Architectural Considerations for IoT Device Security in the Home

A guide for ISPs specifying CPE devices

Version 0.9

Abstract

Guidelines on security considerations about the use of IoT devices in a typical end user network are discussed in this document. Connectivity for these devices to the carriers' network is generally provided via CPE. It is intended for Internet Operators (ISP) who are specifying requirements for these CPE devices. The document also provides practical advice on the currently available technologies that can be used.

Introduction

By 2025 it is predicted that around 75 billion devices will be capable of Internet connectivity.[1] As more homes make use of the Internet of Things (IoT), it will become more important to establish secure approaches that allow for ease of 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. It took just one more year to get to over 200 million devices.[2] There are a great many different types of devices that will connect, and the standards to address their needs are beginning to mature.

This document focuses on several key aspects:

1. Securely introducing the device to the network
2. Seeing that it gets the access it needs (and no more)
3. Retrospection with regard to whether the device is behaving appropriately
4. Some principles device manufacturers should follow to insure user safety and privacy

A key principle relating to consumers is that they should not be asked questions to which they are not likely to either understand or know the answer. Thus the architecture must provide for some third party, either a service provider or a firewall vendor to provide expertise necessary to reduce the number of interactions to only those that are absolutely necessary and within the capabilities of the consumer to address. Furthermore, the number of interactions with the user should be kept to a minimum.

[1] https://www.softwaretestinghelp.com/iot-devices/

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

Another key principle is it should be assumed that every IoT device will have vulnerabilities. Therefore, a layered approach is required where the device manufacturer and the network security provider work together to protect the consumer.

This paper focuses on **wireless** onboarding. Future versions may also address wired onboarding.

Trust Assumptions

The basis of much of this document is that some service can be trusted to inform the consumer about what devices are joining the network and what access they need. That service has to be trusted by the user, and is in fact acting as the user's agent. That service is receiving from CPE a view of what devices the user has on the network, and also is controlling the CPE to limit device access. The service may also be seeing what flows are generated. If this is a typical Internet provider, most – but not all – of those flows are already visible.

Securely introducing the device to the network

The network onboarding process of a new device offers the best opportunity to initiate several processes that can help to prevent the device from being compromised and/or reduce the impact a compromised device can have on the internet and the internal network.
The current practice is that new IoT devices are added to a consumers network like any "normal" device like a PC, a smartphone or a tablet. However IoT devices usually run unattended, and any misbehaviour is generally not detectable by the user in everyday operation.

To simplify monitoring, mitigation, and quarantining processes; the internal network should be segmented so that different classes of IoT devices can be logically isolated.

When a device is going through the onboarding process its device type/class should be identified (also see fingerprinting) - ideally with approval of the consumer - and be placed in an appropriate network segment. The CPE or an associated agent should keep a database of devices that either are or will be onboarded.

Network onboarding based on device type or brand

In many home appliance situations the onboarding process typically 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 may be unencrypted, or it may be encrypted using a well-known Private Shared Key (PSK).

3) The home owner downloads an appliance-specific app to their phone. The app takes control of the phone's wifi[3], 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.[4] The appliance is now online.

Should the consumer change PSKs, the onboarding process must be repeated for all connected devices. Should a device misbehave and be quarantined based on that PSK, the homeowner could find themselves unable to manage any other devices that share the same PSK. This method also requires a smartphone app for each type 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.

Use of Per-Device Private Shared Keys (PSK)

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:

1) the router is certain that no other device can impersonate the device, so long as the key in the device has remained secure,
2) If the device misbehaves, the router can isolate the device without affecting other devices.

Per-Device PSKs are not commonly used today due to the lack of automation. Typical users can not remember a single PSK today; without automation per-device PSKs are untenable.

The next section deals with ways to provide per-device PSK. This can in theory be implemented via Device Provisioning Protocol (DPP), provided the "configurator" app and AP can provision the unique PSK for a given device. Per-device PSK is not possible when using device-specific app-enabled onboarding unless there is an API from the phone app to the home router (as described below). When using WPA-PSK, CPE should support multiple PSKs.

