PubNub®

A New Approach to IoT Security

5 Key Requirements to Securing IoT Communications

The Internet of Things promises to bring everything from microwaves to pacemakers and shipping fleets online, leverage enormous amounts of new data, and ultimately, make our world smarter, easier, and more efficient. As an estimated 50 billion new devices come online in the next 5 years, **Gartner Research lists security as the #1 challenge to making the Internet of Things a reality.**

Why?

Because in order to be useful, IoT devices must make realtime bi-directional connections to the internet, and that type of communication is challenging to secure. Whereas security protocols and best practices for servers, personal computers, and smartphones are well-understood and broadly adopted, security for IoT devices is nascent and rarely sufficient.

It's a hacker's dream come true.

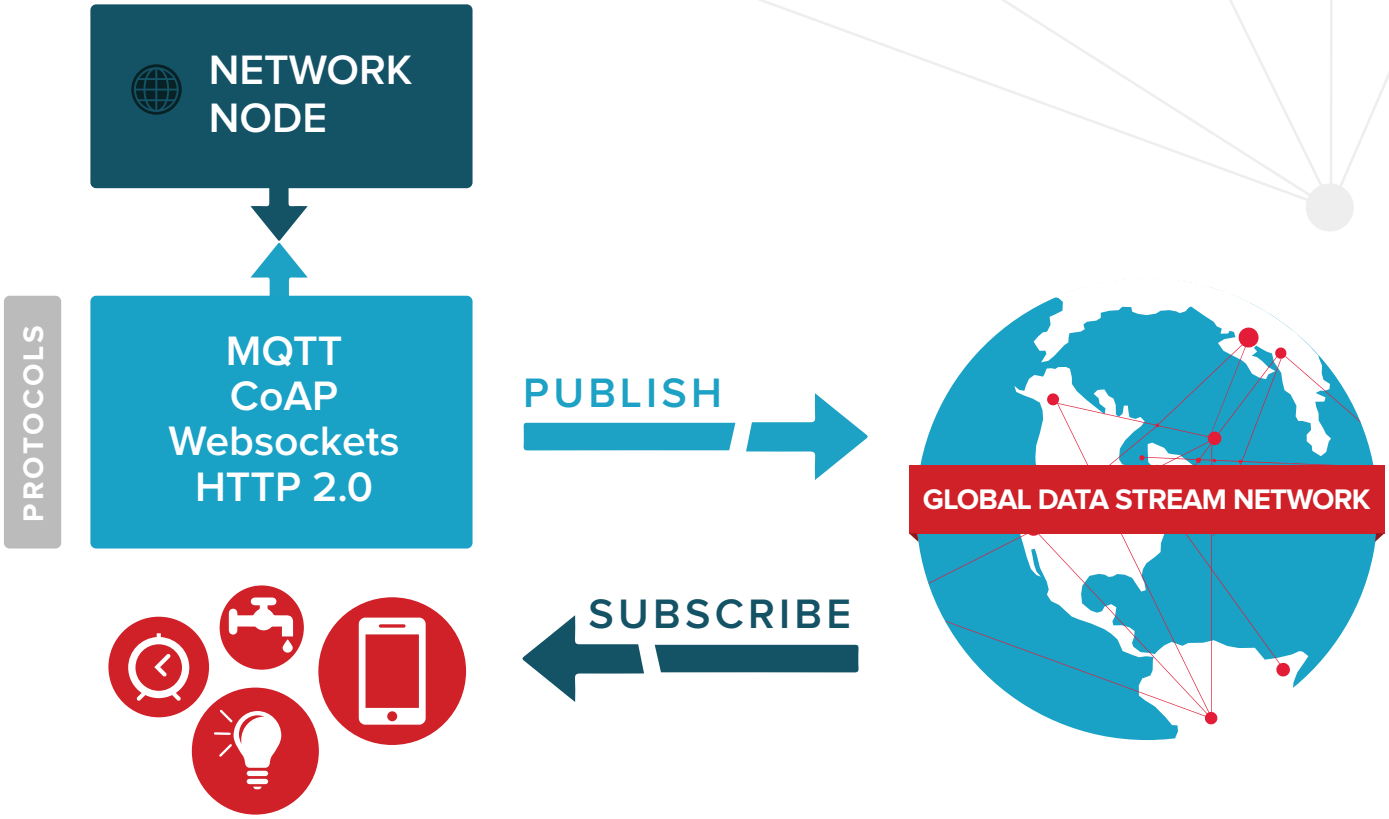
To combat this impending security crisis, we need a robust security model that works across the many different paradigms of device communication. Additionally, the security model should enable devices to be plug-and-play for end consumers -- we can assume that if any component of the security model requires consumers to set their devices up and keep their software and firmware up to date correctly, the model is seriously flawed.

