April 2015

IBM POINT OF VIEW: INTERNET OF THINGS SECURITY

The connectivity of "things" presents an exciting environment for innovation and opportunity, but also a broad set of security challenges and threats.

INTRODUCTION

This document describes IBM®'s comprehensive view of security and privacy for Internet of Things (IoT) systems. In a November 2014 report, analysts estimated that IoT will represent 30 billion connected "things" by 2020, growing from 9.9 million in 2013[1]. The ubiquitous connectivity of things that enrich our lives, businesses, and organizations, such as thermostats, medical devices, automobiles, and industrial equipment, presents an exciting environment for innovation and new business opportunities. This expanded computing environment also presents a broad set of security issues and threats. A world of connected things makes them, the data they produce and use, and the systems and applications that support them, potential attack points for malicious actors. Potential attacks include obtaining private or confidential data, manipulating or controlling devices, or confusing or denying service to applications that use and supply data within IoT systems.

The risks for Industrial IoT systems that support manufacturing, energy, transportation, and other industrial sectors of the economy, are even more challenging. As industrial things become connected to the Internet to enable broader visibility, control, and condition-based maintenance, they also become vulnerable to security attacks. There are published reports of Supervisory Control and Data Acquisition (SCADA) systems and Industrial Control Systems (ICS) being hacked. "After finding more than 60,000 exposed control systems online, two Russian security researchers found vulnerabilities that could be exploited to take full control of systems running energy, chemical and transportation systems."[2]

An IoT system consists of devices (things) that communicate with other devices, applications, and services that use a variety of protocols, and that expose application programming interfaces (APIs) to access data and services across the Internet. Devices range from basic individual sensors that are directly connected to the Internet or that are connected via some form of simple gateway, to more powerful, and sophisticated processing nodes capable of autonomous processing. For example, a connected vehicle is a complex device that consists of different electronic subsystems and sensors that can process autonomously, but can also connect wirelessly to the Internet.

There are different requirements for IoT security depending on the risk profile of the system being secured. The security needs for a consumer IoT system to measure and control a watering system for garden plants are different from the needs of a complex, mission-critical, enterprise petroleum drilling or pipeline operation that involves IoT-connected valves and pumps. The drilling and pipeline operations must include safety-critical systems to protect the business, the environment, and human life. The risks and costs of compromise for the drilling operation are far greater than those of the home garden watering system. As such, comprehensive security measures, expertise, analysis, testing, and management are necessary. For those organizations that are building IoT systems at the high end of the security risk and complexity scale, experienced subject matter experts are needed to provide guidance on the design and operation of such systems. The topic of IoT security is popular, with many people and organizations providing views and insights. IBM first mentioned security and IoT in an IBV Study[3] published in June 2014. Other entities are also publishing information and viewpoints as well, including a recent document from the Open Web Application Security Project (OWASP)[4], as well as some consortium groups, such as the Industrial Internet Consortium (IIC)[5], Allseen Alliance[6], and builditsecure.ly[7].

A critical consideration for security of IoT systems (or any IT environment) is that the system cannot depend on the constant integrity of every connected device to ensure the ongoing integrity of the whole system. The design and security features of the IoT system assume that individual devices might be compromised (no security is foolproof), and still be able to function securely with one or more compromised devices.

INTERNET OF THINGS SYSTEM ARCHITECTURE

There are many possible configurations for IoT systems. In some IoT systems, all devices are directly connected to the Internet, and each device is responsible for its own local security.

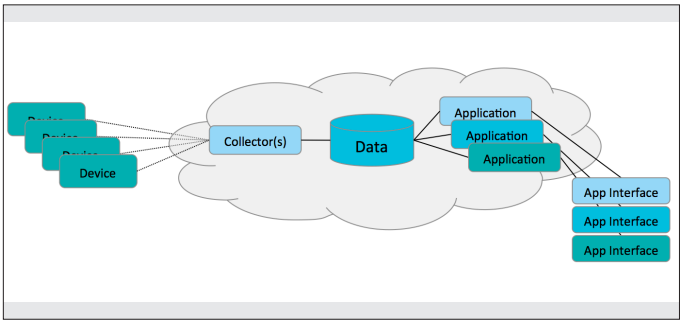

Figure 1: IoT system with devices that are directly connected

In other IoT systems, devices might be connected locally to an aggregation node that acts as an intermediary, or gateway, to aggregate data from locally connected devices. The gateway filters and intelligently reacts to data, and sends and receives data or commands to and from the Internet. A gateway device is used to connect previously unconnected devices, older devices, and insecure devices. It can also provide operational efficiency by allowing multiple devices to share a common connection.

The gateway device might be responsible for managing security on behalf of the locally connected devices as a proxy for the other devices that are connected to the outside world. The role of the gateway is a critical element of the security system, since it manages its connections to the downstream devices and must assure their authenticity.

