




Internet of Things security architecture

 24 minutes to read Contributors  

In this article

<https://docs.microsoft.com/en-us/azure/iot-fundamentals/iot-security-architecture>

When designing a system, it is important to understand the potential threats to that system, and add appropriate defenses accordingly, as the system is designed and architected. It is important to design the product from the start with security in mind because understanding how an attacker might be able to compromise a system helps make sure appropriate mitigations are in place from the beginning.

Security starts with a threat model

Microsoft has long used threat models for its products and has made the company's threat modeling process publically available. The company experience demonstrates that the modeling has unexpected benefits beyond the immediate understanding of what threats are the most concerning. For example, it also creates an avenue for an open discussion with others outside the development team, which can lead to new ideas and improvements in the product.

The objective of threat modeling is to understand how an attacker might be able to compromise a system and then make sure appropriate mitigations are in place. Threat modeling forces the design team to consider mitigations as the system is designed rather than after a system is deployed. This fact is critically important, because retrofitting security defenses to a myriad of devices in the field is infeasible, error prone and leaves customers at risk.

Many development teams do an excellent job capturing the functional requirements for the system that benefit customers. However, identifying non-obvious ways that someone might misuse the system is more challenging. Threat modeling can help development teams understand what an attacker might do and why. Threat modeling is a structured process that creates a discussion about the security design decisions in the system, as well as changes to the design that are made along the way that impact security. While a threat model is simply a document, this documentation also represents an ideal way to ensure continuity of knowledge, retention of lessons learned, and help new team onboard rapidly. Finally, an outcome of threat modeling is to enable you to consider other aspects of security, such as what security commitments you wish to provide to your customers. These commitments in conjunction with threat modeling inform and drive testing of your Internet of Things (IoT) solution.

When to threat model

[Threat modeling](#) offers the greatest value when you incorporate it into the design phase. When you are designing, you have the greatest flexibility to make changes to eliminate threats. Eliminating threats by design is the desired outcome. It is much easier than adding mitigations, testing them, and ensuring they remain current and moreover, such elimination is not always possible. It becomes harder to eliminate threats as a product becomes more mature, and in turn ultimately requires more work and a lot harder tradeoffs than threat modeling early on in the development.

What to threat model

You should threat model the solution as a whole and also focus in the following areas:

- The security and privacy features
- The features whose failures are security relevant
- The features that touch a trust boundary

Who threat models

Threat modeling is a process like any other. It is a good idea to treat the threat model document like any other component of the solution and validate it. Many development teams do an excellent job capturing the functional requirements for the system that benefit customers. However, identifying non-obvious ways that someone might misuse the system is more challenging. Threat modeling can help development teams understand what an attacker might do and why.

How to threat model

The threat modeling process is composed of four steps; the steps are:

- Model the application
- Enumerate Threats
- Mitigate threats
- Validate the mitigations

The process steps

Three rules of thumb to keep in mind when building a threat model:

1. Create a diagram out of reference architecture.
2. Start breadth-first. Get an overview, and understand the system as a whole, before deep-diving. This approach helps ensure that you deep-dive in the right places.
3. Drive the process, don't let the process drive you. If you find an issue in the modeling phase and want to explore it, go for it! Don't feel you need to follow these steps slavishly.

Threats

The four core elements of a threat model are:

- Processes such as web services, Win32 services, and *nix daemons. Some complex entities (for example field gateways and sensors) can be abstracted as a process when a technical drill-down in these areas is not possible.
- Data stores (anywhere data is stored, such as a configuration file or database)
- Data flow (where data moves between other elements in the application)
- External Entities (anything that interacts with the system, but is not under the control of the application, examples include users and satellite feeds)

All elements in the architectural diagram are subject to various threats; this article the STRIDE mnemonic. Read [Threat Modeling Again, STRIDE](#) to know more about the STRIDE elements.