Device Provisioning Protocol

Device Provisioning Protocol (DPP) aka Wifi Easy Connect[5] 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 making the public key accessible to the device owner, typically through a QR code. The owner can then prove they are in possession of the device by having its corresponding public key and the device can prove to the owner that it

[3] If this sounds like a security issue, it is. Apple does not allow this permission, making the user experience significantly more complex.

[4] Again, the app winds up with access to the phone's list PSKs for most networks!

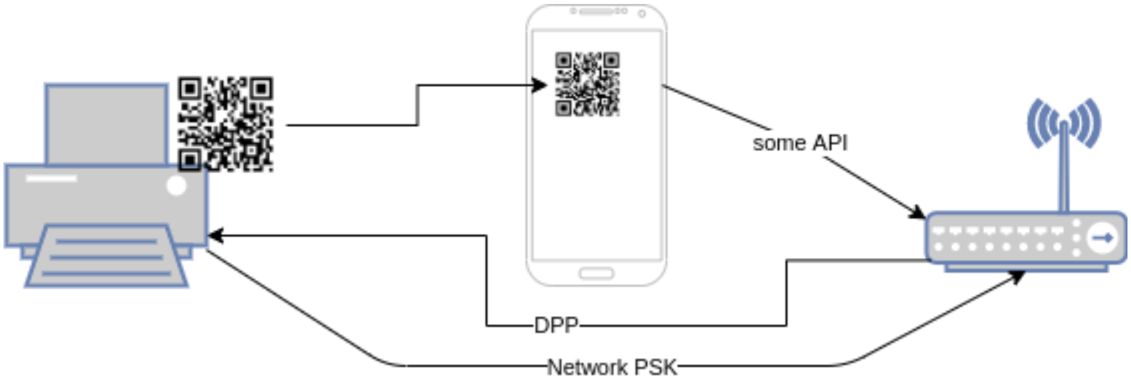
[5] https://www.wi-fi.org/discover-wi-fi/wi-fi-easy-connect

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.

DPP provides a simple mechanism that allows onboarding wireless devices (and specifically IoT) devices to a network without the need to enter credentials. The user interface in DPP is provided by an application or app on a specific device (i.e. a smartphone) that acts as a controller for the network onboarding process. One can extend the basic user interaction to offer more granular access restrictions that can further help to increase the networks resilience to attacks.

To consumers, 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 chip set 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 to the router.

Router-Led DPP Activity

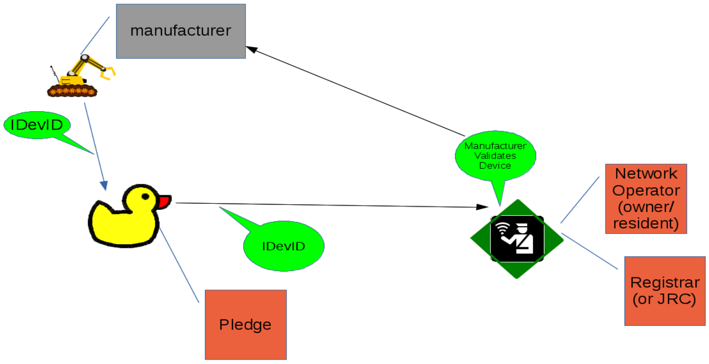


In this scenario, the phone is only used to scan the QR code. The phone then uses some API to talk to the router, and the router then sends the special 802.11 public frames to the device, completing the DPP handshake.   The router is then able to provision whatever PSK it deems appropriate. In this model, only the router needs to support the DPP frames. Furthermore, the router has established a trusted communication path with the device, 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.

Sometimes the consumer may wish to change per-device PSKs. In this case, some form of coordination between each existing end device and the router would be required. That may involve resetting the device and/or rerunning DPP. Devices that have individual PSKs are easier to identify and control. Revoking the corresponding PSK of a misbehaving device will block it from accessing the network.

