IoT Security: Problems and a Centralized Adaptive Approach as a Solution

Philip Cothery
Eastern Michigan University

Jeremy Bauer
Eastern Michigan University

Priyanka Meharia
Eastern Michigan University

Justin Yee
Eastern Michigan University

Biswajit Panja
Eastern Michigan University

The Internet of Things (IOT) is the interconnection, and communication of technology devices. This leads to speedy advancements in technology, but unfortunately invites in room for major security vulnerabilities. One facet of the Internet of Things, is smart home devices. Smart home devices are those that are utilized within one's home to improve quality of life. As all these devices communicate with each other, more and more security risks are developed. In this paper we review the existing security issues and solutions for IOT, and propose and approach to centralize the communications through a singular hub, acting as a central command for the smart home devices, allowing that hub to be the main secure point in the smart home network.

Keywords: IoT security, smart home

INTRODUCTION

The "Internet of Things" is a vastly growing and ever changing technological revolution introduced into the world back in the late 1990s. It is a term to communicate the interconnection of technology devices. However in the years since its first use the Internet of Things has grown well past anything researchers could have ever imagined. In todays the internet of things is used to describe a large range of devices varying from single small sensors, to refrigerators, all the way up to full nationwide power infrastructures. However what people most closely relate this phrase with is the ever growing field of "smart" devices. These are devices that are used to improve our everyday lives and add a new technological advancement to doing everyday mundane tasks. These devices include things like our smartphones, which has essentially

become computers in our pockets, watches and fitness monitors, refrigerators, voice assistants, like Alexa, Siri, Mycroft, and google home, and so many more devices. These devices alone don't make up the Internet of Things however, all of these devices communicate with each other, sending signals and commands back and forth in order to accomplish a task. For example, you could set up your Amazon Alexa device so that when you say "Hey Alexa, turn on the living room lights." Alexa will send a request to your smart light bulbs in your living room telling them to turn on. While this is a simple example, you can set these devices up to do many more powerful things, like telling a camera to unlock your doors when you approach, turn your alarm off, and many other quality of life functions. While this all seems like a good idea, and that it makes life substantially easier, it also invites hackers and cryptanalysts to attempt and break your network.

Judging by the massive power and access Internet of Things devices have, like the ability to unlock your doors, and turn off your alarms, one would hope that they would be pretty resilient to cryptanalysts attempts to break into them. Unfortunately, this is not the case for many devices. With the ever changing and massive demand for these devices many manufactures have put security on the back burner, in favor of new improvements, features, and fast release times. This has left a huge hole in the security of these smart devices and your home that is using them. As the current system works, every device has to connect to each other and send data back and forth, often time unencrypted. This invites challenge from cryptanalysts, who either wish to prove their skills, or gain access to your network and use it for further criminal activity, like spying on and robbing you. This creates an urgent need for an improvement in the security of these IoT devices.

Multiple different proposed ideas have come over the years, however they have many security flaws. In this paper, we have proposed a centralized system that runs specialized software to be a secure middleman for all of these data connections to travel through. Allowing for a scalable system that is it doesn't suffer from a weakest link vulnerability. As all devices will connect to the main hub and the hub will then send data to the required devices. With this style of architecture, the hub can have very strong security and pick up the slack left behind by each device.

EXISTING WORK

With the increasing rise of devices involved in the Internet of Things (IoT), the need to find a security solution to secure the communication between devices is growing ever more prevalent. This is where IoTurva[1] comes in, as a proposed security protocol to secure the communication between devices. A large issue that comes with IoT devices is that most devices are developed by small fast paced teams, employed by large enterprises or small start-ups that normally leads to rushed deadlines and limited resources at the developers disposal, leading to developers that have little care about security and consumers who often don't care about or fully understand the risks at hand. IoT devices are often used to monitor or perform delicate tasks, and often communicate to each other through a simple if-this-then-that interface (e.g If thermostat reads temperature less than 20 degrees, then turn on heater). This interdependence between the devices opens up an exploit that cryptoanalysist can use to gain access and compromise your network of devices. Current Network Intrusion Detection Systems (NIDS) that are use to secure devices are ineffective on device to device communicate due to both their approach at blocking traffic, match-action, and their need for huge databases that hold malware and attack signatures to recognize malicious behaviors. The match-action system is a security style that involves the threat needing to be detected and then blocked, however it is hard for this system to distinguish between a genuine request from device to device and a malicious one.

IoTurva approaches the problem of device to device security through its two component system, Turva Gateway and Turva Services. Turva Gateway serves to be a connection point for all devices and to act as a sort of central command accepting in and sending out messages to and from devices. It also acts a a filter against anomalous network interactions. The Gateway connects is light weight and can be powered on something as small as a raspberry pi, and connects to the devices through a virtual switch. It is broken up into 3 modules; monitoring, classification, and enforcement. Monitoring searches the network and adds new devices as necessary, classification communicates with the Turva Services for the latest security

protocols to be implemented, and enforcement generates openflow rules for the network. Turva services are used to manage network security and device functionality. Services is broken up into two main components; classification model and contextual engine. The Classification model is responsible for deciding whether traffic is normal or malicious. The contextual engine is responsible for collecting information for devices and users, which it then passes off to the classification model to aid in its determination of normal versus malicious requests.

