Architectural Considerations for IoT Device Security in the Home

Version 0.9

Abstract

Consumers need the means to manage IoT devices in their home networks. Specifically, we look at several emergent technologies, beginning with how devices and home networks are introduced through the Wi-Fi Alliance's Device Provisioning Protocol and the IETF's Bootstrapping Remote Key Infrastructure. Once a device is connected, it has to be protected. We discuss both learned and declared profiling mechanisms such as Manufacturer Usage Descriptions. To detect and remediate attacks, we discuss attack signature technologies. Key to all of this is the need to leverage the relationship between the consumer and a party such as a service provider or a firewall vendor, such that the information presented to the consumer is comprehensible and actionable. This document is intended for Internet operators (ISPs) who are specifying requirements for these CPE devices; it also provides practical advice on current technologies that can be used.

Content

1 Introduction 3
1.1 Trust assumptions 3
1.2 Manageability assumptions 4
2 Securely introducing devices to the network 4
2.1 “Legacy” Onboarding Mechanisms 4
2.2 Use of per-device PSKs 5
2.3 Device Provisioning Protocol 5
2.4 Bootstrapping Remote Secure Key Infrastructure (BRSKI) 6
3 Providing appropriate access to the device 8
4 Monitoring device behavior and mitigating threats 10
4.1 Existing technologies 10
4.2 Reporting and mitigation 11
5 User interactions 11
6 Deployment Models 12
6.1 CPE devices provided by ISPs 12
6.2 Second home routers purchased by consumers 12
7 Conclusion 12
References 13
Additional resources 13

1 Introduction

By 2025, some estimate that 75 billion devices will have network connectivity.[1] As more homes make use of the Internet of Things (IoT), it will become increasingly important to establish secure approaches that allow for simple management of access to the home network by consumers. The rates of adoption for certain capabilities are growing exponentially. It took Amazon four years to reach 100 million Alexa-enabled devices, and only one year later it passed 200 million devices.[2] Many different types of devices will soon connect to the Internet, and the standards to address their needs are beginning to mature.

This document focuses on several key aspects:

- Securely introducing the device to the network
- Seeing that it gets the access it needs (and no more)
- Monitoring device behavior and mitigating threats
- Some principles that device manufacturers should follow to ensure user safety and privacy

A key principle is that users should not be asked questions that they are unlikely to know the answer to (or are unlikely to understand in the first place). Thus, the architecture must allow for some third party, either a service provider or firewall vendor, to provide to the user the expertise needed to limit those interactions.

Another key principle is to assume that every IoT device will have vulnerabilities. A layered approach is therefore required, where the device manufacturer and the service provider or firewall manufacturer work together to protect the user.

Important Aspects This Document **Doesn't** Cover

This document is focused on how the network can be used to protect the device. We do not cover what device manufacturers need to do to protect their products and consumers. There are a number of standards including those of NISTIR 8228, ENISA IoT Security Recommendations, and the IOT Security Foundation Best Practices that cover these aspects in great detail.

1.1 Trust assumptions

The technology described in this document makes several base assumptions about trust. The first is that either the CPE or a firewall sitting in front of the CPE is trusted by the consumer. The second assumptions that there is an existing relationship, either through an app or through a contract, between the user and the provider of that CPE or firewall that can be leveraged to maintain an inventory of authorized devices on the local network.

[1] https://www.softwaretestinghelp.com/iot-devices/
[2] https://www.cnet.com/news/amazon-sees-alexa-devices-more-than-double-in-just-one-year/

1.2 Manageability assumptions

Another assumption of this work is that most consumers lack the training, experience, and even necessary information about their own devices to manage their own networks, and they require the expertise of service providers and firewall vendors to assist them. For this to happen, the manageability capabilities that those vendors need must be available in their equipment. Those capabilities include, e.g., traffic monitoring, device status information, including counts of what packets have been dropped and why and authentication success and failure information. The interfaces to collect the information must be present, as must interfaces to change configuration.

2 Securely introducing devices to the network

The network onboarding of a new device offers the best opportunity to initiate processes that can help to securely integrate the device into the home. Today, new IoT devices are added to a consumer's network like any "normal" device, such as a PC, smartphone or tablet. IoT devices often lack input or output interfaces that the consumer might use to enter or establish long term credentials. Thus, an automated way to establish credentials for the device is needed.