Bootstrapping Remote Secure Key Infrastructure (BRSKI)

“BRSKI” is an IETF standards track specification[6] for zero-touch onboarding of devices. It was originally conceived of to onboard Enterprise and ISP class switching devices into data centres without requiring any physical access to equipment. It is also intended for use in Industrial IoT applications where there is some kind of a network operator to setup and maintain the required infrastructure and relationships.


manufacturer
IDevID
Manufacturer Validates Device
Network Operator (owner/ resident)
IDevID
Registrar (or JRC)
Pledge

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 RFC8366 format voucher in order to validate that the network is an appropriate owner. In professionally run networks (ISPs, Enterprises, and Industrial IoT), this network operator knows which kinds of devices (from which manufacturers) they have purchased, 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.

The sales relationships that BRSKI envisions might not apply easily in the home. In order for BRSKI to succeed in the home the BRSKI Registrar must find its way into the Home owners home router (or other device, such as a Home NAS), in order to manage the ownership relationships of the home owner. 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.[7]

Manufacturer Installed (Birth) Certificates

BRSKI explicitly requires every device (called a “pledge” until it is enrolled) to come with a manufacturer installed certificate. Manufacturer-specific onboarding apps may also require this 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 manufacturer maintained private CA.

[6] 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.
[7] For example, see https://minerva.sandelman.ca/

BRSKI explicitly deals with the transition of trust, while the manufacturer specific methods include the appropriate trust anchors in the app itself.

Providing appropriate access to the device

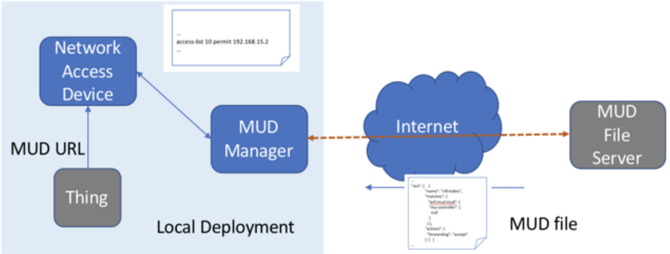
Of the tens of billions of devices that are being connected, any single IoT device will typically need access to only a handful of other endpoints. There are two challenges to providing that correct access:

1. Establishing with confidence what that access should be.
2. Providing the capabilities to limit access to that subset of other endpoints and services.

To address the first question, the CPE can learn by observation what the device is. Such fingerprinting approaches involve observing DHCP, MAC address, Multicast announcements, and similar characteristics to establish what one thinks the device is. Advanced techniques might also look at traffic flows, TLS options used to communicate, and other behavioural information.

Either the CPE itself will process all of this information, or it will send the information upstream for further analysis. In the latter case, a communications channel is required. An open question is whether that channel should be standardised. A number of standards already exist to provide this sort of information.   Two common formats are PCAP and IPFIX.
This learned model presents a challenge in that 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. The same information may also be used to analyse whether a device is remaining in profile. In addition, devices might lie or otherwise obscure information that is used to fingerprint.

An alternative approach is for the device or its manufacturer to declare 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 an access list that can be localized. It can also be used to share other information about a device, such as how to find a software bill of materials (SBOM).



The above diagram 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. That could be the service provider or a firewall vendor. The key is that a control path is needed between the network access device such as CPE and the MUD manager. 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 device implement it. 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 one another (east/west control).

Once access requirements are understood, they must be deployed to CPE. Most CPE equipment has basic firewall capabilities to limit north/south access. Only **some** CPE has the capability to limit east/west access. However, that sort of limited access is critical, in case one home device infect another.

