Best Practices Would Make Things Better in the IoT

Article *in* IEEE Security and Privacy Magazine · May 2020

DOI: 10.1109/MSEC.2020.2987780

5 authors, including:

Behnood Momenzadeh
Indiana University Bloomington
4 PUBLICATIONS **0** CITATIONS
SEE PROFILE

L. Jean Camp
Indiana University Bloomington
278 PUBLICATIONS **2,428** CITATIONS
SEE PROFILE

Best Practices Would Make Things Better in IoT

Behnood Momenzadeh, Helen Dougherty, Matthew Remmel, Steve Myers, L. Jean Camp
School of Informatics and Computing, Indiana University, Bloomington, Indiana, 47405

***Abstract*—Internet of Things (IoT) security appears to depend on the kindness of strangers; including in discovering, disclosing, and mitigating vulnerabilities. Consequently, multiple organizations have published best practices for producing secure IoT devices. We analyzed two very different hubs, identified vulnerabilities, and detail how these best practices would have prevented these flaws.**

***Index Terms*—IoT, Internet of Things, Best Practices, IoT Security, IoT Best Practices**

I. Introduction

One response to the exploitation of vulnerabilities in IoT devices (including instances of well-known, avoidable bugs) has been the creation of IoT best practices. Would these best practices make a difference if followed? To answer this, we selected two very different hubs: Sen.se and Samsung. One system is arguably the most closed hub on the market, designed to interact only with its own sensors. The second system is highly interoperable, working with Amazon, Apple, and Android devices. The targeted markets are consequently very different, with Sen.se targeting specific vulnerable populations and Samsung offering interoperability to all. The hubs are organizationally different with one system from a new entrant targeting the IoT/cyberphysical systems domains and one from a large, established manufacturer moving into the IoT space. Through simple security evaluations, we found both had vulnerabilities. Unfortunately, these vulnerabilities are acute for each hub: the hub targeting sensitive populations is subject to data manipulation, and the one with the broadest interoperability is at risk for botnet enrollment.

Ideally, best practices should address the requirements necessary to provide security and privacy in IoT. Some of these practices are purely technical, part of the construction and design of the devices. Some requirements are inherently organizational, including the disclosure of vulnerabilities.

With a straightforward evaluation using standard readily available tools, we found a number of vulnerabilities that could have easily been addressed by known best practices. To examine the two very different consumer IoT hubs, we used a union of the best practices from the guidelines that existed at the time of the analysis, illustrating in which cases these would have mitigated or prevented the identified vulnerabilities. The extant best practices, if properly used, could have mitigated some of the vulnerabilities. Although there is a need for research on subtle issues in IoT security, even putatively advanced hubs fail to meet the lowest security standards provided by common checklists.

II. Motivation

Perhaps the most well-known and widely reported IoT failure is the use of hard-coded passwords that enabled the Mirai botnet. It was the subject of widespread press reports (e.g., [1]). Another powerful example is the vulnerabilities found in St. Jude Medical's cardiac devices. These could allow attackers to apply incorrect pacing or real-time shocks [2]. When the vulnerabilities were disclosed to St. Jude, they simply responded that the claims were untrue and the devices were not insecure. The vulnerabilities were later confirmed by FDA.

Both of these vulnerabilities are a result of failing to follow basic, well-known security practices. Guidelines for companies entering the IoT domain and companies engaging in Internet-connected devices from previous offline models attempt to codify the actionable security advice. Unchanged weak default passwords have long been known to be risky. The Consumer Reports' Digital Standard addressed the need for no fixed password with their criteria, that "requires the user to set a good password." [3]. Similarly, the need for an open vulnerability reporting system is a common best practice. Consumer Reports takes this further, with their criteria being, "The company is willing and able to address reports of vulnerabilities." The IEEE Building Code for the Internet of Things also identifies the need for reporting and responding to vulnerabilities under the rubric of *Managing Obsolescence and Sunsetting* [4]. At a high level the IEEE Building Code for IoT was developed to provide system guidelines in the form of questions. These questions would be useful for design and could have improved the hubs we examined. Other best practices take the form of recommendations instead of questions.

We repeated the first step in the analysis of the efficacy of best practices in preventing malware attacks in IoT by Dingman et. al. [5] Using the same sources, we derived the same set of best practices. The set of best practices is the union of the following sources: 1) Federal Trade Commission (FTC) [6],2) National Highway Traffic Safety Administration (NHTSA) [7], 3) Federal Bureau of Investigation (FBI) [8], 4)the Online Trust Alliance (OTA) [9], 5) National Institute of Standards & Technologies (NIST) [10], and 6) Open Web Application Security Project (OWASP) [11]. Each of these organizations has developed a set of best practices for secure and reliable IoT, albeit from different perspectives and for different domains. The resulting union of the 131 best practices is 56 unique recommendations. The IoT Building Code recommendations are included in our union of best practices; while the Consumer Reports' Digital Standard is in our evaluation. The subset of the 56 best practices that we consider can be categorized as addressing five areas: privacy and authentication (twelve), system operation (ten), device policies (three), vulnerability mitigation (five), and device operation (ten). Neither the internal threat analysis, auditing practices, nor organizational practices could be observed in our analysis.

