Moving Towards Best Practice for Living in the Internet of Things

The *Best Practice for Living in the Internet of Things* workshop took place August 3-4, 2017 in Seattle, Washington. This work was supported by the National Science Foundation under CNS 1565375 and through the generous support of Microsoft Research. The Security & Privacy in Informatics, Computing, and Engineering (SPICE) center at Indiana University and the University of Washington Tech Policy Lab provided additional sponsorship. This report is intended to communicate the majority view of participants in working groups at the workshop; in instances where the working groups' views differed, the report attempts to identify all points of view.

The goal of this workshop was to chart a path from the current woefully insecure state of the art to the state we envision. Reaching a secure state from the current state requires overcoming very serious gaps. This includes the gap between the cryptographic threat models and user risk preferences; the distance between cryptographic implementations and device capabilities; and between the usability of the devices in the home and expectations of the home's occupants. The method was an introductory ground-setting keynote followed by intensive collaboration leveraging the participants' collective expertise.

The exploration in the workshop was structured around defining the initial gaps, then focusing on the ground that must be covered to breach these for a secure IoT. The starting point of this report is an examination of the efficacy of six current sets of best practices to three well-known IoT security events. This analysis illustrates, as did the workshop's introductory discussion, that we are far from the ideal of a trustworthy IoT. Following this introduction, the core of the report is structured around the four focus areas from the workshop: (i) getting the cryptographic infrastructure right, (ii) building secure code on that infrastructure, (iii) enabling people to leverage the capabilities in that code safely, and (iv) enabling recovery when attacks happen. Finally, we close with four "next steps", which will ground our events over the next four years.

The goals and targets of the workshop were the following.

(i) Cryptography: Do we need specific standards for constrained devices? If there were agreement on a standard, diffusion would remain a challenge. Given the long-lived nature of some installations, how important is cryptographic agility? Within the larger challenges of cryptographic agility, what is the role and timeframe for post quantum? How important are even lower-powered block ciphers and hash functions? At what levels should APIs be focused? How much cryptography should be directly implemented in hardware versus software? To what degree do we need to provide security against sophisticated adversaries that own the devices? Should entropy generators be standardized in hardware, or should we use noisy sensors as inputs to traditional PRGs? We will solicit, prioritize, and attempt to answer questions along these lines.

(ii) Improving code and design: What is required for secure code in the different visions for IoT across the industry? What can be assumed in terms of trusted hardware? Do developers need readily available secure code examples, perhaps coordinating with the cryptography breakout group? Or do developers need libraries that implement secure communications? What is missing in terms of openly available and highly usable evaluation tools for the security and privacy of code on specific IoT platforms? Where should developer-centered research focus in the next four years?

(iii) Usability: Methods and standards for design and evaluation. There are multiple standards and heuristics for usability, but these are often not applicable to IoT, for example, because of the assumptions of modes of interaction. What tools exist or are needed for evaluation of the usability? What current heuristics apply to IoT? Are there any that apply to all the major proposed IoT platforms?

(iv) Recovery: Vulnerability mitigation, patching, and device isolation will be needed in the medium term, regardless of what happens in the long term. The vulnerabilities equity process makes assumptions that simply do not hold in IoT. Patching is not a solved problem. This breakout group will coordinate with the breakout group on usability.

These concerns are not cleanly separable. As the cryptography working group opined, what is needed from human-focused researchers is not so much frameworks but more detailed “useful access control models”. We did not choose to develop a set of best practices. Best practices now are aspirational, not operational. Even known, basic practices from the eighties are not adopted (e.g., unchangeable weak passwords in shipped products). There are current, effective best practices and standards efforts. The goal is to contribute to those efforts, rather than trying to create a new effort.

Our core findings identified the need for cryptographic agility, missing incentives for secure code, empowerment and protection for home consumers, and security transparency for all participants in the IoT ecosystem. One consensus recommendation was the need for case studies not only of large-scale disasters, but also near misses and smaller-scale problems.

(i) Cryptography: Crypto agility is the core of a future secure IoT infrastructure. This is not an insurmountable challenge. There are strong industry forces for (as well as against) improved security. There are promising developments, with physically unclonable functions increasingly well understood, and the fact that long-term reliable sources of entropy in the IoT are a solvable challenge.

One way to support better practice is to provide a taxonomy to understand which devices can participate in different spaces, in terms of specific security requirements. Once there is agreement on the needed requirements and thus components, there is a further need for provenance and pedigree of cryptographic components. If developers must understand cryptography to succeed, then we as cryptographers have failed.

(ii) Usability: One perspective is that if end users are required to think about cryptography, then developers have failed. Consumers in the IoT are not empowered; with even basic information about specific products, information about the security practices of producers is often absent.

One problematic component about usable privacy and usable security is that the goal of least surprise may imply privacy or may imply data flows for customization. The conflict between customization and data protection is not resolved for the creation of personalized compositional threat models. Notifications and warnings are also open areas of research. An ideal situation might be a culture of feedback that informs but does not overexpose nor overwhelm. Transparency and identification of appropriate decision points are core challenges in home-based IoT.

Recovery must be a component of this transparency. The total cost of ownership of any device includes recovery, cost of device, operator requirements, expected lifetime, cost of disposal, cost of access, data exposure, and cost of updating. This cost should be visible to the customer at the moment of purchase.

(iii) Recovery: Some participants argued that plans for recoverability must not wait for a distributed, robust, cryptographically enforced trust infrastructure, but must be built on what is available now; others argued that moving forward without such an infrastructure would be pointless or Promethean. It was observed that currently hubs range from effective vectors for recovery to ill-designed hindrances.

There are strong arguments for micro-segmentation. Patching a component for recoverability may not be feasible, yet potential for isolation exists. There was strong disagreement about expiration dates. There is no system for consumer support for when a company goes out of business, and expirations may exacerbate this. Environmental cost must also be considered when discussing expiration dates.

Success (and failures) in other domains can inform recoverability in the IoT. Given the history of laptops, we know that devices should have automatic online updates to the extent possible without requiring user interaction. This is aligned with the culture of feedback concept from usability.

Failsafe for a home-based IoT environment is not well understood: door open or closed; video available or disabled? Without that understanding, developers and designers are left with inconsistent or ad hoc requirements.

(iv) Improving code and design: The core topics in improving code were incentive alignment and programmer support. The role of platforms in the ecosystem was integral to both threads. Platforms have a larger, and critical, role to play in developer support, feedback, and code management. The ideal environment would make it more difficult to provide insecure code than secure code; this requires information and incentives. There are ongoing research and industry efforts towards this goal, as identified below. Documentation is a challenge for which progress is particularly needed. This is unsolved in desktop and mobile domains, and IoT arguably has greater annotations requirements, particularly for recovery.

Many of the observations transcended the boundaries between the four workshop-defined dimensions, so the details of the summary topics below may appear in multiple sections.

1 The State of IoT

From light bulbs to cars and refrigerators to children’s toys, consumer devices are increasingly connected to home networks and to the Internet. With this connectivity comes many benefits for the homeowner: Connected sensors and remote control can allow homeowners to, for example, save on heating bills while still coming home to a warm house on days when they break from their normal routine, turn the oven or stove on or off from their office, or unlock the front door when the pet sitter arrives. However, poor device security and unanticipated interactions also bring new opportunities for malicious actors and, to date, IoT security has been woefully inadequate. If homeowners can control thermostats, appliances, and egress then there is a risk that attackers can do the same. Even refrigerators play a role in global ecrime, providing denial of service for hire and credentials for sale (Holm, 2016; Geller, 2015; Morgner et al., 2016). Another major contributor to insecurity in the IoT is “smart” televisions, which give a new dimension to watching the watchers (Gavrilut et al., 2016).

