Remote Code Execution in The Smart Home Environment

A Study of the Potential Threats of Remote Code Execution on Today's 'Real World' and 'Connected' Smart Home Appliances

Mentor:
Ming Chow
https://github.com/mchow01

David Pointeau
https://github.com/dpoint01

December 14, 2015 ~ COMP 116 - Final Project ~ Tufts University

Abstract

The rapid growth of the paradigm of Internet of Things (IoT) has affected the way we use software to accomplish new feats or simplify current actions; it has also affected our concept of “home”. With a plethora of new smart home appliances, our homes are becoming clusters of small networks where previously static/unconnected objects are now inter-connected. This inter-connectivity and automation makes our lives simpler but also opens up to new possible threats. One of these threats is remote code execution. I will review various potential threats caused by remote code execution on what I call “real world” and “connected” appliances. These type of smart appliances are vulnerable to remote code execution and can be the entry points to attacking the entire smart home environment. Ultimately, I hope to raise awareness of the security threats present in smart homes. This paper will primarily focus on remote code execution which is just one of many possible attacks on smart appliances. If nothing is done to secure this new thriving sector, the abbreviation IoT might soon be used to refer to the “Internet of Targets”.

Contents

[1] For info about product: https://nest.com/
[2] For info about product: http://getvera.com/controllers/veralite/
[3] For info about product: http://overseas.hikvision.com/en/index.html

I. Message To The Community

In May of 1998, seven young men sat in front of a panel of senators in Washington D.C. Their goal was very simple: make the general public aware of how insecure the internet was. “If you’re looking for computer security, then the Internet is not the place to be” warned Mudge, one of the seven hackers who came to testify that day [1c]. What the hacker group L0ft did in 1998 embodies what we have to today concerning the evolution of smart homes. While everybody was too focused on the new exciting functionalities and services provided by the World Wide Web, L0ft was able to look in another direction and see what was at the core of the internet’s survival: security.

I decided to write on the topic of smart homes because it is a booming industry that could lead to amazing breakthroughs and services. Yet, in the same way L0ft was concerned with the security of the internet as it made its way to the masses, I am deeply concerned about how smart home security issues could compromise this whole new industry - even before its widespread. Just like L0ft, we have to act preemptively and evaluate the future risks that come with this new industry instead of dreaming of its future advantages.

We now have to account for the hacker at our doorstep.

II. Introduction

As the concept of smart homes is becoming increasingly popular, many manufacturers are joining in to surf on this new technological wave. These manufacturers understand that their potential customers are hungry for new services and applications that could simplify their day-to-day life. In this respect, the development cycle for these new appliances rarely takes in account security measures because it is primarily focused on deploying marketable functionalities. Security is too often left on the sideline.

I will start by providing an overview of the concept of smart homes and provide some information on the kind of software/hardware smart appliances are built with. I will then present three different attacks that were performed on appliances that are currently on the market: The NEST Learning Thermostat, the VeraLite smart home hub, and Hikvision DVRs. All three attacks, even though quite different, can be placed under the umbrella of remote code execution. Following these case studies, I will discuss potential solutions - both preemptive and counteractive - that could mitigate remote code execution attacks on the smart home environment. Finally, I will comment on the current evolution of smart homes and provide a statement on how they are not as safe as they should be.

III. Key Definitions

Smart Home: Commonly refers to residences that have electrical devices that facilitate certain aspects of daily life. These devices can be controlled remotely via a computer, a phone, or a smart home hub. It is the regroupement of these appliances that composes the smart home.

Smart Home Hub: A device used to centralize and unify all connected devices in a smart home. The user can access and control all his smart appliances through the hub's central interface.

Remote Code Execution: In the context of this paper, remote code execution refers to the ability of an attacker to execute malicious code on a designated target remotely. The attacker essentially aims to gain control of the device. With the control of the device, an attacker can perform any desired command. This can include directory traversal, escalation of privileges, a DDoS attack, or just simply giving specific commands to the device's hardware such as unlocking doors or disabling a security system.

Real World Smart Appliances: Appliances composed of software and hardware that respond to their surrounding environment through the use of various sensors. E.g. thermostats or smart locks.

Connected Smart Appliances: Appliances that function through the use of a network. These appliances are either connected to a local network and/or the manufacturer's designated network. As a result of being connected to a network, connected appliances are also linked to other connected appliances on the same network.

IV. Overview of Smart Homes

A. Smart vs Traditional

Consider the following scenario. You arrive back home after a relaxing two-weeks vacation. You are exhausted and the only thing you want to do is sleep. Yet, before you can make it to your bed, you have get your home back up and running. We have all been there. First you disable the security alarm by manually entering your four digit password. You then head to the basement to turn the heat back on. You also have to go to each room in your house to open the shutters - one by one. After performing these tedious tasks, you make yourself a cup of tea from the kitchen and go to the living room to enjoy some music on your stereo system. That's life in a traditional home. Now take the same scenario and apply it to a smart home.