Three of the best practices address development practices, which may or may not have been followed and cannot be determined without a business audit. Four of them address threat analyses and auditing requirements; eight identify best organizational practices. As a result these are not included in the following analysis.

Our goal was to use two very different hubs to evaluate the state of security in basic IoT hubs and, if vulnerabilities were found, to determine if the best practices would have prevented or mitigated those vulnerabilities. In order to do this we examine the hubs to identify vulnerabilities, disclose these, and determine if best practices would have been mitigating. To make this determination, we engage in basic testing and then attempt responsible disclosure. After this, we report on how the systems aligned with the best practices.

If the role of best practices is to prevent any security incidents this is a standard that can only be honored in the breach; that is, they will certainly eventually fail. However, if the standard for the value of best practices is that they would improve the current state of the IoT, then identification of vulnerabilities through standard security testing is an appropriate method of evaluation.

The framework by Zhang et al informed our threat modeling [12]. This research focused on naming and authentication, but did not address operational issues such as security of updates. Similarly the Manufacturers Usage Device, under development as an IETF standard, provides an effective mechanism for coordinating access control, but does not address other basic security measures [13]. Hossain et. al [14] focused on extending previous frameworks to include a taxonomy of vulnerabilities in IoT space. Their contribution was the creation the smallest possible list of security requirements on device, one which is subsumed in the union of best practices.

III. Focus of Study

There is not a widely accepted definition of consumer IoT hubs, nor a widely adopted testing framework. Here the definition of hub is any device that connects to other devices and provides a point of contact between these devices and the external internet while not providing an additional service. We sought to choose hubs that are very different for our analysis.

SmartThings and the Sen.se hub (which Sen.se has branded as 'Mother') provide the basic functionality to interconnect, monitor, and interact with other devices in the home. Neither device provides additional connectivity, so their characterization as hubs instead of devices is reasonable. Sen.se touts the Mother as a way to "protect your home from intruders"; Samsung advertises a variety of webcams, sensors, and door locks that pair with SmartThings. Both are marketed as providing personal safety, and it would be reasonable to expect some minimal level of computer security given this focus.

Beyond marketing the the IoT systems as protection against home invasion, the design and implementation decisions in SmartThings and the Mother demonstrate the manufacturers' concepts of an adversary as being inherently outside the household. This is implemented by using role based access control, with outsiders being untrusted and home owners most trusted. Each hub has a primary administrator and "owner" of the hub (in a technological sense, ignoring any economic transactions in which the hub is involved). The owner has complete administrative access to the hub and the devices with which it is paired. Through the hub's app (on the web or mobile), the administrator can establish relationships between devices, set up automatic actions of devices, monitor the system, and make changes at will.

SmartThings' other administrator accounts are invited and authorized by the owner and have similar control over the system. Because they are authorized by administrators (the owner or otherwise), Samsung assumes they are entirely trustworthy for the hub with which they were invited to interact.

The other category of trusted users is "members" or "guests" of the household in which the hub is placed. These are assumed to interact with the devices paired with the hub (and in some cases the hub itself), but they have no access to administrative controls and do not interact with the web interface or app. They are trusted to interact with the IoT network in the house and access its resources as well as to use the same outward facing network as the hub.

The trusted non-owners in the household are effectively placed at the same access level as the non-human members of this model: the devices connected to the hub. These devices are authorized by an administrator to be part of the network, paired with the hub under the administrator's control, and must be approved by the owner. Just like a household member, these devices should be allowed to use the network but should not have any administrative control over other devices or the network.

Sen.se refers to the devices (sensors) in its network as "cookies" and the hub as a "Mother". Only Sen.se sensors are compatible with a Mother. Sen.se allows multiple sensors associated with one hub to have different administrators. For example, sensors associated with one Mother may also be associated with another account in the app (say, a family member of the administrator) distinct from the hub. In that case the user associated with the sensor will not have access to the Mother directly, just that particular sensor and the information associated with it under the aegis of the phone app. The Sen.se use cases have focused on care-giving and home domains so the focus on distributed external access is certainly reasonable.

Comparison of hubs		
	Samsung	Sen.se
Access control of device	yes	yes
Zigbee 802.15.4	no	yes
Wifi 802.11	yes	no
LAN	yes	no
Complex access models possible	yes	no
Interoperable with many devices	yes	no
Interoperable with other systems	yes	no

TABLE I: Sen.se is a closed system with access control over only data. SmartThings can operate with many manufacturer's devices and has multi-level access control.