Monitoring device behavior and limiting its access

Once a device is connected to a network there is always the possibility that an attack will succeed against it. If that happens, the device may start behaving as a malicious actor itself. 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 black-lists
- **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
- **Anomaly based**: Malicious traffic is detected by being significantly different than what is considered normal behaviour for this device. For instance:
    - Deep learning or other artificial intelligence that summarises 'normal' traffic, combined with thresholds that would mark activity as anomalous.

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 such lists can grow quite large, this analysis is usually done centrally, either through a VPN or by sending a summary of traffic to a central server, and reliant on a subscription service model.

Most existing technology in this field uses a combination of the allow/blocklist and signature-based approaches. Open source examples of these are Snort, Zeek, and Suricata. 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[8] contains a Distributed Adaptive Firewall, where suspicious traffic is collected and analysed centrally. Resulting additional protective 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.

True anomaly-based detection is still an active field of research. The SPIN project[9] is a platform for research and development of securing home networks, and contains an experimental example module which shows this in action. Its evaluation model is rather simple: it compares the number of packets and destination against the average of the device, and blocks the device when this exceeds a certain threshold.

All of these approaches generally protect north-south traffic, traffic out of and into the local network. East-west traffic, traffic between the devices, is currently not often taken into account, is quite necessary, and is available on some newer CPE. With VLANs, local networks can be separated into smaller clusters, thereby providing reduced attack surfaces for devices on the local network.

Reporting and mitigation

Once an anomaly has been detected, a mitigation and reporting mechanism is required. This reporting occurs in two phases: first to some technical support function, typically offered by the vendor or ISP, that can assess the risk to the consumer and to others. The second phase of reporting is what gets presented to the consumer. The consumer is unlikely to be the first point of contact because some expertise is required to provide consumers with meaningful remediation options. This technical support function should decide what action gets taken and what mechanisms are appropriate: in short, who gets notified and when.

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 (RFC8782) 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.

[8] https://turris.com
[9] https://spin.sidnlabs.nl

User Interactions

As mentioned above, the number of user interactions should be kept to a minimum. There are three!! possible user interactions. One of these is through a portal on the CPE. In this case, the user must directly connect on the local network to the CPE. Another approach is where the CPE that has a control interface into a cloud connector, which in turn 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 that standard and it may or may not be necessary.

To minimise user interactions, developers should consider whether the person onboarding a device is the owner. If so, owners will know what is being onboarded. Otherwise, an interaction with the owner may be warranted. This addresses the access control requirements.

Putting It All Together

In almost all cases discussed above there is some device on the home network which is already trusted by the homeowner (or possibly the ISP) that has a role in the security of the IoT device.

ISP provided secure CPE devices

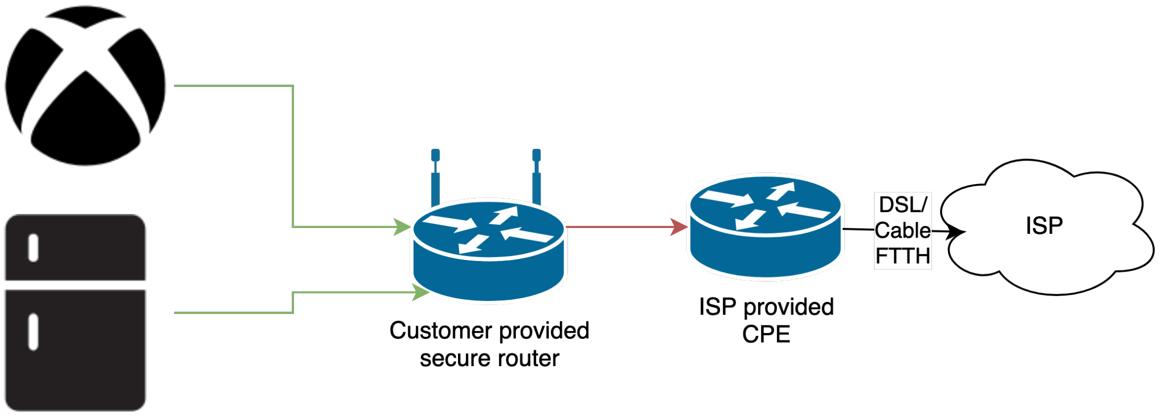
The best situation is that the CPE already includes all of the needed components.
Most of the components are available today. For instance in the openwrt.org project,and industry associations such as the prplFoundation.org. Some ISPs have commissioned their own router hardware (or purchase it from entities like turris.cz), and they can easily include the right packages and permissions immediately.
For other ISPs, they purchase complete solutions from vendors. At least half of those vendors are just shipping code from openwrt.org, and again, could be persuaded to include the right components today.

