**Project Title:** Potential Security Risk from Internet of Things Devices.

Author

Oisin Michael Farrell. <[oisinf@kth.se](mailto:oisinf@kth.se)>

Abstract

# Introduction

The purpose of this paper is to check the potential security risk caused by Internet of Things (IoT) devices. This research paper will give me a better understanding of the communication between devices and will solidify the fundamentals of the networking that happens between these devices.

## Theoretical framework/literature study

An IoT device is a piece of hardware that sends some sort of data to other devices via a network [1]. IoT devices can come in many forms such as smart fridge or a smart bulb, the possibilities are endless. In 2019 there was 26.66 billion active IoT devices in the world with this number set to reach 35 billion by 2021 [2]. It is estimated that 127 IoT devices are connected every second [2]. IoT devices can be extremely useful. One such IoT device is the TP Smart bulb. The functionality of the smart bulb according to the TP-Link website:

* **Dimmable (Adjust with Ease) –** Manage the lighting at your fingertips to create the perfect atmosphere in your home at any time.
* **Dimmable (Pre-set for Convenience) –** Found your favourite lighting for watching movies? Save it as a setting you can simply select for future movie nights.
* **Schedule (Set Your Times) –**Create a regular schedule to turn on/off with the brightness you set.
* **Schedule (Sunrise and Sunset Modes)** – Automatically activate your light according to your time zone.
* **Voice Control**– Manage the lights with your voice commands as Tapo works with Amazon Alexa and Google Assistant.
* **Remote Control**– Take your phone out for the pocket and open the Tapo app to control your light remotely [6].

This is really advantageous to the user as it gives them control of devices at the click of a button. They do not even have to be in the same room to use devices. If you have internet access you can turn on or off lights. You could leave work and potentially have a nice dimly lit reading room waiting for you at home and if you had the TP-Link smart plug perhaps even a cup of coffee waiting for you. IoT devices are revolutionary, and they are changing the way we interact with out appliances, but they do have one potential drawback. Like most devices connected to the internet there is always a threat of security issues.

According to a tweet sent out by TP-Link UK, the twitter page for TP-Link, there has been some “security vulnerabilities” in the past [3]. This is an issue for the safety of the user. Although you might think that it is no big deal have an online hacker flick on and off your lights, there are more malevolent reasons to hack someone’s IoT device. If the hacker has access to your data, then they can see your historical data. This might reveal common times that your lights are off and might reveal to a potential intruder when you are likely to be out of the house so that they could break in undetected.

An example of such a security breach caused by an IoT device was when a North American casino was hacked via the thermometer in their aquarium. The hacker penetrated the network via the IoT device and was able to steal some data from the network [4]. If this happens to you and your data was leaked, would you feel safe?

## Research question

The aim of this paper was to find out if IoT devices can pose a potential security risk to their users. My research question is thus: Can IoT devices threaten our privacy?

# Method

To find the answer to this question I decided to take an experimental approach. The idea was to emulate a TP-Smart Bulb on a Raspberry Pi and sniff the packets that the devices send to each other. To emulate the communication between the smart plug and the user I set up a client server environment using Python’s Socket library. In the experiment my Windows machine will act as the server and the Raspberry Pi will act as the client. I have set it up in this was because in theory multiple smart plugs can be connected to one user. A user will use their phone or laptop to control multiple devices and so in the client server relationship they are the server.

Text

Description automatically generatedText

Description automatically generated

Figure 1, Client and Server Python code [7].

Figure 1 shows the server and client python programs. One thing worth mentioning here is that this is not an exact representation of the TP-Link smart bulbs that you would find in the real world. This is just my interpretation as it would be too complicated to make a one-to-one replication. Another important detail is that I cannot be exactly sure of the security measures that are used in the smart bulb. On the TP-Link website they state that they use “commercially reasonable safeguards” such as encryption and secure socket link (SSL) protocols but that they “cannot fully ensure the security of any information you transmit” [5]. As I am unable to program what I do not know some security measures may have been omitted in my implementation that might be present in the actual smart bulb. Therefore, the experiment is not an exact representation of the security that might be used in the actual smart bulb but for research purposes should suffice as it gives an overview of IoT devices in a general sense.

