universität
innsbruck

Department of Computer Science
Security and Privacy Lab
Course: 703647-0 18S PS3 Netzwerksicherheit

SEMINARTHESIS

Security of Smart Light Systems: Covert Channel on IoT Light Bulb

Bennett Piater 01418470
Julia Wanker 01314695

Innsbruck, July 2018

Contribution Statement

Our overall work repartition was fairly equal. Bennett spent more time working on the hardware, Arduino, and so on, which Julia compensated by doing more of the initial research and paper structure. The rest was a mix of hands-on experimentation together, for which we met, and evenly split analysis and writing work which we cross-checked.

Abstract

In this paper we attempt to reproduce the functionality extension attack on IoT light bulbs initially conducted by Eyal Ronen and Adi Shamir [15] . This type of attack uses the light bulb's brightness change to exfiltrate data from air-gapped networks. We implemented the attack on the widely known Philips Hue White lightning system and were able to prove that those systems are potentially vulnerable to such a covert channel attack. We evaluate our findings with regard to the results of Ronen and Shamir and give practicable recommendations.

Contents

1 Introduction **1**

2 Related Work **2**

3 Communication with Light **4**

4 Covert Channel on IoT Light Bulb **5**

4.1 Experimental Setup 5

4.1.1 Transmitting Setup. 5

4.1.2 Receiving Setup. 6

4.2 Attack Description 7

4.2.1 Controlling Smart Light Bulbs 7

4.2.2 Crafting PWM Signal 8

4.2.3 Getting Data 8

4.3 Limitations 9

5 Conclusion **10**

1 Introduction

Internet of Things (IoT) devices have a poor security in general. The most recent state-of-the-art security survey was performed by Zhang et al. [18]. They provide a detailed analysis of vulnerabilities and defense mechanisms and suspect either a lack of expertise, or outright neglect of security design from the part of vendors. This insecurity extends to smart light systems.

Dhanjani [5] exploited several issues in the connection setup between smart lights and their controller as well as encryption flaws in the ZigBee Light Link (ZLL) protocol, which is used for communication between the controller and the light bulbs. He used these flaws to perform Denial-Of-Service (DoS) attacks, which could cause blackouts e.g. in hospitals and other critical infrastructure. Fortunately, the impact of his attack was limited to its short physical range.

Morgner et al. showed that the flaws in ZLL are more serious: They were able to control ZLL-certified light bulbs from up to 36 meters [12]. To make things even worse, Ronen et al. [16] were able to replace the firmware on smart lights with a worm able to spread to nearby bulbs over ZLL. This brings the scenario of a drive-by *war-flight* attack, infecting the smart lights in an entire block or city, into the realm of possibility.

Most of the attacks on smart lights, including the aforementioned ones, focused on *disabling* (i.e. DoS) or *ignoring* the intended functionality of the devices (e.g. to build bot nets). While these kinds of attacks definitely pose a security threat, they are not particularly interesting because they do not differ much from attacks on other IoT devices, or even on any networked general-purpose computer.

However, in an attack vaguely similar to the more recent (and more famous) data extraction using routers status LEDs [9], Ronen and Shamir introduced the category of *functionality extension attacks*, which (mis)use the functionality provided by a device to perform something not intended by its designers — like the old engineering definition of *hacking*. They used a smart light to build a *covert communication channel* which is unlikely to be

detected unless explicitly looked for. Also, and probably of even more immediate interest to Israeli security researchers, visible light is a very promising exfiltration channel for air-gapped systems because of its small apparent threat.

Because this is an interesting, unique and not currently reproduced piece of research, we decided to try to reproduce it.

This paper is organized as follows: First, we cover some preliminaries and relevant related work in Sections 2 and 3. Then, we will proceed to give a detailed description of our attack in Section 4. Finally, we will draw conclusions and give an outlook in Section 5.