This section focuses on **wireless** onboarding. Future iterations of this document may also address wired onboarding.

2.1 "Legacy"[3] Onboarding Mechanisms

In many home appliance situations, the onboarding process works as follows:

1. A button or control on the device enables the onboarding process.
2. The device becomes an access point for a specific WiFi SSID. This might be unencrypted or encrypted with a well-known Private Shared Key (PSK).
3. The consumer downloads an appliance-specific app to their phone. The app takes control of the phone's WiFi[4], changes to the above well-known SSID, and then executes some appliance-specific API.
4. The app takes control of the appliance, and usually copies the PSK from the phone to the appliance.[5] The appliance is now online.

Should the consumer change PSKs, the onboarding process must be repeated for all connected devices. If a device misbehaves and is quarantined based on that PSK, the consumer could find that they are unable to manage other devices that share the same PSK. This method also requires

[3] We say "Legacy" but this is a general case today.
[4] If this sounds like a security issue – it is. However, not permitting automatic WiFi control makes the user experience significantly more complex.
[5] Again, the app ends up with access to the phone's list of PSKs for most networks!

a smartphone app for each brand of IoT device. Finally, if the security of one device is broken, the network can be accessed by any device using that key. This model is **not** recommended in the future.

2.2 Use of per-device PSKs

Per-device L2 network segments can be accomplished by giving each device a unique PSK instead of using a single PSK for every device on the local network. This accomplishes two things:

- The router is certain that no other device can impersonate the device, provided the key in the device has remained secure.
- If the device misbehaves, the router can isolate the device without affecting other devices.

The average PSK today consists of a large string of letters and numbers, making management of per-device PSKs untenable without automation. The next section deals with ways to automate per-device PSKs. This can be implemented via Device Provisioning Protocol (DPP), provided the "configurator" app and access point can provision the unique PSK for a given device.

2.3 Device Provisioning Protocol

Device Provisioning Protocol (DPP), also known as WiFi Easy Connect[6], is a voluntary industry standard introduced by the WiFi Alliance. DPP simplifies device onboarding by having the manufacturer imprint a public/private key pair in the device and provide the public key to the consumer, typically through a QR code. The consumer can then prove that they are in possession of the device by having the corresponding public key, while the device can prove to the owner that it has the associated private key. Thus, mutual authentication is established, and the device can be configured with appropriate credentials for the owner's network. In some cases, the device can provide additional information to the network, like a MUD URL.

To users, DPP appears very similar to device-type or brand-specific methods. However, DPP uses 802.11 public frames rather than IP frames over a private network. While DPP envisions apps on phones directly provisioning endpoint devices, because of chipset issues in phones, it is more likely that a home router management app will be able to make use of a custom API to communicate the device capabilities directly or indirectly via a cloud connector to the router.