In the experiment the Raspberry pi is set up to emulate a TP-Link smart bulb connected to a light source which reports the light status at regular intervals. I then sniffed the packets being sent between them. For the first part experiment Wireshark was run on one of the machines involved in the exchange. The reason for this was that none of my devices were capable of promiscuous mode which would allow you to sniff the packets of another machine on a local network. This is not really any concern for our privacy as we now assume that the attacker must have access to our machine to sniff our traffic. To get around this issue and demonstrate how an attacker would sniff your packets I set up a man in the middle (MITM) attack by performing an Address Resolution Protocol (ARP) poisoning my own machine. This is done on my local network and should not be used for illicit activity. It is only used here to demonstrate how an attacker would cause damage in theory.

First, let us define a MITM attack. Put simply, a MITM attack is the interception and or modification of the communication between two parties [9]. ARP spoofing is one type of MITM attack. According to a lecture given by Markus Hidell, ARP spoofing refers to sending fake or gratuitous ARP replies and goes on to say that ARP poisoning is possible as ARP replies are not authenticated [10]. This makes it easy to falsify or poison ARP caches on the local network. To perform the MITM attack for this project I used dsniff’s arpspoof tool. This too allows you to attack a target and have their traffic sent to your machine. For this part of experment, my local windows machine is the target and my Raspberry Pi is the attacker. On the Raspberry Pi I used the following command to trick my Windows machine and my router into forwarding traffic to the Raspebrry Pi:

arpspoof -i INTERFACE -t targetIP routerIP

While this command, and the same command but with targetIP and routerIP swapped, are running the targets ARP cache is poisioned, causing traffic to flow through the Raspberry Pi. This is where the attacker can sniff the packets and potentially cause harm.

Diagram

Description automatically generated

Figure 2, ARP spoofing diagram inspired by Introduction to Computer Security course, lecture 5 slides [10]

Figure 2 shows the ARP spoofing that I done for the experiment. As you can see, there is an attacker, a target, and a router. The attacker sends false ARP messages to both the router and the target. These ARP messages are accepted as there is no authentication so then their ARP cache is poisoned, and both the router and target will send data through the attacker. The arpspoof commands for this example would be:

arpspoof -i INTERFACE -t 192.168.1.2 192.168.1.1

arpspoof -i INTERFACE -t 192.168.1.1 192.168.1.2

The interface parameter depends on the network type. It can be wlan or eth depending on whether you are using wireless LAN or ethernet. In my case it was eth0 as I am connected via physical ethernet.

# Results and Analysis

Three packets were successfully sent from the Raspberry Pi to the server and were recorded.

Text

Description automatically generated

Figure 3, Terminal of Server.

Figure 3 shows the running code for the sever which in a real-life scenario would be TP-Link’s supporting app Tapo. As you can see, the Raspberry Pi, IP address ending in 128, connects to the server. This simulates a user setting up their Smart Bulb. Information sent between an IoT device and a server or companion app will not look like that of figure 1, this just gives an overview of the communication between the two devices. Instead, there would be a nice graphic user interface (GUI) that makes it look nice.

Text

Description automatically generated

Figure 4, Wireshark packet of message between Raspberry Pi and Server.

Above is a screenshot of a packet containing the message from figure 2. Highlighted in blue is the message that was sent between the IoT device and the server. As you can see the message can be interpreted easily as it is in plain text. Hopefully in the real smart bulb this information is encrypted. Another thing to note is that the packet itself is timestamped. This means that even though the time is not directly sent in the message it can be recorded. This information could be used to create a record of when lights are commonly on or off. This is very unsafe as it can alert potential intruders when you will likely be out.

Text

Description automatically generatedText

Description automatically generated

Figure 5, Normal vs Poisoned ARP table.

