Teardown of the TP-Link Kasa Smart Wi-Fi Plug KP115

Benjamin Simmonds (5233344), UNSW Canberra

July 2021

Abstract

As the Internet of Things (IoT) continues to take the world by storm with an estimated deployment footprint of 20 billion globally, security of these devices is a major area of concern, with direct impacts to economic and citizen safety. To better understand the severity of the situation, this paper presents the findings of a security analysis of a contemporary Smart Plug device purchased only weeks ago from Bunnings Warehouse. The results were alarming, a useless encryption scheme combined with no authentication, makes it a trivial exercise to gain complete control over the device. Lack of regulations in this space, combined with strong economic drivers for manufacturers to push IoT devices to the mass market quickly may be contributing factors.

Contents

Introduction	2
Aim Methodology Tools and Equipment	3 3 3
Observations Port scan TP-Link Device Firmware and Server Daemon Decompiled Kasa Smart Android App	4 5 5 5
Conclusion	6
Appendix A: Enumeration of TP-Link Smart Home Protocol Commands	7
Appendix B: Programmatic Control using Python Script Switch on	8 8 8 9 9 9 9 9 9
Appendix C: nmap scan results	9
References	10

Introduction

This paper steps through a "tear down" (a methodical decomposition of a system) of the TP-Link (Kasa) Smart Wi-Fi Plug Slim model KP115, which was purchased from Bunnings for about \$20 in July 2021.

Goals and questions:

- Evaluate firmware and software makeup
- Analyse the app-to-device network communication employed
 - Specific protocols and ports
 - The onboarding or discovery process
 - The use of any WAN (cloud) based services
- Assess any potential security mechanisms used, such as encryption
- If any are found to exist, practically demonstrate vulnerabilities



Figure 1: The TP-Link/Kasa Smart Plug under assessment (Co (2021a))

TP-Link is the umbrella company of Kasa, a consumer device manufacturer with a focus on IoT connected "smart devices".

Some basic reconnaissance (TP-Link 2021) shows the "Kasa Smart Wi-Fi Plug Slim" is an actively supported product line and markets many consumer desirable qualities such as ease of use and setup, controllable from anywhere, voice control, an integrated timer for scheduling and even boasts "Multiple Safety Guarantees".

Aim

Methodology

The security assessment was structured based on the lifecycle of an industry based offensive methodology, by MITRE Corporation (Corporation (2021)).

- 1. Reconnaissance: research on functionality, software eco-system available, documentation or patents on protocols, similar product lines, CVE databases
- 2. Resource development: create cloud account, install the android app, understand base functions, reverse engineer android app, design test bench
- 3. Initial access: establish connectivity to the smart plug via rogue access point (evil twin)
- 4. Discovery: network service scanning
- 5. Collection: MITM AP network sniffing
- 6. Command and control: reverse engineer the TP-Link Smart Home protocol (TSHP) and fabricate commands

The specifications (Co (2021c)) highlight the device operates over 2.4GHz 802.11 b/g/n. This eliminated use of Bluetooth related protocols, focusing the design for intercepting network traffic to Wi-Fi only. It turns out a majority of IoT devices deployed in the wild are Wi-Fi based. One theory is that this is driven by cost saving measures by manufacturers, due to the ubiquity and affordability of Wi-Fi (Junior et al. (2019)).

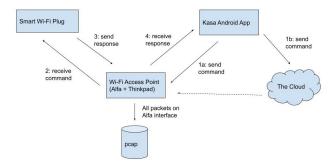


Figure 2: High level design to intercept all network activity

Tools and Equipment

The security assessment made use of the following.

Equipment:

- A Kasa Smart Wi-Fi Plug Slim (KP115) i.e., the device under evaluation
- Thinkpad X220 running Arch Linux
- Alfa AWUS036NHA USB Wi-Fi interface
- Google Pixel 4A Android Smartphone

Tools and software:

- The official Kasa Smart Android App
- A copy of the firmware HS110(US)_V1_151016.zip running on the device (Co (2021b))
- Wireshark and tcpdump
- binwalk
- tplink-smarthome.lua: a custom wireshark LUA dissector (Stroetmann and Esser (2018b))

- tplink_smartplug.py: a proof of concept that implements the weak XOR autokey encryption used by the TP-Link Smart Home protocol (TSHP) called (Stroetmann and Esser (2018c))
- Firmware for the Atheros family of network devices (for the Alfa)