Different elements of the application diagram are subject to certain STRIDE threats:

- Processes are subject to STRIDE
- Data flows are subject to TID
- Data stores are subject to TID, and sometimes R, when the data stores are log files.
- External entities are subject to SRD

Security in IoT

Connected special-purpose devices have a significant number of potential interaction surface areas and interaction patterns, all of which must be considered to provide a framework for securing digital access to those devices. The term “digital access” is used here to distinguish from any operations that are carried out through direct device interaction where access security is provided through physical access control. For example, putting the device into a room with a lock on the door. While physical access cannot be denied using software and hardware, measures can be taken to prevent physical access from leading to system interference.

As you explore the interaction patterns, look at “device control” and “device data” with the same level of attention. “Device control” can be classified as any information that is provided to a device by any party with the goal of changing or influencing its behavior towards its state or the state of its environment. “Device data” can be classified as any information that a device emits to any other party about its state and the observed state of its environment.

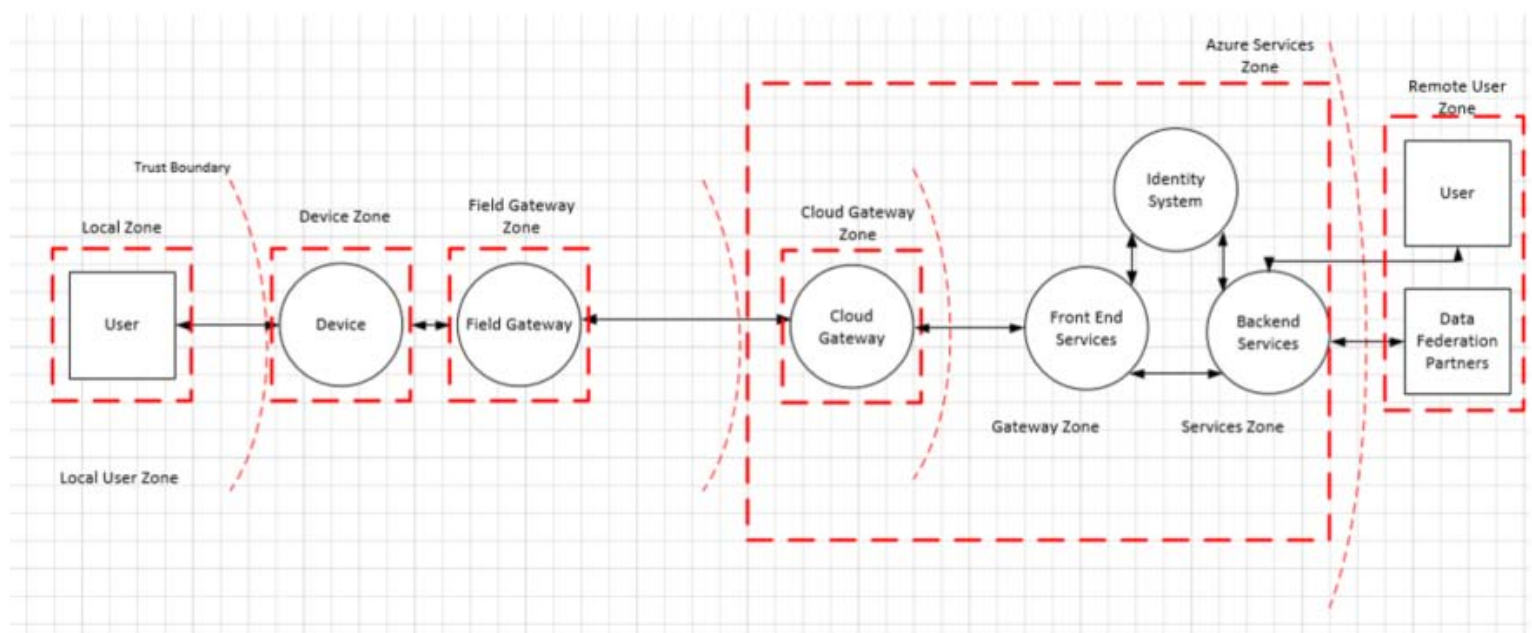
In order to optimize security best practices, it is recommended that a typical IoT architecture is divided into several component/zones as part of the threat modeling exercise. These zones are described fully throughout this section and include:

- Device,

- Field Gateway,
- Cloud gateways, and
- Services.