NIDS uses a set of states received from packet headers to identify whether the traffic should be sent through or dropped, which assumes each device to be independent and only allows a device to change its own state. However IoT devices often need to be able to communicate and change their own state as well as the state of the other devices, which lead to IoTurva's proposed security profiles, which use each devices actions to determine whether it secure, suspicious, or unsafe. It then uses these classifications, and data gathered from users and other devices to decide whether to let a request through or not. For example if the network has a CCTV camera that opens your garage door when it detects a car in the driveway, it may have its category set to suspicious, and in add on rules that may involve, time of day, location of owners smartphone, or numerous other security features to ensure that the car in the driveway is one that is supposed to be accessing the garage door.

The Internet of Things (IoT)[2] takes devices from our surrounding environment and makes them active participants by sharing information between each other. It is predicted that by 2020, there will be over 50 billion IoT devices. The rising number of devices poses new security, privacy, and trust threats. The phrase IoT can be boiled down into 6 categories: Consumer services (smart homes/objects), smart energy (smart meters and grids), smartphones and tablets, internet connected cars, wearable devices, and wireless sensor networks. A few of the main technologies used in IoT are, Radio Frequency Identification (RFID), Wireless Sensor Networks (WSN), and Cloud Computing. RFID can be used to transmit data wireless using tags that are essentially an electronic bar code. WSN are small autonomous devices that are distributed to monitor physical environment conditions. Finally Cloud Computing allows for data storage. The structure of an IoT network is comprised of three main parts, the Application Layer, Network Layer, and Physical Layer. The Physical layer consists of the actual hardware that makes the connections to each other through some form of communication technology (i.e. Wi-Fi, Ethernet, Bluetooth). These devices should all have some sort of Universally Unique identifier (UUID). The Network layer consists of network interfaces, communication channels, network management, information maintenance, and intelligent processing. While there is no standard protocol for IoT networking the most commonly used ones are MQTT 3.1 and Constrained Application Protocol (CoAP). This layers main responsibility is gathering information from the physical layer and sending the data to other systems on the physical layer or or outside the network. The application layer is used to store data, and communicate outside the device oriented systems through the use of different applications. One major security issue of the IoT is that due to devices requiring the ability to run on very limited power it isn't possible to use the standard TCP/IP protocol for the devices.

Security in IoT Devices is crucial due to the sensitive nature of the data that is passed around between devices, the network, and the users themselves. It is very important for the network to have all 3 of the major CIA principles (confidentiality, integrity, and Availability). Another key component for the security if the network is trust. There needs to be a level of trust between each IoT layer, as well as a level of trust with the user. IoT is vulnerable to a multitude of different types of attacks. Physical attacks attack the actually hardware (devices) on the network. Network attacks which involve compromising the network between the devices. Software attacks are the main type of security vulnerabilities, and consist of Trojan horses, worms, viruses, spyware, and malicious scripts. Finally there are Encryption attacks, which focus on breaking the encryption scheme that is ran on the system.

Proposed protection against these security vulnerabilities needs to be broken down into securing each of the 3 components of an IoT system. to protect the physical layer, IoT companies should look into implementing secure booting, device authentication, data integrity, data confidentiality, and anonymity protocols. At the Network layer protocols used should include data privacy, routing security, and data integrity. Finally the Application layer can be secured using Access control lists, firewalls, data security,

and anti-virus/spyware/adware programs. All layers would benefit from the implementation of risk assessment, intrusion detection, securing of the premises, and a trust management system.

The Internet of Things contains many different types of devices, of varying different complexities and sophistication. The networks that these devices use to communicate are as broad and complex as the devices themselves. These are two main reasons why Internet of Things devices are so open to security vulnerabilities and attacks. In order to help secure our data and these devices we need to drastically change how we currently approach Information Technologies security. Devices should have End to end (E2E)[3] data protections because data is continuously getting shared between devices and remote networks. This increased traffic of data leaves it vulnerable to getting stolen more often. IoT devices are organically connected to each other, and those things are dynamically changing. In this situation it is crucial that devices maintain a certain security level, to keep outside rouge devices from connecting into the system. The process this would follow would be called Secure Things Orchestration. Security platforms for Multi-Level things should be supplied. The Internet of Things is made up of many devices, ranging from things as small as a simple sensor to as large and complex as smartphones and everything in between. If one device is a weak point in a system, then it offers a way into the network with the capability of moving laterally from one device to another. Because this is such a large issue, every device must have a secure SW execution environment provided. However, since the devices are all varying complexity and computing power, it would never be possible to write a one size fits all security protocol for all devices. So, we must develop multiple protocols to accommodate all sizes and complexity of devices. One of the biggest security flaws in the Internet of Things is the user. It is unrealistic to expect the user to know and understand the complex nature of security and privacy policies. This is why either easy to use systems need to be developed or automatically applied security protocols must be implemented

In this article[4], it provides blockchain as a solution to improve IoT security. “Blockchain is a database ledger that stores registry of assets and transactions across a peer to peer (P2P) network.” Blockchain has chained blocks of data that have been time stamped and validated by miners. The blockchain uses elliptic curve cryptography (ECC) and SHA-256 hashing to offer a strong cryptographic proof for data authentication and integrity. “The block data contains a list of all transactions and a hash to the previous block.” The design structure is made up of the block header and the block body which holds a list of transactions. The block header contains various fields, one is a version number to track software of protocol upgrades. The header also contains a timestamp, block size, and the number of transactions.