2 Related Work

Ronen and Shamir [15] discovered that sensitive data like e.g. passwords and keys can be extracted even from air-gapped networks using IoT light bulbs within the same network and the principle of visible light communication (VLC). The creation of a covert communication channel was tested on two different IoT lightning solutions. One was the widely known Philips Hue White lightning system, formerly called Philips Lux. The idea is to switch between two very close brightness levels which further allows to interpret the corresponding pulse width modulation (PWM) signal as logical zero and logical one, respectively. Philips comes with 255 different brightness levels and uses a PWM frequency around 20 KHz. This requires measuring differences of only 200 nanoseconds (ns) and thus high end measuring equipment is needed, including sensitive light sensors in order to actually see the difference in luminosity. When changing brightness levels, a measurable phase shift is created in the light sensor's frequency output which allows to further decode the sent data. Assuming that light in offices is turned on for at least 12 hours a day, around 10 KB each day could be leaked which is actually enough for e.g. passwords and keys.

The general idea of using light emitting diodes (LEDs) for data exfiltration from air-gapped networks was also explored by Guri et. al [9]. Their idea was to control the router's activity and status LEDs via a malicious

script running on the router which accesses the routers General Purpose Input Output (GPIO) controls in order to change the LEDs blinking pattern according to the data which a potential attacker wants to extract. To be able to receive the data the LEDs need to be in the line of sight of the attacker. Therefor Guri et. al tested two different approaches. Firstly, they used a video camera to determine on and off periods. When using a camera the maximal bit rate depends on the type of camera, i.e. the best possible solution is using a high-speed camera as for example a GoPro. With this a transmission rate of 800-960 bit per second (bps) can be achieved when using eight LEDs. A much higher bit rate could be obtained using an optical sensor, as Ronen and Shamir [15] did. In that case Guri et. al were able to transmit 4000 bps, again using eight LEDs.

VLC in general is of high interest nowadays and may also be in future since it solves some of the limitations we meet with traditional wireless transmission means. For one, it is less expensive since no radio frequency (RF) units are needed. Furthermore, the light signal is free of any health concerns, it is also conductible in RF sensitive areas, e.g. planes, and the bandwidth is not limited, and so on [7]. Further, with the rise of IoT and thus also smart light solutions, LED bulbs will again become more and more prevalent. VLC has by now been studied for decades. By that time various different modulation schemes exist in order to encode data. Indeed, the most standing to reason sort of modulation is the On-Off Keying (OOK) modulation scheme, but on the other hand OOK brings limitations on the conductible data rates. Several researches have thus been made on Orthogonal Frequency-Division Multiplexing (OFDM) together with higher level modulation schemes [7, 17]. With IoT light bulbs the modulation issues can be disregarded, since the PWM signal cannot be modulated directly. But, as it was already described in the beginning of this section, we can still reach enough bandwidth for extracting sensitive data.

3 Communication with Light

Before we can look at how a functionality extension attack on a smart light system can actually be leveraged, we need to understand how communication over light works.

In VLC [11, 17] the visible part of the electromagnetic (EM) spectrum, namely the visible light, is used for communication purposes. VLC is a subset of optical wireless communication technologies, like infrared. In order to conduct VLC, simple white LED bulbs are needed. The actual transmission of data is based on the fact that LEDs can be switched on and off at such a high rate, that those intensity changes cannot be seen by the human eye.

As known from the video game section, human eyes are capable of seeing flickers above 30 Hz. The difference between 30 frames per second (fps) and 60 fps can easily be noticed. Flicker rates which go beyond 60 Hz can though not be detected by the human eye. So in the context of light communication, especially covert light communication as needed in our case, it is important to flicker at over 60 Hz. Those quick flickers allow constant illumination besides the ability of data transmission.

The flickers are done by rapidly switching between on and off states and adjusting the duty cycle. The transmission of data is further allowed by interpreting the on period as logical one and the off period as logical zero. In order to actually get the encoded data out of the light signal the duration and frequency of the light flickers need to be measured. This is done using a light sensor since these are capable of accurately distinguishing between the on and off states and measuring the duty cycle. Further light sensors are robust to other light sources or any other noise.