Customer provided second Home Router device

Some customers find that their ISP provided CPE is inadequate. Either it does not have the WiFi range, or it is missing some feature, or they simply do not trust the ISP. Some jurisdictions have a legal requirement that customers can choose to provide their own CPE equipment. In the Cable connected Internet space, the CPE router and the Cable Model are often integrated, and it is sometimes hard for customers to find an equivalent third party device. Those devices are also often not open to the customer.

The Fibre to the Home (FTTH) market is sometimes more open, but as many installations are really GPON deployments, the GPON “ONT”[10] may require specific optics and the ISP often does not allow customers to provide their own.

Stacked CPE Routers



The above diagram shows two router in “stacked” format. If the outer, ISP provided CPE can not be put into pass through mode, then the red link above will be IPv4, and the customer’s equipment will usually experience two layers of NAT44. Unless the customer provided router includes a VPN for access to IPv6 services[11], then the customer likely will not have IPv6. The passthrough mode of operation therefore has significant benefits, as it puts the customer controlled router right on the Internet. This usually means that IPv6 is properly supported, and no functionality interferes with applications that consumers view as critical (e.g., games, movies, conferencing, etc).

The passthrough mode of operation has problems as well: if the ISP is also providing IPTV services, then it is unlikely that the IPTV signal will get through the customer provided secure router.[12].

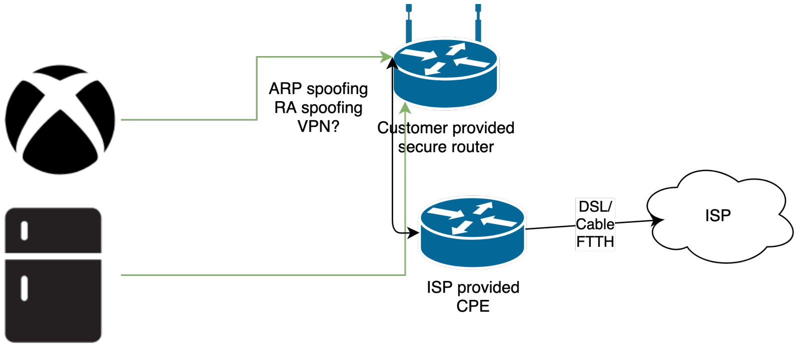
Impersonating CPE Routers

An alternate mechanism has appeared on the market from a number of vendors where a second router is added to the home network, but not in a way stacked fashion. These routers are “one-armed” routers in that they attach to the existing CPE router with a single cable, and then “take-over” the network.

[10] Optical Network Terminals --- i.e. the “modem” equivalent for GPON

[11] Such as https://ungleich.ch/u/products/viirb-ipv6-box/

[12] https://support.bell.ca/Fibe_TV/Receivers/What_is_Bell_Fibe_TV .



The way that this works is that the original ISP provided CPE continues to offer DHCP and Routing Advertisements on the LAN. But, the new router forges ARP responses for the "192.168.1.1" address, forcing all local traffic to the new router.[13] A reason for doing this is so that the traffic can be (selectively) forwarded through a privacy enhancing VPN. Some solutions provide their own WIFI interface, while others are able to take over the CPE provided WiFi as well. The results are sometimes inconsistent as this solution depends upon either beating the ISP provided CPE to answer the ARP, or for end devices to accept whichever ARP reply they last saw as being valid. This method essentially exploits the lack of L3 security!