"The crux concept for IoT manufacturers is this: hardening devices against intrusion is a good first step, but it is nowhere near a complete security model." The strategy that we propose in this white paper is to leverage a secure data stream network and its accompanying services to provide enterprise-level end-to-end security for IoT devices. Doing so shifts the primary burden of securing billions of new devices from hardware manufacturers into the network layer, which is far more flexible and robust for ongoing security.

With this network-first security strategy in mind, this white paper details best-practice design patterns and tactics for implementing a secure data stream network network to enable bi-directional communication for the Internet of Things. It also explains the critical security requirements of such a network, each of which plays a unique role in securing IoT applications and connected devices.

Requirement 1 Devices Must Not Have Open Inbound Ports

For one device - say, a server - to push data, another device (i.e. an IoT device) has to be listening. In a traditional model, the listening device will open an inbound port and wait for data to be pushed. While this can work in some scenarios, it is a massive risk for IoT as these ports must remain open indefinitely. The security risks of leaving inbound ports open include malware infections, modification or theft of data, DoS attacks, and arbitrary code execution.



Let's be very clear: any device on the Internet with an open inbound port will be attacked. It's a matter of when, not if.

Devices connected to a secure IoT network should make only outbound connections. These connections are not vulnerable to the kind of attacks that open inbound ports are. The outbound-only design pattern eliminates one major threat to IoT devices. To support this design pattern, we'll also need to use a publish/subscribe communication design so devices can send data bi-directionally. With the communication design pattern articulated, how do we make sure it scales to handle the

"Any device on the Internet with an open inbound port will be attacked. It's a matter of when, not if."

unprecedented amount of data those 50 billion new IoT devices will create?

Secure and reliable communication that uses protocols like MQTT, CoAP, WebSockets, and HTTP 2.0 is able to power publish/subscribe communication between devices with no open ports. Regardless of which protocol is used, opening a connection outward and leaving it open is of primary importance, followed by using publish/subscribe as the paradigm for communication for that connection. To address the needs of IoT scale, the publish/ subscribe connection should be managed by high-performance servers distributed throughout the world (a data stream network) with multiple points of presence.