The final member in this trust/adversary model is the adversary themselves, the outsider. The outsider is not a member

of the household and not associated with a member of the household. Although not necessarily malicious, the outsider is nonetheless not entirely harmless, and their gaining knowledge of or access to the hub's network is unappealing.

In the next few sections, we describe our analysis of the security of each hub. We began with an out of the box experience, observing the interactions with the hubs via packet capture. We identify vulnerabilities known to exist is the protocols used by the hubs, and anomalies in how the hubs used different protocols. We then sought to manipulate the data as transmitted, and on the device itself.

A. Analysis

For each hub we began with the out of the box experience. We implemented an initial examination of the first connection using PCAP (Packet Capture). These initial packets were mainly used for setting up the device. We implemented a Man-In-The-Middle (MITM) attack to observe traffic. Access control is a widely agreed-upon best practice, so we also evaluated the access control model. After our analysis, we attempted to engage in responsible disclosure.

We document our method and results in each of these steps. We close by evaluating the union of best practices in light of our findings with these two hubs. We conclude that, despite being imperfect, the best practices would have had an immediate impact on security of the hubs had these been adopted.

B. Samsung Hub

The Samsung SmartThings can connect to a wide range of devices. It requires a direct connection via an Ethernet cord to the home internet router and accepts connections through wireless, Zigbee and ZWave from devices. Samsung also noted in the documentation that SmartThings supports Bluetooth LE, but the feature was not yet activated as we write this document. It interacts with devices from different manufacturers.

Our examination of the connection setup via PCAP (Packet Capture) data revealed multiple potential vulnerabilities and some eccentricities in the hub's 802.11 wireless traffic.

Each device may have its own way of connecting and communicating with SmartThings but for 802.11 devices the hub broadcasts a UPnP (Universal Plug and Play) SSDP (Simple Service Discovery Protocol) packet to advertise its presence.

Previous research by Fernandes et. al on Samsung SmartThings analyzed the mobile application of SmartThings in detail [15]. As a result our analysis focused on the simple installation of the device and its wireless interactions.

As soon as the hub connected to the internet, it immediately sent a DNS query for Amazon Web Services (AWS) server dc.connect.smartthings.com. The hub connected and apparently performed initial setup tasks before making another DNS request for fw-update.smartthings.samsung.com, also an Amazon-hosted server, to check for and install extended firmware updates. After installing the updates, it immediately sent another DNS query for the server to which it had just finished interacting. The transfer of firmware update was done through http with no authentication or integrity check on the network layer.

After completing that check-in process, the hub connects to the Samsung server on AWS. These data are sent using a connection protected by TLS v1.2. The certificate details show cryptographic choices that are not optimal. The integrity of the certificate is verified with a SHA1 hash. SHA1 has long been considered broken in theory in that the creation of collisions appeared tractable [16]; it is now also broken in practice [17]. This choice of hash function is exacerbated by the lifetime of the certificate. The certificate is issued in 2015 and does not expire until 2025.

We identified straightforward vulnerabilities in this out of the box analysis. DNS resolution is a well known attack vector. The firmware update data was not encrypted using TLS thus risking both the integrity of the data and the authentication of its source. It is worth mentioning that we could not reverse-engineer the firmware update entirely. In addition, any denial-of-service attack on DNS servers would lead to a denial of service for the purchaser of the hub. A hub that could not configure could not function.

One of the main concerns arising from our analysis is not a vulnerability of the hub, but a vulnerability of the whole system. Specifically, SmartThings uses SSDP and UPnP to interact with devices. UPnP is difficult to secure. The current recommendation for addressing vulnerabilities is to block the SSDP protocol on the router [18].

SmartThings *requires* devices to have UPnP enabled in order to connect to the hub. As a result, all of these devices have UPnP enabled, which creates a large attack surface across the entire system. There have been a number of CVEs dedicated to discussing vulnerabilities caused by UPnP libraries. CVE-2012-5958 to CVE-2012-5965 identify a vulnerabilities while CVE-2013-0229 and CVE-2013-0230 enumerate MiniUPnP vulnerabilities. Although these are not particularly recent announcements, many devices were still using outdated UPnP libraries. Since then, the number of devices has greatly increased [19]. The vulnerabilities reported in the CVEs above could be utilized to run arbitrary code on a vulnerable device. This amplifies the importance of vulnerability reporting and updating. For example, we integrated a Samsung smartcam as an example device to work with the SmartThings hub. The smartcam uses a vulnerable outdated version of UPnP library, and thus exposes other devices connected to the hub.

C. Sen.se Mother

Sen.se Mother can connect only to devices built by Sen.se to be compatible. It is designed to be a closed system. The market segment targeted for the smaller device we evaluated is the homes of older adults who have health challenges and are seeking to age in place. The company ships the device with a selection of sensors each of which corresponds to a feature that can be used in conjunction with the Sen.se mobile app. It is possible to extend any of the functions of the pre-made apps, or design new ones entirely using the Sen.se API and python library in the Sen.se ecosystem. The Mother is not WiFi-enabled; it is necessary to plug the device directly into the network using a provided Ethernet cable.