In the case of smart lights, the PWM signal cannot be changed directly, since the light intensity is changed by sending commands over the manufacturers Application Programming Interface (API). Those commands internally modulate the pulse width of the light signal. Thus, by sending two close brightness commands, we can achieve the same effect as with traditional VLC and covertly transmit data. Different to traditional VLC, a logical one is represented by the higher brightness level whilst a logical zero is represented by

the lower brightness level. Fortunately, since internally the luminosity of the LEDs is again changed by adjusting the PWM signal, light sensors are again capable of measuring the differences.

4 Covert Channel on IoT Light Bulb

Our goal was to reproduce the attack from Ronen and Shamir [15]. Therefore, we tried building a covert channel using the Philips Hue White lighting system. With our experiment we were able to prove that data can be covertly transmitted using the Hue light bulb. On the transmission side we used a laptop running our python script over which the Hue API is accessed in order to send brightness commands. At the receiver side we used a light sensor which converts the light intensities to a frequency signal. That signal was forwarded to an oscilloscope which further sent the received frequency output to our laptop where we plot the results in order to validate the sent bits.

In the following sections we first describe the setup components and their functionality. After that, we elaborate the actual attack.

4.1 Experimental Setup

Our proof of concept was implemented using affordable equipment which costs less than 1100.- €. Further, we did not need to run any unauthorized code on the light bulbs since the control over the Hue API suffices to create covert flicker effects. Figure 1 shows our overall receiving setup. An Arduino was used as power supply and for configuring the light sensor, which the PicoScope had direct access to.

4.1.1 Transmitting Setup.

We used the Philips Hue White light bulbs for our experiment [13]. We bought the starter kit which contains two E27 9 Watt 806 Lumen bulbs together with a bridge which allows to remotely control the bulbs using e. g. a laptop. In order to set up the smart light system the bridge needs to be

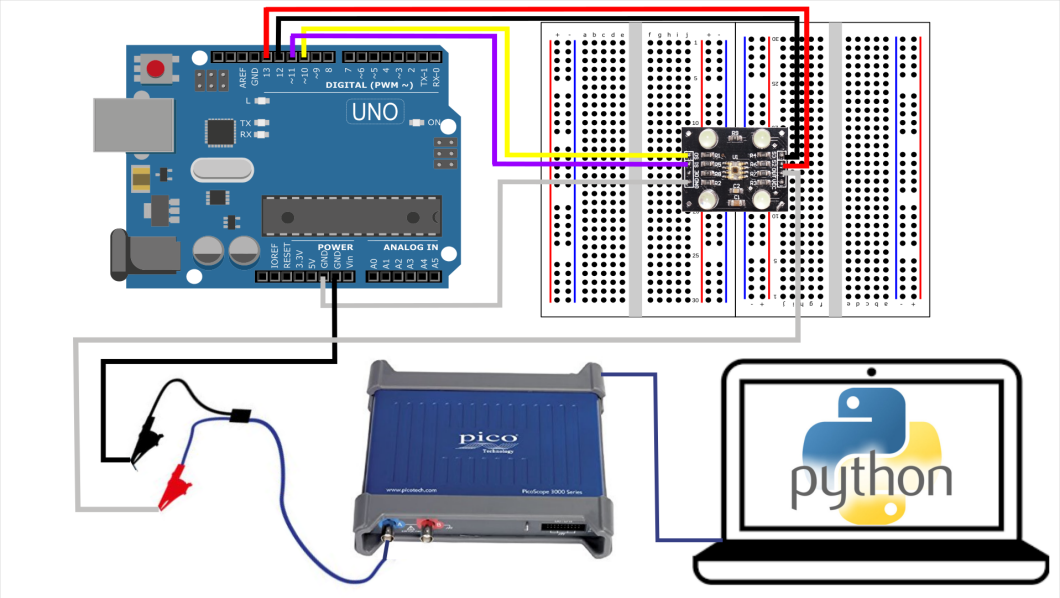

Figure 1: Experimental setup for measuring the lights' frequency output and reading the sent bits out of the received signal

connected to the user's network using Ethernet. Once connected, the user can send brightness-change commands from any device within the local network using the Hue API. We made recourse to a command line based version of the Hue API [1], which allowed us to easily send the commands via the python script we used for realizing the experiment. The bridge further forwards the commands over a RF transmitter to the light bulbs using the ZLL protocol.