[6] https://www.wi-fi.org/discover-wi-fi/wi-fi-easy-connect/

Router-Led DPP Activity

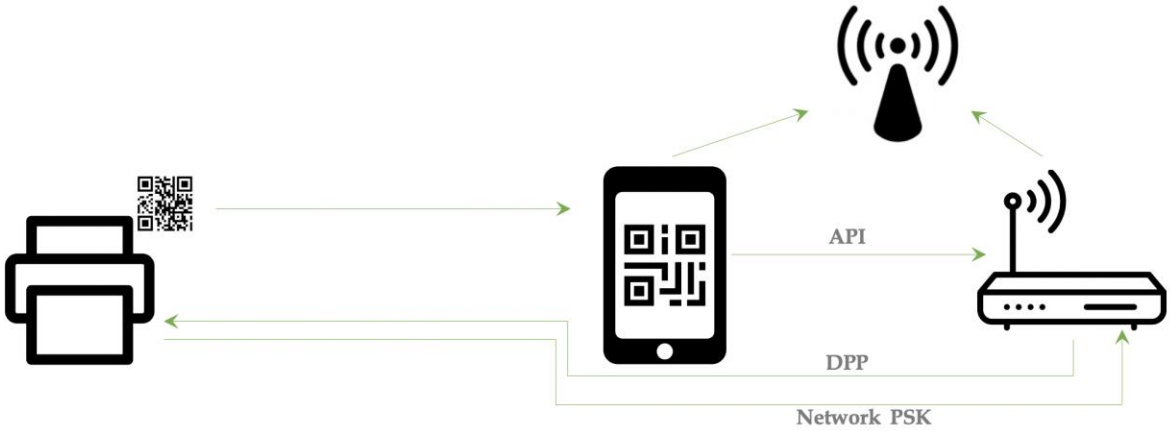

Figure 1: Device onboarding via Router-Led DPP

In a scenario (Figure 1), the mobile phone is only used to scan the QR code providing the public key. The phone then uses an API to talk to the router, and the router sends the special 802.11 public frames to the device, completing the DPP handshake. The router is then able to provision whatever PSK it deems appropriate. In this model, only the router needs to support the DPP frames. Furthermore, the router has established a trusted communication path with the endpoint that is being onboarded, over which it may exchange network-related configuration or state information. Router vendors are advised to check with their PHY and driver suppliers for compatibility with DPP. Exposing an API to a phone requires significant security considerations around how trust between those two devices is established.

The consumer may sometimes wish to change per-device PSKs. In this case, some form of coordination between each existing end device and the router would be required. This might involve resetting the device and/or re-running DPP. Thereby, devices with individual PSKs are easier to identify and control. Revoking the corresponding PSK of a misbehaving device will block only *that* device from accessing the network.

2.4 Bootstrapping Remote Secure Key Infrastructure (BRSKI)

"BRSKI" is an IETF standards track specification[7] for zero-touch onboarding of devices. It was originally meant to onboard enterprise and ISP-class switching devices into data centers without requiring physical access to equipment. It is also intended for use in industrial IoT applications where there is some kind of network operator to set up and maintain a private certificate authority

[7] Draft-ietf-anima-bootstrap-keyinfra, waiting for references in the RFC-editor Q. Also see https://www.sandelman.ca/SSW/ietf/brski-links for more explanatory material.

(CA), an Enrollment over Secure Transport (EST) server[8], and an Authentication, Authorization, and Accounting (AAA) service.

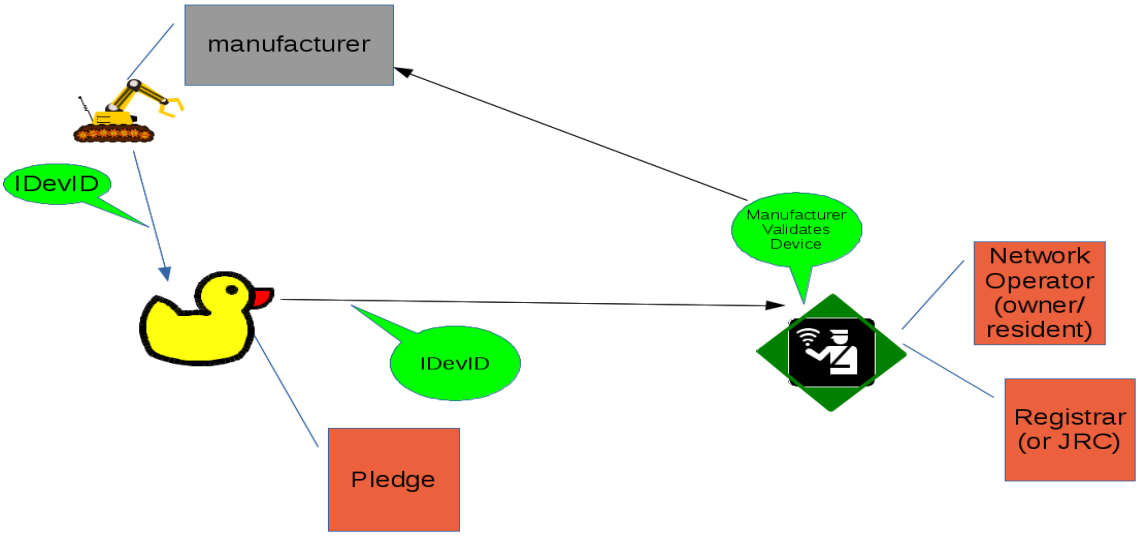

Figure 2: Device onboarding via BRSKI

BRSKI uses a manufacturer-installed IEEE 802.1AR certificate (IDevID) in order for the network to validate the identity of the device. The device uses an RFC 8366 voucher to validate that the network is an appropriate owner. In professionally run networks (ISPs, enterprises, and Industrial IoT), the network operator knows what kinds of devices they have purchased from which manufacturers and may even know the set of serial numbers to expect. They might not know which serial number will go where, or the order in which the boxes will be opened. From an alternate point of view, the manufacturer is aware, via automation of their sales process, to whom they have sold devices. This can introduce a challenge with resales where prior registration may be required.