{"body": {"stateFrequency": 180, "smileLed": "255,0,0", "serverUrl": "in.sen.se", "rightLed": "255,0,0", "rfPower": 100, "leftLed": "255,0,0", "serverPort": 80, "maxConn": 24, "soundLevel": 50}, "type": "gateway", "resource": "state", "method": "post", "auth": "MO001EC0E7519F"}

```
{
    "body": [
        {"repetition": 0, "type": "sound", "probability": 100, "sequence": [
            {"soundId": "check", "soundLevel": 50}
        ]}
    ]
    , "type": "gateway", "resource": "animation", "method": "post", "auth": "MO001EC0E7519F"
}
```

Fig. 1: Packets we used to generate Mother's Smile and Playing a sound

Sen.se Mother can connect to its own sensors and devices, which Sen.se confusingly refers to as *cookies* in their literature. Sen.se sells discrete devices each of which is marketed and designed to have a specific purpose, although a single Sen.se device may have a range of different sensors (e.g., thermometers, motion sensors). The data from these sensors are sent through Mother to the cloud for analysis. We will refer to these as *sensors* from here on out; however, any comparisons with Sen.se documentation should translate this to 'Sen.se cookies'.

Again we began with packet capture of the configuration. We scanned the ports to find out more information about the services and operating system on the device, but found that unused ports did appear to be closed.

We connected Sen.se sensors to the device. When one or more sensors have been added to the app that controls the account which connects to the cloud, the sensors can be bound to an installed application. All sensor data that is received by the Mother is sent to the cloud, and the cloud updates the dashboard or mobile app. Actual decisions about notifications or alerts are made in the cloud.

According to the official frequently asked questions page on the website, sensors are able to hold data on board until in range of a Mother, at which point data will be sent to the cloud via the nearest Mother, "even if the Mother is not yours." This feature might raise a red flag to someone who may not feel comfortable having their data sent through the private network of someone they do not know and without their permission. It also offers a fairly easy method of attack. For example, consider a house with a Mother hub which uses a sensor to monitor when the front door is opened and closed. (This use is recommended by Sen.se to protect those dealing with dementia by addressing the common risk of confused departure from the safety of home.) If an attacker were aware of the Mother, the attacker can simply place another hub close to the house's door. It could notify the attacker that the home is empty; and could further block notification of a caregiver that a sundowning elder is outside, alone, and in a state of confusion. Conversely, some may not be comfortable with other users' data being sent to the cloud through their own private network without their permission. There is no security between the sensor and the hub: no authentication, no message integrity, and no confidentiality. It is simple to gain access to sensors' data, tamper with data, or send fake sensor data. As an example, the packet for playing a sound and generating a Mother's smile is shown in Figure 1.

The observations of the packet exchanges with the external network resulted in some surprising observations in how Sen.se uses TLS. The device does establish a TLS connection to the cloud server as soon as it is powered on, and uses it for the UDP DNS query. Instead, as soon as data are to be sent, the hub generates another connection which is completely insecure. Unlike the first socket, this secondary socket is unencrypted. We could successfully analyze and reproduce packets related to the accelerometer, Mother eyes (used for giving notifications), Mother smile (used for giving notifications), sounds, and also the packets confirming another sensor has been added to the network of the Mother.

IV. Applying Best Practices

Having identified the sources of the best practices in Section I above and implemented a security evaluation of the hubs, we now address the issue of the efficacy of these best practices. We itemize the best practices, and note if these were followed. If they were not followed, we check if there was a vulnerability associated with the failure to follow the best practice.

We can characterize most of the the best practices as either met in practice (with *yes*), or clearly not followed (indicated by *no*). Others were not determined in our analysis and are indicated by an empty space. Such practices may simply be inapplicable given the design of a hub, which is indicated as *n/a*. There are also some best practices which cannot be so simply classified. For those we include a letter, e.g., a, and then add corresponding text immediately under the table. There are two uses of the indicator to show that these are open questions. Use of the best practices by the IoT hub creators may have required additional investigation of these to find vulnerabilities; as it is, the easy to identify vulnerabilities verified the need for Best Practice adoption well before we reached that state of analysis.

Each best practice is followed by an indicator of the source, based on our enumeration of the sources in Section I.

Were Best Practices for Device Operation Followed?		
Best Practice	Samsung	Sen.se
Disable UPnP (3,6)	no	yes
Lifecycle Monitorings (2,6)		no
Minimize open ports (2,3,6)	yes	yes
Obfuscate firmware (2,4,6)	yes	
Write-only logs (2,6)		
Tamper evident or resistant (1,2,4,5,6)	yes	no
Secure sensitive messages (1,2,4,6)	*a*	no
Disable unused features (2,4,6)	no	yes
No multi-device credentials (2,5,6)	yes	
Unique per-device crypto keys (2,4,5)	yes	yes

TABLE II: Use of Best Practices for Device Operation

a) The Samsung hub communicates to the users' phones or tablets through the cloud, not through unprotected messaging protocols.