Figure 5 shows the normal and poisoned ARP table for my Windows machine. On the left you can see normal ARP table. The default gateway, IP address 192.168.10.1, has a corresponding physical address of a4-91-b1-a5-d1-e6. However, when I ARP spoof from my Raspberry Pi, which we know has IP address 192.168.10.128, the physical address of the router has changed from a4-91-b1-a5-d1-e6 to b8-27-eb-93-cb-43. This corresponds to the physical address of the Raspberry Pi and not my router. You can clearly see that there has been some poisoning as both the IP of the default gateway and the IP of the router point to the same physical address, that of the Raspberry Pi. This means that the attacker could now intercept and even alter packets without needing access to your actual machine.

# Discussion

From the experiments it is evident that there is a potential security risk from IoT devices. However, like I mentioned before, this is not a completely true representation of the TP-Link IoT devices. TP-Link claim that they use SSL and encryption which will change the contents from plain text to a form that an attacker will not be able to read. This reduces the risk of data being intercepted by the wrong people. Another security measure taken by TP-Link regarding their smart bulb is ‘away mode’. This mode simulates someone being home to discourage any potential intruders.

The experiment was done with a focus on the TP-Link smart bulb. TP-Link is a company that generated 94 million dollars in revenue in 2020 [8]. You would expect them to have good security measures in place. However, this is just one example. Many other IoT devices may not have the same level of security and thus might be more like the experiment in this report and thus it can be concluded that IoT devices can potentially threaten our privacy if the correct security measures are not set in place. Therefore, this report concludes that there is cause for concern surrounding the safety of our IoT devices and that security should be the focal point of these devices as information can so easily fall into the wrong hands.

Doing this research paper has thought me many things. It strengthened my understanding of internetworking and how devices communicate over a network. If I had more time, I would like to perform the same experiments using an actual IoT device and not an emulated version. I would also like to compare different IoT devices with varying levels of security.

References

[1] – J. Clark, What is the Internet of Things (IoT)?, IBM.com. <https://www.ibm.com/blogs/internet-of-things/what-is-the-iot/> (accessed 2 May 2021)

[2] – G. Maayan, The IoT Rundown For 2020: Stats, Risks, and Solutions, Jan 13, 2020, securitytoday.com. <https://securitytoday.com/Articles/2020/01/13/The-IoT-Rundown-for-2020.aspx?Page=2> (accessed 2 May 2021)

[3] – P. Schousten, TP-Link offers way to add local API back, November 23 2020, Home-assistant.io. <https://www.home-assistant.io/blog/2020/11/23/tplink-local-access/> (accessed 2 May 2021)

[4] – G. Marks, A Casino Gets Hacked Through a Fish-Tank Thermometer, April 14, 2021, entrepreneur.com. <https://www.entrepreneur.com/article/368943> (accessed 2 May 2021)

[5] – TP-Link, Privacy & Terms of Use, October 12, 2020, tp-link.com. <https://www.tp-link.com/se/about-us/privacy/> (accessed May 2, 2021)

[6] – TP-Link, Smart Bulbs, tp-link.com. <https://www.tp-link.com/se/home-networking/smart-bulb/tapo-l510e/> (accessed May 4, 2021)

[7] - Techwithtim, (5 April 2020), Python Socket Programming Tutorial,

[online video] Available at: <https://www.youtube.com/watch?v=3QiPPX-KeSc&t=2565s> (Accessed May 4, 2021)

[8] – Dun & Bradstreet, D & B Business Directory, dnb.com. <https://www.dnb.com/business-directory/company-profiles.tp-link_usa_corporation.00f45eca47a75bd85a8be9dc987d62cf.html#:~:text=Tp-Link%20USA%20Corpo> (accessed May 4, 2021)

[9] – Dan Swinhoe, What is a man-in-the-middle attack? How MitM attacks work and how to prevent them (Feb 13, 2019), csoonline.com. <https://www.csoonline.com/article/3340117/what-is-a-man-in-the-middle-attack-how-mitm-attacks-work-and-how-to-prevent-them.html> (accessed May 10, 2021)

[10] – Markus Hidell, Network Security ITCP/IP Attacks, kth.se. <https://canvas.kth.se/courses/22612/files/4125635?module_item_id=330017> (accessed May 10, 2021)