A 2015 Federal Trade Commission (FTC) report points out, “…most of the sensors currently present on the market are not capable of establishing an encrypted link for communications since the computing requirements will have an impact on a device limited by low-powered batteries” (FTC, 2015, footnote 55: 13). As Clarke et al. note, the sheer heterogeneity of the IoT domain creates unique challenges, and this is exacerbated by the connectivity requirements (Clarke et al., 2014, pp. 2637).

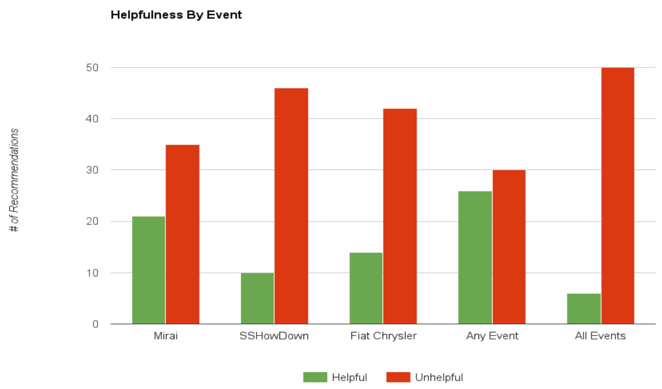
Research is needed on how to provide IoT security property reflection and, once extant, how to leverage this in a secure way to allow things such as automatic pairing and inclusion in networks of devices. Secure inclusion without a hub, a mesh network, or the user being able to program in the device beforehand would offer a foundation for constructing a secure home.

Guidelines exist for designing less vulnerable "things". Before implementing the workshop, Camp and Dingman did a high-level analysis of the state of best practices in IoT (Dingman et al., 2016). We considered six sources of "Best Practices" respectively produced by the Federal Trade Commission (FTC), the National Highway Traffic Safety Administration (NHTSA), the Federal Bureau of Investigation (FBI), the Online Trust Alliance (OTA), the National Institute of Standards & Technologies (NIST), and the Open Web Application Security Project (OWASP). Together, these best-practices documents offer 131 best practices that, after integration and removal of duplicates, result in 56 unique best-practice recommendations. This union of the recommendations is included as Appendix B.

Each of these 56 recommended practices was then evaluated against three recent large-scale events involving the (in)security of deployed IoT devices. We characterized as "helpful" those which, had they been followed, would have prevented, mitigated, or eased recovery from the attack; we characterized as "unhelpful" those that were followed but had no impact on incident existence, damage, or recovery or those which, even had they been followed, would have had no impact on the same.

The three incidents we examined were the Mirai botnet, SSHowDown, and the FiatChrysler vulnerability. Mirai, which captured the world's attention, was the most obvious choice. Mirai is a botnet that infects and conscripts vulnerable IoT devices like DVRs and IP cameras into a botnet for launching distributed denial of service (DDoS) attacks. The Mirai botnet was responsible for some of the largest DDoS attacks ever recorded, not only because of its size, but as a result of the sheer number of infected devices that are online at any given time; indeed, IoT devices like cameras and routers are rarely off and can, therefore, be leveraged to attack targets at any time (MalwareTech, 2016).

Over half of recommendations (30 out of 56) would not have helped in any of the three events. This does not imply that these are inherently bad ideas, only that the current best practices do not address the most common problems. For example, all the best practices documents consistently recommended TLS or other transport-layer encryption. This is important and should be part of any such document. However, in Mirai attackers used known credentials. In SSHowDown, attackers used default credentials or their default absences. Transport encryption was used; but only by the attackers. In the case of the automobiles, establishing the control channel required neither leaked data nor authentication. Figure 1 illustrates our aggregate findings in the form of a histogram.



Helpful or Not Helpful Best Practices

The essential fact is that we have not yet even begun to meet the minimal level where best practices can be adopted. Basic authentication practices dating from the eighties are not adopted (e.g., change the password). The same approaches that failed to address human and socio-technical factors in the relatively homogenous domain of well-trained engineers in a workplace will likely be inadequate for the more diverse reality of families at home.

The IEEE Building Code for IoT (Lindqvist and Locasto, 2018) took a different approach, with the goal of providing guidelines to system vendors and integrators, primarily in the form of reflexive design questions and needed outcomes. Several components of that report independently emerged during our secure code working group. In particular, our discussion aligned with the goal that "system owners and designers need a collection of procedures that establish a foundation for, and the presence of, these safety properties relatively quickly". Secondly, the recovery issue aligns with the insights in "managing obsolescence and sunsetting". Our discussions on contextually aware risk estimates were addressed in that report as goals for "flexible isolation and awareness of the risk of physical boundaries".

In addition to these results, there are continuing standardization efforts. The Alliance for Internet of Things Innovation (AIOTI) is focused on standardization and interoperability in the digital single market. The AIOTI has provided guidelines that connect the IoT with the EU's General Data Protection Regulation (GDPR), and these may offer a global model for securing information, but the current focus is on aligning policy not platforms. The Internet of Things European Research Cluster is a complementary but not formally coordinated research component for the IoT in the European Union and the United Kingdom. In contrast, in Japan the IoT Acceleration Consortium does not have security or privacy in the charter or the working groups. However, the overall focus on societal change will inherently address these.

Thing-to-Thing Research Group (T2TRG) is designed to build on IETF standards, and integrate the consortia and industry groups to take standards into infrastructure. It is distinguished from W3C Working Group on the Web of Things by scope and the goal of extending beyond network boundaries. There are activities in the area of security and interoperability but the extent of integration of these is not readily observable. We did not perceive a large degree of overlap with current or planned efforts.

As an extension or complement, the Open Connectivity Foundation exists specifically to resolve the complex combined challenges of interoperability and security.

Lack of interoperability is often claimed as security benefit; however, a direct comparison of an open hub and closed ecosystem found the security of the closed system much weaker. The merger of the AllSeen Alliance and the Open Connectivity Foundation (OCF) to sponsor the combined IoTivity and AllJoyn offer the promise of making credentialing and cryptography invisible to developers. There was a strong consensus with the need to provide developers with improved tools and interactions, and any effort from this workshop should complement, inform, and be informed by the OCF.

To complete the circle of organizations with industry, the Consumer Reports efforts place the concerns of the end user at the center of their system design. Consumer Reports brings to the standards efforts decades of unique experience in home safety, device reliability, and consumer expectations, while the industry organizations have greater strengths in networking and interoperability.

2 Cryptography for IoT Infrastructure

Infrastructure takes a very long time to change, yet choices for algorithms, key strengths, and lifetimes are being made today. We have already seen SHA1 in an IoT hub, so the urgency of addressing this recognition in practice is abundantly clear.