The sales relationships that the BRSKI operating model envisions might not apply easily to the home. For BRSKI to succeed here, the BRSKI Registrar must find its way to the consumer's home router or other device (such as a home NAS), in order to manage the ownership relationships of the consumer. This functionality is similar to that provided by DPP. However, it includes features that may appear complex, such as a private Certification Authority. The BRSKI Registrar functionality fits nicely into a container on existing CPE.[9] More likely the registrar would run in a cloud-based service.

Manufacturer-installed (birth) certificates

BRSKI explicitly requires that every device (called a "pledge" until it is enrolled) come with a manufacturer-installed certificate. Manufacturer-specific onboarding apps may also require this

[8] Pritikin, et al, *Enrollment over Secure Transport (EST)*, RFC 7030.
[9] For example, see https://minerva.sandelman.ca/

certificate if the communication between the app and the device is based on TLS (for instance HTTPS). In both cases, the certificate will be from a private Certificate Authority (CA) that is maintained by the manufacturer. BRSKI explicitly deals with the transition of trust from the manufacturer to the local environment, while manufacturer-specific methods include the appropriate trust anchors in the app itself.

3 Providing appropriate access to the device

When a device goes through the onboarding process, its device type/class should be identified, and it should be placed in an appropriate network segment. Ideally this should happen with the consumer's approval. The CPE or an associated agent should keep a database of devices that have already been, or will be, onboarded.

Of the tens of billions of devices that are being connected, any single IoT device will typically need access to only a handful of other endpoints. There are two challenges to providing correct access:

1. Determining the endpoints the device may contact, and the type of traffic that may run between the device and those endpoints.
2. Providing the capabilities to limit access to that subset of other endpoints and services.

There are two approaches to address the first challenge: a learned model and a declared model. Both models attempt to limit a device's network access so as to reduce its threat surface to various forms of attacks by devices that have no business talking to it.

With the learned model, CPE can learn by observation what the device is. Such fingerprinting approaches involve observing DHCP requests and responses, MAC addresses, Multicast announcements, and similar characteristics to establish what one thinks the device is. Advanced techniques such as machine learning might also look at traffic flows, TLS options used to communicate, and other behavioral information in order to make a determination.

Either the CPE itself will process all of this information, or it will send the information upstream for further analysis.[10] A number of information format standards already exist. Two common formats are PCAP and IPFIX. The same information may also be used to analyse whether a device is remaining in profile, doing only what it is supposed to be doing.

This learned model presents a challenge: either the CPE must do substantial amounts of processing, or a copy of (at least some) communications must be sent upstream for processing. It is thus resource intensive, depending on how much information is used to identify device access requirements. In addition, devices might lie or otherwise obscure information that is used to fingerprint.

[10] An open question is whether the channel used to communicate this information should be standardised.

The declared approach is for the device or its manufacturer to state outright what it is and what sort of access it requires. This is the approach taken by Manufacturer Usage Descriptions (MUD) [RFC 8520]. MUD can be used to provide deployments with a generalized access list that can be localized to a specific network. It can also be used to share other information about a device, such as how to retrieve a Software Bill of Materials (SBOM). MUD can specify what Internet sites to allow a device to access (sometimes termed north/south control), and what devices in the home should be permitted to talk to each other (east/west control).