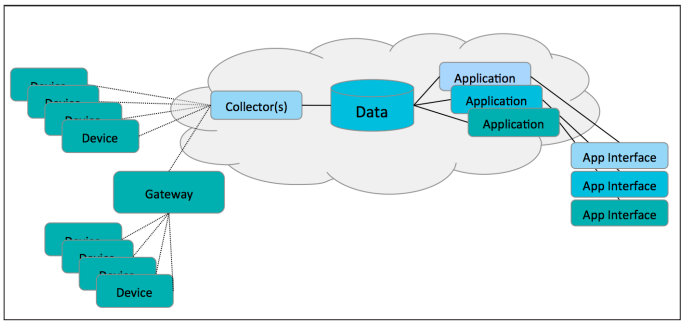

Figure 2: IoT system with devices that are connected through a gateway device

For example, connected vehicles contain many sensors and processors that are themselves unsecured and connected only to the local controller area network (CANbus) in the vehicle. One subsystem, typically the telematics, or infotainment subsystem, acts as the communication gateway between the vehicle and the outside world. This subsystem aggregates data from the other vehicle subsystems to communicate to the Internet, and interprets commands or data that are received from the Internet. The subsystem redistributes the data and commands through the local CANbus to the other vehicle subsystems. In an industrial environment, such as a manufacturing facility, it is common to find devices that are connected by means of existing industrial protocols, such as Modbus, Profibus, or DeviceNet, to a local gateway device. The local gateway might aggregate data, filter data and perform local analytics. It can also connect to a cloud or back-end server to propagate data up to higher-level systems and analytics.

The devices that connect to the cloud might not be a single entity, but might consist of hierarchies of connected Internet nodes. The applications that support the devices might be distributed across multiple hardware nodes for reasons of scalability, performance, or fault tolerance, but appear as a single logical source/destination as far as the connected devices are concerned.

There are also IoT systems that communicate in a peer-to-peer or meshed model. In these systems, there are unique security characteristics to consider, along with risks, threats, and attacks to address. These environments are challenging due to the constraints of their peer-to-peer operating environments. The devices often

operate on low power, with a low level of network communication, and a relatively low level of computing, storage, and memory capability. Devices might move between disconnected and connected states, and between different peer-to-peer collections of devices.

The IoT system might be connected to other systems, such as back-office systems, other linked IoT systems, government, or municipal systems, and services that are provided by parties that operate in the Internet. The entire ecosystem of devices, networks, and application systems must be considered within the security scope of the IoT system.

Another point of view is that of a human user of connected things, in this case, a typical consumer. This consumer has access to many things through a mobile device. The device becomes a window on their connected things world and a potential point of security vulnerability.

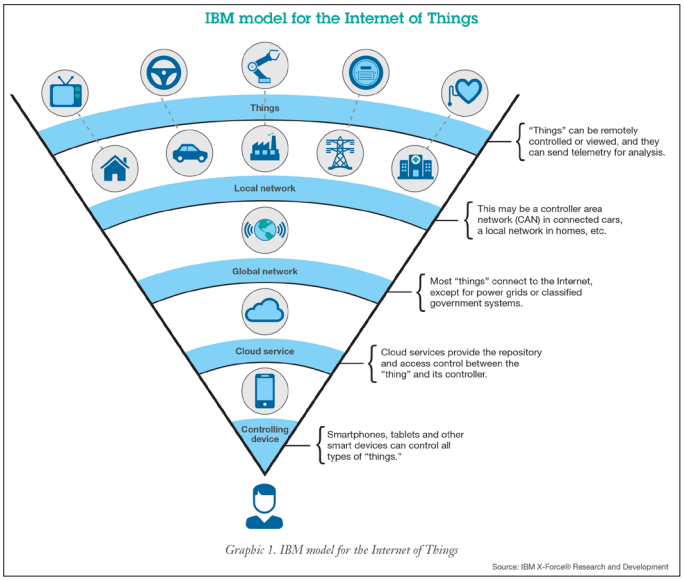

Figure 3: Human view of Internet of Things (Source: X-Force Research and Development)[8]

As has been suggested, there are a wide range of risks, threats, and attacks that apply to IoT systems. In Figure 4, these attacks are annotated onto the high-level system architecture diagram from Figure 2. Some of these threats are familiar, such as man-in-the-middle attacks, application vulnerabilities, and information leakage. Denial-of-service attacks – on either the applications or the devices – are also a threat. There is the added threat of rogue devices or devices taken over with a "zombie" being used to carry out denial-of-service attacks against other systems in the IoT environment.

Protections against attacks and exploits are also numerous and in many cases well known. Some of these include OS integrity checks, authentication/authorization, anomaly detection, and secure development and delivery. As Figure 4 shows, different sets of protections apply to different areas of the IoT system.

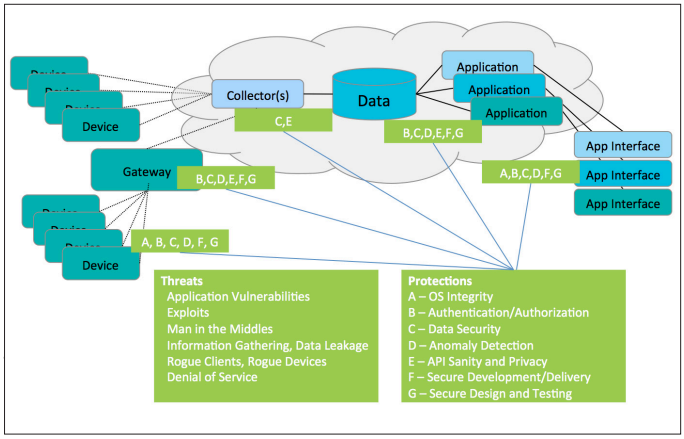

Figure 4: IoT system with threats and protections annotated

Throughout this document, the issues and techniques for managing security in IoT systems are addressed from two broad perspectives:

- Makers of Things - The design and manufacture of secure IoT systems and devices
- Operators of Things - The secure operation of IoT systems that are deployed

MAKERS OF THINGS - DESIGN AND MANUFACTURE SECURELY

DESIGN FOR SECURITY

Key takeaways:

- Apply Secure Engineering principles to the design of connected devices and the environments in which they operate.
- Defense in depth – have multiple layers of defense in the solution.
- Devices are "in the wild" and now part of the attack surface.
- Devices that were isolated before are now connected, which considerably broadens the potential significance of any security breach.
- Fail-safe modes of operation must be assured for devices, even if they become isolated from communication with other parts of the environment.