Yet strong cryptography has risks as well as benefits. PKI may be used to enhance anticompetitive approaches or ensure secure interoperability. In the first case, there are advocates for closed protocols and DRM-encumbered, walled-garden model of mobile devices, as players seek to lock consumers into one platform. Conversely, IoT device and appliance manufacturers have every reason to be interoperable with all platforms. Additionally, the major players (including Microsoft, Cisco, Intel, Cox, Verizon, and Comcast) are members of the Open Connectivity Foundation, mentioned above. At the network standards layer, the IETF has a working group designing a Manufacturers Usage Description (MUD) proposal that will enable whitelisting for IoT devices from any manufacturer.

Post-quantum cryptography was a major discussion point, due to the long lifetime of durable IoT goods like televisions, furnaces, water heaters, and refrigerators. One commonality in post-quantum cryptography is relatively large key sizes, and the resource-limited nature of IoT means that this may be a future constraint. Of course, noting the state of IoT above, IoT in the home will likely not be the site of early adoption of post-quantum cryptography. Hardware manufacturers may ignore or openly reject any such requirements.

One challenge for adoption of post quantum cryptography is the lack of universal agreement about which post-quantum standards are acceptable. Identification of such standards would go far to address the issue. While NIST has a complete process, adoption of early lightweight post-quantum cryptography standards does not seem likely in the near term.

As one contribution of this workshop, a strong opinion was that hash-based signatures have some potential to bridge pre- and post-quantum cryptography infrastructures; likewise, cryptography based on super-singular elliptic curve isogenies could potentially reuse existing circuit for classical elliptic curve-based cryptography.

One observation from the workshop was that discussions of post-quantum often stand in for larger discussions of cryptographic agility. Quantum computing may never arrive. Post-quantum cryptography has become not only a topic but also shorthand for agility in the face of other possible challenges. Not only fundamental mathematical breakthroughs (for example in factoring, algebra) but also significant changes in algorithms could require an immediate, agile response. The uncertainty associated with post-quantum cryptography should not prevent more immediate investments in cryptographic agility.

Questions on the appropriate balance between doing cryptography in software versus hardware are moving targets. From the industry perspective, nation-state attackers are not worth protecting against in the IoT home domains; however, nation-state attackers could use insecure IoT devices as a springboard to attack a country's critical infrastructure and so policymakers should consider how protecting against such attackers might be possible. Industry may focus more on protecting against well-equipped competitors than nation states.

A core challenge in IoT is entropy in low-power devices. Entropy is not easy, as noted by Lenstra and coauthors (Lenstra et al. 2012) in a widespread analysis of public keys. The generation of keys cannot be made outside of the expectation and contexts of use. And with the IoT, that context could be the lifetime of a durable appliance (i.e., decades). The same physical reality that makes the selection of algorithms and even storage of keys problematic also offers the promise for the use of physical sources of entropy. For example, lava lamps and cloud patterns can provide randomness (Noll, Mende, Susodiya, 1998).

Physically Unclonable Functions (PUFs) go a step further than simply providing entropy. An early and easily understood example of PUFs was a proposal to scatter fiber optics into paper so that a unique result occurred when the light hit the paper to authenticate its uniqueness. Moderns PUFs simply create outcomes based on physical systems that are easy to evaluate, but only when given the physical system. The output of a PUF should look like a random function so that it unpredictable for an attacker. PUFs can be affordable and are an area of needed innovation. Long-term reliable sources of entropy in the IoT are a solvable challenge.

Yet even with the additional physical dimensions and possible real-world sources of entropy, the question of the minimal amount of cryptography that is needed remains contextual, and thus an open question. What is needed is a taxonomy to understand which devices can participate in different spaces with specific security requirements. Such a taxonomy can define minimum requirements as well as the cost/benefit requirements for different choices. It could be used to classify cryptographic primitives (including physical components) according to their applicability to application domains and the requirements (e.g., the number of gates needed to implement them in hardware). The taxonomy could

An example of a true random generator for the physical world is the images recorded from the wall of lava lamps, which is one source of entropy used by CloudFlare.

be evaluated according to its ability to provide guidance to developers of low-power devices such as "rather than using the most common primitive for this job, the second most common may be a better fit as it uses far fewer gates".

Industry is addressing some issues and, in this case, the immediate need is communication with academics so that we can update our curriculum and training. For example, should lightweight ciphers like Simon and Speck be integrated into computer security courses; should we work to put these into pervasive and ubiquitous computing curricula?

There are already extant tools and methods that are not being adequately used in the IoT. Formal methods exist to verify the correctness of cryptographic implementations. There is a need for methods to establish the pedigree of cryptographic components (both hardware and software implementations). The further need for establishing a pedigree for code that is built atop cryptographic libraries or using hardware implementations is further addressed in the secure coding section.

The cryptography groups identified two open questions that must be answered by the usability group. First, what is the minimum requirements for a trusted base for an IoT device? Individuals should be able to express the degree to which they rely on these devices. Second, should devices have a wipe button? If so, how should such wipe functionality of triggered; for example, should it be a time-based wipe (e.g., wipe if the device is unused for a long period), if it changes power source, with credentials still on board should it be auto-wiped? Strong encryption versus short-lived key is an issue that combines usability and context, as does automatic re-keying, re-enroll, re-certification issues. In any case, due to a wide range of processing power and lifetimes, crypto-agility is critical.

3 Usability

A common belief is that if end users must think about cryptography, then we as developers and security professionals we have already failed. We have extended this at the workshop with the recognition that if developers must understand cryptography in the IoT, then the battle for security is similarly lost. The metaphor of airline travel arose, truly a modern miracle in which safety is such a routine that airlines compete (one using the Lord of the Rings theme, another using other famed actors) to engage those ensconced in that miracle to simply use their seatbelts. If everyone on the airplane had to understand the Bernoulli principle for air travel to function, it would be as hazardous to safety as today's Internet is to security and privacy. Similarly a secure and safe IoT at home can not rely upon individuals all understanding cryptography. Usability and security are not in opposition, and usability is achievable.

Usability goals have been met when a user receives the privacy they want with the security they need. Our concept of usability is grounded in the concept of least surprise. Least surprise combines usability, functionality, and risk. Humans cannot be kept entirely out of the loop and thus usability is a core problem. If a product is not usable it is not secure, and vice versa; if a person cannot control their settings or easily access an application, then they will try to avoid making those setting more secure because it is so difficult; if a person believes something is secure when it is not, then it creates a false notion of security that can be more dangerous than a person knowing that they are insecure. Building for such a goal requires understanding the expectation of the user in terms of privacy and security.

Usable security and least surprise may imply privacy or may imply data flows for customization. Usability ideally increases autonomy, meaning that ease of use increases one's abilities to control the technology without decreasing one's desired agency. If an individual cannot do anything about surveillance or information sharing, then that technology cannot be said to be usable. Yet data compilations can increase usability, when a system can personalize to usage patterns, it may be usable. Lack of privacy is likely to inhibit adoption so there is some incentive alignment with industry.

An important research arena for the future is understanding appropriate mental models (Mohamed et al., 2017; Lin et al., 2012; Cranor, 2008; Yee, 2002; Rimmer et al., 1999), as further addressed under usability. Users should not be thinking about cryptography per se; users cannot be expected to have an intuitive knowledge of access control but rather the access control should "just work" as expected.