It is possible to do this for IPv6 as well. It works the best when the ISP provides no IPv6, as then there are no competing networks. IPv6 otherwise supports the concept of multiple routers sending router advertisements: the customer provided router simply provides higher priority RAs than the CPE provided router. It may be for the same prefix, or for another one that goes through a VPN.

Some principles on device safety and privacy

One concern about safety and privacy is the rapidly-changing IoT landscape. Devices are sometimes manufactured, used and abandoned in life-cycles that can be shorter than the standards development process! There are also difficult regulatory and legal constraints which are not well understood and will vary from jurisdiction to jurisdiction. GDPR is the obvious example.

The principle of algorithm agility is explained in ITU-T Recommendation Y.4807[14]. This explains why those developing and deploying IoT platforms need to ensure these systems have the flexibility to keep up with advances in telecommunication/ICT security and cryptography. However it deliberately does not provide guidance on specific cryptosystems, standards or algorithms since these are continually changing because the security landscape is continually changing. In short, an algorithm or key length that is thought to be "secure" today could be considered "insecure" tomorrow.

[13] https://www.privacyhero.com/ is one such solution. There are quite a few, mostly sold as privacy enhancing VPNs.
[14] https://www.itu.int/rec/T-REC-Y.4807-202001-I

Manufacturer User Descriptions (MUD)[15] [RFC8520] provide a way for manufacturers to document the network behaviour of their IoT products: which ports they use, what servers they contact and so on. These can then be incorporated into the network's access policy so that anomalous behaviour by IoT devices can be detected or even stopped. While this is a good starting point. it is not clear yet how MUD descriptions will be incorporated into the access devices -- typically DSL or cable boxes -- at the edge of the customer network. Most end users are unlikely to understand MUD descriptions or what they mean. Much work remains to be done with MUD to improve the security and privacy outcomes. An added complication is some IoT devices are likely to use HTTPS for communication outside the local network, making it difficult to understand what data are being transferred.

The UK's National Cyber Security Centre (NCSC) has published a code of conduct for consumer IoT security[16]. It's written in clear, non-technical language that's aimed at manufacturers, providers, developers and retailers. The document provides a number of guidelines on good practice. These include obvious sensible measures:

- Use unique device passwords that cannot be set to a universal default value
- Provide a vulnerability disclosure policy
- Ensure software gets updated
- Securely store security-sensitive data such as authentication credentials and Personal Data
- Use secure communications protocols - presumably based on TLS
- Minimise exposure to attack surfaces
- Provide software integrity - for example digital signatures for update and patches
- Be resilient to outages
- Monitor telemetry data such as logs and provide appropriate reporting/alert mechanisms
- Validate all input data and guard against buffer overflows
- Make it simple for consumers to install and maintain IoT devices
- Provide convenient ways for consumers to delete their Personal Data or limit how that is used

A similar document is needed for end-user/consumer guidance: how to make informed choices when purchasing and using IoT devices, good password hygiene, keeping software up to date, be alert for unusual behaviour (unexpected network traffic, activity from normally "quiet" devices) and so on.

There is an obvious role here for consumer associations and other organisations that regularly test and review mass market products. Their efforts can help users make informed choices when buying IoT devices. For instance, a recent study by the Dutch Telecom Agency

[15] https://tools.ietf.org/html/rfc8520

[16] https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/773867/Code_of_Practice_for_Consumer_IoT_Security_October_2018.pdf

(Agentschap Telecom) “Report on IoT Device Security” could be taken into consideration when buying IP cameras and similar devices. These reviews should take the security properties and features of connected devices into account in their test scores.

In its most basic form this sort of advice could offer details of how well or poorly devices support features for secure onboarding and updating. Consumer-friendly summaries of full security reviews would be even better.

Many governments are already considering some degree of regulation of IoT device safety and security. It seems reasonable to expect such frameworks will emerge.

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