The Hue White bulbs have 255 different brightness levels, which forced us to sample the output at a very hight rate in order to determine the changes in the light frequency output. But, on the other hand, due to the minor difference between two close levels, the changes were imperceptible to the human eye, which worked for us.

4.1.2 Receiving Setup.

For measuring the changes in light intensity we used the TAOS TCS3200 Color Sensor [4]. The sensor consists of an array of photo diodes where each is capable of filtering red, green, blue or clear white light. We set up the sensor to measure the clear white luminosity output of our Hue bulbs. The

sensor contains an internal oscillator in order to convert the light's intensity output to a corresponding square-wave frequency signal. The TCS3200 is capable of communicating directly to an Arduino micro controller.

Unlike Ronen and Shamir [15], we used the Arduino board as power source only, and the PicoScope to measure the brightness. This was necessary because the advanced API functionality which the former used to craft a PWM signal is no longer available [15], and we had to directly distinguish existing brightness levels using their PWM profile.

In order capture the output from the light sensor, we used the PicoScope 3205D MSO since it is capable of sampling 10 MS/s, which we need in order to accurately measure the light sensor's frequency output which lies around 800 KHz. Actually, the PicoScope is able to sample up to 1 GS/s when using one channel and 500 MS/s with two channels, but for our needs only 10 MS/s suffice.

4.2 Attack Description

The following paragraphs give a detailed description of the main steps to realize such an attack. Therefore, we first need to look more precisely at the functional principle of smart light bulbs. Further we have a look at how smooth brightness changes are achieved and how we actually get data out of the received signal.

4.2.1 Controlling Smart Light Bulbs

Smart light bulbs consist of three main components: (1) a RF receiver, (2) a processing unit and (3) LEDs and LED drivers. The communication with the controller is ensured through the *RF receiver* and relies on the ZLL protocol. The received commands are further forwarded to the *processing unit* which interprets the processed signal and controls the LED by modulating the pulse width. The PWM allows the different dimming factors. When sending a brightness change command via the Hue API, this automatically forces the processing unit to generate the corresponding PWM signal. The PWM is

sent to the *LED drivers* which further turn the LEDs on and off at a very fast rate such that those changes in the duty cycle cannot be seen by the human eye. Since Hue comes with 255 brightness levels which need to be differentiated smoothly, a PWM with a frequency around 20 KHz is used [15].

4.2.2 Crafting PWM Signal

Since we can no longer craft a custom PWM signal using the Hue API, we had to get along with the PWM used for dimming. Thus, our goal was to use close brightness levels and attempt to distinguish them from their PWM profile.

Because of the great amount of brightness levels, we had to measure very small off periods of about 200 ns. This could be done with the described light sensor as well as the PicoScope, since the light sensor's output is around 800 KHz, which we could easily sample at 10 MS/s with our PicoScope.

4.2.3 Getting Data

Our primary goal was to distinguish adjacent brightness levels that cannot be distinguished by the human eye. For this purpose, we performed a Short-Time Fourier Transformation (STFT) to transform the voltage sinusoid obtained from the light sensor into the frequency-over-time domain.

As can be seen in Figure 2, the difference between brightness levels is clearly visible upon optical inspection of the graph. For comparison, we found that it took around 10–15 levels of separation for the naked eye to distinguish 2 brightness levels, depending on the absolute brightness. In the best images, the PWM profiles were clearly recognizable.

However, we had a rather hard time consistently reproducing images. Distance from the light to the sensor, angle, and variations in external light, in decreasing order, made it harder to clearly distinguish the brightness levels and to choose a starting brightness. In particular, the square curve tended to become a muddy sinusoid, which we attribute partly to the fact that we perform STFT on the data. We found that distance between sensor and light had a strong impact on the correct choice of STFT window size and intensity

of the brightness levels between which we switch, with increasing distance requiring more energy and/or a smaller window.