By design, data transmitted by IoT devices connected to the blockchain network will always be cryptographically proofed and signed by the true sender that holds the unique public key and GUID (Global Unique Identifier), therefore ensuring authentication and integrity of transmitted data. Furthermore, all transactions made to or by an IoT device are recorded on the blockchain distributed ledger and can be tracked securely. “Blockchain smart contracts have the ability to provide a decentralized authentication rules and logic to be able to provide single and multiparty authentication to an IoT device.” Smart contracts can also provide a more effective authorization access rules to connected IoT devices with less complexity when compared with traditional authorization protocols like Role Based Access Management (RBAC), OAuth 2.0, OpenID, OMA DM and LWM2M. The data privacy can be safeguarded by using smart contracts which set the access rules, conditions, and the time allowed for certain individuals or group of users or machines to own, control, or have access to data at rest or in transit. Furthermore, the smart contracts could give the right to update, upgrade, patch the IoT software or hardware. With blockchain, key management and distribution are eliminated. Each IoT device would have its own unique GUID and asymmetric key pair once its installed and connected to the blockchain network.

With the exponential[5] growth of devices connected to the internet, security networks is one of the hardest challenges for network managers. In this article they present two ways to improve security. First, they present a new SDN (Software Defined Networking) based architecture with or without infrastructure, that they call an SDN domain. A domain includes wired network, wireless network and Ad-Hoc networks. The second architecture consists of sensor networks in an SDN based network and in a domain. Lastly, they interconnect multiple domains to improve security.

To improve IoT security this article recommended a distributed SDN (Software Defined Networking) security solution. The concept of this architecture, first is a controller that manages the security of one SDN domain. Second, is to extend the first controller to include multiple controllers with respect to the available resources on each control platform. Third they extend the distributed control architecture by interconnecting all SDN domains via order controllers, which will lead to a secure model for the IoT. This architecture guarantees the security of the entire network with the concept of grid security embedded in each controller to prevent attacks. These SDN controllers behave like security guards of the SDN domain protect the network. Each of the controllers of each of the domains exchange their security rules. The controllers only know the policy of their own domains. When a node wants to communicate with another node from a different domain, the flow will be forwarded to the Security Controller, or Border Controller. The border controller will ask each neighbor controller if it knows the location of the information. With a security controller in every domain, it can prevent users from opening unauthorized services.

It is important that smart device firmware[6] is always kept up to date to resolve security vulnerabilities, improve functionality, add new features and fix bugs. In this article it talks about a Gateway architecture supported by web-services for automatic device and network configuration and automatic system updates to improve IoT security. For this gateway architecture approach, it requires a gateway and cloud-based services. When a new device is connected to the network, the gateway will use the device ID to interrogate a trusted web server to find the details of the device, like its functionality, commands, and any firmware updates that are available. Most auto-configuration approaches require a lot of information to be stored on the devices and for the device to be able to implement a deep protocol stack. However, with their approach a device ID and web service ensures that the information is easily available, and it can stay up to date.

Furthermore, the authors propose an approach that is like auto-configuration, but it relies on two key components. The first is to implement a web-based service. The service will be provided by the manufacturer or a trusted third party and it will be identified during the auto-configuration process. The web service will maintain the latest versions of software and firmware which can be pushed to gateways also identified during auto-configuration. The web service will be able to recognize vulnerabilities and download patches. The gateway will manage the update process locally. It can auto-schedule updates locally at appropriate times. The gateway would also be able to rollback information if something unexpected happened. In addition, it can respond automatically to vulnerabilities, for example by blocking network access to an unsafe device until a patch is available. Lastly the gateway acts as a firewall to protect smart devices from cyber threats.

The Internet of Things encompasses many interconnected devices[7]. One application of all these connected devices is the Smart Home Concept. In this concept devices that range from Televisions and Appliances to Smart Locks and Lights all communicate with one another giving total freedom, usability and control of the home to the user. Because of this amount of control, it is necessary that security and user authentication is of the upmost priority.

The way in which these devices connect with each other is through a protocol of communication layers. One way to describe it is as a "network of networks" (Khawla and Tomader 2) which includes the Physical & Data Link, Network, Transport and Application Layers. Data is generated by smart devices and shared between these layers every second, all of which could potentially contain important and private information. In addition, many smart devices found in a Smart Home use cloud computing for ease and convenience, however this opens up another opportunity for a breach in security.