Zones are broad way to segment a solution; each zone often has its own data and authentication and authorization requirements. Zones can also be used to isolation damage and restrict the impact of low trust zones on higher trust zones.

Each zone is separated by a Trust Boundary, which is noted as the dotted red line in the following diagram. It represents a transition of data/information from one source to another. During this transition, the data/information could be subject to Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service and Elevation of Privilege (STRIDE).



The components depicted within each boundary are also subjected to STRIDE, enabling a full 360 threat modeling view of the solution. The following sections elaborate on each of the components and specific security concerns and solutions that should be put into place.

The following sections discuss standard components typically found in these zones.

The Device Zone

The device environment is the immediate physical space around the device where physical access and/or "local network" peer-to-peer digital access to the device is feasible. A "local network" is assumed to be a network that is distinct and insulated from – but potentially bridged to – the public Internet, and includes any short-range wireless radio technology that permits peer-to-peer communication of devices. It does *not* include any network virtualization technology creating the illusion of such a local network and it does also not include public operator networks that require any two devices to communicate across public network space if they were to enter a peer-

to-peer communication relationship.

The Field Gateway Zone

Field gateway is a device/appliance or some general-purpose server computer software that acts as communication enabler and, potentially, as a device control system and device data processing hub. The field gateway zone includes the field gateway itself and all devices that are attached to it. As the name implies, field gateways act outside dedicated data processing facilities, are usually location bound, are potentially subject to physical intrusion, and has limited operational redundancy. All to say that a field gateway is commonly a thing one can touch and sabotage while knowing what its function is.

A field gateway is different from a mere traffic router in that it has had an active role in managing access and information flow, meaning it is an application addressed entity and network connection or session terminal. An NAT device or firewall, in contrast, does not qualify as field gateways since they are not explicit connection or session terminals, but rather a route (or block) connections or sessions made through them. The field gateway has two distinct surface areas. One faces the devices that are attached to it and represents the inside of the zone, and the other faces all external parties and is the edge of the zone.

The cloud gateway zone

Cloud gateway is a system that enables remote communication from and to devices or field gateways from several different sites across public network space, typically towards a cloud-based control and data analysis system, a federation of such systems. In some cases, a cloud gateway may immediately facilitate access to special-purpose devices from terminals such as tablets or phones. In the context discussed here, "cloud" is meant to refer to a dedicated data processing system that is not bound to the same site as the attached devices or field gateways. Also in a Cloud Zone, operational measures prevent targeted physical access and are not necessarily exposed to a "public cloud" infrastructure.

A cloud gateway may potentially be mapped into a network virtualization overlay to insulate the cloud gateway and all of its attached devices or field gateways from any other network traffic. The cloud gateway itself is not a device control system or a processing or storage facility for device data; those facilities interface with the cloud gateway. The cloud gateway zone includes the cloud gateway itself along with all field gateways and devices directly or indirectly attached to it. The edge of the zone is a distinct surface area where all external parties communicate through.

The services zone

A "service" is defined for this context as any software component or module that is interfacing with devices through a field- or cloud gateway for data collection and analysis, as well as for command and control. Services are mediators. They act under their identity towards gateways and other subsystems, store and analyze data, autonomously issue commands to devices based on data insights or schedules and expose information and

control capabilities to authorized end users.

Information-devices versus special-purpose devices

PCs, phones, and tablets are primarily interactive information devices. Phones and tablets are explicitly optimized around maximizing battery lifetime. They preferably turn off partially when not immediately interacting with a person, or when not providing services like playing music or guiding their owner to a particular location. From a systems perspective, these information technology devices are mainly acting as proxies towards people. They are “people actuators” suggesting actions and “people sensors” collecting input.

Special-purpose devices, from simple temperature sensors to complex factory production lines with thousands of components inside them, are different. These devices are much more scoped in purpose and even if they provide some user interface, they are largely scoped to interfacing with or be integrated into assets in the physical world. They measure and report environmental circumstances, turn valves, control servos, sound alarms, switch lights, and do many other tasks. They help to do work for which an information device is either too generic, too expensive, too large, or too brittle. The concrete purpose immediately dictates their technical design as well the available monetary budget for their production and scheduled lifetime operation. The combination of these two key factors constrains the available operational energy budget, physical footprint, and thus available storage, compute, and security capabilities.