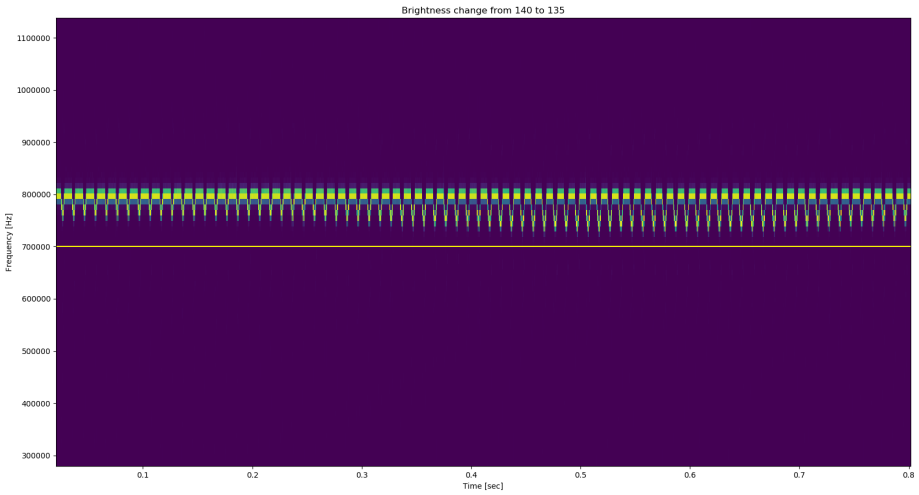

Figure 2: Brightness change from levels 140 to 135 (and back) sampled at 10 MS/s. Note the smooth fading.

We were not able to robustly automatically distinguish brightness levels, but that may be because we spent most of our time trying to do that with the Fourier-transformed data when it may have been better to analyze the sinusoid from the light sensor directly. In any case, this is a solvable signal processing problem and more a question of time.

With that done, the next step would be to select brightness levels for 0 and 1, and maybe a third in between the two to be used as delimiter and reduce the need for synchronization. All that would then be left to do is to apply a suitable channel coding to the data, send appropriate brightness commands to the light, write out the recognized 0s and 1s, and decode the data.

4.3 Limitations

Communicating with the light bulbs over the Hue bridge brings some limitations with it [15]. For one, the bridge or the LED drivers implement some

smooth fading feature in order to avoid sharp brightness changes. Due to the automatic fading we cannot see phase shifts in our signal output, which makes it harder to analyze.

Furthermore, the bridge restricts the rate of commands which can be sent within the system. While this does not impact the functionality of our proof-of-concept since the command rate doesn't determine the flickering frequency, it does limit the bandwidth of the channel. Thus, in case such an attack should actually be leveraged, one may need to access the ZLL communication directly in order to circumvent the rate limit.

5 Conclusion

IoT devices in general are largely insecure and should be trusted at most as much as any other device in a network. Smart lights in particular are interesting because they are networked LEDs and therefore could be used to build a covert communication channel. Additionally, visible light is often the only part of the EM spectrum that is allowed to leave an air-gapped system. Therefore, the attack that we replicated in this paper could potentially be used to exfiltrate data from air-gapped network.

Ronen and Shamir [15] showed that specially-crafted PWM profiles could be used to build a covert channel and mentioned that the PWM from dimming could be used once the method they used was no longer available. Our findings confirm that argument: Using the PWM, we were able to distinguish brightness levels indistinguishable to the human eye. Therefore, Smart LEDs lend themselves to building covert channels.

More work is required to make this attack practical, but it is definitely feasible even with limited resources. Given better signal processing, calibration, and a good channel encoding, the full transmission sequence could be automated in a very reasonable time frame. To make the attack truly practical, better range would be required, for example by putting the light sensor in the focal point of a telescope focused on the light bulb [15] and maybe using a more sensitive sensor. That telescope would be the most expensive required piece of hardware by several orders of magnitude.

The biggest takeaway from this is that smart lights are not to be trusted in secure environments. Better use traditional lightning solution within sensitive environments. If for some reason it is necessary to use networked lights, standard security practices (for networked and IoT devices) should be followed: Change passwords regularly, disable unused services, isolate IoT devices onto their own network(s) or use specially tailored engines to curtail the IoT devices' controlling rights, and raise awareness that these devices are networked general-purpose computing systems. Additionally, seriously consider to not allow visible light to exit the air-gapped room or building by designing it without windows.