The two main security threats to Smart Homes and devices therein can be categorized into either Passive or Active Attacks. Three common attacks are Denial of Service, Eavesdropping and Hijacking. These attacks can be made on specific devices, connected devices or the entire network of devices itself. Due to the amount of devices and connections that could potentially be found in a smart home there is almost no limit to the amount of security breaches and failures that can occur.

Various security solutions could be implemented to mitigate and hopefully avoid fully many of these threats to Smart Homes. These solutions can range greatly. One example is a "User Privacy-Enhanced Security Architecture" (Khawla and Tomader 5) in which personal information is only transmitted over separate private networks, through firewalls and message authentication. Another proposed solution is a

security monitoring system in which alarms can send notifications to the owner when devices are behaving irregularly.

With the rapid increase in smart technologies and interconnected devices that make up the Internet of Things, there is much to debate concerning the security of Smart Homes. A person's home is private and must be protected and respected. A breach in a Smart Home isn't just a breach into a person's private information but into their very life. All potential threats must be considered and all possible solutions explored.

Most smart devices, apps[8] and platforms are cloud based with many more moving in that direction. Cloud servers and cloud computing can be a very valuable technology that is both convenient for both the consumer and developer. However, devices and platforms that are too reliant on the cloud can be problematic when the question is asked: what happens when there is no connection to the cloud?

Internet speeds and connections have improved over recent years; however there is no service that has 100% up-time. It is a certainty that at some point, whether due to accident or natural causes, a connection to the cloud via the internet will fail in some form. It is at these points when the devices and platforms that rely on cloud connections become almost useless. Some of the most popular smart platforms such as Samsung SmartThings, Amazon AWS, IBM Watson and Microsoft Azure require internet connections to their proprietary cloud computers to work properly.

When a cloud connection is lost, not only is it a hindrance on the efficiency of the devices that rely on the cloud, but it also poses a security issue for the devices and the network they are operating on. Many cloud-based devices become vulnerable to hijacking attempts while their connections are offline. This means that not only is the user shut out from control over their devices, but that an unauthorized user may be controlling them.

There are many possible solutions to this problem. First, the smart platform should be able to detect the loss of cloud connections and properly distribute this information to the user as well as any devices on the network. Secondly, a service transfer in which the local hub takes the role of the cloud, but on a local network, until a reliable connection to the cloud can be reestablished. And finally, devices should be developed to utilize cloud servers can computing but not completely rely on such connections.

With the emergence of smart devices and the Smart Home concept[9], the network of IoT devices and those that make them are growing rapidly. The global market size for these devices is expected breach $10 billion within the next 2-3 years due to a larger amount and variety of devices that are being manufactured by more companies each year. Growing with the popularity of smart home devices is also the concern of security breaches on the consumer level. Recent studies indicate that some smart device development frameworks, such as Samsung's SmartThings, has already been proven to have multiple flaws, one such flaw allowing malicious Smart Apps more privileges than granted by the user. A suggested solution to this problem is a system concept called HoMonit, designed to work with Samsung SmartThings, and potentially other frameworks, to monitor smart devices and apps through encrypted traffic across a network.

One of the most popular smart home frameworks, Samsung SmartThings, is widely used by many developers of smart devices and apps. In this framework the hub mediates all communications between devices connected to the SmartThings framework and serves as a gateway to the cloud. SmartThings supports a variety of communication protocols including ZigBee and Z-Wave. However, several security-critical design flaws have been found.

The first and foremost flaw is over-privileged access. This comes from when an app requests access to a certain command or attribute of a capability within SmartThings device. Instead of being granted solely that attribute, it will always be granted that entire capability, and thus having access to other commands that it was not originally requesting and that the user may not be aware of.

The second most-critical flaw found in this framework is event spoofing. Event objects are handled in the SmartThings cloud and contain identifiers of the hub and the devices. A malicious smart app could gain this knowledge and spoof an event. Since this event is handled in the cloud, the SmartThings framework deems it as legitimate and it can then be sent to all smart apps and devices on the network corresponding to that event.

HoMonit[10] is a system that observes traffic sent to and from the cloud by eavesdropping on wireless communication packets such as ZigBee and Z-Wave. HoMonit is comprised of two components, one that extracts the expected logic of smart apps from their source code and another that identifies the misbehavior of apps by comparing inferred behavior based on the extracted logic and comparing it to the traffic gained by eavesdropping. It uses the Deterministic Finite Automaton (DFA) to characterize the logic of the smart apps from their source code. Then HoMonit uses a simple sniffer and Universal Software Radio Peripheral (USRP) to collect packets and monitor the app communications.

Successfully utilizing and implementing a system such as HoMonit could go a long way to making smart home frameworks more secure and thus making the overall experience better for the consumer.