If something “goes wrong” with automated or remote controllable devices, for example, physical defects or control logic defects to willful unauthorized intrusion and manipulation. The production lots may be destroyed, buildings may be looted or burned down, and people may be injured or even die. This is a whole different class of damage than someone maxing out a stolen credit card's limit. The security bar for devices that make things move, and also for sensor data that eventually results in commands that cause things to move, must be higher than in any e-commerce or banking scenario.

Device control and device data interactions

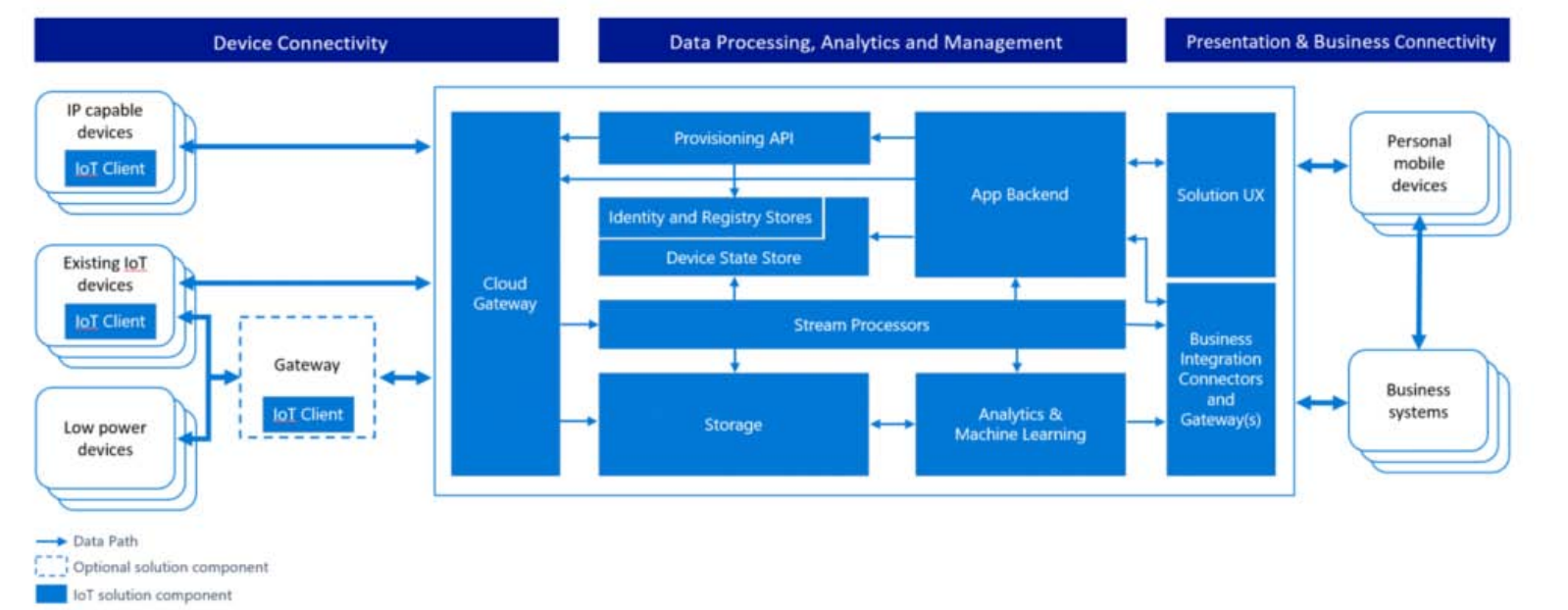
Connected special-purpose devices have a significant number of potential interaction surface areas and interaction patterns, all of which must be considered to provide a framework for securing digital access to those devices. The term “digital access” is used here to distinguish from any operations that are carried out through direct device interaction where access security is provided through physical access control. For example, putting the device into a room with a lock on the door. While physical access cannot be denied using software and hardware, measures can be taken to prevent physical access from leading to system interference.