The purchase of an IoT item is a commitment for an ongoing interaction with a company as data and code are shared. Consumers need information both about specific products and producers. Two concepts described mostly cleared by Solove (Solove, 2005) were found effective in our discussion: data exhaust and ethical debt. These were used to characterize item-specific and producer-specific information in a manner that included not only security and privacy but also other dimensions of contextual trust and data exposure. Interaction designers must be able to clearly articulate assumptions about access that can be transacted into actionable controls for developers: RBAC, ABAC, Chinese Wall (Jin et al., 2012; Yuan et al., 2005; Ferraiolo et al., 1995; Brewer and Nash, 1989). These clear models can then be a foundation for clear non-technical communication to those living in the IoT home.

Meaningful options and meaningful communication of these options are necessary for living in the IoT. People have the initial option to not purchase or install an application if they do not agree with its permissions. Users are rarely given the option to change these permissions. Once they have become used to or dependent on an application, they typically have no choice when it comes to accepting changes in terms and permissions. We contrasted the easy availability of GPS and the embedded tracking of users in iOS and Android. Meaningful communication requires both meaningful options and accurate product and producer data. With agency, it means that a person has control of their privacy. With usability, it means their perception of control is correct. People come with different capabilities and understandings of how to use different platforms. Thus, two people with the same system will have very different experiences of agency and different expectations. People may also interpret the usability and the agency of a platform differently, with one feeling overwhelmed and one feeling empowered by the same options.

The support needed for the homeowner must be adequate for nontechnical, busy people to meet the expectations of the system. Having acknowledged that there was general agreement that the first line of defense is in the home, discussions started on risk level at networks based on device type and what kind of scams are mapped to the home environment. The ISP's may be able to inform customers about a potential infection in a network via email, text message, or phone calls, but these notifications should be sent through a device that is not infected. The problem is how to indicate which devices are infected. Presently, such notifications are far too generic; effective notices require specific guidance on how to fix the infection. Unless the consumers are aware of the type of the problem, they will not be able to solve it.

Different stakeholders have different usability needs. One person may value privacy over security while another may value the opposite. Yet one person's purchase will impinge another's person's security in a shared space. IoT and other smart devices affect the interactions between space owners, controllers, and anyone who is in the space. All parts of IoT and all users are responsible for and affected by installation, maintenance, and disposal. People who use space that contains an IoT device are affected.

Not only may there be a discontinuity between the owner and occupant of a space, there is a strong discontinuity between producers and consumers. Visibility and ability to negotiate spheres of control is a critical issue for IoT. Yet visibility requires prioritization, and this, in turn, depends on automation of the creation of personalized compositional threat models. Individual hubs are the natural place for the compositional threat models that are needed to inform recovery. For example, if a deadbolt is programmed to open based on a voice command from the home audio interface (e.g., Amazon Alexa), that happens to be physically located near a home's entranceway, then it may be possible for a visitor to issue voice commands to unlock the door from outside, which is clearly a security flaw. The flaw here is not with the Alexa system, which makes no security claims, nor is it with the door lock, which has an encrypted and authenticated channel to its hub within the home. It is with these in combination and in context. Any rating systems for security must include the matrix that shows the devices it is used with.

Notifications still exist in several forms and yet, in their current form, they have proven inadequate for the IoT context. Once a home is infected it is easy for devices to then be commandeered to install more malware. What notification works? Actionable notifications are needed yet on the web we are still engaged in general guidance. Because of the shared nature of spaces, there is a need for social feedback and human negotiation.

The ideal situation would be a culture of feedback that informs but does not expose nor overwhelm. All participants had experienced the phenomena of a password policy being experienced by a legitimate user as a denial of service attack against legitimate use. Social engineering and cognitive limits as models illustrate how a failure to design for usability is an attack vector. In social media, usability surpasses security and usability is a cost to security and vice versa. The design, implementation, and application of policy should be clear on what is and is not secure. It should be clear what it means when their devices or applications are attacked. Open questions include the role of friends and neighbors. There should be a cumulative price of risk and exposure. Make sure that people, for example, know that ignoring a problem will cost them more later. A cumulative price for being insecure was also addressed in the recovery section, where the total cost of ownership included stakeholder resources.

When people connect themselves to the Internet they put themselves at risk just as when driving an automobile or engaging with other technologies. When people isolate themselves, they decrease their risk level from the technology but may increase other risks, e.g., home security or health. The allocation of risk and potential to place others at risk could not be resolved at this event. The group discussed the fact that meaningful ratings may not be possible across domains or even stakeholders. One possible response to the limits of ratings is to focus on a path forward, that provide both ratings and appropriate metaphors. (Denning et al., 2013) As one participant noted, “We need pathways to change, not more best practices.” An end goal for that path would be a culture or security or a culture of safety. Security may not be measured the same way safety is, particularly given the interaction of security and safety.

Security is a critical part of any device and we must build it correctly in order for any product to succeed. Making code quality a measure of security may be a necessary next step. There are key points when a consumer does have an option (purchasing, installation, and maintenance) but the consumers either do not know how to make the decision or cannot make it. Transparency and identification of appropriate decision points are combined challenges. Most applications and products do not offer the consumer a clear and understandable way to view varying privacy levels. Different transparency points are also critical to the recovery of systems.

4 Recovery

The recovery group began by addressing the basic definition of recovery. For an individual participant, it may simply mean resetting. If it means returning to the previously uninfected operational state, this would simply create a cycle of infection, recovery, and vulnerability. Regardless of the definition, the recovery group suggested that recoverability will get you through times of no crypto better than times with crypto will get you through times with no recoverability. Thus, plans for recoverability must not be delayed waiting upon a distributed robust cryptographically enforced trust infrastructure but rather built on what is available now.

The cryptography group did not fully concur with this, asking how one recovers when most private and personal information has been and continues to be handed to attackers, or if a safe state is unknown. This reflects a core reality of recovery in the case of IoT interactions between concepts of device security and information security. Without an initial safe secure state, recovery is only a first step to security. Additional hardening and improvements are also required.

Patching a component for recoverability may not be feasible. In these cases, risk mitigation will be required. Such mitigation may be based on isolating or limiting the functionality of an affected device. If these options are chosen for recovery, this may be indistinguishable from subversion from the perspective of the homeowner. Unexpected behavior could come from subversion or recovery, and the homeowner should be able to distinguish these.

With appropriate safeguards, either in terms of threat modeling to identify key risks or additional defenses to mitigate vulnerabilities, it may be appropriate to not update systems even with known vulnerabilities. A system that cannot be updated requires containment. A system that need not or can not be updated should be purchased with the assumption of lack of reliability and confidentiality. Every system has assumptions built around it—and security vulnerabilities are likely violations of those assumptions.

There was disagreement about the concept of an expiration date. The support for expiration dates is that these enable clear guarantees for consumers. When a consumer can trust a device for a known time, that consumer can benefit fully from the device. Expiration dates can provide management cycles that allow consumers to plan. The only responsibility of the consumer is to replace a device after it expires, which is well within consumer capabilities. Without expiration dates, the risk includes consumer expectations of end-of-life markets that never materialize.

The opposition to end of life as a recovery ideal is that a primary function of such a standard is mandating new purchases. It also creates strong perverse incentives to build devices that cannot be patched, must be disposed of, and thus industry sets the time for consumer purchases. Consider the case of iPhones, where older phones can't connect to App Store. This illustrates that forced upgrades are feasible on the high end; however, older phones continue to be used in an insecure mode with untrustworthy stores. In addition to the perverse incentives, there are issues of recycling. Drowning in physical devices negated by code is made more important by the fact that such products will be disposed of en masse. Expiration dates address the issue of security but create strong perverse incentives, and come at vast environmental costs. An alternative, which was to make a failure to update expensive through liability for failure, was also a subject of disagreement.