IBM has extensive experience on the topic of security technologies and has contributed significant technology and thought leadership to the space. IBM published its internal best practices for software assurance and cyber supply chain security in the IBM Secure Engineering Framework (SEF)[9]. IBM's SEF can be applied broadly for use, not only for software application development, but also with connected devices and IoT systems.

IoT devices must be designed to be securable, and must be enabled to be secure by default. Security starts at the design stage of the device, with an analysis of the potential attack surfaces for the device. Threat modeling, and identifying which threats are mitigated and how they are mitigated are also part of the design process.

The expected and required operating conditions for the device must be considered from the perspective of the communications and operating characteristics. For example, if the emitted electro-magnetic field (EMF) radiation from the device's processor might be used to infer the computations that are being performed, this could inform an attacker of the security processing that is used. This external operating characteristic can become an attack point to consider. Eliminating this type of potential attack might require special packaging be accounted for in the design of the device. Alternatively, the allowable operating conditions might be defined to specify that the device must be physically secured so that an EMF sensor cannot be physically located near the device.

Secure communications capabilities must be incorporated into IoT devices. Existing, tested, analyzed, and updated secure communications protocols, such as SSL/TLS and Diffie-Hellman key exchange, can be reused. Kerberos, known symmetric and public/private key cryptographic algorithms, and secure hashing algorithms, can also be reused. Teams must be sure to use secure communications protocols that are remediated against known vulnerabilities, such as Poodle, Heartbleed and FREAK, and apply changes that are made to these implementations in a timely manner.

As the capabilities of devices increase and the information that these devices generate, transmit, receive, process, and consume also increases, the importance of having secure processing capabilities embedded in the devices also increases. There is a need for devices to be able to prove their unique identity and use that identity in setting up secure communications with partners, whether those are peer devices or services running elsewhere in the environment.

IBM is working with device and processor manufacturers on methods for securing the lifecycle of IoT devices. The lifecycle starts with the insertion of cryptographic material into the processors that are used in IoT devices and identification of the processors into a secure registry at fabrication time. The lifecycle continues with registration of these devices in a secure registry during processor installation into IoT devices at device manufacturing time. And the lifecycle continues further with the activation of these devices at deployment time by users. The secure registry allows for the removal and retirement of devices, when devices leave the active state or are taken out of service. The secure registry service exposes appropriate and secure programming interfaces (APIs) to allow for the secure interaction with the registry by processor fabricators, device manufacturers, and users.

IoT development teams must embrace good security coding guidelines to ensure that an environment that is easy to penetrate is not introduced. Many sources of secure coding guidelines are available. Tools like IBM Security AppScan[10] to verify and enforce them are also available.

Teams can include a security viewpoint in system design models, and use threat modeling[11] to anticipate potential threat vectors and to design in protections and mitigations. Consider using a system modeling tool with a security viewpoint and threat modeling

profile, such as IBM Rational® Rhapsody®[12], a UML/SysML design tool.

Avoid making assumptions on data that is passed through APIs; check all data. A common security vulnerability is invalid assumptions that are made at component interfaces. Two common examples of the risks of not checking data at component interfaces are buffer overflow attacks and SQL injection attacks. Both are avoidable with proper input parameter checking, either for appropriate bounds or for content within parameters.

In today's rapidly paced environment, many development teams turn to open source components to reuse existing implementations to expedite their core functional deliverables. Open source is a boon to rapid development, but it also provides a fertile breeding ground for vulnerabilities. There are many documented instances of security vulnerabilities that are found on IoT devices. These vulnerabilities are due to using open source components with known vulnerabilities, such as Heartbleed/OpenSSL. Because these vulnerabilities are documented and widely deployed, hackers can easily target them on the devices. Organizations should rigorously track all open source component and version dependencies. There are databases of known vulnerabilities that are published and updated regularly, such as IBM's X-Force Vulnerability Research Database[13] and the National Vulnerability Database[14]. Having a means to manage and update devices when security exposures are detected is a critical element of any IoT architecture. The vulnerabilities in systems stem from the struggle to update the systems as these vulnerabilities are uncovered. The open source components, in general, are responsive to making fixes available. These fixes, even when available, must be distributed to all of the locations where open source has been deployed..

In many environments, practicing defensive coding and doing threat modeling are not enough. A careful record of all of the changes that are made to the system must be maintained to enable an audit of the environment after a breach. In some industries and organizations, keeping change records might be mandated. Use of the right advanced software change and configuration management environment or application lifecycle management (ALM) tools can help support traceability and audibility when responding to a failure or a breach. IBM Rational Team Concert[15] provides a sophisticated model for tracking change sets that enables auditing and tracing changes to a fine-grained level.

A part of designing for security is making sure that security policy requirements are defined, considered, and addressed. This design includes defining appropriate operating environments/conditions for the devices and the overall IoT system. The design also includes defining necessary enforcement mechanisms and checks to see that the required conditions are in effect.