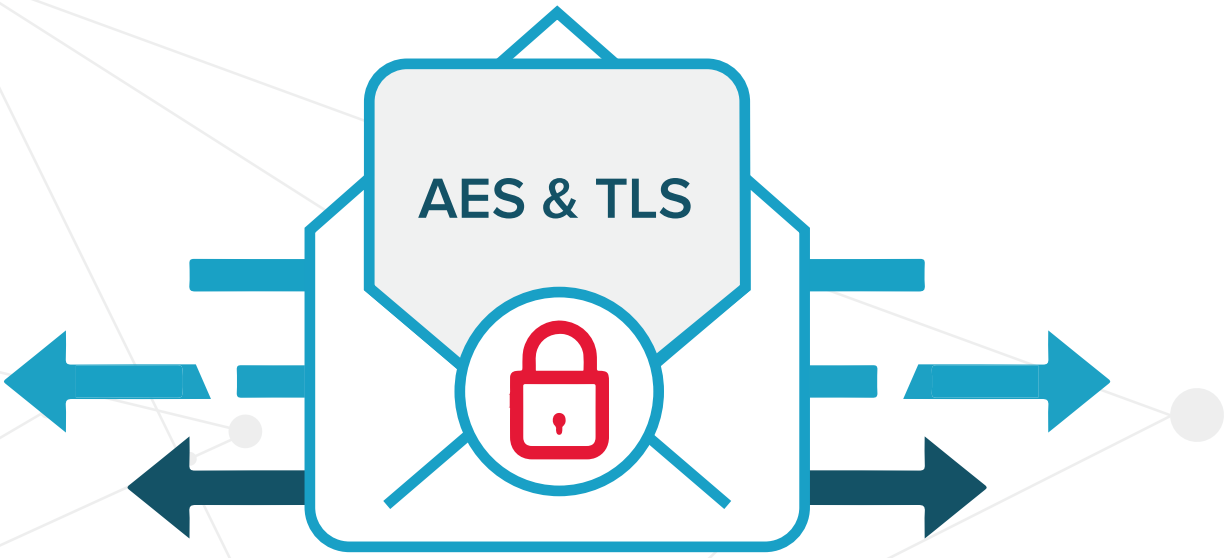
Requirement ② End-to-End Encryption

Transportation Layer Security (TLS) is an industry standard communication layer for sending encrypted data over a wide area network (WAN) that can be paired with AES encryption to provide true end to end security. TLS/SSL protects the top level of data streaming between devices, encrypting the data from device to device at the endpoint when the data is transferred. While TLS/SSL is suitable for data transmission security, data generated from IoT devices is still vulnerable over the network unless it is encrypted. For true end-to-end security, the data itself should be encrypted with the Advanced Encryption Standard (AES) encryption specification.

AES encryption works in conjunction with keys that can be distributed and encrypted at an endpoint -- and only devices with encryption keys can decrypt the data as it is pushed and received. In such a robust security scenario, the network provides full end-to-end security, alleviating hardware manufacturers of a significant security burden.

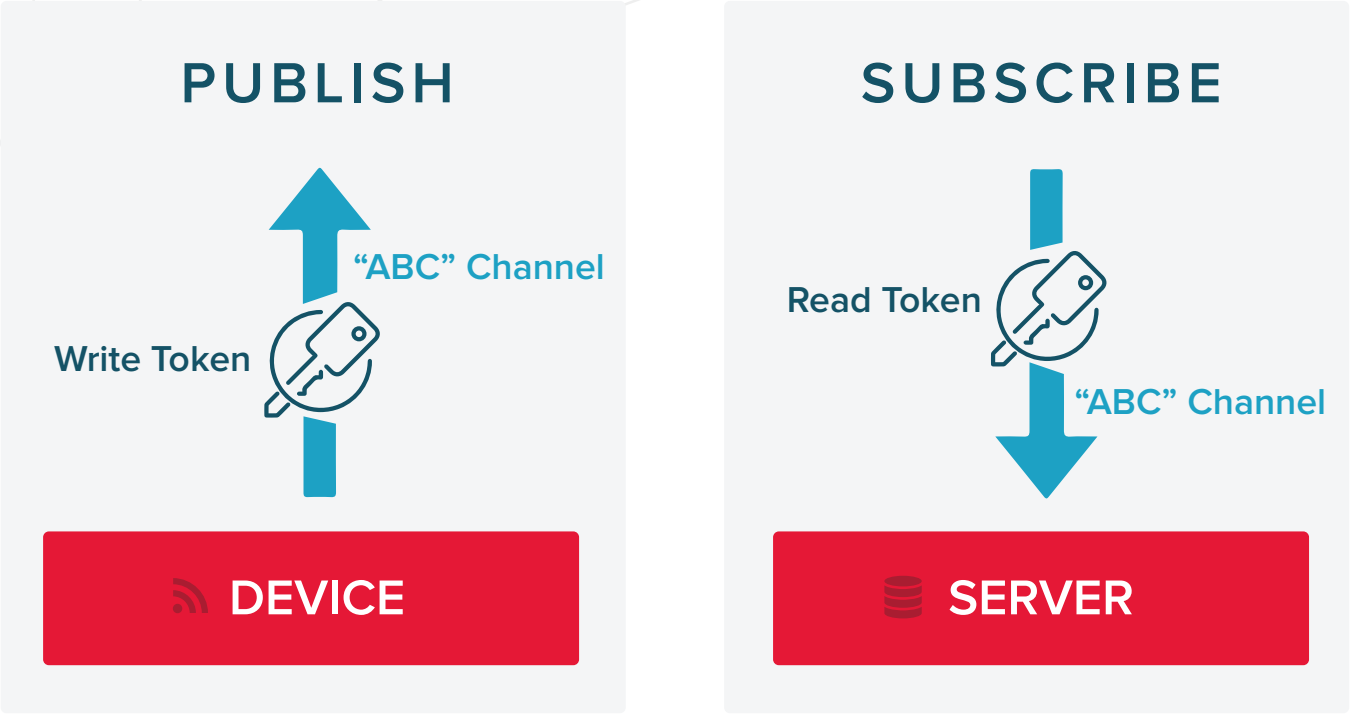
It is also worth noting that there may be scenarios in which full encryption could be limiting - for instance, if a midpoint device wanted to read part of a message to filter or analyze it. With full encryption, this would not be possible, but with a message/envelope paradigm, it is.