Figure 3: Equipment used (from left to right) an Alfa AWUS036NHA, the Kasa Smart Plug KP115 and Linux ThinkPad

After configuring the Atheros OS driver for the Alfa card, a Wi-Fi hotspot named "zeit8023" was setup:

\$ nmcli dev wifi hotspot ifname wlx00c0ca991c64 ssid zeit8023 password 13371337

\$ nmcli d			
DEVICE	TYPE	STATE	CONNECTION
wlx00c0ca991c64	wifi	connected	Hotspot
enp0s25	ethernet	unavailable	
lo	loopback	unmanaged	
wlp3s0	wifi	unmanaged	

Having installed the "Kasa Smart" Android app from the Google play store on my Google Pixel 4A, the Google Pixel 4A Smartphone was connected to the hotspot, and TP-Link Smart Plug plugged into a power socket. The smart plug booted up automatically, and launched the pairing process.

On the ThinkPad, I configured Wireshark to profile all traffic flowing over the Alfa card. Using the "Kasa Smart" app on my Android I performed the following functions:

- 1. Switched the smart plug on.
- 2. Switched the smart plug off.
- 3. Switched the smart plug on.
- 4. Created a schedule to turn the switch off at 20:45
- 5. Switched the smart plug on.
- 6. Switched the smart plug off.

Observations

Some high-level facts that were immediately drawn from the network capture:

- The MAC address of the smart plug is c0:06:c3:09:e0:61
- The IP address assigned to the smart plug is 10.42.0.126
- The IP address assigned to the Android Pixel Smartphone is 10.42.0.202

Port scan

After capturing the network trace, a port scan was conducted against the smart plug device, which revealed port 9999 was open – see Annex: C nmap scan results.

Analysing the port 9999 related traffic in Wireshark revealed that many TCP conversations were taking place between the Android Smartphone (the app) and the TP-Link Smart Plug. The message data appeared to contain indecipherable binary data. Using CyberChef I attempted to decode the binary data using several common encoding and encryption methods, to no avail.

TP-Link Device Firmware and Server Daemon

Running **binwalk** on a copy of the firmware running on the smart plug (Co (2021b)), showed it was running a Linux 2.6 based kernel, and several other popular embedded sub-systems:

- U-Boot Bootloader 1.1.4
- Linux Kernel 2.6.31
- A squashfs based filesystem

Examining the contents of the filesystem, discovered the following files of interest:

- /bin/busybox v1.01 (2015.10.16-03:17+0000)
- /usr/bin/shd

It turns out (Stroetmann and Esser (2018a)) that /usr/bin/shd is the daemon server process responsible for handling the port 9999 related traffic, and is a C binary:

shd: ELF 32-bit MSB executable, MIPS, MIPS32 rel2 version 1 (SYSV), dynamically linked, interpreter /lib/ld-uClibc.so.0, corrupted section header size

At this point I decided to abandon reversing the smart plug shd server process any further.

Decompiled Kasa Smart Android App

Given the app is the command center for all its connected smart devices, I conceptualised that it would need to encrypt the "command data" in whatever scheme was being used.

Using "Apk Extractor" on the Google Pixel Smartphone, I was able to extract the "Kasa Smart" app's apk package, and it turn it's Java bytecode. With some help of security researchers (Stroetmann and Esser (2018a)) I quickly discovered the key encryption function:

```
public static byte[] m7377b(byte[] bArr) {
    if (bArr != null && bArr.length > 0) {
        int i = -85;
        for (int i2 = 0; i2 < bArr.length; i2++) {
            byte b = (byte) (i ^ bArr[i2]);
            i = bArr[i2];
            bArr[i2] = b;
        }
    }
    return bArr;
}</pre>
```

This simple encryption function uses the XOR bitwise operator ($^{}$) and washes each byte of the "command data", starting with the hardcoded IV value of -85 (or 171). Each iteration, the next cleartext data byte is XOR'ed with the previous cleartext byte (i.e. i = bArr[i2]). Involving the actual data as the key is known as an autokey cipher (Foundation (2021)).

Given the use of a hardcoded nonce (or IV), this process can easily be reversed to decrypt the message data.