Special considerations must be made in device design for fail-safe operating modes. A connected device must be able to continue safe operation, even when the device determines that either it, or the network it is communicating over, or the other devices and systems it is communicating with, might be compromised. This need to be able to continue to operate in a safe manner is one of the biggest differences in designing security into IoT systems. There is a significant difference between a mobile device user who is unable to check the weather or stock prices as opposed to an industrial water pump that is unable to evaluate current conditions to determine what speed to run to protect the lives of those living downstream.

Designing for security must also consider both information technology (IT) and operations technology (OT) elements. Part of the IoT system operates in relatively controlled conditions while a large part of the environment operates in less controlled environments, susceptible to extreme weather conditions and vulnerable to attacks. This condition suggests that the reset/update of devices depends more heavily on unattended over-the-air processing since on-site adjustment is likely to be difficult at best.

It should now be obvious what a constant fight it is to keep a system secure. In the Internet of Things, we naturally tend to focus on the vulnerability of and the need to protect the devices. But we cannot forget the bigger picture - devices are often part of larger "systems of systems". IoT devices are especially vulnerable because they are often accessible outside of physically controlled environments. Even with the best protections, it is not possible to ensure that whole fleets of devices remain resilient and uncompromised over time. There is not a 100% guarantee of security at an individual device level. Because of the nonzero likelihood of the eventual compromise of one or more devices, the designer of an IoT system must assume that devices can be compromised. The system must continue to function with one or more compromised devices, while trying to isolate and remove the vulnerability created by the compromised devices. For example, a device might be compromised by being physically removed (stolen or broken in to) and then reverse engineered. Such attacks can be made expensive by using hardware-based cryptography and

burned-in digital certificates/keys. However, these attacks cannot be ruled out. Therefore, in designing and testing the system, you must consider and run scenarios in which devices get compromised. This process ensures that the system is able to identify, isolate, and report the compromised devices while the rest of the system remains operational.

DESIGN FOR PRIVACY

Key takeaways:

- Employ data separation, segregation, redaction, and data transform techniques to remove personally identifiable information.
- Unique device identifiers can be considered personally identifiable in some situations.
- Use ephemeral and separate identifiers in communications and data storage. Isolate associations with unique device identifiers and with unique personal information.

The data that flows to and from things and that might be stored on things or their controlling devices is often sensitive. Drivers might connect their mobile phones to the in-vehicle infotainment system, which has access to their contact information, email, and text messages. With financial applications on mobile phones, credit card information might be accessible to the vehicle. Credentials to access home automation and industrial control systems can also be exposed if not properly protected.

Information that is collected from devices might be used to identify who or what was in which location, at which time, and doing what operation/task/act. This level of detail about what is going on in the world is new. It raises valid concerns about how such data is handled, who has access to this data, and what people and organizations might be allowed to do with this data.

The computing industry has been dealing with Personally Identifiable Information (PII) in the medical and financial records space for many years. Data privacy is not new, but what is new is the volume of information and the detail that the volume of information provides. IBM InfoSphere Guardium[16] and IBM InfoSphere Optim[17] solutions have specific capabilities for handling data privacy. These tools provide centralized controls for real-time data security and monitoring, fine-grained database auditing, automated compliance reporting, data-level access control, database vulnerability management, auto-discovery of sensitive data and static and dynamic masking on demand.

Organizations that are building IoT solutions need to consider data privacy as they build their solutions. From the data models that are used to store information to the external interfaces that are exposed to partners, users, and consumers, the questions of what data, in what format, with what granularity, must always be asked and privacy-sensitive answers provided.

As information flows from device to data collection system, the information must be protected. Within the data-center, PII should be separated from other data elements so that the information does not pervade the entire environment. Privacy and separation of concerns for information has been considered and addressed in the past. Consider the use of techniques, such as Multi-level Security (MLS), to prevent unauthorized access to information or to inferred knowledge from correlations among individually non-identifying data.

Special consideration must be given to the potential for using information from multiple data sets such that PII might be inferred, even if it is not stored. These issues have been considered and addressed in security work done in the medical and financial records management areas and can be applied in IoT systems also.

The Internet of Things is also opening up new business models in brokering or providing access to the information that is collected from sensors and devices. The data is often made available by surfacing a programming interface, such as a RESTful services interface. The programming interface defines the data elements that are provided as parameters and the output data that is returned. Each interface must be evaluated for its potential disclosure of PII. For example, a query for the number of users with wearable devices to track their fitness routines might inadvertently disclose the names of the devices. This disclosure might, because of a naming convention, be correlated back to a user name, such as "Jane Doe's Fitbit®". In these cases, special safeguards must be in place to prevent inadvertent disclosure. Other examples of safeguards include returning only aggregated information (averages and variances over a sufficiently large sample set), and transforming or genericizing information that would uniquely identify a person or device.

Another consideration that is related to data privacy is data retention policy. With an increasing amount of information that is collected, the potential increases that some of this information

might be used at some point in the future. One way to avoid this situation is to have good data retention and disposal policies that include the active removal and deletion of information that is no longer needed. There are good reasons to actively delete information as soon as it is legally acceptable to do so.

TEST FOR SECURITY

Key takeaways:

- Security testing techniques apply to devices as they apply to any other software systems.
- Code analysis, ethical hacking, and other techniques apply to devices and device-side code.
- Hostile environment testing extends beyond physical hostile conditions to include communications and networking hostile conditions.
- If the code is correct, as validated by testing, the attack surface shrinks.

Testing for security vulnerabilities must be an integral and organic part of any IoT implementation. Security testing techniques that are commonly used for software systems apply to IoT devices and infrastructure.