In this scenario, the message body is encrypted with AES, but the surrounding envelope, which can contain key data to be used midstream, is only encrypted at the endpoints with TLS. Using this strategy, IoT manufacturers can easily ensure full end-to-end encryption for sensitive data while simultaneously allowing for clever design patterns that leverage “envelope” data for mid-stream processing and analysis.



Requirement ③ Token-Based Access Control

While AES and TLS/SSL can be used to encrypt the data as it is being transferred, another major challenge is fine grain access control over who and what can transmit and receive data. With potentially millions of devices trying to listen to the correct channels and topics, it is extremely inefficient and insecure to ask end devices to filter out topics they don't subscribe to. Instead, the network should handle the bulk of this task.



Within the publish/subscribe paradigm, a token-based access control approach can be used to distribute tokens to devices to grant access to specific data channels. This approach enables fine-grained control over which tokens are created, which devices receive those tokens, and to which data those tokens grant access. It also enables centralized control over when and how tokens are revoked, cutting off data stream access from non-paying customers, for example.

In doing so, the network effectively serves as a traffic cop, both authorizing device access and managing which devices can speak and listen on the network based on the tokens the network distributes.

Requirement ④ Device Status Monitoring

In both consumer and industrial IoT, it is critical to actively monitor the online/offline status ("presence") of devices. When a device such as a home security monitor, oil field sensor, or home appliance disappears or stops sending and receiving data, the owner or monitoring system needs to know about it. An offline device could mean local tampering is taking place, or a broader issue like a power or Internet outage has occurred.

IoT metadata tracking requires a separate, secure data channel to stream presence data for each device, which can be customized to stream online/offline status as well as other custom states such as temperature, acceleration, or geolocation. Each aspect of the device's state can have its own publish/subscribe channel to stream a "heartbeat" through the network, which can then be used for alerts and other action triggers.

"An offline device could mean local tampering is taking place, or a broader issue like a power or Internet outage has occurred"

For instance, a remote door lock could alert its owner of a change in lock state only if the owner's phone is not within 20 feet of the front door. Or, if an array of sensors at a solar power plant go offline, the network could immediately dispatch a technician to investigate the problem.

Realtime, highly reliable presence/status monitoring gives both consumers and IoT manufacturers the peace of mind they need to trust the 50 billion new devices coming online in the next 5 years. Without it, adoption and usability will very likely suffer as consumers and industries are reticent to pass over major pieces of their lives and businesses to "smart" devices they cannot monitor.