In a smart home environment, all of the aforementioned actions can be performed from the comfort of your couch through one or several applications. Yes, even the tea and music part. A smart home aims to simplify these actions by automating routines, storing and learning about the user's preferences, and by centralizing control. A crucial component of smart home appliances is that they are remotely controllable. By being remotely accessible, these appliances make it possible for users to perform specific actions without being physically present. While remote accessibility benefits the user, it also benefits potential attackers. In a way, the smart home's greatest functionality can paradoxically become its strongest flaw.

B. Characteristics of Smart Appliances

Surely, one of the most repeated word in this paper is the word "smart". So what makes smart appliances so smart? How do they differ from normal electronic devices? Although there exists a wide range of smart products out there, most of them share key characteristics. As previously mentioned, remote accessibility is a very common trait found in smart devices. The user is often able to control the smart device remotely using different protocols. This remote connection can be done via bluetooth or via a web/mobile application specific to the product. Wi-Fi is also an option, but a costly one. Indeed, Wi-Fi is a high-bandwidth network that requires a lot of power - that's why data-heavy activities such as streaming a movie drastically drains a computer's battery. For smart devices to be Wi-Fi compatible, they either need a long-lasting battery or a power source. A connected smart device also has the ability to be aware of the other smart devices connected on the network. This awareness allows for interconnectivity of devices but also for effective attacks as we will see in the Nest Thermostat example.

Another key attribute of smart devices is their ability to sense and interact with their environment. While their software and hardware are static fixed at design time (apart from software updates/patches), smart devices also collect data through a variety of sensors that can detect temperature, air moisture, vibration, movement, location, etc… This data enables the smart device to become a real world device that interacts dynamically with its environment. Yet, in order to be fully smart, these devices need to be automated - that is the ability of a device to automatically perform a task according to sensor input or a given trigger. This enables the device to work almost as a reflexive system: when this happens, do that. Finally, one important trait to note is that many of these devices have third party components in them. This means that the manufacturing of the hardware and/or the deployment of the code was done by different companies with different security policies. This mix-and-match of code/hardware can create additional security vulnerabilities since it increases the number of potential entry points for attacks.

V. Remote Code Execution Case Studies

A. The NEST Thermostat

TrapX Labs, an independent research team in cybersecurity, exposed key security flaws in the NEST Learning Thermostat at the BlackHat 2014 conference. The AOA they released on the subject revealed that they were able to use the NEST Thermostat as a vector to compromise an entire network. Even though they acknowledged that the NEST device was relatively secure compared to most IoT objects, they were still able to gain access to the target's home network. They were then able to performs various instantiations of a remote code execution attack: gain access to personal files on a laptop, browser history, directory traversal, knowing when the house was empty, etc…

The NEST Thermostat is a home automation device that controls the temperature of your home or business. The NEST device does so by collecting data about you: when you change the temperature, your schedule, when you are home and when you are not. While the device itself is designed for one very specific purpose, the data it collects can be used in multiple ways. In their Automated Test Bed Facility (ATBF), the TrapX team took the device apart and analyzed each of its hardware components. They decided to use the mini-USB port as their chosen hardware backdoor. They then managed to get root access through the port by performing an "SSH reverse Tunnel"[4]. Once they gained root access, they installed a custom-made tool chain (by cross-compiling multiple hacking tools from various platforms and languages), and wrote a basic ARP Spoofing attack. Through this ARP attack, they were able to redirect all traffic meant to the

[4] http://unix.stackexchange.com/questions/46235/how-does-reverse-ssh-tunneling-work

user's computer to the attacker's machine. Using the Libcrafter and LibPcap libraries [6], they saved the captured traffic to a pcap file and extracted all valuable information. This attack not only shows the vulnerabilities of one device but also how a single device can be used as an entry point to compromising almost anything on the network. The toughest part of this hack was to cross-compile multiple tools across different platforms to have one single modular tool chain. The rest of the ARP spoofing was relatively straight-forward. The NEST device they targeted is definitely one of the most secure on the market. Yet, today hackers move faster and with more funding. They are able to exploit the slightest opportunity and often find those rare vulnerabilities manufacturers did not consider.