All IoT projects must undergo a range of tests to verify functional operation in accordance with design specifications. These tests include verification of the security mechanisms and services that are incorporated into sensor devices, and the infrastructure that is communicating with those devices.

Several phases of testing can be conducted, including:

- The Unit Test verifies that a component of the solution performs as designed in isolation.
- The Function Verification Test verifies that a composite solution operates in accordance with written specifications.
- The System Verification Test verifies the integration and operation of components within the full solution environment.

Security testing can be done during all test phases. Security testing can include automated testing tools, such as IBM Rational® Software Analyzer[18] and IBM Security AppScan[19]. Security testing that uses ethical hacking techniques can also be used. A wide range of testing techniques are available to validate whether a system is secure or not. It is important to repeatedly test for resistance to attacks since new attacks are developed even after a product or solution is created and released. In addition to testing in development and quality assurance phases, testing of IoT systems in production settings is recommended. Just as devices are subjected to extremes of physical operating conditions, so do these devices need to be subjected to extremes of computational conditions. These conditions include resistance to denial-of-service and jamming-style attacks where a flood of information is sent to a device in an attempt to confuse, overpower, or disable the device.

Where appropriate, solutions might undergo outside analysis and testing, including certification as specified by the Common Criteria[20]. Resources are available from IBM X-Force to help with penetration testing. IBM X-Force researches and monitors the latest Internet threat trends, develops security content for IBM customers, and helps by advising customers and the general public on how to respond to emerging and critical threats.

CONTINUOUS DELIVERY MODEL

Key takeaways:

- Problems and vulnerabilities will be detected after the devices are manufactured, delivered, and deployed.
- In-service updates to device-side code will be necessary.
- Plan for and utilize continuous delivery techniques for device-side code.
- Special considerations are necessary for determining when to apply/enact/enable updates.

Agile and Development Operations (DevOps) methods are popular in the software industry, and for good reason. These methods allow for early delivery of useful capabilities to customers, enable speedier feedback from users, and allow for faster adjustments and updates to those capabilities. Software product delivery becomes more of a continual flow of deliveries rather than infrequent product releases, migration planning, and cutovers. With the advent of services-based environments where products and features are delivered as services, frequent updates to the offering that is "in production" is the new norm.

There are pros and cons to such environments. However, making update and delivery of functions continual, and, eventually, a "non-event", enables a deployment and delivery path to assist with reacting to security-related issues found in products that are delivered.

All forms of security attention are important: prevention, detection, reaction, and treatment. Continuous delivery makes reaction and

treatment much easier than mechanisms used in the past, such as product releases, patches, and fix packs. Many of the same techniques that are used for agile and DevOps methods in software development can be applied to IoT systems development and delivery. One important difference is that the code that is running and that needs to be updated is not in a controlled data center or server environment. Rather, the code is running in the field, in routers, gateways, sensors, and other devices. These devices are moving or fixed, always connected or occasionally connected, and have varying degrees of storage and computing capability. But it is still code that is running in those devices, and it will contain issues or vulnerabilities that will be discovered after the products are deployed into the field. Since there is no chance of 100% prevention, there will be a need to perform over-the-air (OTA) updates of the systems to address the issues that are discovered.

The sooner organizations can build a continuous delivery model for the code that is running on their devices, the sooner they can enable earlier and faster deliveries of capability to their customers, with frequent updates and feature additions. This paradigm requires validating the updates that are received over-the-air, including the use of code signing and validation techniques. These technologies are not new, but they need to be applied to the space of systems/device development and deployment.

There are unique challenges to supporting over-the-air updates for devices in the field. In particular, the devices must remain operational while an update is applied. Either that or the devices must have sufficient logic to delay processing/applying an update until the devices are in a location, time, and environment that is appropriate to apply the update. The devices must have a fail-safe fallback mechanism, including checks on the running system to back off a change that is found to be acting inconsistently.

Many devices include open source software as part of the code that is running on the device. Device manufacturers should maintain lists of open source components that are used, so that when a vulnerability is identified in one of those components, an update can be made available quickly to the device owners/operators. Also, device manufacturers must ensure that communication procedures are established with the device owner/operators to allow for rapid response should any vulnerabilities be uncovered. There are existing avenues for publishing and responding to these vulnerabilities that include US-CERT[21] and the Common Vulnerability and Exposures[22] formats.

ENSURE INTEGRITY IN MANUFACTURING AND DELIVERY

Key takeaways:

- Device delivery encompasses a complete supply chain.
- Follow the existing guidance for securing the supply chain for device manufacturing.

A trusted supply chain must include focus on effective management of design, manufacturing, transportation, fulfillment, import and export, intellectual property management, support, and maintenance. IBM leads the global focus on supply chain security and is a founding member of the Electronic Industry Supplier Code of Conduct. IBM contributed to an Open Group[23] standard on Supply Chain security.

A trusted supply chain should ensure that suppliers use the following guidelines:

- Commit to defined supplier conduct and security principles.
- Submit to periodic assessments.
- Commit to remediation actions if found to be out of compliance.
- Ensure the robustness, stability, performance, and security of components.
- Ensure that software and firmware development libraries and documentation have proper access controls.
- Provide certification of originality by documenting the source of all delivered components.

An important element of the supplier assessment process is a security risk assessment. The intent of the security risk assessment is to identify all components that make up the overall supplier risk – offering, process, and business risks. Risk characteristics are identified to help assess the security risk level. Mitigation strategies can be addressed as part of the assessment process.