Were Best Practices for Device Policies Followed?		
Best Practice	Samsung	Sen.se
Account recovery, reset (4,5,6)	yes	yes
Privacy policy transparency (1,4)	no	no
Lifecycle policy transparency (2,4,6)	no	no
Vulnerability reporting system (1,2,4)	*b*	no
Validate updates before patching (2,4,5,6)	*c*	
Apply patches quickly (1,2,3,4,5,6)	yes	no
Encryption at rest(1,4,6)	yes	no
Minimize physical ports (2,4,6)	no	yes

TABLE III: Best Practices for Device Policies & Risk Mitigation

Both manufacturers fail to follow practices related to policy transparency. Privacy policy transparency requires an easily accessible readable policy.

b) At the time of this research, Samsung did have a reporting system. We used social networks to find the vulnerability reporting system. We would not consider it easy to locate nor prominent. This system is not reachable at the time of writing, but it was operational at the time of the research.

c) Please see the discussion in Section III-B where we describe firmware updates by Samsung before our disclosure.

Were Best Practices for System Operation Followed?		
Best Practice	Samsung	Sen.se
Network isolation & segmentation (2,3,6)	yes	no
Defense in depth (identified risks) (1,2,4)		
Prevent unauthorized access (1,2,4)	yes	no
Lifecycle Support (2,4,6)		no
Transport encryption (1,2,4,6)	*d*	no & *e*
DMARC policy with rejection (4)		
Firewall functionality (3,6)	yes	yes
Connection request notification (4)	yes	yes
Restrict dangerous operations (2,4)	yes	no
Encrypt all device messages (1,2,4,6)	*d*	*e*

TABLE IV: Best Practices for System Operation

d) Samsung does provide transport encryption. However, 'yes' seems a strong statement because the weakness of the cryptography, as noted above. Not only has SHA1 been considered broken, but also 10 years could be considered more than enough time to find a collision. Simply having a Boolean requirement is not adequate.

e) As we mentioned earlier, Sen.se does establish a TLS connection, but they do not use the connection for further communications. They establish another insecure http connection and use that connection to send and receive the data. Just like Samsung SmartThings, this is another example which shows that a simple check-box is not adequate.

Were Best Practices for Privacy Followed?		
Best Practice	Samsung	Sen.se
Minimize data collection (1,4,5,6)	no	yes
Anonymize collected data (4,5,6)	yes	*f*
No PII in error messages (6)		
No default passwords (1,2,3,5,6)	yes	yes
Allow password change(3,4,5,6)	yes	yes
MAC safety (1,2,3,4)	yes	
Secure password storage(1,4)		
Brute force defense (1,4)	yes	
Credential change notification (4)	yes	yes
Multi-user access control (2)	g	yes
Require strong passwords (3,4,6)	yes	yes
Use two-factor authentication (1,3,4,6)	no	no

TABLE V: Best Practices for Privacy & Authentication

In the union of best practices there was some disagreement about the need to require or simply allow password changes. We choose to use the weaker requirement, as it subsumes the other. A requirement to change passwords would be a stronger recommendation.

f) We can not completely verify if Sen.se anonymizes the collected data. The data that we observed (and changed) during our experiment did not include any clearly identifiable information. As with Samsung we did not examine the app.

g) Samsung addresses the issue of guests, as individuals to whom limited access should be provided. Given the target market is the home, this may provide a level of protection against hostile or incompetent visitors and other contextual potential vulnerabilities.

A. Vulnerability Management

Best practices, including those from Dingman et al. [5] as well as the IEEE Building Code and Consumer Reports' Digital Standard, indicate that it should be straightforward to report vulnerabilities and these should be acted upon. For both hubs, we found vulnerabilities using standard, basic security analysis. In both cases these were communicated to the organizations. Having reported the technical failures above we describe the success and failure of reporting vulnerabilities.

1) SmartThings Samsung: In January the research team contacted Samsung and received an immediate response. Specifically we were introduced to the security leads in the IoT and B2B teams, and provided public keys for secure interaction in the future. Their second response included a pointer to the then-active http://security.samsungmobile.com/smrreport.html. It is worth mentioning that we had missed

this link due to the fact that it was on the Samsung mobile webpage instead of Samsung itself. Samsung SmartThings was advertised on the Samsung website, while the vulnerability disclosure web page was hosted only on Samsungmobile's domain. At the time the page defined the format for reporting vulnerabilities. We submitted the report in their required format, both on their website and attached to the email thread. We received an immediate response.