As you explore the interaction patterns, look at “device control” and “device data” with the same level of attention while threat modeling. “Device control” can be classified as any information that is provided to a device by any party with the goal of changing or influencing its behavior towards its state or the state of its environment. “Device data” can be classified as any information that a device emits to any other party about its state and the observed state of its environment.

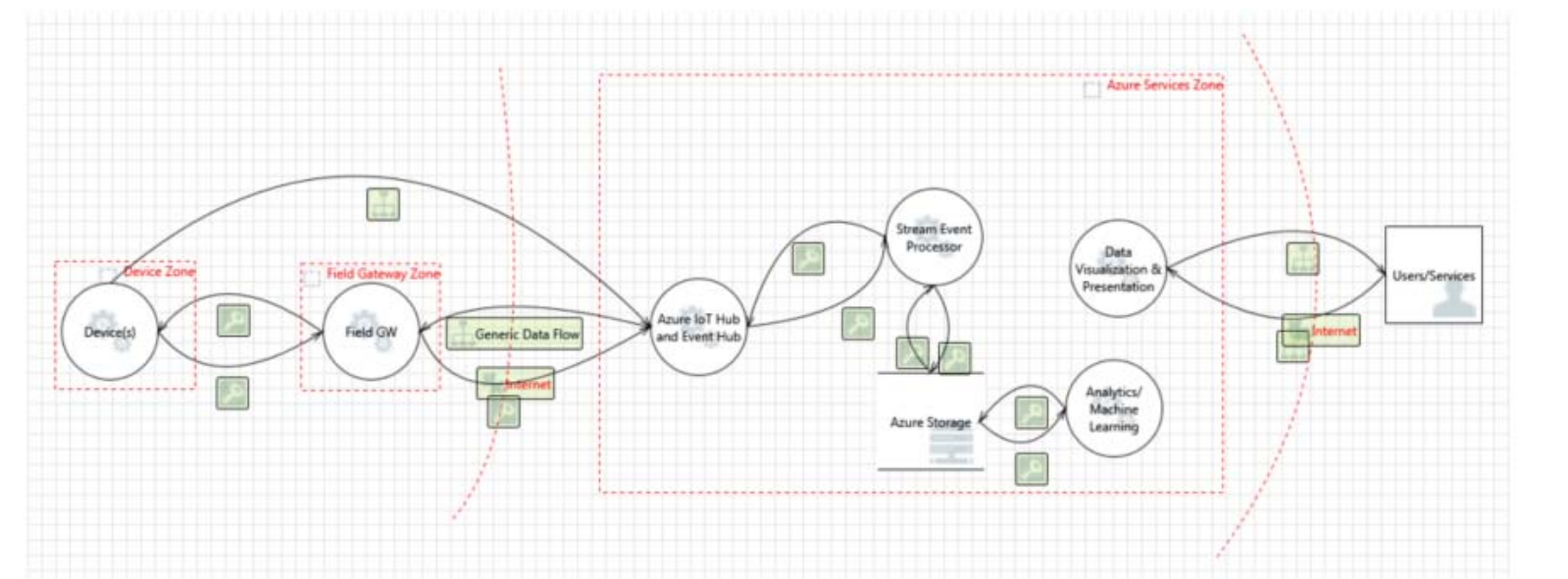
Threat modeling the Azure IoT reference architecture

Microsoft uses the framework outlined previously to do threat modeling for Azure IoT. The following section uses the concrete example of Azure IoT Reference Architecture to demonstrate how to think about threat modeling for IoT and how to address the threats identified. This example identifies four main areas of focus:

- Devices and Data Sources,
- Data Transport,
- Device and Event Processing, and
- Presentation



The following diagram provides a simplified view of Microsoft’s IoT Architecture using a Data Flow Diagram model that is used by the Microsoft Threat Modeling Tool:



It is important to note that the architecture separates the device and gateway capabilities. This approach enables the user to leverage gateway devices that are more secure: they are capable of communicating with the cloud gateway using secure protocols, which typically requires greater processing overhead than a native device - such as a thermostat - could provide on its own. In the Azure services zone, assume that the Cloud Gateway is represented by the Azure IoT Hub service.