Securing manufacturing and delivery is about securing processes, procedures, and supply chains. Secure manufacturing is also about the physical security of the production environment where devices and systems are produced. Be sure to secure the production environment of these systems. Vulnerabilities might be injected into IoT devices due to an infected or compromised assembly/manufacturing line. There have been examples of electronics equipment with embedded vulnerabilities. These vulnerabilities were inserted during manufacturing because some of the manufacturing systems had, themselves, become infected or

compromised.

IBM Global Business Services can help optimize, audit, and secure supply chains in many industries.

OPERATORS OF THINGS - OPERATE SECURELY

HARDEN THE DEVICE

Key takeaways:

- Defense in depth - have multiple layers of defense in the solution.
- Enable a means of isolating compromised subsystems so that the overall solution remains available.

Device development, test, and manufacturing teams can do everything they can to prevent problems with their devices, but history shows us that no matter how much is done in prevention, there is always a chance of a vulnerability being uncovered and an attack being launched. One successful method of defending against attacks is to employ defense in depth techniques. Whether these techniques are firewalls in data centers or packet filtering inside home routers, they represent forms of defense in depth. Having multiple layers to get through adds to that depth, and can also be used to isolate compromised devices or systems.

To help harden the device, or more appropriately, the environment in which the device is running, use gateways and routers to isolate potentially vulnerable devices from other parts of the network. Each of these routers and gateways can be used to isolate one side from the other. For example, for a compromised device on the far side of the gateway, the gateway might be used to mute the information/data/noise coming from that device. A gateway or router can also be useful in blocking the majority of potential network communications with devices that are running behind that gateway or router.

The gateways or routers are also attack points in these environments. They are subject to the same hardening, continuous delivery, and over-the-air update requirements as the devices they are protecting and through which they are channeling information.

Gateways and routers might also serve as monitor points in the network, using sensor data feed, to determine the health of communications between devices and services-based applications.

It is useful to be able to set and update a policy that governs access rights to the device along with defining appropriate and inappropriate access (inbound or outbound). Consider adopting an endpoint management solution that allows control over security policy on devices, such as IBM's Unified Endpoint Management[24]. Endpoint management systems might not work on small or low-power embedded devices, so move the endpoint management as far through the system as possible. Ensure that you at least have endpoint management on the gateways.

SECURE THE COMMUNICATION CHANNELS

Key takeaways:

- Communication paths between devices and systems must be secured.
- Network types and connections might not be trusted.
- Follow the established guidance for each protocol that is used.
- IP communications are typically protected using SSL/TLS.

In IoT systems, a wide range of networking communication mechanisms are used. The mechanisms include local area networking using low-power, low-range methods, such as Bluetooth, Bluetooth Low Energy (BTLE), 6LoPAN, Zigbee, and others. The mechanisms also include local area networking using WiFi, to wide area networking using 2G, 3G, and 4GLTE.

The level of protection that is provided by the different networking models varies widely, as does the security of the networks that a device may encounter as it moves through the physical environment. IoT systems must still be able to set up secure communications through this wide variety of networking mechanisms.

Communications in IoT systems generally comes down to using either HTTP-based (REST-style calls) communications over TCP network connections or some form of event-based communications, also over an IP networking stack. Event-based models include DDS, CoAP, and MQTT formats. Event-based communications models typically use a UDP model rather than TCP to reduce networking-induced connection or data transmission latency.

In both HTTP-based and event-based models, the use of SSL/TLS to set up protected communications is pervasive. This model uses a combination of cryptographic algorithms to establish a secure

communications channel. It allows most of the logic that is running on devices, in gateways, and in cloud-hosted systems to assume a secure communications channel, and to focus on providing the capability of the device or application.

AUDIT AND ANALYZE USAGE PATTERNS

Key takeaways:

- Prevention will not address all issues.
- Detection is needed so that reaction and treatment can be accomplished.
- Use existing log analysis techniques to identify and respond to anomalies.

In the computing industry, it is impossible to predict or even prevent all possible attacks on a system. The need to detect, react, and treat situations that arise becomes as important as designing, implementing, and deploying systems with security in mind. Here again, there is much existing capability in the computing industry to bring to the IoT environment.

Managing computing environments requires monitoring system behavior, detecting situations that require attention, and reacting to those situations. There are both near and real-time reactions, and longer-term analysis and reporting to consider. There is the need to detect active attacks and respond to those attacks. These situations could arise from rogue devices, denial-of-service attacks from outsiders, or a persistent attack on a device or a set of devices that are running in the environment. By actively monitoring the usage patterns of the system, anomalies in behavior can be detected and appropriate responses can be enacted. Having monitoring tools in place to watch the system is only as effective as the active usage of those monitoring tools. You must watch and react to situations, not just monitor and log events. The capabilities in tools, such as IBM Security QRadar® SIEM[25] (Security Information and Event Management), can provide this type of audit and analysis.

There are potential threats that may take place over an extended period. In these cases, the system's behavior must be observed to understand the common or expected behavior patterns. The system must be actively monitored to determine whether some set of events is anomalous. The use of anomaly detection techniques can be used to identify potentially compromised devices if those devices operate/act/emit information that is inconsistent with their normal operating behavior. Tools, such as IBM Operations™ Analytics - Log Analysis[26], contain capabilities that are needed to monitor environments over an extended period to determine whether the system is acting within or outside of expected behaviors.