As a proof of concept, a Python implementation tplink_smartplug.py and custom Wireshark LUA dissector tplink-smarthome.lua (Stroetmann and Esser (2018a)) was used to conveniently decrypt and analyse the autokey encrypted conversations taking place between the app and the smart plug.

Dispert 9999.55 (plink smarthere None Destination Proceed Length Info 20.0.63143 10.42.0.322 Destination Proceed 147.86222 - WHM (PSA, ACK) Sep-1 Ack-1 Min-05535 Len- 20.0.63143 20.0.63143 10.42.0.325 DOLK 0.125 PTLDM: SMARTHERE(JSDM 147.86222 - WHM (PSA, ACK) Sep-1 Ack-1 Min-05535 Len- 20.0.63143 20.0.63143 10.42.0.325 DOLK 0.0125 TTLDM: SMARTHERE(JSDM 107.86222 - WHM (PSA, ACK) Sep-1 Ack-4 Min-05535 Len- 20.0.630157 20.0.63143 DOLK 0.0126 DOLK 0.0126 TTLDM: SMARTHERE(JSDM 107.86224 - MIN (PSA, ACK) Sep-1 Ack-10177 Len- 20.0000 - 44176 (PSA, ACK) Sep-1 Ack-10177 Len- 20.0000 - 44176 (PSA, ACK) Sep-1 Ack-10177 Len- 20.0000 - 90000 - 44177 (PSA, ACK) Sep-1 Ack-10177 Len- 20.0000 - 90000 - 44177 (PSA, ACK) Sep-1 Ack-10177 Len- 20.0000 - 90000 - 44177 (PSA, ACK) Sep-1 Ack-10177 Len- 20.0000 - 90000 - 44177 (PSA, ACK) Sep-1 Ack-10177 Len- 20.0000 - 90000 - 44120 (PSA, ACK) Sep-1 Ack-10177 Len- 20.0000 - 90000 - 44120 (PSA, ACK) Sep-1 Ack-10177 Len- 20.0000 - 90000 - 44120 (PSA, ACK) Sep-1 Ack-10177 Len- 20.0000 - 90000 - 44120 (PSA, ACK) Sep-1 Ack-10177 Len- 20.0000 - 90000 - 44120 (PSA, ACK) Sep-1 Ack-10177 Len- 20.0000 - 90000 - 44120 (PSA, ACK) Sep-1 Ack-10177 Len- 20.0000 - 90000 - 44120 (PSA, ACK) Sep-1 Ack-10177 Len- 20.0000 - 90000 - 44120 (PSA, ACK) Sep-1 Ack-10177 Len- 20.00000 - 90000 - 44120 (PSA, ACK) Sep-1 Ack-10177 Len- 20.00000 - 90000 - 44120 (PSA, ACK) Sep-1 Ack-10177 Len- 20.000000 - 90000 - 900000000000000000000	Len-622, JawaScript Object Kotat cm-100, JawaScript Object Kotat Len-49, JawaScript Object Kotat Len-104, JawaScript Object Kotat Len-64, JawaScript Object Kotati Len-62, JawaScript Object Kotati	 Seg-1 Acke-94 Win-2027 Len-622, Jun Seg-1 Acke-94 Win-2023 Len-105, Jun Seg-1 Acke-107 Win-2014 Len-15, Jun Seg-1 Acke-107 Win-2014 Len-16, Jun Seg-1 Acke-107 Win-2014 Len-49, Jun Seg-1 Acke-107 Win-2014 Len-49, Jun Seg-1 Acke-107 Win-2014 Len-49, Jun Seg-1 Acke-107 Win-2015 Len-00, Junu 	147 48372 - 9999 576 9999 - 48372 159 48375 - 9999 163 9999 - 48376 156 48382 - 9999 103 9999 - 48382	TPLDK-SWRTHING/JSON TPLDK-SWRTHING/JSON TPLDK-SWRTHING/JSON	10.42.0.125 10.42.0.282 10.42.0.126	Source 10.42.0.382 10.42.0.125	Time 29 0.643147 30 0.699157
