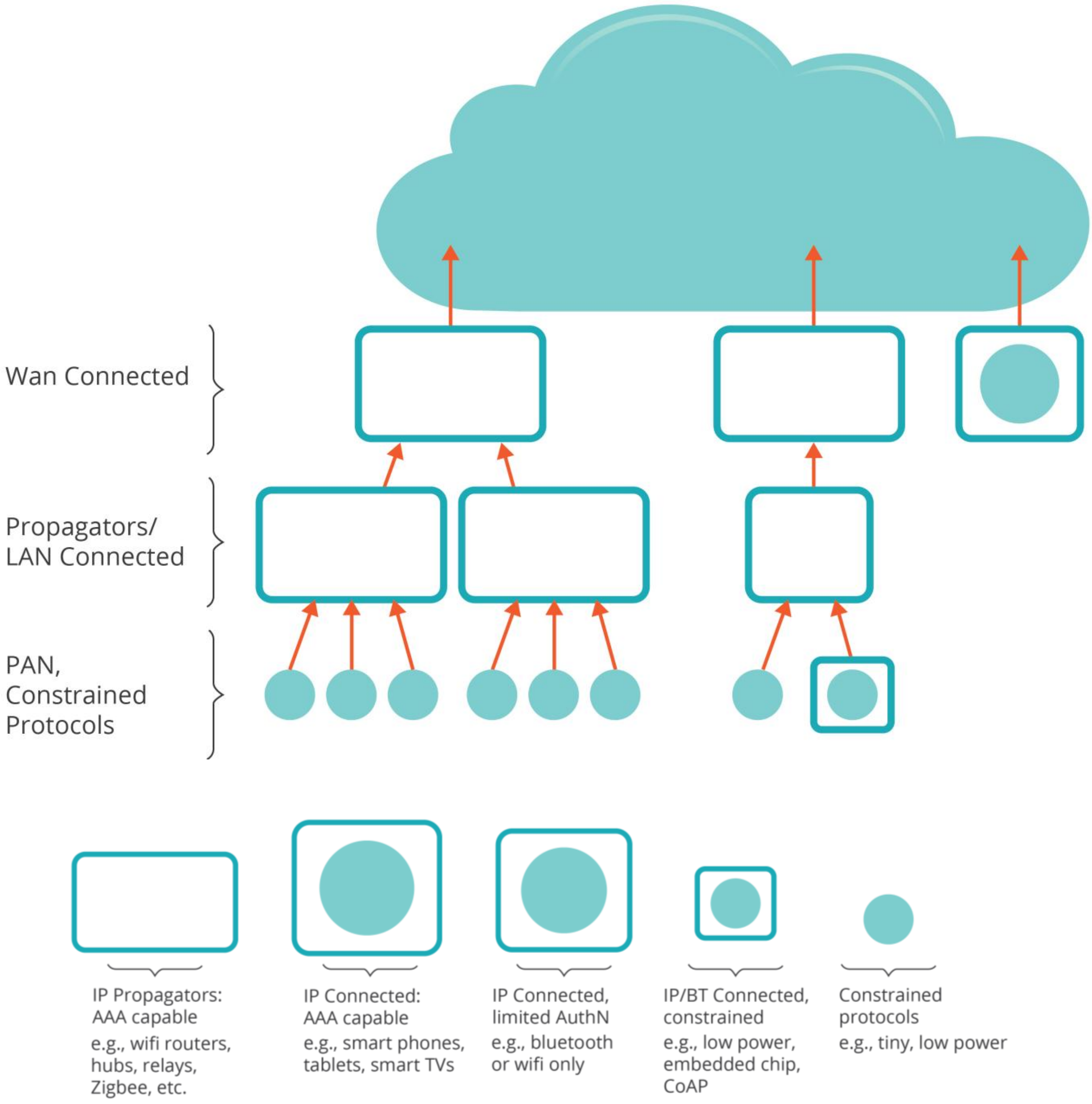

Pill Camera
Implantable Transceiver
Blood Pressure Monitor
Is there any security-relevant data to collect here?
Gateway Aggregates Data
Does Gateway Store Data? Is the health organization responsible for securing this device?
Patient-owned; no security monitoring
What data is considered Private?
Is the data trusted?
Analytics
Cloud
How long to store? Encrypted
Is there any security-relevant data in aggregate?
Consumer Demarcation point?
Security Operation Center
EHR
Scheduling System
Pharmacy System
Do we monitor what content is delivered? Accessed?
Content Delivery direct to device
Doctor's Office

As can be seen, it is important to understand what components within the IoT ecosystem will actually provide audit data feeds and which components should actually be mined for anomalous behavior within their operational data stream. As an example, considering which components are owned by a consumer will allow for a plan to captured and analyze appropriate data (e.g. failed logins). It is also important to ensure that no sensitive (privacy-related) information is included in the audit logs unless it is protected using sufficient security safeguards (e.g. encryption).