There are proposed solutions at different layers, transparency and isolation are both needed for recovery. There are strong arguments for micro-segmentation. Network segmentation is a popular solution to mitigate information leakage. Software-defined networks offer value in device-specific micro-segmentation. There is the network boundary model, with the manufacturer usage description (MUD) creating a whitelist for each device, giving each device its own constrained view of the network. There are LAN solutions, where devices in the home monitor each other. Similarly, the winners of the Federal Trade Commission contest for protecting the IoT in the home focused on transparency and isolation. The winning submission was mobile app, "IoT Watchdog" by Steve Castle which is designed to identify misbehaving devices: enabling homeowners to identify the devices on their networks, ensuring that only the correct devices are connected.. The other recognized submission, Persistent Internal Network Containment (PINC) system, focused on the isolation of devices not only in terms of their connection with outside devices, but also between devices in the home. These different approaches illustrate some of the challenges of recoverability, with the first providing actionable guidance by integrating the phone and all devices security monitoring into a single app. The second provides isolation and protection, but not user support for mitigation should a failure occur.

It could be possible for individual frameworks to compete on security, for example, could "works with Alexa" come to have implications with respect to recoverability? Here the value of distinguishing between approved devices versus interoperable devices becomes clear. There may be a role for internet service providers to offer recommended or approved devices based on their own observation of their ecosystems.

IoT is a space that attracts app builders, small manufacturers, and other innovators. One way in which IoT is not unique is that most innovations are ultimately unsuccessful. There is no system for consumer support for when a company goes out of business. There is an argument for code escrow, in case a company goes out of business. There is also an argument for a default for entities that no longer work with the dominant model; for example, defaults for different hubs if there is no MUD entry. This must be addressed. End-of-life issues were also discussed in secure code, as well as being identified in the IEEE Report (Lindqvist and Locasto, 2018).

Independently from the usability group, the recovery discussion also brought forward the issues of transparency. Recovery identified that there are necessarily multiple dimensions of transparency. The first is transparency at time of purchase. This component of transparency includes expected lifetime, end of life, and expectations of support. The second component of transparency is at vulnerability. The third is transparency at recovery, including guidelines for recovery. Should it be clear to owners if and when devices need to be updated? Or are updated? The final component is transparency at end of life. Communicating that a device is no longer supported is critical.

There is also a need to consider ecosystems of devices. This requires considering updates across multiple devices (e.g., some updated, some not, and the updated ones violate the assumptions of those that have not yet updated.) The total cost of ownership for a device may vary based on the IoT home environment with which it will be integrated. The total cost of ownership of any device includes recovery, cost of device, cost of education, expected lifetime, cost of disposal, cost of access, and cost of updating. This total cost should be transparent. Regardless of the approach taken, there is a need to inventory risk of devices, and devices in their particular context.

IoT requires thinking beyond access control to include information control and state control. Access control mode may change based on events, where state determines access. It is safety-critical for a door to open in the event of a fire, but it is equally safety-critical that it not be possible to completely open and unlock a house by cutting the power feed at the outside meter. Examining information flow may make it possible for the appropriate state to be determined, at which time access control decisions can be made. Information flow control may be a richer approach than traditional access control. Unlike personal devices, homes have “visiting” events. There was a focus on the risks of the Amazon in-home delivery service, but there are also guests, service providers, roommates, and family members. Valet systems for temporary access can resolve some of these problems, for example, adding a temporary code for one-time use restricted to a given time period. The natural approach from a computer science perspective is to enable predefined groups, meaning the creation of some general schema that people may fall into. Such a schema should be visible to the user as templates, specifically; the user must not be required to fit an ontology to their lives to get the correct results. Each family or household will have its own patterns and its own anomalies. For example, Airbnb guests and children would require different levels of access to IoT devices in a house and the children of Airbnb guests a third level. A household dealing with one should be able to detect inappropriate use, even when such uses would be appropriate in another household. These combine sociological issues with usability issues and technical issues to implement in the system. Recovery requires both determining the policy and enforcing the policy. The second may prove more difficult.

Most devices should still be able to function basically without being connected to the Internet, as an Internet outage should not be a complete denial of service event; moreover, if there is a connection failure, it will clearly be synchronized throughout the household. Thus, per-device recovery at any blackout would not be feasible for the individual household. Isolation itself may be useful as a form of recovery. The concept of having a flat tire, where there is limited ability to drive and then different cost of recovery; for example, a drive-flat tire that is resilient for fifty miles once punctured but must be entirely replaced by a new purchase, versus a spare tire in the car requiring immediate replacement, but then may be equivalent to the one that is replaced. What model is appropriate for connectivity loss or isolation depends on the device and the context. Failsafe for a home-based IoT environment is not well understood.

Given the history of laptops, devices should have automatic online updates to the extent possible, which are automatic, without user intervention, at an appropriate time/context. Manufacturers may decide to not have automatic updates as an option, without periodic user input, but that should not be a default without contextual justification.

Any hub should validate the legitimacy of updates (if possible) and firewall devices from external adversarial input. For any device in their system that has been compromised, a hub can provide multiple services. The first service is, of course, identification: a system should develop mechanisms to know if the device has been compromised. The second service is the enforcement of policies about outbound communications from compromised devices, e.g., firewall outbound actions from compromised devices. One possible forward is a central security manager running on top of any hub or router [Simpson et al., 2017].

Success (and failures) in other domains can inform recoverability in the IoT. The recovery group largely agreed, for example, that the issue of roots of trust is a solved problem. (The cryptography group did not concur.) In any case, issues of roots of trust, economic incentives, and challenges of end-of-life systems existed before IoT, and lessons learned from those domains should not be neglected. There are two primary sources of IoT devices. The first is manufacturers who made physical items and are new to connectivity. The security community could provide training or guidelines, from threat modeling to sample code. The second is Internet companies who are familiar with shipping fast and patching later. Learning between the “I” and the “T” requires reaching across industry boundaries. Consider the issue of manifests: how to create an industry-readable manifest; how to verify it; and how much can you detect tampering in manifests? The automotive industry is an example for manufacturers with expertise in certification and manifests. In that case, the manufacturer is responsible for what is in the box. The automotive industry understands how to ship with manifests, and the IT industry understands how to threat model and build roots of trust. Code integrity and verification is needed for recovery to be reliable.

5 Code

There can be no principle of least surprise if the developers themselves are surprised by their own code. There are many courses of this. Most development in the embedded space is not done at the command line, but rather in integrated development environments for opaque platforms. There is a tremendous amount of shared space, and in that shared space developers may inherit technical debt from previous projects. For example, if a developer wanted to make a privacy commitment for an IoT device integrated with a mobile app, such an assertion may not be possible. One potential way to mitigate this risk to developers it to make it possible to bring in the least amount of code needed; that is, support pulling in the least possible. Incentive alignment, education, and empowerment are all components of creating secure code.