Device and data sources/data transport

This section explores the architecture outlined previously through the lens of threat modeling and gives an overview of how to address some of the inherent concerns. This example focuses on the core elements of a threat model:

- Processes (both under your control and external items)
- Communication (also called data flows)
- Storage (also called data stores)

Processes

In each of the categories outlined in the Azure IoT architecture, this example tries to mitigate a number of different threats across the different stages data/information exists in: process, communication, and storage. Following is an overview of the most common ones for the "process" category, followed by an overview of how these threats could be best mitigated:

Spoofing (S): An attacker may extract cryptographic key material from a device, either at the software or hardware level, and subsequently access the system with a different physical or virtual device under the identity of the device the key material has been taken from. A good illustration is remote controls that can turn any TV and that are popular prankster tools.

Denial of Service (D): A device can be rendered incapable of functioning or communicating by interfering with radio frequencies or cutting wires. For example, a surveillance camera that had its power or network connection intentionally knocked out cannot report data, at all.

Tampering (T): An attacker may partially or wholly replace the software running on the device, potentially allowing the replaced software to leverage the genuine identity of the device if the key material or the cryptographic facilities holding key materials were available to the illicit program. For example, an attacker may leverage extracted key material to intercept and suppress data from the device on the communication path and replace it with false data that is authenticated with the stolen key material.

Information Disclosure (I): If the device is running manipulated software, such manipulated software could potentially leak data to unauthorized parties. For example, an attacker may leverage extracted key material to inject itself into the communication path between the device and a controller or field gateway or cloud gateway

to siphon off information.

Elevation of Privilege (E): A device that does specific function can be forced to do something else. For example, a valve that is programmed to open half way can be tricked to open all the way.

Component	Threat	Mitigation	Risk	Implementation
Device	S	Assigning identity to the device and authenticating the device	Replacing device or part of the device with some other device. How do you know you are talking to the right device?	Authenticating the device, using Transport Layer Security (TLS) or IPSec. Infrastructure should support using pre-shared key (PSK) on those devices that cannot handle full asymmetric cryptography. Leverage Azure AD, OAuth
	TRID	Apply tamperproof mechanisms to the device, for example, by making it hard to impossible to extract keys and other cryptographic material from the device.	The risk is if someone is tampering the device (physical interference). How are you sure, that device has not been tampered with.	The most effective mitigation is a trusted platform module (TPM) capability that allows storing keys in special on-chip circuitry from which the keys cannot be read, but can only be used for cryptographic operations that use the key but never disclose the key. Memory encryption of the device. Key management for the device. Signing the code.
	E	Having access control of the device. Authorization scheme.	If the device allows for individual actions to be performed based on commands from an outside source, or even compromised sensors, it allows the attack to perform operations not otherwise accessible.	Having authorization scheme for the device

Field Gateway	S	Authenticating the Field gateway to Cloud Gateway (such as cert based, PSK, or Claim based.)	If someone can spoof Field Gateway, then it can present itself as any device.	TLS RSA/PSK, IPSec, RFC 4279 . All the same key storage and attestation concerns of devices in general – best case is use TPM. 6LowPAN extension for IPSec to support Wireless Sensor Networks (WSN).
	TRID	Protect the Field Gateway against tampering (TPM?)	Spoofing attacks that trick the cloud gateway thinking it is talking to field gateway could result in information disclosure and data tampering	Memory encryption, TPM's, authentication.
	E	Access control mechanism for Field Gateway		

Here are some examples of threats in this category:

Spoofing: An attacker may extract cryptographic key material from a device, either at the software or hardware level, and subsequently access the system with a different physical or virtual device under the identity of the device the key material has been taken from.

Denial of Service: A device can be rendered incapable of functioning or communicating by interfering with radio frequencies or cutting wires. For example, a surveillance camera that had its power or network connection intentionally knocked out cannot report data, at all.