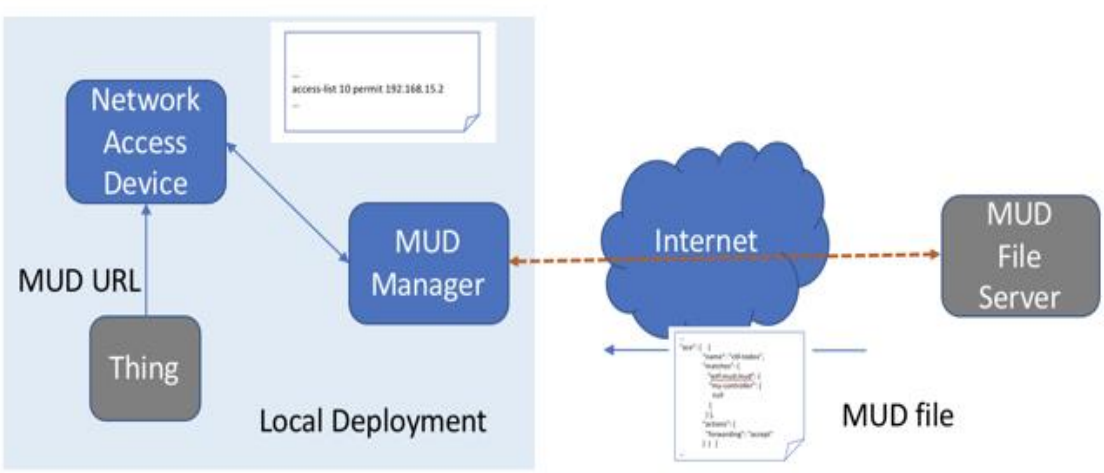

Figure 3: General MUD architecture

Figure 3 represents the general MUD architecture. In a consumer environment, either the network access device serves as a MUD manager, or, more likely, some service is playing that role. This could be provided by the service provider or a firewall vendor. The key is that a control path is needed between the network access device, such as the CPE, and the MUD manager for purposes of the CPE sharing MUD information and the MUD manager providing access rules the CPE should implement. Furthermore, a communication channel is needed between the MUD manager and the consumer for approval, as discussed below. Because MUD is a declarative approach, it is less resource intense on its own, and may be more authoritative. However, it requires that the IoT endpoint announce a URL (such as via DHCP or LLDP).

Once access requirements are determined, they must be deployed to CPE. Most CPE equipment has basic firewall capabilities to limit access to and from the Internet. Only **some** CPE has the capability to limit access between devices in the home. However, that sort of limited access is critical, in case one home device infects another.

4 Monitoring device behavior and mitigating threats

Once a device is connected to a network, there is always the possibility that an attack will succeed against it. If this happens, the device itself may start behaving as a malicious actor. There are several general approaches to detect and mitigate such cases:

- **Allow/blocklist based** – Malicious traffic is detected by its destination. For instance:
  - Comparing UDP/TCP destination to known deny-lists ("blocklist")
  - Validating that UDP/TCP traffic destination matches MUD profile
  - Performing reverse DNS lookups to map network target to domain blacklists

- **Signature based** – Malicious traffic is detected by its properties. For instance:
  - Detecting when devices on a LAN are initiating spoofed UDP traffic
  - Inferring profile based on MAC fingerprinting
  - Performing DPI
  - Performing Netflow analysis
  - Examination of certificates and TLS parameters

- **Anomaly based** – Malicious traffic is detected through abnormal device behavior. For instance, deep learning or other artificial intelligence that summarises "normal" traffic, combined with thresholds that would mark activity as anomalous.

4.1 Existing technologies

There are several efforts that attempt to provide some of this functionality. In general, these tend to use either allow/blocklist or signature-based approaches, using lists similar to anti-virus tools. Since these lists can grow quite large, this analysis is usually done centrally, by sending a summary of traffic to a central server, and relies on a subscription service model.

Open source examples of this approach are Snort, Zeek, and Suricata.[11] Several companies also supply "secure routers", which provide this functionality, usually accompanied with a subscription model for rulesets, or even a full VPN for cloud-based analysis.

The Turris Project[12] contains a distributed adaptive firewall, where suspicious traffic is collected and analysed centrally. The resulting additional firewall rules are distributed to all connected routers. This can protect home networks, and with sufficient deployment, provide an avenue to mitigate large-scale attacks as well.

Anomaly-based detection is still an active field of research. The SPIN project[13] is a platform for research and development on securing home networks that contains an experimental module that

[11] https://www.snort.org, http://zeek.org, https://suricata-ids.org
[12] https://turris.com/
[13] https://spin.sidnlabs.nl/

compares the number of packets and their destination to an average for the device, blocking it when this exceeds a certain threshold.