Requirement 5 User-Friendly Setup and Upgrades

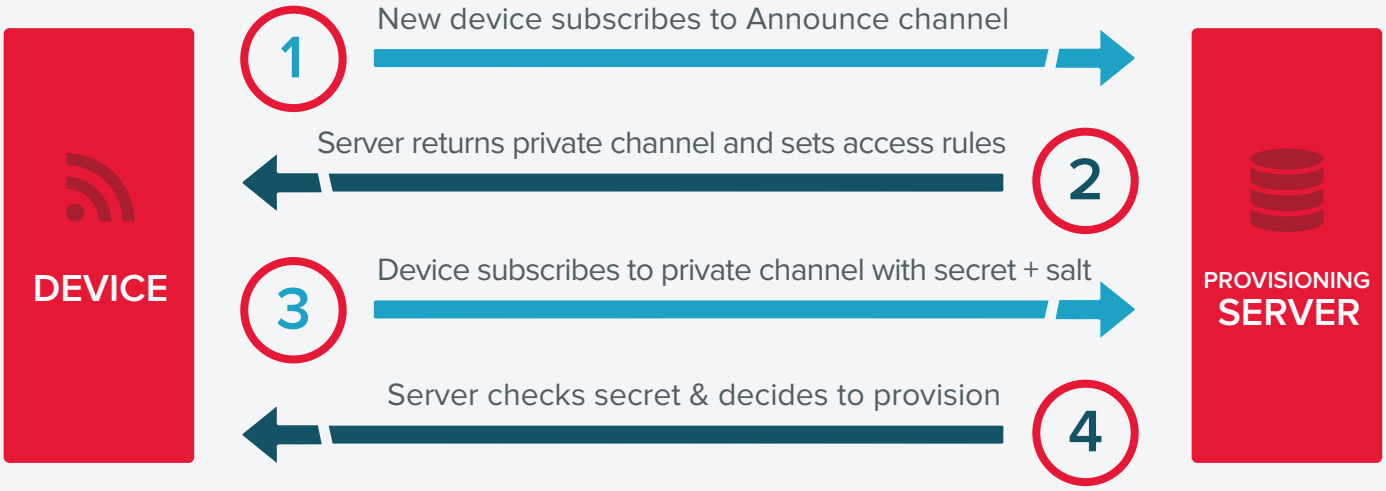
Thus far, we have assumed that the IoT devices in question are operational and connected to the internet. It's time to address the process of getting devices up and running, and keeping them up to date with software and firmware upgrades.

Imagine this scenario: a consumer purchases a system of 6 wi-fi enabled cameras with motion sensors for home security. The customer expects that these cameras will work like any other peripheral device once they are plugged in and connected. Today, that expectation is rarely met.

"A publish/subscribe paradigm makes it easy to securely set up and provision IoT devices"

Instead, the customer is responsible for getting the cameras around their home firewall that blocks their connection, making them broadcast to the correct port, keeping them up to date with the latest software and security updates, and a host of other challenges that are technically far above the ability of an average consumer. The security vulnerability in this paradigm is obvious: more than likely, the customer will never install updates to patch any security vulnerability if they get the device properly set up in the first place.

