IoT Security Proposals

Phillip Hallam-Baker
phill@hallambaker.com

There are two primary IoT security concerns:

- Intentional sabotage during manufacture
- Unintentional vulnerabilities

The primary focus of this note is the second issue. Some proposals are also relevant to the first

A major point to be borne in mind when evaluate security is to know which type of security concern is being addressed: *Confidentiality*, *Integrity* or *Availability*. While consumers are most likely to be concerned that their devices might bey 'spying' on them (confidentiality), the biggest threats to the system come from various forms of denial of service (availability) and these are among the most difficult to address.

Cryptography offers an impressive but limited set of security controls. Even the best cryptography does not provide security against a device compromise. Cryptography allows us to limit the set of concerns we need to consider at the network core to availability and defeating traffic analysis, but compromise of an end point threatens everything.

How to deploy best security practice

Far more important than the question of which technologies to deploy is the question of how to deploy them. There is no shortage of Internet security technology but only a small fraction is deployed.

It is instructive to consider the typical mode of software development of IoT devices. Most device manufacturers consider themselves as being first and foremost hardware providers. Software is often if not usually an afterthought. This is particularly the case where low production cost IoT devices are concerned where the 'manufacturer' is often little more than an assembler of parts designed and made by others. Such devices are typically shipped with software 'borrowed' from a competitor which in turn did little more than rebrand the reference software provided by one of the component manufacturers. Such reference code in turn representing little more than an illustration of the capabilities of the devices using resources cut-and-pasted from various sources found on the net.

The same is true of process control equipment. PID controllers typically use essentially the same hardware and software that was developed in the early 1980s when self-tuning features were added to even older microprocessor-based devices.

Carrot measures

Rather than attempting to change the IoT software development process, we should use it to our advantage. The IoT manufacturers would be just as happy redistributing secure code written by other people than the insecure code they ship today.

Commercial and open source efforts are already underway to produce IoT software stacks that are better suited to their purpose than existing alternatives. Governments could 'nudge' these efforts toward providing secure software at comparatively low cost by issuing appropriately designed challenge grants.

Recommendation: Challenge Grants – Charter a body to offer challenge grants for development of secure IoT device cores applying modern security controls.

Stick measures

Even if all new software being developed is secure, this will leave a long tail of legacy systems that manufacturers will continue to produce until faced with an incentive to invest resources in developing something better. For example, almost every electronic garage door opener and electronic car door opener sold in North America and the EU uses a chipset originally developed in the 1970s. The cryptography used to implement the use of 'rolling codes' was known to be insecure when the systems were designed. The systems were broken in the 1980s and reports of this being 'discovered' have appeared periodically ever since.

Ford, GM, Chrysler are not going to invest a cent in improving the security of these systems unless they are forced to. Calling the CEOs to televised congressional hearings to explain themselves would be a satisfactory means of effecting change but one that would likely result in industry taking a very narrow view of the problem, only looking at the specific issues raised.

Recommendation: Liability - Legislate to place liability for IoT insecurity on a specific party. Which party is chosen does not matter provided that they have the ability to influence product development and can be held accountable.

Recommendation: Approvals - Work with existing approvals bodies to create an Underwriter's Lab or BSI Kite Mark for IoT device security.

Specific technical measures

The list of how to improve security is endless. The following measures are high leverage.

Egress control

The most common form of DDoS attack today is SYN flooding in which a network of bots attempt to overwhelm a target by sending a large volume of TCP/IP SYN packets. Each of which consumes resources at the target node. SYN packets are TCP/IP control packets used in establishing new sessions. Even under heaviest use conditions, an entire MIT dormitory is unlikely to source more than a few hundred SYN packets a second. In a DDoS attack, a residential connection attempts to generate millions of SYN packets per second.

Broadband Internet subscribers typically connect to the Internet through a residential gateway which typically provides a limited set of firewall functions. Adding egress control to these requirements would provide substantial benefits at negligible cost. Rate limiting sourcing on SYN packets and equivalents would have negligible impact on users except to render them a less attractive target for botnet herders.

Recommendation: Egress Control – Require broadband providers to implement egress control.

Managed code

While script injection attacks have gained in popularity over the past decade, buffer overrun exploits remain one of the top two vectors for network attack. Moreover, buffer overrun exploits continue to dominate attacks resulting in root access.

The solution to this problem was well established before the problem was created. Array bounds checking and other compile time security checks as supported in modern computing languages such as Java and C# provide a robust solution.

Recommendation: Managed Code – Require use of managed code in all challenge grant and approvals initiatives.

Overlay cryptography

One of the chief concerns raised at previous CCN conferences and in the wider industry community is the risk of ‘backdoor crypto’. Compromise of keys embedded in a device during manufacture represents a concentration of value attractive to a wide range of attackers ranging from nation states to organized crime. Offline key generation presents numerous practical and security challenges.

These concerns have traditionally favored key generation on the device after manufacture. But this has frequently resulted in devices generating keys based on a small pool of ergodicity meaning that they are easily guessed by an attacker.

A technology that I developed to address these concerns is Key Co-Generation which may be performed for any public key cryptographic systems in the Diffie Hellman family including elliptic curve variants. This is a simple protocol that takes advantage of a feature of the DH system which means that a party that only has knowledge of two public keys can determine the public key that would result from adding the two corresponding private keys.

Applying the key co-generation approach, an administrator adding a new device to their personal collection, generates a new key pair for that device which is sent securely to that device during provisioning of the device to their collection. The device then combines this key pair with the one provided during manufacture to determine the composite key pair to be used by the device in the context of the user's collection. This has the advantage that the composite key is secure against most forms of attack unless *either* the user *or* the manufacturer performed the process correctly[1].

The provisional patent on this technology has now expired without a full application being submitted. It is therefore unencumbered to the best of my current knowledge.

Recommendation: Technology Grants – Incentivize development of similar unencumbered technology proposals to extend the state of the art in cryptography as currently practiced with new inventions and by mining the corpus of existing work that is out of patent.

[1] There remains a small window of vulnerability in this protocol that a manufacturer may exploit if they are able to observe the registration process. This may be mitigated through additional controls in the registration process.