References

[1] hueadm - Phillips Hue Admin CLI Utility. GitHub. License: MIT. Online, `https://github.com/bahamas10/hueadm`; accessed 19. June 2018.

[2] K. Angrishi. Turning Internet of Things (IoT) into Internet of Vulnerabilities (IoV): IoT Botnets. *arXiv preprint arXiv:1702.03681*, 2017.

[3] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou. Understanding the Mirai Botnet. In Engin Kirda and Thomas Ristenpart, editors, *USENIX Security Symposium*, pages 1093–1110. USENIX Association, 2017.

[4] DFRobot. TCS3200 RGB Color Sensor For Arduino. Website. Online, `https://www.dfrobot.com/product-540.html`; accessed 23. June 2018.

[5] N. Dhanjani. Hacking Lightbulbs: Security Evaluation of the Philips Hue Personal Wireless Lighting System. 2013. Online, `http://www.dhanjani.com/docs/Hacking%20Lighbulbs%20Hue%20Dhanjani%202013.pdf`; accessed 19. June 2018.

[6] M. Donno, N. Dragoni, A. Giaretta, and A. Spognardi. Analysis of DDoS-Capable IoT Malwares. In Maria Ganzha, Leszek A. Maciaszek, and Marcin Paprzycki, editors, *FedCSIS*, pages 807–816, 2017.

[7] H. Elgala, R. Mesleh, H. Haas, and B. Pricope. OFDM Visible Light Wireless Communication Based on White LEDs. In *2007 IEEE 65th Vehicular Technology Conference - VTC2007-Spring*, pages 2185–2189, April 2007.

[8] A. Grau. Can you trust your fridge? *IEEE Spectrum*, 52(3):50–56, 2015.

[9] M. Guri, B. Zadov, A. Daidakulov, and Y. Elovici. xLED: Covert Data Exfiltration from Air-Gapped Networks via Router LEDs. *CoRR*, abs/1706.01140v1, June 2017.

[10] C. Kolias, G. Kambourakis, A. Stavrou, and J. M. Voas. DDoS in the IoT: Mirai and Other Botnets. *IEEE Computer*, 50(7):80–84, 2017.

[11] T. Komine and M. Nakagawa. Fundamental Analysis for Visible-Light Communication System using LED Lights. *IEEE Transactions on Consumer Electronics*, 50(1):100–107, Feb 2004.

[12] P. Morgner, S. Mattejat, and Z. Benenson. All Your Bulbs Are Belong to Us: Investigating the Current State of Security in Connected Lighting Systems. *CoRR*, abs/1608.03732, 2016.

[13] Philips. Hue White Starter Kit E27. Website. Online, `https://www.philips.at/c-p/8718696449554/hue-white-white-starter-kit-e27`; accessed 19. June 2018.

[14] F. Restuccia, S. D'Oro, and T. Melodia. Securing the internet of things in the age of machine learning and software-defined networking. *IEEE Internet of Things Journal*, pages 1–1, 2018.

[15] E. Ronen and A. Shamir. Extended functionality attacks on iot devices: The case of smart lights. In *2016 IEEE European Symposium on Security and Privacy (Euro S&P)*, pages 3–12. IEEE, 2016.

[16] E. Ronen, A. Shamir, A. Weingarten, and C. O'Flynn. IoT Goes Nuclear: Creating a Zigbee Chain Reaction. *IEEE Security & Privacy*, 16(1):54–62, 2018.

[17] Z. Yu, R. J. Baxley, and G. T. Zhou. Brightness control in dynamic range constrained visible light OFDM systems. In *2014 23rd Wireless and Optical Communication Conference (WOCC)*, pages 1–5, May 2014.

[18] N. Zhang, S. Demetriou, X. Mi, W. Diao, K. Yuan, P. Zong, F. Qian, X. Wang, K. Chen, Y. Tian, C. A. Gunter, K. Zhang, P. Tague, and Y. Lin. Understanding IoT Security Through the Data Crystal Ball: Where We Are Now and Where We Are Going to Be. *CoRR*, abs/1703.09809, March 2017.