There is a range of possible approaches to create incentives. One possibility is transparency, as addressed in both the usability and recovery sections. Such transparency can create marketplace incentives for security. At another extreme, there is strict liability, where the developers put themselves on the line for the code. There is a space between today's wild west and the strict regulatory environment of the Federal Drug Administration that would be optimal for IoT, but we do not have the data to locate this optimal zone. The workshop did not result in calls for specific regulation. Conversely, without some sort of pressure to do the right thing, security practitioners cannot make it easy enough to do software assurance. One source of incentive is in the platform providers, returning to the issue of interoperable as opposed to approved or recommended components.

The ideal environment would make it more difficult to provide insecure code than secure code; for example, embedding input validation into APIs. The next step is the provision of libraries for common points of failure, for example, trivial to use libraries that allow integration and validation of an implementation of input validation. This confirmed the discussion of the cryptographers' breakout session, which addressed the need for available, easy to locate, and easy to use libraries of primitives.

In addition to developer support, annotation standards are needed. Annotations are needed for review and confirmation, yet code documentation is an unsolved challenge. It is very difficult to validate code that is written without assessment as a goal, and it is rare to find such code. Providers of connectivity and hubs may seek a minimal level of annotation and documentation in code for approval or recommendation. Verifiability and verification of code is a research challenge.

One possible approach to the assessment of code is creating mechanisms for annotations, so that different stakeholders need not repeat verification. Annotations would also be useful in helping other developers compare, share, and leverage previously-written code. Annotation is a more difficult issue than correctness, as every developer would like correct, usable code but there is not the same intrinsic desire for annotated code. Best practices and community standards are components of the annotation challenge.

The ability to rank devices and toys on different levels of security could be improved by annotation. Large organizations struggle with annotation for their internal code, so this is a large challenge for open platform where there is no compulsory power over developers. The definition of a minimum for a given context could be an appropriate basis for developer-centric best practices.

Platforms have a larger role to play in developer support, feedback, or management. The ability of a platform to offer baseline security (e.g., Azure, AWS, iOS) is in opposition to the incentive to recruit as many offerings and recruit as many developers as possible. Coordination of a few platforms may be a more effective way forward than traditional regulation. Proactive coordination could improve overall security and prevent harm, there is a role for academic and industry cooperation. Such cooperation to address code flaws has the potential to prevent future unwanted regulation.

There is not a culture currently of sharing cybersecurity failures/near missed right now. With IoT, new companies are entering the IT market and are starting by repeating near misses and mistakes of other domains. Cross-industry coordination is a role for academic and public-sector leadership; however, finding the right scale and right people is difficult. One step forward to learning from specific cases is to curate case studies in security failures and near misses (Bair et al., 2017). There is much to be learned from traditional curating, library practices, and reporting requirement in physical domains. Currently, case studies are embedded in other publications, provided without adequate detail in press reports, and published in different domains. The participants saw these are closer to transportation reports of accidents and near misses, focused on engineers, further from disclosure requirements to users. The goal here would not be transparency to current users but to build a body of measurement-based, observed empirical cases in order to avoid future failures. Another dimension where IoT can learn from the large body of work in cyber-physical systems is in modeling physical effects of devices. One way to conceive of the code challenges is that what is needed are tools and systems to uncover effect on different levels of the stack, implications of engineering decision.

Individually verified code can be combined in a manner that creates emergent failures. As an IoT example, an early IoT light switch was coded using a library where the incorrect use of dependencies resulted in overheating and potential risk of fire. The issue in that was that the expectations of the interaction of the libraries was incorrect, in fact, this interaction of libraries is so common that library interaction results in long-term storage of master secrets in Android (Lee and Wallace, 2018). Defining developer expectation of code in highly variable context is an open question. The interaction of physicality makes this more difficult.

Improving the training and expertise of developers is a rich area for industry and academic collaboration. While the explicit university degree or industry certification models are both too heavy, badges have great potential. Developers with the skills to easily identify the minimal library and permissions, use the tools that provide the correct cryptography, annotate, and provide provenance are extremely valuable. Badges can be developed and monetized to improve code quality, and such a system could be incentive-aligned for all the stakeholders.

Secure Coding & Developer Usability

The usability and the secure coding groups had significant overlap, as developers are people too. Both breakout groups discussed the critical need to provide developers the tools needed to support nontechnical users in the home. From the developer's perspective, security needs to be in the architecture to be usable, usable tools to automate security in the developer process (e.g., SSDLC Vericode plug-in; TLS 1.3; EVREST, crypto toolkit). Bypassing security is easier than being secure in most cases. Working with the secure code and cryptography areas, we can make the most secure action the easiest action. (Seacord, 2005; Howard and LeBlanc, 2003)

Usability cannot be made distinct from quality of code, just as cryptography cannot be made distinct from usability if either is to be correct. Usability from a code and engineering perspective requires comprehensible annotation that identifies or supports developer identification of properties that matter in their own context. A coordinated effort is needed to make it easier to develop systems without the current common, even chronic flaws. Ubiquitous education of developers is a widely supported proposal. However, coding is a global industry from industrial development of cryptographic libraries to kids on tablets. Everyone will not be a security expert. If they are learning by example, there is a need for better examples. When developers seek answers to security problems, much of the online guidance is flawed and insecure (Acar et al, 2016).

From a developer perspective, the easiest thing to do is import entire libraries just for one function. This overuse of libraries and permissions is a chronic problem in mobile development (Felt et al., 2011; Lu et al., 2012). That this is known as a generic problem does not mean that each developer knows when they are engaging in this practice. Developers cannot know if they are deviating because the extent of inclusion of vulnerabilities via reference is itself an open research question. One possible step forward is regular reports on code use and library inclusion, analogous to transportation or other infrastructure reporting. Libraries have different constraints, different resource constraints, and update constraints. Even annotation which makes clear which libraries are being used and which are included for convenience would be an improvement to common practice.

Developer-centered design is a subset of the larger domain of user-centered design, and can be informed by that literature. (e.g., Sasse et al, 2001, Cranor, 2008, Camp, 2003).

6 Next Steps

The intensive collaborative workshop surfaced multiple directions where subsets of our participants believe high impact is possible.

First, is developer support for embedding cryptography, this combines the results from secure coding, usability, and cryptography working groups. Second is the need to move forward, however initially inadequately, in developing interactions and ratings. As an industry, we must move beyond warnings. The third is creating easy to locate verified secure code with visible provenance, which requires industry and academic research and practice. Finally, identification of the role of isolation in recovery is critical. Recovery that throws individuals off the network would be a security failure.

Infrastructure takes a very long time to change, and choices for algorithms, lengths, and lifetimes are being made today. We have already seen SHA1 in an IoT hub. Developers need point-and-click directions to the best cryptographic choice. Ideally, this would include guidance to developers of low-power devices that enable informed choices not just for keys but also for primitives. Not only would this encourage using some primitive rather than the most common primitive for reasons of cost, IC real estate, or power consumption. More than post quantum cryptography, infrastructure components need cryptographic agility.

One possible topic for a future workshop is collaboration towards moving cryptographic choices not only away from the end user, but also removing the burden from the developer. The combination of cryptographic agility and emerging post quantum standards means developers require at the least a front-end that chooses cryptographic algorithms and implementations given the device processing strength, the expected lifetime of the device. Common parameters are needed for this goal, including how to discuss the minimal capacities of a device. Such a tool could integrate the potential role of hash-based signatures to bridge pre- and post-quantum cryptography infrastructures. One way forward may be to come up with a taxonomy to understand which devices are capable of participating in different spaces with specific security requirements. The current work on community-driven open standards that integrate coding and credential provision by AllJoyn requires that any academic work be integrated in, or at least complementary to, that larger effort.