29 0.43143 10.42.0.322 10.42.0.325 10.42.	Len-622, JawaScript Object Kotat cm-100, JawaScript Object Kotat Len-49, JawaScript Object Kotat Len-104, JawaScript Object Kotat Len-64, JawaScript Object Kotati Len-62, JawaScript Object Kotati	 Seg-1 Acke-94 Win-2027 Len-622, Jun Seg-1 Acke-94 Win-2023 Len-105, Jun Seg-1 Acke-107 Win-2014 Len-15, Jun Seg-1 Acke-107 Win-2014 Len-16, Jun Seg-1 Acke-107 Win-2014 Len-49, Jun Seg-1 Acke-107 Win-2014 Len-49, Jun Seg-1 Acke-107 Win-2014 Len-49, Jun Seg-1 Acke-107 Win-2015 Len-00, Junu 	147 48372 - 9999 576 9999 - 48372 159 48375 - 9999 163 9999 - 48376 156 48382 - 9999 103 9999 - 48382	TPLDK-SWRTHING/JSON TPLDK-SWRTHING/JSON TPLDK-SWRTHING/JSON	10.42.0.125 10.42.0.282 10.42.0.126	10.42.0.282 10.42.0.125	29 0.683147 30 0.699157
20 0.4210 10.42.0.125 10.42.0.202 TPLDMC-SMATHEME(JSDM 570 5000 4.41272 (FML Soc) 560.44.0.424 Mon-2027 77 5.175005 10.42.0.212 10.42.0.207 TPL2MC-SMATHEME(JSDM 500 5000 440.2172 (FML Soc) 560.44.0.4272 10.42.0.202 10.42.0.207 10.42.0.210 10.42.0.207 10.42.0.210 10.42.0.207 10.42.0.207 10.42.0.210 10.42.0.207 10.42.0.210 10.42.0.207 10.42.0.207 10.42.0.210 10.42.0.207 <td< th=""><td>Len-622, JawaScript Object Kotat cm-100, JawaScript Object Kotat Len-49, JawaScript Object Kotat Len-104, JawaScript Object Kotat Len-64, JawaScript Object Kotati Len-62, JawaScript Object Kotati</td><td> Seg-1 Acke-94 Win-2027 Len-622, Jun Seg-1 Acke-94 Win-2023 Len-105, Jun Seg-1 Acke-107 Win-2014 Len-15, Jun Seg-1 Acke-107 Win-2014 Len-16, Jun Seg-1 Acke-107 Win-2014 Len-49, Jun Seg-1 Acke-107 Win-2014 Len-49, Jun Seg-1 Acke-107 Win-2014 Len-49, Jun Seg-1 Acke-107 Win-2015 Len-00, Junu </td><td>575 9999 → 48372 959 48375 - 9999 183 9999 → 48376 196 48382 - 9999 103 9999 → 48382</td><td>TPLONK-SINARTHOME/JSON TPLONK-SINARTHOME/JSON</td><td>10.42.0.202</td><td>10.42.0.125</td><td>30 0.699157</td></td<>	Len-622, JawaScript Object Kotat cm-100, JawaScript Object Kotat Len-49, JawaScript Object Kotat Len-104, JawaScript Object Kotat Len-64, JawaScript Object Kotati Len-62, JawaScript Object Kotati	 Seg-1 Acke-94 Win-2027 Len-622, Jun Seg-1 Acke-94 Win-2023 Len-105, Jun Seg-1 Acke-107 Win-2014 Len-15, Jun Seg-1 Acke-107 Win-2014 Len-16, Jun Seg-1 Acke-107 Win-2014 Len-49, Jun Seg-1 Acke-107 Win-2014 Len-49, Jun Seg-1 Acke-107 Win-2014 Len-49, Jun Seg-1 Acke-107 Win-2015 Len-00, Junu 	575 9999 → 48372 959 48375 - 9999 183 9999 → 48376 196 48382 - 9999 103 9999 → 48382	TPLONK-SINARTHOME/JSON TPLONK-SINARTHOME/JSON	10.42.0.202	10.42.0.125	30 0.699157