4 Steps for Securely Provisioning IoT Devices



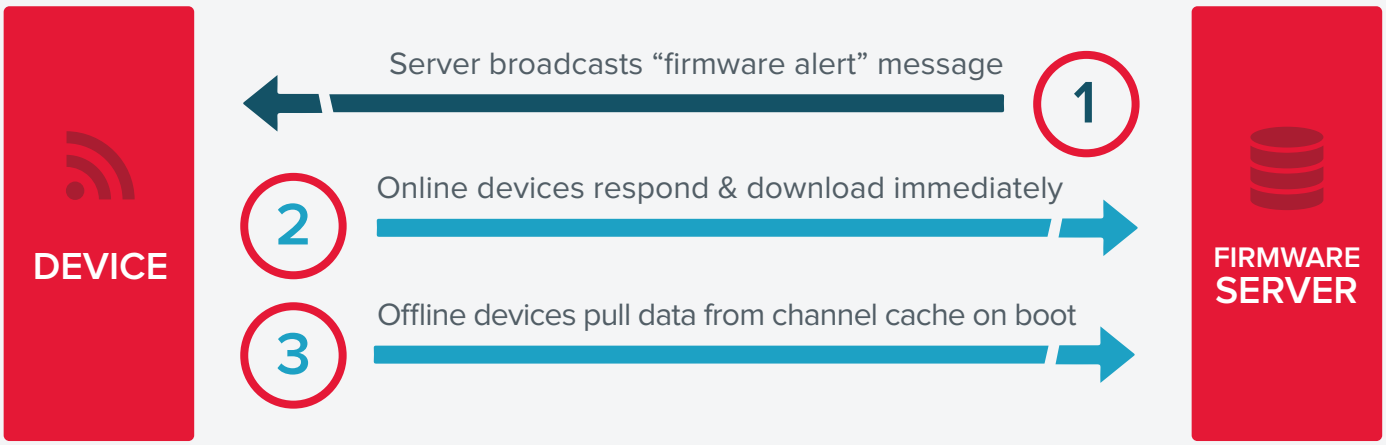
A publish/subscribe paradigm that uses the standard outbound ports 80 and 443 makes it easy to securely set up and provision IoT devices. When it's plugged in, the device wakes up, subscribes to a designated Announce channel and announces itself to the data stream network. The server then returns a private channel on which the device and server can communicate. The server can then set access rules on the channel and provisions from there. All of this happens immediately, giving the consumer the plug and play experience she expects.

Once a device is set up and provisioned, it's also important to implement a way to securely update firmware for that device. If consumers have responsibility for downloading and installing firmware updates, they will likely not download critical updates or may be vulnerable to malicious updates from unauthorized sources. To counter this eventuality, manufacturers should use the device's secure publish/subscribe channels to instruct the devices to download and install firmware updates when they become available.

The design model for realtime firmware updates in the field begins with the server broadcasting a firmware alert message on a channel that all devices can read securely. The master server then instructs the devices how to access and install the update.

When paired with presence monitoring, the IoT manufacturer can be sure that if an individual device is offline, it will receive the firmware update message from the network as soon as it boots back up. Online devices install firmware updates immediately. All this communication can also use the end-to-end encryption and token-based access control methods detailed in previous sections of this paper to make the firmware upgrades as secure, accurate, and automatic as possible.

3 Steps to Secure, Remote Firmware Upgrades



Moving Forward

To reap the full promise from the Internet of Things, both consumers and industry need to be convinced that the data newly-connected devices gather and use is safe. Manufacturers that ignore this reality risk not only the adoption of their new devices but the entire reputation of their company.

With the understanding that security is essential for IoT trust and adoption, manufacturers are faced with a choice: to attempt to harden security in-house, device by device, for the entire lifecycle of every product -- or to offload the bulk of security onto the network that transmits data to and from their devices. This white paper has made the case that it is advantageous to shift as much security burden onto the network is possible. Doing so will decrease time to market for new devices, increase user adoption, and lower the ongoing risk of securing 50 billion new IoT devices. IoT manufacturers can choose to reinvent the wheel, or they can leverage a secure, global Data Stream Network and take advantage of the massive economies of scale such a service can offer.

PubNub®

PubNub is a secure global Data Stream Network (DSN) and easy to use API that enables our customers to connect, scale, and manage IoT devices and realtime apps. With over 70 SDKs for every platform, 250ms worldwide data transfer times, and scalability for hundreds of millions of devices, PubNub's unique infrastructure gives you the ability to easily build and operate world-class realtime applications and IoT devices. PubNub is headquartered in San Francisco.

Start Your Free Trial Today @
www.pubnub.com/get-started