PROPOSED APPROACH

It is extremely difficult to ensure that any device has complete security and can operate with no downtime or errors. It is even more difficult attempting to maintain a network of devices, which are constantly communicating with each other and cloud services through the internet with little to no human interaction, such as in a Smart Home. The Smart Home is a concept in which dozens of devices from laptops and TVs to refrigerators and appliances can operate in an efficient manner either through human interaction via user interface, app, or vocal commands, or no human interaction at all by reacting and predicting behavior by the user. In this concept dozens of connections are formed by all the devices on the network and the internet, with each of the connections being vulnerable to security risks, downtime and communication failures. This can be a challenge when considering that not all modern smart devices have adequate security and are manufactured by a number of different companies that utilize different standards of communication, operation and quality. We propose a concept as a solution to these problems in the form of a centralized communication gateway that would act as a moderator, observer and securer of a Smart Home.

The first problem, one of the highest priority, which our gateway would handle is security. Unfortunately, more often than not, security is not thought of with much importance by developers of software and hardware for smart devices. Many companies such as Apple or Google boast very impressive security on their 1st party devices as they have been operating in the technology and communication businesses for quite some time. However, many companies are now developing smart versions of their products, such as appliances and TVs, which have no background with technology and communication. It is with these devices where security becomes an issue. Due to the constant communication that forms the network of devices that make up a Smart Home, one breach in security, even with a simple device such as a smart refrigerator, could put the whole network at risk. How our device would handle this problem is by acting as a gateway. All communications in a Smart Home network, between devices and the cloud, would be initially sent and received by our device. This could be done with two different methods. The first, the passive method, would be to monitor the communications being sent between devices by eavesdropping on the encrypted packets passing through the gateway and comparing it to the expected behavior of the app sending this information. It could notify the owner if any communications seem irregular without inhibiting the speed of the network, however, it would not stop any malicious commands being sent between devices. The second, the active method would also monitor all communications passing through the gateway but would not allow any packets of data to pass through the gateway without being verified with expected app behavior. This would be the most secure method as it could theoretically stop all unauthorized communications as well as isolate and quarantine hijacked devices, however, this method would undoubtedly slow down communications across the network which lead to unhappy users. It is possible that the gateway could switch between the two methods during different hours of normal user activity such as when the user is away from the house or in the middle of the night. While possibly not completely secure, a Smart Home that communicates through a gateway such as our device would be much more secure than one without.

The next problem that our gateway would address is device incompatibility. With the rapid rise in demand for smart devices in recent years there has also been a rise in the number of types of smart devices

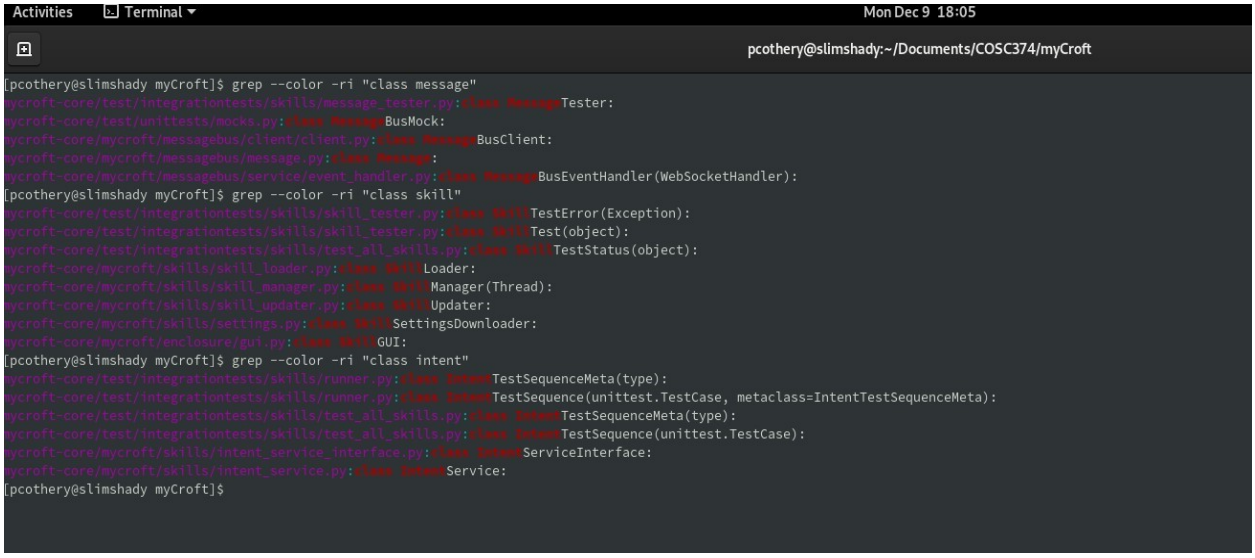
being produced as well as companies that produce them. If one were to assume that the average Smart Home would be made up of at least ten or more smart devices, it is not unreasonable to think that they would be made by different companies and therefore be utilizing different methods of communication and encryption. A large majority of smart devices today are made with the assumption that they will be interacting with popular brands such as Samsung or Apple, but not all devices are prepared to be compatible with devices lesser known companies. This can disrupt communications between smart devices. Our gateway would seek to nullify this issue by accepting all forms of communication from registered devices on the Smart Home network and repackaging the data in a more device-friendly form before sending it to the desired point.