The Samsung engineering team confirmed receipt of the report. We did not receive a formal closure of the matter. However, the Samsung SmartThings hub received an update which appears to have addressed some of the issues we mentioned in our report. We say that this appears to be the case, as one change observed was the use of TLS for updates on an installed device. The certificate used for communication between the hub and the cloud stayed the same.

2) Mother Sen.se: In May repeated attempts were made to contact the people on LinkedIn who would be responsible for Sen.se. The initial contact informed the founder of the company over LinkedIn. Specifically the research lead said, "My IoT group has found a technical issue which I believe could fairly be characterized as vulnerability. I am *certain* it is something you would want to fix. I would like to practice responsible disclosure." There was no response. There was never a response about this vulnerability from Sen.se to the researchers, on LinkedIn or any other platform. We delayed submission and publication. After one month an author met a senior employee of the investors in Sen.se and inquired as to his interest in this vulnerability. He expressed strong interest, as the same platform is being used for smart cities. In a smart cities context, the lack of authentication or integrity in the data would also be problematic but that threat model is not a focus of our research. The senior employee served as a conduit to Sen.se. The organization denied the existence of vulnerability; then when it became undeniable, they firmly responded that it would not be remediated.

As we mentioned before there is no security between the sensor and the hub: no authentication, no message integrity, and no confidentiality. However, the company continues to advertise for safety-related applications; for example, the Med-Peanut is targeted at medication compliance and monitoring. Our observations were that data are sent in the clear to other hubs, the cloud, and thus presumably caretakers and physicians. Mother does not encrypt nor secure any data during transmission. The request that we examine their other products was the final communication between our team and Sen.se, although we had responded asking how to build a productive collaboration.

V. Discussion

At the end of this evaluation, we concluded that best practices offered increased security, as even advanced IoT hubs focused on personal safety lack basic security practices. One open question is whether the vulnerability analysis we implemented is grounded in practice. Are the vulnerabilities noted here legitimate real-world concerns? We would argue that this was a most basic analysis using publicly available tools. Had the best practices been followed, perhaps an advanced analysis could have identified more subtle vulnerabilities. As it is, the vulnerabilities identified here are not in any way obscure, unpredictable, or even contextual or emergent. There is a wealth of research needed on contextual vulnerabilities that result from different instantiations; but there was no need for such work given the state of the market. One might argue that in terms of risk communication Sen.se is the worst case in that the Sen.se devices appear cleanly designed and are marketed to vulnerable populations, yet are insecure. In terms of disclosure, study of Sen.se was not the worst case in that there were no threats to the researchers.

Recall that in Samsung, updates were not encrypted using TLS. This vulnerability would have been prevented by the 'Best Practices for System Operation'. Another best practice which could affect these patches is 'Validate Updates Before Patching'. We could not reverse engineer the update packets and as a result could not forge a packet to contain a malicious update and as a result we can not verify or refuse update validation. It is worth mentioning that, there were chunks of Samsung certificate in the earlier parts of update packet which could suggest a verification with the server. A cryptographically weak protection using TLS was used to transport the data to the cloud.

The best practices suggest disabling UPnP under the 'Device Operation' category. This was not done.

Samsung did have a vulnerability reporting system and template. Once we located these (on a different domain than the device is offered) it was possible to report and confirm reporting of vulnerabilities. They were partially addressed in the next update.

Samsung SmartThings did not use Bluetooth to connect to any device; however, they had not disabled the Bluetooth on the hub. Although we did not find any vulnerability with Bluetooth, best practices suggest disabling the unused features under 'Device Operation' category.

In a clear violation of multiple best practices, Sen.se did not encrypt the data sent to the cloud. The best practices that could have prevented this vulnerability were 'Tamper Evident or Tamper Resistant', 'Secure Sensitive Messages', and 'Encryption at Rest'.

The vulnerabilities created by the choice of Sen.se Mother to use the closest hub to send data from sensors to the cloud could have been avoided by 'Prevent Unauthorized Access', 'Network Isolation and Segmentation', and 'Tamper Evident or Tamper Resistant'.

Upon learning of the structural vulnerability of the system, Sen.se declined to take action except for expressing a lack of confidence that they had resolved the problem in their other product lines. Given the ease of the attack and the domains in which Sens.se is marketing the hub we examined (health care and personal safety), this illustrates that problems remain in providing even the most minimal security in IoT hubs. In fairness we note that Sen.se clearly asserted in multiple communications that Mother uses 'Transport Encryption'; although no data is encrypted in transport.

There is not always an identifiable match between a given best practice and a specific vulnerability. In part this is because

some of the best practices may be read as a checklist, so that a weak or flawed implementation can arguably be seen as fulfilling the best practice. The case of TLS as used by Sen.se supports this conclusion. Another contributing factor, beyond lack of specific technical details, is that the union of the best practices essentially requires threat modeling that is implemented carefully and managed well. So while the sum of the best practices would identify and mitigate risks, a direct one to one match between practice and risk is not always possible.