4.2 Reporting and mitigation

Once an anomaly has been detected, a technical support function should decide what action is taken and what mechanisms are appropriate in order to determine a mitigation (in short – who gets notified and when, and what is to be done). The consumer is unlikely to be the first point of contact, because a certain expertise might be required to make use of meaningful remediation options. Reporting should occur in two phases: first to the technical support function, typically offered by the firewall vendor or ISP can assess the risk to the consumer and others; second, the firewall vendor or ISP should report to the consumer.[14]

User Services Platform (TR-369), is a standard for device lifecycle management that includes device monitoring and alert management. Slightly more limited in scope, the Distributed Denial-of-Service Open Threat Signaling (DOTS) Signal Channel Specification (RFC 8782) also provides a method of requesting mitigation actions from a router. This does not include full remediation information for consumers, but it could be used to take mitigating actions immediately.

5 User interactions

As mentioned above, the number of user interactions should be kept to a minimum. There are three possible options to communicate with a consumer. One of these is through a portal on the CPE. In this case, the consumer must directly connect to the CPE on the local network. Another approach is where the CPE has a control interface in a cloud connector, which is in contact with the consumer via an app. A third approach is where the app connects directly to the CPE. These approaches are not mutually exclusive. While standards like TR 369 and NETCONF provide some of the necessary capabilities, different CPE manufacturers may or may not make use of these protocols.

To minimise user interactions, developers should consider whether the user onboarding a device is the consumer or home owner, rather than a visitor or family member. In this way, the consumer can know what is being onboarded.

[14] One area that we do not explore, but others are exploring, is that of governance: which parties are responsible and accountable for the various aspects of maintaining security posture of the home and devices in the home?

6 Deployment Models

In almost all cases described above, there is some device on the home network which is already trusted by the consumer (or possibly by the ISP) that has a role in the security of the IoT device. We give some examples below.

6.1 CPE devices provided by ISPs

In this case, the service provider has included the CPE as part of their Internet service. Most of the components necessary to onboard and protect IoT devices are available today through such distributions as the OpenWrt Project[15] and industry associations such as the prpl Foundation[16]. Some ISPs have commissioned their own router hardware or purchased it from a supplier who can provide the right packages and permissions. Other ISPs purchase complete solutions from vendors. Many of those vendors are just shipping code from the OpenWrt Project and could be convinced to include the respective components today.

6.2 Second home routers purchased by consumers

For many Internet services such as cable and fibre to the home (FTTH), the CPE router and modem are often integrated. Some consumers find that these CPE devices are inadequate. Either the CPEs lack certain features such as decent Wifi range, or the consumers simply do not trust their ISPs. Some jurisdictions have a legal requirement that consumers can choose to use their own individually procured CPE, while in other cases, specific optical requirements for FTTH make it difficult to impossible for the consumer to select their own hardware. CPE and firewall vendors should take care: if two cascaded home routers are used, if both are offering security services, it is possible that their policies will conflict or be confusing.

7 Conclusion

The Internet of Things requires a different perspective from consumers, service providers, and others. The role of the service provider in protecting not only the consumer but the community through the externalities introduced by devices that are either unsafe or likely not to stay safe needs to be carefully considered. We have proposed methods that permit devices to securely onboard in a scalable fashion that leverage existing relationships in a way that makes it easy for a consumer to understand. This requires an expanded role for either the CPE or firewall provider.

[15] https://openwrt.org/
[16] https://prplfoundation.org/

References

- https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/773867/Code_of_Practice_for_Consumer_IoT_Security_October_2018.pdf
- https://www.cisco.com/c/en/us/solutions/collateral/internet-of-things/white-paper-c11-743623.html
- https://www.itu.int/rec/T-REC-Y.4807-202001-I
- https://www.wi-fi.org/discover-wi-fi/wi-fi-easy-connect
- Report on IoT Device Security: https://www.agentschaptelecom.nl/binaries/agentschap-telecom/documenten/rapporten/2019/09/25/rapport-digitale-veiligheid-van-iot-apparatuur/Report+on+IoT+Device+Security.pdf

Additional resources

- https://docs.oasis-open.org/cacao/security-playbooks/v1.0/csd01/security-playbooks-v1.0-csd01.pdf