Our last problem that our gateway would solve is the reliance on outside connectivity in the form of cloud connections. Most smart devices utilize or even completely rely on cloud connections to operate. Cloud connections are not always reliable for a number of reasons involving downtime from natural causes or interference. Our gateway could potentially negate this reliance by acting as a temporary cloud connection when one cannot be established.

It is through these three solutions that we believe our gateway concept would make Smart Homes not only more secure, but more efficient and reliable, all of which is demanded and expected by consumers in today's market.

ETHICAL HACKING OF IOT DEVICE

FIGURE 1
ETHICAL HACKING OF MyCROFT

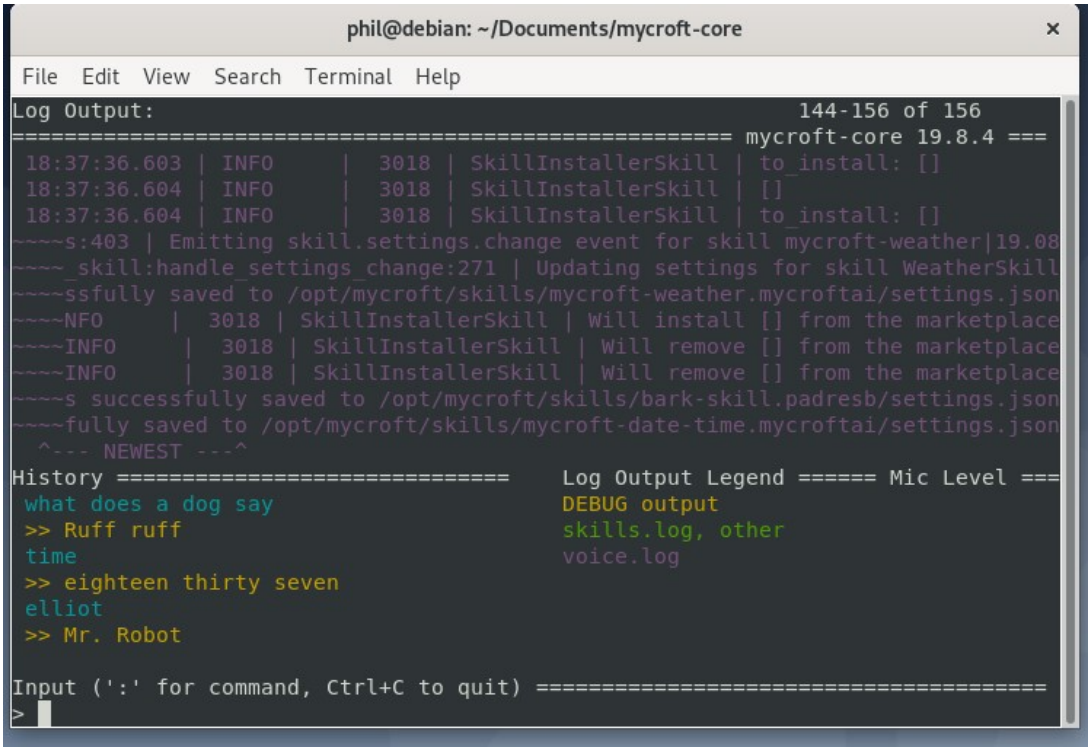


To show how hacking may work in IOT devices. We used MyCroft voice assistant. The first step we took to try and manipulate the MyCroft voice assistant was to download the open source code form MyCroft.ai. We then spent a few days combing through the different files to figure out what would be useful in our attempted take over of the device. We decided on attacking this device through the use of their skill implementation, which is done through skill creators uploading their skills on to github where then anyone could download them. This opened up room for vulnerability as there is no official quality control on these skills and if we could inject malicious code into one of these skills we had a high likely hood of being successful in taking over a device. With this information we all went on our own to download and install the MyCroft linux version onto our virtual machines in order for us to have an environment to test these skills and further our knowledge on them. After we each had MyCroft up and running and a few

sample skills installed we decided to open up the source code for these skills to see what they are comprised of. These source files are broken up into a few different files that consisted of "intent" files, which are the list of words and phrases you would tell your MyCroft in order to trigger your skill to start. There are also "utterance" files, which is the response templates that MyCroft follows for your skills, while we did change these around a bit these were not ultimately of much interest to our specific works. Finally, the most important file for us was the "__init__.py" file, the main source python file of the skill. This is where we can inject our code in attempts to alter the MyCroft's functionality. Upon diving into this python file, we found a couple things that we wanted to take a closer look at. The main one was the Message. We were able to tell that this was what we told MyCroft to get the skill to trigger, but we couldn't figure out how to deconstruct it in a useful way to us. This is what lead us to using the Linux command "grep --color -ri "class message" to try and find the source file behind what a message is. To break down this command, grep is a very powerful command used to find patterns of text inside of files. the --color modifier just makes it easier for humans to read, by color coating the findings, the -ri is two separate modifier flags "r" and "i", where the "i" tells it ignore case, or look for pattern without caring if it's the letters are lowercase or uppercase, while the "r" tells it to look recursively, meaning if there is a folder where you are searching, dive into the folder and check the files in there too. Finally the "class message" is the pattern or the string of characters that we are searching for in each file. So the command will return the file name and location (in purple), the pattern (in red) and the remainder of the line (in white) for every instance it finds. There is another field that we left off of our grep search, which allows you to specify where to search, however since we just wanted to search the current directory we were in, we didn't need this field. Below is a picture of this command being ran with a couple different search parameters.