One difficulty in helping individuals make cryptographic choices is that these are embedded in certificate hierarchies; so, one part of this is making cert-chains more usable. Such an approach could also be reversed to indicate that achieving levels of security given developer's description of certain costs, power, memory, communication, or IC real estate (area). This reverse of these directions would allow vendors to quickly determine if they can provide the security they want with the hardware they're planning.

Meeting the principle of "least surprise", is a challenge when people are unfamiliar with the technology. Heuristic evaluation is a good starting point but can fail when there is no pre-existing knowledge about the human factors. We have begun by building on previous work in mental models and using that to explore interactions, then work with talk aloud protocols using Wizard of Oz, generating new hypotheses and interactions. Options for the GUI for IoT include ambient, embedded, wearable, tactile as well as mobile and traditional computing devices. The discussion of failsafe states and the human aspects of security and privacy included usability, but also identified stakeholder-related issues beyond usability. There are also numerous venues now focused on publishing short and long papers on smart home security and privacy. What is missing, in our opinion, is an opportunity to collaboratively workshop different solutions, with a diverse set of participants.

Straightforward coding failures are a chronic problem with mobile devices. StackOverflow and search results are a major source of insecure "solutions" to coding problems. What may be needed is collaboration between the major platform providers—e.g., Google, Microsoft, Apple, Amazon, Samsung, and others—to create a verified seal of security for code snippets. This must also include Google, Bing, Yahoo and StackOverflow to ensure that those verified solutions are the first developers see when searching. This is a valuable place to support the use of formal methods to verify correctness of crypto implementations. Today this creates problems with the apps that will interact with the IoT. It is also visible in our early analysis of hubs (as Sen.se opens a TLS connection, then sends the actual data unencrypted via a different web socket). Do we need libraries or training? Do we need code samples or card games? We are testing developer education through card games and plan to move towards a prototype badging system.

Finally, systems will fail. Recovery requires visibility and isolation. The power to isolate devices and the ability to easily identify vulnerable devices may be more valuable to attackers than defenders. Network service providers, network defenders, and consumer advocates are the critical stakeholders for recovery.

One step forward in convening a subset of the participants from those domains to create basic guidance on creating recovery systems when the solutions are more valuable to legitimate participants than attackers, even when these fail.

Acknowledgements

This research was supported in part by the National Science Foundation under CNS 1565252 and 1565375 and through the generous support of Microsoft. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the US Government, the National Science Foundation, Microsoft, University of Washington, Indiana University or

any specific associated participant. This document is intended the consensus of the majority of participants in the working groups.

We would like to acknowledge the participants including Kevin Kane, Carl Landwehr, Gianpaolo Russo, Kate Stewart, Shrirang Mare, Hamed Haddadi, Earlence Fernandes, Ivan Evtimov, Emily McReynolds, Jacob Abbott, Behnood Momenzadeh, Tony Skjellum, Allan Friedman, Katerina Megas, David Balenson, Joshua Streiff, as well as Roy D'Souza and Dave Mercer of Nestlabs. We also thank other participants from Microsoft, Shostack & Associates, and SRI International. We particularly appreciate the financial support of Microsoft Research.

References

Please note that this is in no way a comprehensive literature search on IoT security or related topics. Rather this is the set of references and other works which were explicitly noted in the discussion.

[Acar et al, 2016] Acar, Yasemin, et al. "You Get Where You're Looking For: The Impact of Information Sources on Code Security." *Security and Privacy (SP), 2016 IEEE Symposium on*. IEEE, 2016.

[Bair et al., 2017] Bair, Jonathan, et al. "Voluntary Reporting of Cybersecurity Incidents." (2017).

[Brewer and Nash, 1989] Brewer, David FC, and Michael J. Nash. "The chinese wall security policy." *Security and privacy, 1989. proceedings., 1989 ieee symposium on*. IEEE, 1989.

[Camp, 2003] L. Jean Camp, "Design for Trust," Trust, Reputation and Security: Theories and Practice, ed. Rino Falcone, Springer-Verlang (Berlin) 2003.

[Clarke et al., 2014, pp. 2637] Clarke, James, et al. "Protecting the internet of things." *Security and Communication Networks* 7.12 (2014): 2637-2638.

[Cranor, 2008] Cranor, Lorrie Faith. "A framework for reasoning about the human in the loop." *UPSEC* 8.2008 (2008): 1-15.

[Denning et al, 2013] Denning, T., Kohno, T., & Levy, H. M. (2013). Computer security and the modern home. Communications of the ACM, 56(1), 94-103.

[Dingman et al., 2016] Andrew Dingman, Gianpaolo Russo, George Osterholt, Tyler Uffelman and L. Jean Camp, "Good Advice That Just Doesn't Help", 3rd ACM/IEEE International Conference on Internet of Things Design and Implementation (Orlando FL) 27-30 April 2018.

[EPFL-REPORT-174943, IACR, 2012] Lenstra, Arjen, et al. "*Ron was wrong, Whit is right*." No. EPFL-REPORT-174943. IACR, 2012.

[FBI, 2015] "Internet of Things Poses Opportunities for Cyber Crime." *https://www.ic3.gov/media/2015/150910.aspx*, 2015.

[Felt et al., 2011] Felt, Adrienne Porter, et al. "Android permissions demystified." *Proceedings of the 18th ACM conference on Computer and communications security*. ACM, 2011.

[Ferraiolo et al., 1995] Ferraiolo, David, Janet Cugini, and D. Richard Kuhn. "Role-based access control (RBAC): Features and motivations." *Proceedings of 11th annual computer security application conference*. 1995.

[FiatChrysler] "Fiat Chrysler Issues Recall Over Hacking." *https://www.nytimes.com/2015/07/25/business/fiat-chrysler-recalls-1-4-million-vehicles-to-fix-hacking-issue.html, 2015.*

[FTC, 2015, footnote 55: 13] *The Annual Federal Trade Commission (FTC)* (2015) footnote 55: 13.

[FTC, 2015] "FTC Report on Internet of Things Urges Companies to Adopt Best Practices to Address Consumer Privacy and Security Risks. " *https://www.ftc.gov/news-events/press-releases/2015/01/ftc-report-internet-things-urges-companies-adopt-best-practices*, 2015.

[Holm, 2016] Holm, Eric. "The role of the refrigerator in identity crime" *Cyber-Security and Digital Forensics* (2016): 1.

[Howard and LeBlanc, 2003] Howard, Michael, and David LeBlanc. "*Writing secure code*." Pearson Education, 2003.

[Jin et al., 2012] Jin, Xin, Ram Krishnan, and Ravi S. Sandhu. "A Unified Attribute-Based Access Control Model Covering DAC, MAC and RBAC." *DBSec* 12 (2012): 41-55.

[Lindqvist and Locasto, 2017] Lindqvist, Ulf, and Michael E. Locasto. "Building Code for the Internet of Things." *IEEE Computer Society*, 2017.