Tampering: An attacker may partially or wholly replace the software running on the device, potentially allowing the replaced software to leverage the genuine identity of the device if the key material or the cryptographic facilities holding key materials were available to the illicit program.

Tampering: A surveillance camera that's showing a visible-spectrum picture of an empty hallway could be aimed at a photograph of such a hallway. A smoke or fire sensor could be reporting someone holding a lighter under it. In either case, the device may be technically fully trustworthy towards the system, but it reports

manipulated information.

Tampering: An attacker may leverage extracted key material to intercept and suppress data from the device on the communication path and replace it with false data that is authenticated with the stolen key material.

Tampering: An attacker may partially or completely replace the software running on the device, potentially allowing the replaced software to leverage the genuine identity of the device if the key material or the cryptographic facilities holding key materials were available to the illicit program.

Information Disclosure: If the device is running manipulated software, such manipulated software could potentially leak data to unauthorized parties.

Information Disclosure: An attacker may leverage extracted key material to inject itself into the communication path between the device and a controller or field gateway or cloud gateway to siphon off information.

Denial of Service: The device can be turned off or turned into a mode where communication is not possible (which is intentional in many industrial machines).

Tampering: The device can be reconfigured to operate in a state unknown to the control system (outside of known calibration parameters) and thus provide data that can be misinterpreted

Elevation of Privilege: A device that does specific function can be forced to do something else. For example, a valve that is programmed to open half way can be tricked to open all the way.

Denial of Service: The device can be turned into a state where communication is not possible.

Tampering: The device can be reconfigured to operate in a state unknown to the control system (outside of known calibration parameters) and thus provide data that can be misinterpreted.

Spoofing/Tampering/Repudiation: If not secured (which is rarely the case with consumer remote controls), an attacker can manipulate the state of a device anonymously. A good illustration is remote controls that can turn any TV and that are popular prankster tools.

Communication

Threats around communication path between devices, devices and field gateways, and device and cloud gateway. The following table has some guidance around open sockets on the device/VPN:

Component	Threat	Mitigation	Risk	Implementation
Device IoT Hub	TID	(D)TLS (PSK/RSA) to encrypt	Eavesdropping or interfering the	Security on the protocol level. With custom protocols, you need to figure out how to protect them. In most cases, the

		the traffic	communication between the device and the gateway	communication takes place from the device to the IoT Hub (device initiates the connection).
Device Device	TID	(D)TLS (PSK/RSA) to encrypt the traffic.	Reading data in transit between devices. Tampering with the data. Overloading the device with new connections	Security on the protocol level (MQTT/AMQP/HTTP/CoAP. With custom protocols, you need to figure out how to protect them. The mitigation for the DoS threat is to peer devices through a cloud or field gateway and have them only act as clients towards the network. The peering may result in a direct connection between the peers after having been brokered by the gateway
External Entity Device	TID	Strong pairing of the external entity to the device	Eavesdropping the connection to the device. Interfering the communication with the device	Securely pairing the external entity to the device NFC/Bluetooth LE. Controlling the operational panel of the device (Physical)
Field Gateway Cloud Gateway	TID	TLS (PSK/RSA) to encrypt the traffic.	Eavesdropping or interfering the communication between the device and the gateway	Security on the protocol level (MQTT/AMQP/HTTP/CoAP). With custom protocols, you need to figure out how to protect them.
Device Cloud Gateway	TID	TLS (PSK/RSA) to encrypt the traffic.	Eavesdropping or interfering the communication between the device and the gateway	Security on the protocol level (MQTT/AMQP/HTTP/CoAP). With custom protocols, you need to figure out how to protect them.

Here are some examples of threats in this category:

Denial of Service: Constrained devices are generally under DoS threat when they actively listen for inbound connections or unsolicited datagrams on a network, because an attacker can open many connections in parallel and not service them or service them slowly, or the device can be flooded with unsolicited traffic. In both cases, the device can effectively be rendered inoperable on the network.

Spoofing, Information Disclosure: Constrained devices and special-purpose devices often have one-for-all security facilities like password or PIN protection, or they wholly rely on trusting the network, meaning they grant access to information when a device is on the same network, and that network is often only protected by a shared key. That means that when the shared secret to device or network is disclosed, it is possible to control the device or observe data emitted from the device.