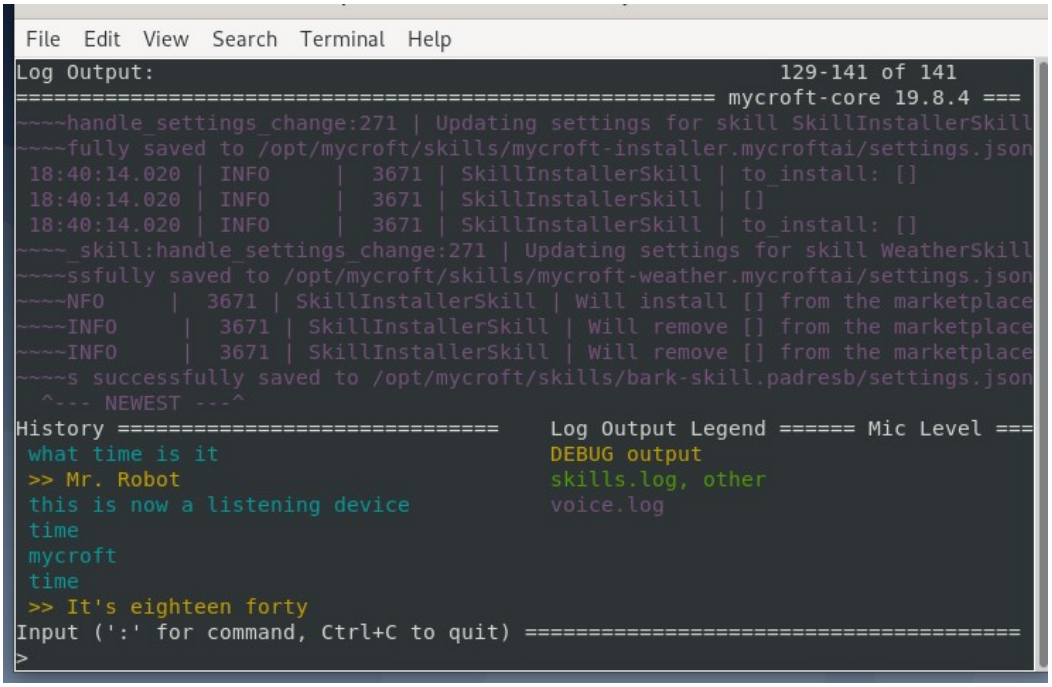
Moving on from that research, we decide to try our hand at manipulating skills. So we started off simple and added our own intent, utterance, and just some simple code telling our program that when it hears the name "Elliot" it will reply with "Mr. Robot" Below is a picture of this in action.

FIGURE 2
STEPS FOLLOWED FOR ETHICAL HACKING OF MyCROFT



We next wanted to see what we could add to this. So our next plan of attack was to take over the default skill intent of "what time is it" which to our surprise was not protected to the time skill, and if we coded a skill with this intent our skill would actually have higher priority. Through our viewing of source code we noticed a command for MyCroft that allowed it to wait for the user to say something inside of a skill, so that the skill can get get responses. We thought that an interesting use of this would be to put it in an infinite loop and turn MyCroft into a makeshift listening device. This allowed us to listen to everything that the MyCroft heard and put it into a text file on the machine. Below is a picture of the listening device in action.