[Lenstra et al., 2012]] Lenstra, Arjen, James P. Hughes, Maxime Augier, Joppe Willem Bos, Thorsten Kleinjung, and Christophe Wachter. *Ron was wrong, Whit is right*. No. EPFL-REPORT-174943. IACR, 2012.

[Lee and Wallace, 2018] J. Lee & D. Wallach, "Removing Secrets from Android's TLS", *NDSS 2018*, San Diego, CA 18-21 February 2018.

[Lin et al., 2012] Lin, Jialiu, et al. "Expectation and purpose: understanding users' mental models of mobile app privacy through crowdsourcing." *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 2012.

[MalwareTech, 2016] MalwareTech, "Mapping Mirai: A Botnet Case Study", *https://www.malwaretech.com/2016/10/mapping-mirai-a-botnet-case- study.html*. 2016.

[Mohamed et al., 2017] Mohamed, Mona A., Joyram Chakraborty, and Josh Dehlinger. "Trading off usability and security in user interface design through mental models." *Behaviour & Information Technology* 36.5 (2017): 493-516.

[Mirai BotNet] *https://github.com/jgamblin/Mirai-Source-Code*, 2017.

[NHTSA, 2016] "U.S. DOT issues Federal guidance to the automotive industry for improving motor vehicle cybersecurity." https://www.nhtsa.gov/press-releases/us-dot-issues-federal-guidance-automotive-industry-improving-motor-vehicle, 2016.

[NIST, 2014] "Framework for Improving Critical Infrastructure Cybersecurity version 1.0", *http://www.nist.gov/cyberframework/upload/cybersecurity-framework-021214.pdf*, 2014.

[Noll, Menda, Sisodiya, 1999] Noll, Landon Curt, Robert G. Mende, and Sanjeev Sisodiya. "Method for seeding a pseudo-random number generator with a cryptographic hash of a digitization of a chaotic system." U.S. Patent 5,732,138, issued March 24, 1998.

[OTA, 2017] "Securing the Internet of Things; A Collaborative & Shared Responsibility." *https://otalliance.org/initiatives/internet-things*, 2017.

[OWASP, 2015] "OWASP Internet of Things (IoT) Project." *https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project*, 2015

[Rimmer et al., 1999] Rimmer, Jon, et al. "Examining Users' Repertoire of Internet Applications." *INTERACT*. 1999.

[Sasse et al, 2001] Sasse, M.A., S. Brostoff, S. and D. Weirich, "Transforming the `weakest link' --- a human/computer interaction approach to usable and effective security", BT Technology Journal, 19:3, Springer, 2001.

[Seacord, 2005] Seacord, Robert C. *Secure Coding in C and C++*. Pearson Education, 2005.

[Simpson et al., 2017] Simpson, Anna Kornfeld, Franziska Roesner, and Tadayoshi Kohno. "Securing vulnerable home IoT devices with an in-hub security manager." Pervasive

Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on. IEEE, 2017.

[Solove, 2005] Solove, D. J. (2005). A taxonomy of privacy. U. Pa. L. Rev., 154, 477.

[SSHowDown] "SSHowDowN Proxy attacks using IoT devices." *https://www.helpnetsecurity.com/2016/10/13/sshowdown-proxy-attacks*, 2016.

[Yee, 2002] Yee, Ka-Ping. "User interaction design for secure systems." *Information and Communications Security* (2002): 278-290.

[Yuan et al., 2005] Yuan, Eric, and Jin Tong. "Attributed based access control (ABAC) for web services." *Web Services, 2005. ICWS 2005. Proceedings. 2005 IEEE International Conference on*. IEEE, 2005.

Appendix A: Workshop Schedule

3 August 2017

Time	Event
12:00-1:30	Lunch (CSE 303)
1:30 -2:00	Welcome (CSE 305) Jean Camp, Yoshi Kohno
1:30 – 2:30	**Federal Standards & Guidance in IoT** There is on the order of a thousand pages of guidance from the Federal Government as regulatory, security, and defense agencies struggle with providing guidance on security for the Internet of Things. The opening keynote will provide a baseline for participants to allow each of us to locate and delve into the most appropriate standards. Allan Friedman
2:30 – 3:00	Coffee & Tea (CSE 303)
3:00 – 4:30	First Breakout: Common Ground The charge for the first breakout group is to create a menu of possibilities. Ideally, we create short term possibilities for effective actions that can be implemented and tested. This includes identifying possible champions to track the ideas through the final report.
4:30 – 5:30	Breakout Presentations Each group will present the tasks that we can set aside as being adequately addressed by existing documents. This is likely to be some minimal set of requirements.
	Reception (CSE Atrium)

4 August 2017

9:00-9:20	Flash Panel of Breakout Leaders Breakfast & Overview Distribution (305) Coffee Available until lunch (303) Each breakout lead will offer a topic from their own group that they feel requires reflection by another.
9:30-12:00	Breakout: Contested Ground & Gaps Coffee Available until lunch (303) Second breakout set on selected subtopics, the goal is to find conflicts within the breakouts
12:00-12:30	Breakout Presentations Each group will present the areas where there was significant dispute not only about the content of a best practice, but also about its effective existence.
12:30-1:30	Lunch (CSE 305)
1:30 - 2:30	Breakout: Working Groups Breakout groups reform, including smaller groups on new topics.
2:30 – 3:00	Coffee & Tea (CSE 305)
3:00 – 4:30	Final Breakout: Working Groups These breakout groups will be a mixture of previous groups, for cross-community dialogue. Gaps are often invisible until one tries to step into the space. Disagreements are expected.
4:30 – 5:30	Breakout Presentations: The Map Forward Each breakout identifies key research challenges, key directions for overcoming those challenges, new collaborations, opportunities for future collaborations and sign up for any working groups that might continue online.
	Drafting

Appendix B: Union of Best Practices

Development Practices

Security from design phase (SDL security)
Evaluate 3rd party component security
Use current protocols and standards

Device Operation

Disable UPnP
Lifecycle Monitoring, characterize operations to detect anomalies
Minimize open ports
Obscure firmware
Write-only logs
Tamper evident or tamper resistant
Secure sensitive message with device-based encryption
Disable unused 3rd party components and features
Eliminate multi-device credentials
Unique per-device crypto keys

Device Policies

Secure account recovery or secure and private reset
Privacy policy transparency
Lifecycle policy transparency
Encryption at rest
Minimize physical ports

Vulnerabilities

Vulnerability reporting system
Validate updates before patching
Apply patches as soon as feasible

System Operation

Network isolation / segmentation
Defense in depth (identified risks)
Prevent unauthorized access
Lifecycle Support
Transport encryption
DMARC policy with rejection
For devices with the processing power, include firewall functionality
Connection request notification
Restrict scope of dangerous operations
Encrypt all device messages

Privacy

Minimize data collection
Anonymize collected data
No PII in error messages

Threat Analysis

Threat modeling / risk assessment before adoption
Consider device fitness for purpose in threat analysis
Consider interactions with external/aftermarket devices
Penetration testing / security audits before shipping

Authentication

No default passwords
Allow / require password change
MAC safety
Secure password storage
Brute force defense
Credential change notification
Multi-user access control
Require strong passwords
Use two-factor authentication

Organizational Practices

Restrict/remove debug access when shipping
Allow disabling physical ports
Prioritize product security communication
Restrict administrative/root access
Dedicated security design staff
Powerful product security executive
Manufacturers should join industry consortia
Require security expertise in every team