The issue of vulnerability disclosure indicates that the least observable best practices may in fact be the more important. One use of these best practices could be that producers might identify, for each product, which practices are followed. This could use the model of a nutritional label, and like nutrition, be subject to sanction if the statements were found to be untrue.

VI. Conclusions

Currently, IoT security appears to depend on the kindness of strangers, including in disclosure and responsiveness to disclosure. Based on our results using the extant and available best practices would be a significant improvement. In some domains best practices are contested (e.g., passwords) and in others best practices are more statements of goals than practices that can be implemented. In the case of best practices for IoT there are identifiable results for the actionable items that would comprise compliance. We also identified a weakness in the checklist approach for best practices, in that Sen.se could check the "use TLS" box but provided no protection. Mother opened a connection, didn't use it, and left the data unprotected.

Specifically, our conventional penetration testing identified predictable security weaknesses in confidentiality of data and integrity. The vulnerabilities we identified were particularly problematic given the adversarial models implicit in the two IoT hubs.

Having discovered vulnerabilities, we described our disclosure efforts to determine if the companies were aligned with best practices in terms of disclosure. We then compared the vulnerabilities and their disclosure with the union of those best practices available at the time of our research. Examining the hubs using a lens of these best practices, we argue that the vulnerabilities are preventable and would have been avoided had best practices been followed.

Given that the security state of IoT in general is in a vulnerable state and devices are still lacking basic security practices, having vulnerability reporting systems is particularly valuable. As IoT diffuses, quick responses to vulnerabilities will be ever more important. As IoT implementations (presumably and hopefully) improve, research on emergent risks and contextual risks can add to these best practices to ensure that even the most vulnerable are protected. As future research we are exploring the creation of 'nutrition labels' to indicate the privacy and security of IoT devices. With IoT apps there is a baseline in the form of static analysis and permissions. In IoT hubs, the best practices can serve the same role.

Interoperability cannot be blamed for the vulnerabilities we observed, beyond those in UPnP. The closed system was less secure than the open one. We could inject messages into the Sen.se with no use of the app. We could alter the outgoing messages from Sen.se. Confidentiality and integrity were lacking. Sen.se is marketing the device we tested for vulnerable populations and health monitoring, which makes this particularly problematic. We have not examined the system of the same design targeted at Smart Cities.

The open system offered better privacy, it preserved confidentiality and integrity, but lacked security. The device used UPnP (a vulnerable protocol) and does not require a minimum version of UPnP in order to accept a device as compatible. In addition to this, this hub uses a questionable certificate hashing algorithm and lifetime.

The Sen.se hub declares, correctly, that the device uses TLS. Thus any standard that is simply a check box (such as proposed by the Internet of Things Alliance Australia (IoTAA [20])) would find Sen.se to have complied. However, there is no effective decrease in risk with the use of TLS because there is no protection for the data, in the home or outside of it.

In closing, we analysed two very different IoT hubs, and reported the results of that analysis. We illustrated when the current efforts at best practices would have had an impact were these best practices followed. We further show that a simple Boolean requirement for the existence of security features is inadequate. One positive that can be taken from our research is that the extant best practices could provide guidance for the next generation of hubs.

We would like to acknowledge Samsung and Sen.se for their communications about the weaknesses detected. We would like to acknowledge Mathew Millard for his contributions to the analysis of the Sen.se hub. This project is supported in part by NSF CNS 1565375 and CNS 1814518, Cisco Research, and Comcast Innovation Center. 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 National Science Foundation, Cisco, Comcast, Samsung, Sen.se, nor Indiana University. We also would like to thank reviewer three, as should the reader, for their comments significantly improved the paper. We would also like to thank Laura Calloway and Joshua Streiff for their contribution to improving this paper.

References

[1] Roland Dobbins. Mirai IoT botnet description and DDOS attack mitigation. *Arbor Threat Intelligence*, 28, 2016.

[2] Safety communications - cybersecurity vulnerabilities identified in St. Jude medical's implantable cardiac devices and merlin@home transmitter: FDA safety communication, accessed April 2019. https://www.fda.gov/MedicalDevices/Safety/AlertsandNotices/ucm535843.htm.

[3] Consumer reports and disconnect and ranking digital rights and the cyber independent testing lab and aspiration digital standard, May 2017. https://www.thedigitalstandard.org/.

[4] Ulf Lindqvist and Michael Locasto. Building code for the Internet of Things. *IEEE Computer Society*, Sept 2017.

[5] 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)*, 2018.

[6] Federal Trade Commission. FTC report on Internet of Things urges companies to adopt best practices to address consumer privacy and security risks. accessed April 2019. https://www.ftc.gov/news-events/press-releases/2015/01/ftc-report-internet-things-urges-companies-adopt-best-practices.

[7] NHTSA. Cybersecurity best practices for modern vehicles. *Report No. DOT HS*, 812:333, 2016.