5.7.1. The Use of Gateways and Aggregators

Given the constraints of today's IoT devices, a tailored approach to maintaining situational awareness of an IoT ecosystem must be adopted. IoT deployments can include single sensors or consolidations of multiple sensors and other devices. In the latter case, it may be prudent to deploy a multi-layered architecture that consists of endpoints logging to a consolidator or propagator. These propagators can then translate between the protocols used by each device and provide higher aggregation of information for analysis.

Gateways and Aggregators


Wan Connected
Propagators/ LAN Connected
PAN, Constrained Protocols
IP Propagators: AAA capable
e.g., wifi routers, hubs, relays, Zigbee, etc.
IP Connected: AAA capable
e.g., smart phones, tablets, smart TVs
IP Connected, limited AuthN
e.g., bluetooth or wifi only
IP/BT Connected, constrained
e.g., low power, embedded chip, CoAP
Constrained protocols
e.g., tiny, low power

This tiered architecture allows for the placement of highly constrained devices at the edge. These devices typically have minimal capabilities and are connected via some wireless protocol. These devices can connect back to a more capable device which is connected to the organization's Local Area Network (LAN) and then offload collected information to another node for transport. This architecture also allows for an organization to collect security data from devices that are on various distinct network segments across the company. One trait of the IoT is that there will be many different device types spread across many different geographic and logical areas. Capturing and correlating data from all of these devices requires a standardized format for recording and transmitting audit log data.

Information overload is another worrying aspect of the IoT, given the quantity of devices that will collect or process data. Fine tuning the collection of audit data to capture data that is security relevant is important to minimize the risk of too much information triggering false positives and an inability to process information sufficiently to identify true threats.

Complicating matters further is that there is a logical extension of the IoT to the cloud. The IoT can exist in both a physical and virtual form and data should be captured for both.

It is important to make a distinction between the operational data that is captured and communicated by an IoT device, and the security audit data that is required to maintain security awareness of the device. Operational data, for example water temperature collected by a temperature sensor, would not necessarily be security relevant and as such should not be fed upstream for aggregation and analysis.

There will be significant opportunity in the near future to begin understanding the relationship between in-line data (operational IoT data) and security audit data. Tuning data analytics systems to identify potential security events and feeding the filtered output to a SIEM would add significant security value.

5.7.2. Logging of Data

In general, it is important to log data that may indicate that an incident has occurred or will occur. Whenever possible, the following minimum data elements should be logged.

5.7.2.1. What Events to Log

- Failed privilege elevation attempts
- Failed log-ins to the device
- Failed log-ins to the service provider (cloud)
- Failed device-to-device authentication attempts
- Failed database access attempts
- Policy changes
- Privilege use
- Account creation
- Account change
- Failed tunnel negotiation
- Internal state
- On/Off
- Changes in the integrity of appropriate file systems

5.7.2.2. What Metadata to Log

- Start time
- End time
- User
- Peer device id
- Destination device MAC address
- Destination device IP address

- Destination device IPv6 address
- Destination device hostname
- Transport protocol
- Data-link protocol

5.7.2.3. Where to Log

Logging should occur as close to the end-device as possible, although it may not be possible to collect or routinely forward data at some disadvantaged devices. In these instances, maintaining situational awareness through collection of data at IP propagators such as Wi-Fi or other protocol routers, gateways and standard network security devices should be evaluated.

5.7.3. Transport

Ensure that security data from edge devices and aggregators is encrypted and authenticated during transport.

5.7.4. Security Considerations

An important consideration related to the transport of logging data from IoT devices, is that encryption should be applied for the protection of sensitive/PPI data. This includes storage of audit data while at rest on end devices and propagators, as well as the use of encryption for data-in-transit between audit/logging components.

6. Future Efforts

The IoT has limitless capabilities forcing innovation and adoption industry-wide. These opportunities also come with new threat vectors. Efforts to mitigate these security risks will breed innovation in the development of standards and technologies specific to the IoT.

6.1. Standards

An unfortunate characteristic of the current state of the IoT is the lack of standardization across all aspects of the IoT. This can be seen by examining the wide range of communication protocols, messaging busses, processors and even operating systems that can be married together to provide IoT functionality. Organizations today cannot readily purchase packaged IoT systems that support their unique uses cases and as such must work to engineer their own systems. This complexity foreshadows misconfigurations and vulnerabilities within each IoT System. New standards for security assurance of IoT devices may also aid adopters in evaluating the aforementioned threats and risks in deploying IoT. Such schemes (e.g., Common Criteria) are today mainly used only for high-end systems.

6.2. Situational Awareness of the IoT Security Posture

Industry must work on ways to identify security relevant data from the streams of operational data provided by and analyzed within the IoT. This includes work on ways to identify anomalous behavior within the operational data stream provided by sensors, providing a view into whether a single or small group of sensors has been compromised based on a change in their behavioral pattern or inputs outside of the norm. Establishment of standardized APIs between data analytics platforms and SIEMs would provide the ability to capture a holistic security view of an organization's IoT implementations. Tuning data analytics systems to identify potential security events and feed those filtered output to a SIEM would add significant security value.