B. Vera Control VeraLite

The VeraLite smart hub is a device used to centralize control of all IoT on a single network. With VeraLite, users are able to control their thermostat, door locks, lighting, and security system from smart controller. While this provides amazing functionality and great user satisfaction, smart hubs are becoming prime targets for cyber-attackers trying to penetrate a home network. During the summer of 2015, Tripwire's Vulnerability and Exposure Research Team (VERT) uncovered several zero-day flaws in three different smart hub devices on the market, one of these devices being the VeraLite hub. Tripwire's previous research has shown them that manufacturers and software developers - of smart devices or not - often discount the risks of attack vectors using cross-site request forgery (CSRF). That's exactly the type of attack VERT performed on the VeraLite hub.

The goal of this CSRF attack was to gain system-level access to the home network without necessitating the victim to be authenticated to any service on the network. One of the difficulties in this type of attack is that the attacker needs to know the exact IP address of the targeted device - something quite tricky to do on a Local LAN. It is possible to iterate through commonly used LAN addresses using JavaScript but scanning through the 18 million IPv4 addresses allocated for private use will take quite some time [7]. Yet client-side technologies are now capable of revealing local IP LAN addresses. VERT used a technique they refer to as '"Smart CRSF" which is essentially JavaScript code that can be triggered to find out a device's IP address. After detecting an IP address, the JavaScript uses this information to scan the network for a vulnerable device on the LAN. Therefore, the key vulnerability exploited by VERT is the fact that these new smart hubs are IP-enabled. Figuring out the IP address of a smart hub can give you access and control to all the devices on the same network.[5]

C. Hikvision DVR DS-7204

The Hikvision DVR DS-7204 is a network-enabled DVR often used in security camera systems to store recorded security footage. Having taken a quick peek at the product specifications, the DS-7204 accepts the following network protocols: TCP/IP, DHCP, SMTP, RTSP and HTTPS - all of which can be used as vectors to possible scans or attacks. Security Systems, and more specifically security cameras/DVRs, have always been at the top of the potential targets in the IoT bubble. Why? Because they hold sensitive information about your private life and provide a layer of protection from *real* physical harm. In a research conducted by Rapid7 labs, several Hickvision DVRs were found vulnerable to buffer-flow vulnerabilities that leaves the device open to remote code execution. Using this hack, an attacker could delete all security footage stored on the DVR. More importantly, Rapid7 labs also discovered that an

[5] Source-code of VERT's example Smart CRSF attack: http://pastebin.com/UfXpPd5J

infected DVR could also be used as a proxy to access other devices in its local network - including workstations, point-of-sale systems, or any other insecure IoT device. HikVision models are popular across the IPv4 space, owing in part to the fact that users can view the streams remotely with an iPhone app. This makes them even more vulnerable to an attack.

The buffer overflow vulnerabilities found by Rapid7 affect the code that handles real-time streaming protocol (RTSP) requests. None of them require authentication to exploit and can be easily performed by intercepting these requests. Rapid7 has posted denial-of-service proof-of-concept exploits for these vulnerabilities in the RTSP request body handling (CVE-2014-4878) and header handling (CVE-2014-4879). Furthermore, the most interesting exploit is taking advantage of RTSP basic authentication handling (CVE-2014-4880) which could grant full control of the device to the attacker.

VI. Mitigation

In this section, I will present several ways to mitigate - the possibility and/or the effects of - remote code execution attacks on a smart home environment. These measures can be both preemptive and counteractive and concern those making the devices, not those using them. Indeed, I quickly discovered in my research that the underlying problem is that IoT software is often poorly designed and rarely takes in account security as a key determining factor of “good” code. Of course, there are higher level flaws that can be attributed to network issues and the overall architecture of the internet - not to the actual devices themselves. Here are some of the most effective measures to secure smart appliances:

➔ On a network level: the importance of generalizing the **transition to IPv6** is of crucial importance to keeping the IoT secure. The primary function of IPv6 is to allow for more unique TCP/IP address identifiers to exist, now that we've run out of the 4.3 billion created with IPv4. More importantly, IPv6 can run end-to-end encryption.This is one of the main reasons why IPv6 is such an important innovation for the IoT. The encryption and integrity-checking used in current VPNs are a standard component in IPv6 and made available for all connections. Widespread adoption of IPv6 will therefore make man-in-the-middle attacks harder to perform. Therefore, if correctly designed and configured, securing IPv6 will mean securing the IoT.

➔ For the manufacturers: make sure to make thorough **design/security reviews of all third-party components** that compose the device. This is a strong preemptive measure that could avoid a lot of work like patching a discovered flaw. The point of the game here is to never assume that the code you use or write is 100% safe.

➔ For the manufacturers: make sure to have a well designed and **rapidly executable strategy to deploy and integrate software/hardware fixes** to your customer base. As previously mentioned, hackers move fast and always ways find new vulnerabilities in systems. Manufacturers have to be ready for the worst possible scenario. It is not only important to take preemptive measures but also to make sure you can reduce the effects of a succesful attack.