Tro 5.17300F 10.42.0.126 10.42.0.327 TPLINE-SMATHEME(/1504) 105 9999 - 44370 (F94, AK) Social Ach.017 Win-2016 Letter 224 60.35813 10.42.0.326 10.42.0.326 TPLINE-SMATHEME(/1504) 100 4994 - 44370 [F94, AK) Social Ach.017 Win-2016 Letter 224 60.35813 10.42.0.326 10.42.0.326 TPLINE-SMATHEME(/1504) 100 4994 - 44382 F994, AK) Social Ach.017 Win-2016 Letter 246 60.369642 10.42.0.326 10.42.0.326 TPLINE-SMATHEME(/1504) 100 4994 - 44382 F994, AK) Social Ach.017 Win-2016 Letter 246 60.369642 10.42.0.326 10.42.0.326 TPLINE-SMATHEME(/1504) 107 4994 - 4909 (F94, AK) Social Ach.018 Win-3017 Lenter 247 60.701585 10.42.0.326 10.42.0.327 TPLINE-SMATHEME(/1504) 107 6999 - 44334 1994, AK) Social Ach.018 Win-3027 Lenter 247 60.701585 10.42.0.326 10.42.0.322 TPLINE-SMATHEME(/1504) 576 9999 - 44334 1994, AK) Social Ach.018 Win-3027 Lenter 247 60.701585 10.42.0.322 10.42.0.322 10.42.0.322 10.42.0.322 10.42.0.322 247 60.701585 10	Len-45, JavaScript Object Notat Len-106, JavaScript Object Notat Len-45, JavaScript Object Notat Len-63, JavaScript Object Notati	 Soq-1 Ack-107 Win-2014 Len-45, Jaw Soq-1 Ack-1 Win-0535 Len-106, Jaw Soq-1 Ack-107 Win-2014 Len-49, Jaw Soq-1 Ack-1 Win-05535 Len-93, Java 	$183 9999 \rightarrow 48376$ 160 48382 = 9999 $103 9999 \rightarrow 48382$			10 41 0 101	
222 60.28513 10.42.0.22 10.42.0.25 TPLDM::SMATHOME/JSDM 100.0022 - 0999 [PSH, AK] Sonj-1 Acks1 Min-65535 Len- 225 60.2004 10.42.0.22 10.42.0.22 TPLDM::SMATHOME/JSDM 103.9999 - 4832 (PSH, AK) Sonj-1 Acks1 Min-65535 Len- 246 60.00049 10.42.0.22 10.42.0.22 TPLDM::SMATHOME/JSDM 107.0029 - 48204 (PSH, AK) Sonj-1 Acks1 Min-65535 Len- 247 60.201565 10.42.0.125 10.42.0.20 TPLDM::SMATHOME/JSDM 576 9999 - 44304 (PSH, AK) Sonj-1 Acks-24 Min-65535 Len- 247 60.201565 10.42.0.125 10.42.0.20 TPLDM::SMATHOME/JSDM 576 9999 - 44304 (PSH, AK) Sonj-1 Acks-24 Min-65535 Len- 247 60.201565 10.42.0.125 10.42.0.20 TPLDM::SMATHOME/JSDM 576 9999 - 44304 (PSH, AK) Sonj-1 Acks-24 Min-65535 Len- 576 9999 - 44304 (PSH, AK) Sonj-1 Acks-24 Min-63535 Len- 576 9999 - 44304 (PSH, AK) Sonj-1 Acks-24 Min-63535 Len- 576 9999 - 44304 (PSH, AK) Sonj-1 Acks-24 Min-63535 Len- 576 9999 - 44304 (PSH, AK) Sonj-1 Acks-24 Min-63535 Len- 576 9999 - 44304 (PSH, AK) Sonj-1 Acks-24 Min-63535 Len- 576 9999 - 44304 (PSH, AK) Sonj-1 Acks-24 Min-63535 Len- 576 9999 - 44304 (PSH, AK) Sonj-1 Acks-24 Min-63535 Len- 576 9999 - 44304 (PSH, AK) Sonj-1 Acks-24 Min-63535 Len- 576 9999 - 44304 (PSH, AK) Sonj-1 Acks-24 Min-63535 Len- 576 9999 - 44304 (PSH, AK) Sonj-1 Acks-24 Min-63535 Len- 576 9999 - 44304 (PSH, AK) Sonj-1 Acks-24 Min-63535 Len- 576 9999 - 44304 (PSH, AK) Sonj-1 Acks-24 Min-63535 Len- 576 9999 - 44304 (PSH, AK) Sonj-1 Acks-24 Min-6353 Fiberrat II, Src: 56:271-471:78:76:76:76:76:70, Dat: TP-Link,993:81:61 (00:96:25:99:81:61) Internet Protocol Wernie 4, Src: 10.42.0.282, Dat: 18-42.136 Teasentistion Control, Protocol, Scr Port Prot. 9999, Sonj: 1, Ack 1, Len: 106	Les-106, JavaScript Object Notat Les-49, JavaScript Object Notat Les-93, JavaScript Object Notati	[] Seq=1 Ack=1 Min=65535 Len=106, Jaw [] Seq=1 Ack=107 Win=3816 Len=49, Jaw [] Seq=1 Ack=1 Min=65535 Len=93, Jawa	$160 \ 48382 - 9999$ $103 \ 9999 \rightarrow 48382$	TPLDK-SW/THME/JS04		10.42.0.202	77 5.357007