It is important to actively monitor events that are happening in the system, either flowing through gateways, on devices, or in cloud and data-center-hosted computing services. Policies also need to be in place to audit the operation of the overall system. This auditing is a form of "watching the watchers", and provides protection against insider attacks. By having an active audit process, attacking or subverting the system requires some level of collusion between multiple parties to carry out an attack. The level of protection can be raised by increasing the number of audits or levels of auditing built into the system. Systems can routinely log all access attempts. These logs should be maintained for a reasonable period in case forensics are required later to understand the extent of an attack and potential breach.

MAINTAIN AN UP-TO-DATE SECURITY ENVIRONMENT

Key takeaways:

- There are several aspects to cover with security: authentication, authorization, auditing, administration, encryption/decryption, key management, and integrity checking.
- A combination of technologies and processes ensure that the environment remains secure.
- Devices operate in much less controlled conditions than systems running in a data center, cloud, or other controlled environment.

Creating and maintaining a secure environment for IoT applications is related to and dependent on a secure computing environment for all of an organization's computing systems. Elements of authentication, authorization (access control), auditing, and administration apply. An added challenge is that the number of devices is orders of magnitude higher than in the past when working with users and groups, mobile devices, and endpoints. Endpoints are associated with humans that are working in the organization. In the world of IoT, there are many endpoints to consider, with a wide range of capabilities for security support.

In addition to user and device registration, authentication, and access control, there is also the need to manage keys that are used to set up encryption and decryption mechanisms for

authentication, communications, and data storage. Key management, extending all the way to IoT devices, ensures that information can flow in a protected manner from source to destination, and is held securely in between and at rest. The use of a burned-in private key data in a Trusted Platform Module (TPM) within an IoT device can help in performing key management. Specifications for TPM devices have been developed and enhanced by the Trusted Computing Group (TCG)[27].

The capabilities of IBM Identity Management[28] solutions, including the use of IBM Identity when using IBM Bluemix[29], help to maintain an up-to-date security environment for user/group definitions that are associated with the IoT solutions being developed. Additional capabilities in IBM IoT Foundation[30] to support device registration and lifecycle point out some of the base capabilities that are required to maintain a secure environment for devices and applications using those devices to communicate with one another. This includes the secure device registry capabilities that been described previously.

The capabilities of IBM Security Key Lifecycle Manager[31] provide insight into a core set of functions that are necessary for dealing with encryption key management and distribution. While this offering has been used primarily in financial services environments, the capabilities are based on working key management into device-level encryption services. Based on the OASIS Key Management Interoperability Protocol (KMIP)[32], key management can extend to a wide range of devices that are running in a distributed network environment. The first devices to work in this manner were IBM encrypting tape drives. The KMIP protocol was designed to be as lightweight as possible. The KMIP protocol allows for implementations and provides support in a wide range of networking and computing devices.

In addition to managing identity and cryptographic key information, all devices in the environment must be managed and maintained. The devices, gateways, routers, and other infrastructure must be regularly updated to apply all security patches and fixes. In the past, a considerable amount of human intervention was required to interact with the device, gateway, or router directly to perform a firmware or software upgrade, fix, or migration. In the future, due to the increased numbers of devices and the expected frequency of updates, this work will transition from active participation by humans to automated over-the-air update processing. Human intervention will be isolated to exception processing rather than handling and processing each update as it arrives. This suggests an increased level of monitoring and reporting on the status and progress of update processing across the inventory of gateways, routers, and devices involved.

As the price-point of connected devices changes over time, not every device will be worth the cost of keeping it up-to-date and protected from all attacks. Rather, the most cost-effective management approach in this situation is to remove the device from consideration altogether, and configure gateways to ignore/filter out information that is coming from the device. Further, device manufacturers should consider including a "kill-switch" type of capability. The device will still operate in a fail-safe or minimally operational disconnected mode, but will no longer communicate with the network – protecting both itself and the rest of the environment. While there would be a loss of sensor information coming from the device, the environment would be protected from potential attack through a device that could not be patched/fixed to eliminate a vulnerability.

As described earlier in this paper, part of managing an up-to-date security environment will require device manufacturers to participate and respond to security incident reports, just like other network-connected computing devices.

And as with any other part of the computing environment, active management and freshness of login/password information must be considered and addressed across the environment. This consideration includes the active management of this type of information that might be embedded in IoT devices. These aspects are covered by key lifecycle management, identity management, over-the-air updates, device registration, and lifecycle management. There are multiple stakeholders to consider with connected devices: device manufacturers, device purchasers/owners/managers, and device users. Potential third parties, such as network communications providers, and device service and support contractors, are also stakeholders to consider.

CREATE A TRUSTED MAINTENANCE ECOSYSTEM

Key takeaways:

- Follow the existing guidance on setting up and maintaining a secure environment.
- Develop a comprehensive incident response process.

Operating a secure IoT system requires that the people responsible for running the environment are considered for their secure and appropriate activity. For more information, see the section on securing the supply chain, including contractors who are used in operating the environment. Proper maintenance procedures must be followed so that the security and integrity of the systems is maintained.

There must be a clearly defined and communicated incident response process for handling security-related incident reports. This process must ensure that when an incident is discovered and validated, that there are structured remediation plans to close the vulnerability. The process must also ensure that other component owners that might be affected by the vulnerability are informed so that they can execute a remediation plan. With complex combinations of component reuse and solution construction, incident response processes must ensure that all potentially affected components are identified and remediated quickly.