➔ For those who design the consumer facing interfaces and websites used by IoT devices: make sure they are **safe from CSRF or SQL injection attacks** and that the various IP addresses of devices are encrypted in the database. Security should be airtight both in the device itself and the various applications that interact with it.

➔ For product designers and back-end developers: smart devices that work by collecting data should have secure information disclosures that have the end-user's security in mind. That means **not collecting the data that you don't need and encrypting the data you do need.**

Reducing the amount of data stored on these devices or on cloud-based systems will also reduce the number of potential attack vectors.

➔ For cyber-security researchers: **develop specific and modular tool chains** for each new device that is out there. Using effective tools in penetration testing assures that the product designers are aware of the potential security vulnerabilities of their product allowing them to make adjustments before deploying.

These measures are just a sample of the potential solutions that could make the IoT safer. The concept of “Smart Home” is not one of the future but one that it embedded in today’s reality. Yet, in order to safeguard this new exciting technology that could benefit us all, it is in the manufacturer's’ duty to finally put security at the forefront of software design and hardware architecture.

VII. Conclusion

In this paper, I have exposed several simple attacks that were carried out on three smart devices. Each device comes with its own flaws: the NEST Thermostat is bootable via a mini-USB port, the VeraLite is vulnerable to CSRF attacks, and the Hikvision DVR is not protected against buffer-flow hacks[6]. Using these examples as case studies, my goal was to showcase how unsafe smart homes actually are. The evolution of computing towards the IoT is exciting but it also brings in new questions to the table. With more and more “real world” and “connected” devices on our networks we have to put security at the core of the discussion. We can’t think anymore of security in terms of one single isolated device but in terms of clusters of devices connected to a single network. *Today, our networks are as safe as the least-secure device connected to it.* What smart devices manufacturers are currently doing is compromising

[6] I use the present tense, but most of these vulnerabilities have since been fixed by the respective companies

our networks by adding all these unsecure devices to them. If they don't channel their development cycle towards making more secure products, the IoT will soon become the single biggest cyber-attack vector we've ever seen.

We now have to account for the hacker at our doorstep.

VIII. References

[1]

[a] Timberg Craig, "The Real Story of How the Internet Became so Vulnerable." *Washington Post*. The Washington Post, 30 May 2015. Web. 01 Dec. 2015.

[b] Timberg Craig, "Quick Fix for an Early Internet Problem Lives on a Quarter-century Later." *Washington Post*. The Washington Post, 31 May 2015. Web. 09 Dec. 2015.

[c] Timberg Craig, "These Hackers Warned the Internet Would Become a Security Disaster. Nobody Listened." *Washington Post*. The Washington Post, 22 June 2015. Web. 09 Dec. 2015.

[d] Timberg Craig, "The Definitive Account of How Hackers Can Gain Access to Our Cars." *Washington Post*. The Washington Post, 22 July 2015. Web. 09 Dec. 2015.

[2]

"Hacking Nest Thermostat." *Net Security*. Hacking Nest Thermostat, 10 Mar. 2015. Web. 10 Dec. 2015.

[3]

Young, Craig. "Smart Cross-Site Request Forgery (CSRF)." *TripWire*. The State of Security, 15 Sept. 2015. Web. 10 Dec. 2015.

[4]

"Code Injection." *OWASP*. OWASP, n.d. Web. 10 Dec. 2015. <https://www.owasp.org/index.php/Code_Injection>.

[5]
"Command Injection." *OWASP*. OWASP, n.d. Web. 10 Dec. 2015. <https://www.owasp.org/index.php/Command_Injection>.

[6]
TrapX Labs. *Anatomy of an Attack (AOA) Report, "The Internet of Things (IoT) – The Hidden Danger Exposed"*. Tech. TrapX, 16 Mar. 2015. Web. 03 Dec. 2015.

[7]
Tripwire Blog. "Tripwire Uncovers Significant Security Flaws in Popular Smart Home Automation Hubs." *Tripwire*. N.p., n.d. Web. 3 Dec. 2015. <http://www.tripwire.com/company/news/press-release/tripwire-uncovers-significant-security-flaws-in-popular-smart-home-automation-hubs/>.

[8]
"R7-2014-18: Hikvision DVR Devices - Multiple Vulnerabilities." *Rapid7 Community*. Metasploit, 19 Nov. 2015. Web. 3 Dec. 2015. <https://community.rapid7.com/community/metasploit/blog/2014/11/19/r7-2014-18-hikvision-dvr-devices--multiple-vulnerabilities>.

[9]
"Common Vulnerabilities and Exposures." *CVE - (CVE)*. N.p., n.d. Web. 10 Dec. 2015. <https://cve.mitre.org/>.