[8] IoT poses opportunities for cyber crime, accessed April 2019. https://www.ic3.gov/media/2015/150910.aspx.

[9] OTA Internet of Things, accessed April 2019. https://otalliance.org/initiatives/internet-things.

[10] Adam Sedgewick. Framework for improving critical infrastructure cybersecurity, version 1.0. Technical report, NIST, accessed April 2019. https://csrc.nist.gov/CSRC/media/Publications/white-paper/2018/10/17/iot-trust-concerns/draft/documents/iot-trust-concerns-draft.pdf.

[11] OWASP IoT project, accessed Apr 2019. https://www.owasp.org/index.php/IoT_Security_Guidance.

[12] Zhi-Kai Zhang, Michael Cheng Yi Cho, and Shiuhpyng Shieh. Emerging security threats and countermeasures in IoT. In *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, pages 1–6. ACM, 2015.

[13] E. Lear, R. Droms, and D. D. Romascanu. Manufacturer usage description specification. Technical report, IETF Network Working Group, 2017.

[14] Md Mahmud Hossain, Maziar Fotouhi, and Ragib Hasan. Towards an analysis of security issues, challenges, and open problems in the Internet of Things. In *Services (SERVICES), 2015 IEEE World Congress on*, pages 21–28, 2015.

[15] Earlence Fernandes, Jaeyeon Jung, and Atul Prakash. Security Analysis of Emerging Smart Home Applications. In *Proceedings of the 37th IEEE Symposium on Security and Privacy*, May 2016.

[16] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding collisions in the full sha-1. In *Annual international cryptology conference*, pages 17–36. Springer, 2005.

[17] Russell Brandom. Google just cracked one of the building blocks of web encryption (but don't worry). *The Verge, The Verge*, 23, Feb 2017. Accessed Jan 2020.

[18] Shadi Esnaashari, Ian Welch, and Peter Komisarczuk. Determining home users' vulnerability to universal plug and play (UPnP) attacks. In *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*, pages 725–729. IEEE, 2013.

[19] H Moore. Security flaws in universal plug and play: Unplug. don't play. *Rapid7, Ltd*, 8, 2013.

[20] Chris Duckett. Australian IoT tick is to certify a device can be secure, not that it is: IoTAA, Nov 2017.

Behnood Momenzadeh Behnood Momenzadeh is a PhD student in Security Informatics at Indiana University Bloomington. He graduated from University of Tehran in 2014. In his undergraduate degree he focused on data structures and algorithms. His research in PhD revolves around Risk Communication, IoT Best Practices as well as Cryptocurrency Economics. smomenza@iu.edu.

Helen Dougherty Helen Dougherty got her Masters of science in Secure Computing from Indiana University in 2018. She received her bachelor's degree in computer science from Grinnell College in 2016. She is currently a cyber security engineer in Exxon Mobil. htdoughe@iu.edu.

Matthew Remmel Matthew Remmel completed his masters of science in Computer and Information Security at Indiana University Bloomington in 2017. He currently works as an Information Security Consultant at E-gineering LLC. mattremm@iu.edu.

Steve Myers Steven Myers was an Associate Professor in the Department of Computer Science in the School of Informatics and Computing at Indiana University, where he was also a member of the Center for Applied Cybersecurity. Steve Myers completed his PhD (2005) in the Department of Computer Science at the University of Toronto, under the supervision of Professor Charles Rackoff. While completing his PhD he interned in the Mathematical Research division of Telcordia Technologies (formerly Belcore) doing work on secure cryptographic voting. His research interests are in all areas of cryptography, and computer and systems security with a specific interest in phishing and newhetremorphic attacks. He has written tens of papers, led panels, and given invited talks in fields ranging from Cryptography and Computer Security to Distributed Systems and Probabilistic Combinatorics. He left his Indiana University position in 2018. samyers@indiana.edu.

L. Jean Camp L. Jean Camp is a Professor in the School of Informatics, Computing, and Engineering at Indiana University, in Informatics and Computer Science. For 2019, she was at University of California at Berkeley as a Visiting Scholar at the Center for Long Term Cybersecurity. She is a Fellow of the Institute of Electrical and Electronic Engineers. She is a Fellow of the American Association for the Advancement of Science, and has been inducted into the Sigma Xi honor society. She joined Indiana after eight years at Harvard's Kennedy School where her courses were also listed in Harvard Law, Harvard Business, and the Engineering Systems Division of MIT. She spent the year after earning her doctorate from Carnegie Mellon as a Senior Member of the Technical Staff at Sandia National Laboratories. She began her career as an engineer at Catawba Nuclear Station with a MSEE at University of North Carolina at Charlotte. Her research focuses on the intersection of human and technical trust, levering economic models and human-centered design to create safe, secure systems. She is the author of two monographs. In addition, she has authored more than one hundred fifty publications. She has peer-reviewed publications on security and privacy at every layer of the OSI model. ljcamp@indiana.edu.