Physical security of the IoT system overall will continue to be a challenge. By their very nature, IoT devices must operate in demanding and difficult operating conditions, exposed to physical elements, and open to a wide range of attacks. These devices will operate "in the wild", will move around rapidly, and will be subjected to extreme conditions. While this is new territory for large-scale deployment of high-tech electronics, it is not new. Military applications, in-vehicle systems, aircraft avionics, and sensors, along with everything established so far in securing mobile devices has led the way in identifying appropriate practices to follow.

SUMMARY

Security for Internet of Things technologies is at the same time different and the same as security for other large scale computing infrastructures. There are similar problems to solve and techniques to solve them – authentication (device, system/application, and user), authorization, auditing, administration, encryption/decryption, data integrity, and key management. There are new challenges – a wider range of computing device types and capabilities, operating in a less controlled, global environment, and with an expanded set of attack surfaces to secure.

There are challenges to be addressed with IoT security. However, the techniques and technologies that have been honed over many years of research and development can be applied to meet these challenges, extending them as necessary to cover the unique requirements of the Internet of Things.

CONTRIBUTORS

Producing the IBM Point of View: Internet of Things (IoT) Security was a dedicated collaboration across all of IBM. We would like to thank the following individuals for their attention and contribution to the publication of this point of view.

Timothy Hahn	Distinguished Engineer, IBM Analytics - IoT
Sky Matthews	CTO, IBM Analytics - IoT
Lisa Wood	Director, IBM Analytics - IoT
John Cohn	Fellow, IBM Corporate Technical Strategy
Shmulik Regev	Senior Technical Staff Member, IBM Security
Jim Fletcher	Distinguished Engineer, IBM Analytics - IoT
Eric Libow	Distinguished Engineer, IBM Analytics - IoT
Chris Poulin	Research Strategist, IBM X-Force
Katsumi Ohnishi	Distinguished Engineer, IBM Security

FOR MORE INFORMATION

To learn more about IBM Internet of Things, please visit: http://www.ibm.com/software/info/internet-of-things/

REFERENCES

[1] IDC, "Worldwide and Regional Internet of Things 2014-2020 Forecast Update by Technology Split," Doc #252330, Publish date: Nov 2014. http://www.idc.com/getdoc.jsp?containerId=252330

[2] Storm, Darlene, " Hackers exploit SCADA holes to take full control of critical infrastructure," Publish date: Jan 2014. Computerworld. http://www.computerworld.com/article/2475789/cybercrime-hacking/hackers-exploit-scada-holes-to-take-full-control-of-critical-infrastructure.html

[3] IBM IBV Driving Security . http://www-935.ibm.com/services/us/gbs/thoughtleadership/automotivesecurity/

[4] Open Web Application Security Project (OWASP) Top 10 IoT Issues. http://www.owasp.org/index.php/OWASP_Internet_of_Things_Top_Ten_Project

[5] IIC Reference Architecture. http://www.iiconsortium.org/ and IIC Security Working Group Reference Guide – http://www.iiconsortium.org/wc-security.htm

[6] Allseen Alliance. https://allseenalliance.org/

[7] BuildItSecure.Ly. http://builditsecure.ly

[8] X-Force Research and Development. " IBM X-Force Threat Intelligence Quarterly 4Q 2014," Doc # WGL03062USEN, Publish Date: Nov 2014. http://www.ibm.com/security/xforce/downloads.html

[9] IBM Secure Engineering Framework. http://www.redbooks.ibm.com/abstracts/redp4641.html

[10,19] IBM Security AppScan®. http://www.ibm.com/software/products/en/appscan-source

[11] Threat Modeling. http://en.wikipedia.org/wiki/Threat_model

[12] IBM Rational® Rhapsody®. http://www.ibm.com/software/products/en/ratirhapfami

[13] X-Force Vulnerability Research Database. https://xforce.iss.net/

[14] National Vulnerability Database. http://nvd.nist.gov/

[15] IBM Rational Team Concert. http://www.ibm.com/software/products/en/rtc

[16] IBM Infosphere Guardium Data Security. http://www.ibm.com/software/data/guardium/

[17] IBM Infosphere Optim Data Privacy. http://www.ibm.com/software/data/optim/

[18] IBM Rational® Software Analyzer. http://www.ibm.com/software/products/en/ratisoftanalfami

[20] Common Criteria. http://www.commoncriteriaportal.org

[21] US-Cert. http://www.us-cert.gov/

[22] Common Vulnerability Exposures. http://cve.mitre.org/

[23] Open Group – Supply Chain Security. http://www.opengroup.org/news/press/open-group-releases-global-technology-supply- chain-security-standard

[24] IBM Unified Endpoint Management. http://www.ibm.com/software/tivoli/unified-endpoint-management/

[25] IBM Security QRadar® SIEM. http://www.ibm.com/software/products/en/qradar-siem

[26] IBM Operations™ Analytics – Log Analysis. http://www.ibm.com/software/products/en/ibm-operations-analytics---log-analysis

[27] Trusted Computing Group. http://www.trustedcomputinggroup.org/

[28] IBM Identity and Access Manage. http://www.ibm.com/software/products/en/identity-access-manager

[29] IBM Bluemix. http://www.bluemix.net

[30] IBM IoT Foundation. http://internetofthings.ibmcloud.com

[31] IBM Security Key Lifecycle Manager. http://www.ibm.com/software/products/en/key-lifecycle-manager

[32] OASIS Key Management Interoperability Protocal (KMIP). http://www.oasis-open.org/committees/kmip/

IBM Corporation Software
Group Route 100
Somers, NY 10589

Produced in the United States of America
April 2015

Please Recycle

RAW14382-USEN-00