6.3. Information Sharing

Due to the scale of devices within the IoT and the newness of IoT technology in general, there will likely be a stream of 0-day attacks available to exploit weaknesses discovered in IoT implementations. In order to reduce the period of exposure to these new exploits, organizations should consider joining an information sharing and analysis center for the IoT. This would allow for organizational collaboration, sharing of statics and threat reporting across interested and like parties.

6.4. SDP and the IoT

The IoT relies on the cloud for transport in most cases and the traditional notion of perimeter security is becoming obsolete. Research into the marriage of the Software Defined Perimeter (SDP)* with the IoT will provide great benefit towards establishing a layered security approach and network protections to guard against IoT attacks.

6.5. Privacy in the IoT Environment

There are many privacy concerns regarding the IoT that must be examined. One significant concern is the issue of data-capture from sensors to which a consumer is unaware. In these instances, an individual is being watched or tracked by an organization or by another individual, without knowing about it. Many questions exist in this scenario, including what recourse the tracked person has and what responsibility 3rd party organizations have to make sure that the information they collect has not been received without explicit consent.

Finally, it will be important for us to explore which of the suite off privacy requirements (notice, awareness, choice, consent, access, enforcement etc.) can be met through technical controls and which uniquely fall outside the IoT architecture. Similarly, which elements can be captured and transmitted with the captured-data, in terms of attributes, that can be sustained throughout the life of their dataflow across the entire IoT System? For instance, if someone has agreed to a certain use of their data, at capture, how does that determination stay attached to that data as it is shared and transferred and analyzed by the other IoT players? Is it possible to tag data and negotiate "handshakes" between ecosystem partners to ensure that the data is processed only in accordance with the original consent? This includes any re-identification after data has been collected only as an anonymous element.

Appendix A: References

ITU-T Y.2060 Overview of the Internet of Things @ http://www.itu.int/rec/T-REC-Y.2060-201206-I

State of the Market, The Internet of Things in 2015, Verizon @ http://www.verizonenterprise.com/resources/reports/rp_state-of-market-the-market-the-internet-of-things-2015_en_xg.pdf

Industrial Internet of Things Positioning Paper, Accenture @ http://www.accenture.com/SiteCollectionDocuments/PDF/Accenture-Industrial-Internet-of-Things-Positioning-Paper-Report-2015.PDF

IDC Futurescape for Internet of Things, December 2014 @ https://www.idc.com/getdoc.jsp?containerId=prUS25291514

Mitre Common Vulnerabilities and Exposures @ https://cve.mitre.org/

SOHO Wireless Router (In)Security: Tripwire Vulnerability and Exposure Research Team (VERT) Report, 2014 @ http://www.tripwire.com/register/soho-wireless-router-insecurity/showMeta/2/

Dan Geer Security of Things Forum, May 2014 @ https://securityledger.com/2014/05/dan-geer-keynote-security-of-things-forum/

Symantec Corporation Internet Security Threat Report 2014 Volume 19 @ http://www.itu.int/en/ITU-D/Cybersecurity/Documents/Symantec_annual_internet_threat_report_ITU2014.pdf

Opening Remarks of FTC Chairwoman Edith Ramirez Privacy and the IoT: Navigating Policy Issues International Consumer Electronics Show Las Vegas, Nevada January 6, 2015 @ https://www.ftc.gov/system/files/documents/public_statements/617191/150106cesspeech.pdf

Software Defined Perimeter (SDP) Specification Document v1.0 @ https://downloads.cloudsecurityalliance.org/initiatives/sdp/SDP_Specification_1.0.pdf

Privacy and Data Protection Impact Assessment Framework for RFID Applications 12 January 2011 @ http://cordis.europa.eu/fp7/ict/enet/documents/rfid-pia-framework-final.pdf

Article 29 Data Protection Working Party: Opinion 8/2014 on the on Recent Developments on the Internet of Things @ http://ec.europa.eu/justice/data-protection/article-29/documentation/opinion-recommendation/files/2014/wp223_en.pdf

Threat Modeling: Designing for Security by Adam Shostack @ http://threatmodelingbook.com/index.html

Microsoft Developer Network: The STRIDE Threat Model @ https://msdn.microsoft.com/en-US/library/ee823878(v=cs.20).aspx

Microsoft Developer Network: Threat Modeling — DREAD @ https://msdn.microsoft.com/en-us/library/ff648644.aspx

MITRE's Common Vulnerabilities and Exposures @ https://cve.mitre.org/index.html

OWASP's Top 10 Threats for Internet of Things @ https://www.owasp.org/index.php/OWASP_Internet_of_Things_Top_Ten_Project