FIGURE 3
DEMONSTRATION OF TIME MODIFICATION



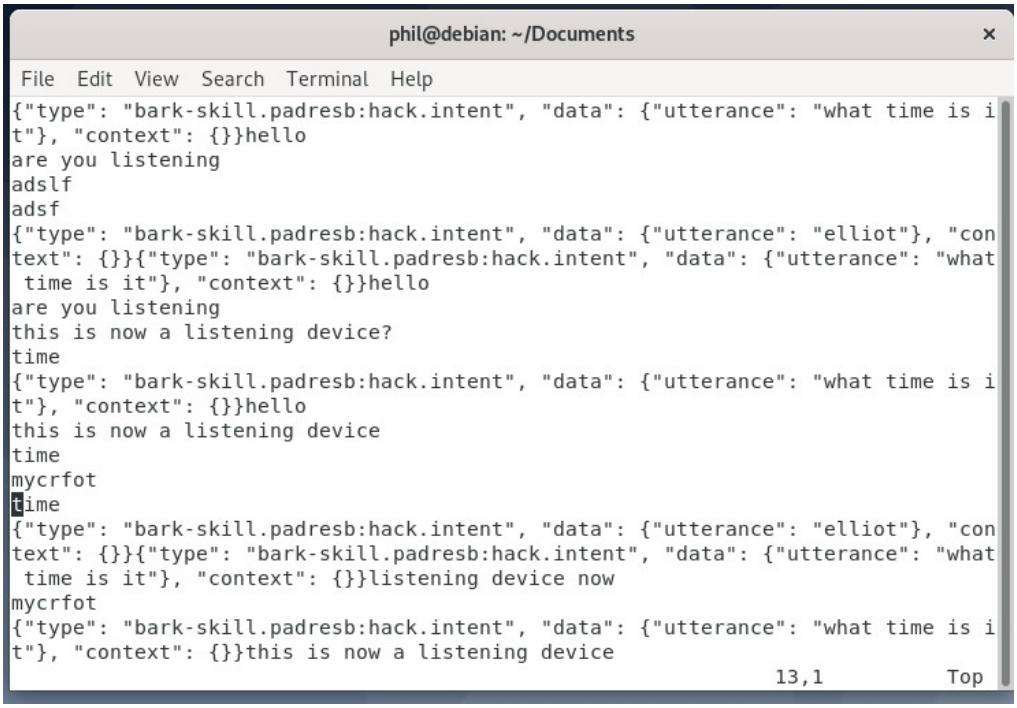
This was just breaking the surface of the power we are able to exploit on through MyCroft skill manipulation. We also did a few test to see if common python libraries still existed with in the MyCroft environment. One major one that we tested was the subproccess library, which allows your code to execute command line arguments. This could be used to find information stored on the device, or cause catastrophic damage. While our proof of concept ended with the listening device, there is limitless opportunity. Below is a image of the "__init__.py" from the bark skill that we manipulated. Our rouge code can be found in the function named handle_hack.

FIGURE 4
CODE EXAMPLES FOR HACKING

```
class Bark(MycroftSkill):
    def __init__(self):
        MycroftSkill.__init__(self)

    @intent_file_handler('hack.intent')
    def handle_hack(self,message):
        f = open('/home/phil/Documents/test.txt', 'a')
        f.write(message.serialize())
        self.speak_dialog('hack')
        res = self.get_response()
        while 'mycroft' not in res:
            f.write(res)
            f.write("\n")
            res = self.get_response()
        f.close()

    @intent_file_handler('bark.intent')
    def handle_bark(self, message):
        self.speak_dialog('bark')

def create_skill():
                                                                    9
```

Below is a picture of the file that the listening device wrote to. It contains a dump of the data in the message, and then anything the device heard during its time as a listening device.

FIGURE 5
DEMONSTRATION OF SUCCESSFUL MODIFICATION OF DEVICES



CONCLUSIONS

In conclusion, the Internet of Things encompasses a large variety devices all across the world that people rely on. In the consumer setting, the Smart Home is a great example of how one's life can be made efficient and convenient through smart devices. However, the more that these devices are integrated into the culture the more that a given person would depend on their devices to be reliable and safe. Unfortunately, the Smart Home is still a new frontier and the smart devices that make up a Smart Home still have many issues and vulnerabilities. With the rapid pace that technology and consumer's reliance on it advances we must strive to perfect these devices. With our proposed concept and others like it, we can begin heading a direction that leads a capable and strong Smart Home that people can rely on.

ENDNOTES

1. Tarkoma, Sasu, et al. "IOTURVA: Securing Device–to–Device (D2D) Communication InIoT Networks." *IOTURVA: Securing Device–to–Device (D2D) Communication InIoT Networks*, 27 Oct. 2017.
2. Ioannis Andrea, Ioannis, et al. "Internet of Things: Security Vulnerabilities and Challenges." *IEEEXplore Digital Library*, July 2015, ieeexplore.ieee.org/abstract/document/7405513.
3. Hwang, Yong Ho. "IoT Security& Privacy: Threats and Challenges." *ACM*, 2015, dl-acm-org.ezproxy.emich.edu/citation.cfm?id=2732216.
4. Khan, Minhaj Ahmad, and Khaled Salah. "IoT security: Review, blockchain solutions, and open challenges." *Future Generation Computer Systems* 82 (2018): 395–411.
5. Flauzac, Olivier, et al. "SDN based architecture for IoT and improvement of the security." *2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*. IEEE, 2015.
6. Lin, Huichen, and Neil Bergmann. "IoT privacy and security challenges for smart home environments." Information 7.3 (2016): 44.
7. HoMonit - Monitoring Smart Home Apps from Encrypted Traffic
8. Zhang, Wei. "Monitoring Smart Home Apps from Encrypted Traffic." Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 8 Oct. 2018, pp. 1074–1088.
9. Khawla, Mazwa, and Mazri Tomader. "A Survey on the Security of Smart Homes: Issues and Solutions." Proceedings of the 2nd International Conference on Smart Digital Environment, 18 Oct. 2018, pp. 81–87.
10. Doan, Tam. "Towards a Resilient Smart Home." Proceedings of the 2018 Workshop on IoT Security and Privacy, 7 Aug. 2018, pp. 15–21.