Spoofing: an attacker may intercept or partially override the broadcast and spoof the originator (man in the middle)

Tampering: an attacker may intercept or partially override the broadcast and send false information

Information Disclosure: an attacker may eavesdrop on a broadcast and obtain information without authorization **Denial of Service:** an attacker may jam the broadcast signal and deny information distribution

Storage

Every device and field gateway has some form of storage (temporary for queuing the data, operating system (OS) image storage).

Component	Threat	Mitigation	Risk	Implementation
Device storage	TRID	Storage encryption, signing the logs	Reading data from the storage (PII data), tampering with telemetry data. Tampering with queued or cached command control data. Tampering with configuration or firmware update packages while cached or queued locally can lead to OS and/or system components being compromised	Encryption, message authentication code (MAC), or digital signature. Where possible, strong access control through resource access control lists (ACLs) or permissions.
Device OS image	TRID		Tampering with OS /replacing the OS components	Read-only OS partition, signed OS image, Encryption

Field Gateway storage (queuing the data)	TRID	Storage encryption, signing the logs	Reading data from the storage (PII data), tampering with telemetry data, tampering with queued or cached command control data. Tampering with configuration or firmware update packages (destined for devices or field gateway) while cached or queued locally can lead to OS and/or system components being compromised	BitLocker
Field Gateway OS image	TRID		Tampering with OS /replacing the OS components	Read-only OS partition, signed OS image, Encryption

Device and event processing/cloud gateway zone

A cloud gateway is system that enables remote communication from and to devices or field gateways from several different sites across public network space, typically towards a cloud-based control and data analysis system, a federation of such systems. In some cases, a cloud gateway may immediately facilitate access to special-purpose devices from terminals such as tablets or phones. In the context discussed here, “cloud” is meant to refer to a dedicated data processing system that is not bound to the same site as the attached devices or field gateways, and where operational measures prevent targeted physical access but is not necessarily to a “public cloud” infrastructure. A cloud gateway may potentially be mapped into a network virtualization overlay to insulate the cloud gateway and all of its attached devices or field gateways from any other network traffic. The cloud gateway itself is not a device control system or a processing or storage facility for device data; those facilities interface with the cloud gateway. The cloud gateway zone includes the cloud gateway itself along with all field gateways and devices directly or indirectly attached to it.

Cloud gateway is mostly custom built piece of software running as a service with exposed endpoints to which field gateway and devices connect. As such it must be designed with security in mind. Follow [SDL](#) process for designing and building this service.

Services zone

A control system (or controller) is a software solution that interfaces with a device, or a field gateway, or cloud gateway for the purpose of controlling one or multiple devices and/or to collect and/or store and/or analyze device data for presentation, or subsequent control purposes. Control systems are the only entities in the scope of this discussion that may immediately facilitate interaction with people. The exceptions are intermediate physical control surfaces on devices, like a switch that allows a person to turn off the device or change other

properties, and for which there is no functional equivalent that can be accessed digitally.

Intermediate physical control surfaces are those where governing logic constrains the function of the physical control surface such that an equivalent function can be initiated remotely or input conflicts with remote input can be avoided – such intermediated control surfaces are conceptually attached to a local control system that leverages the same underlying functionality as any other remote control system that the device may be attached to in parallel. Top threats to the cloud computing can be read at [Cloud Security Alliance \(CSA\)](#) page.

Additional resources

For more information, see the following articles:

- [SDL Threat Modeling Tool](#)
- [Microsoft Azure IoT reference architecture](#)

See also

To learn more about securing a solution created by an IoT solution accelerator, see [Secure your IoT deployment](#).

Read about IoT Hub security in [Control access to IoT Hub](#) in the IoT Hub developer guide.

Feedback

We'd love to hear your thoughts. Choose the type you'd like to provide:

Product feedback

☐ **Sign in to give documentation feedback**

Our new feedback system is built on GitHub Issues. Read about this change in [our blog post](#).

Loading feedback...