225 60.70063 10.42.0.126 10.42.0.27 TPLDW-SWATHWE(7150H 10.9944 - 4005) [F94, A03] Sepi-1 Ach-107 Win-3010 Let 246 60.98645 10.42.0.202 10.42.0.125 TPLDW-SWATHWE(7150H 107 600-4844 - 9906) F94, A03] Sepi-1 Ach-107 Win-3010 Let 247 60.701565 10.42.0.125 10.42.0.202 TPLDW-SWATHWE(7150H 676 9009 - 44304 [P34, A03] Sepi-1 Ach-04 Win-3027 Len- Frame 777 166 bytes on wire (1200 bits), 166 bytes captured (1208 bits) Frame 777 166 bytes on wire (1200 bits), 166 bytes captured (1208 bits) Frame 777 166 bytes on wire (1200 bits), 166 bytes captured (1208 bits) Frame 777 166 bytes on wire (1200 bits), 166 bytes captured (1208 bits) Frame 777 166 bytes on wire (1200 bits), 168 bytes captured (1208 bits) Frame 777 166 bytes on wire (1200 bits), 168 bytes captured (1208 bits) Frame 777 166 bytes on wire (1200 bits), 168 bytes captured (1208 bits) Frame 777 166 bytes on wire (1200 bits), 168 bytes captured (1208 bits) Frame 777 166 bytes on wire (1200 bits), 167 bytes captured (1208 bits)	Len-49, JavaScript Object Notat Len-63, JavaScript Object Natati	[] Seq=1 Ack=107 Win=2014 Len=49, Jaw [] Seq=1 Ack=1 Win=65535 Len=93, Jawa	103 9999 - 48382				
246 68.696493 10.42.0.282 10.42.0.226 TPLINK-SMATHUME/ISSN 147.40284 - 9990 [PSH, ACK] Smg-1 Ack-1 Min-05235 Lem- 247 60.701365 10.42.0.125 10.42.0.282 TPLINK-SMATHUME/ISSN 676 9999 - 44304 [PSH, ACK] Smg-1 Ack-04 Min-3827 Lem- 247 60.701365 10.42.0.125 10.42.0.282 TPLINK-SMATHUME/ISSN 676 9999 - 44304 [PSH, ACK] Smg-1 Ack-04 Min-3827 Lem- 100 (1990) [PSH, ACK] Smg-1 Ack-04 Min-3827 Lem- Frame 77: 160 bytes on wire (1260 bits), 160 bytes captured (1380 bits) Ethernet II, Src: 56:21:41:78:76:10:42:0.282, Dat: TP-Link,90:80:61 (00:86:c3:09:e8:61) Internet Protocol Wrmine 4, Src: 10.42.0.282, Dat: 18.42.0.126 Treasenission Control, Protocol, Scr Port, 48776, 504 Frei, 1399, Seg: 1, Ack: 1, Lem: 106	Les-93, JavaScript Object Notati	[] Seq-1 Ack-1 Min-65535 Len-93, Java					
247 60.701265 10.42.0.125 10.42.0.282 TPLDW: SMWTHOME/150N 676 9009 - 44334 (PSH, ACK) Seq-1 Ack-04 Win-2827 Len- Frame 77: 166 bytes on wire (1266 bits), 166 bytes captured (1266 bits) Ethermet II, Src: S6:271-47:78:Terc1 56:271-47:78:Terc4:23, Dat: TP-Link,99:08:61 (00:86:c1:09:e8:61) Intermet Froitcol Wernise 4, Src: 10.42.0.282, Dat: 18-42.126 Tresentision Control, Protocol, Scr Port, 48776, Bat Part; 8929, Seq: 1, Ack 1, Len: 106							
Frame 77: 160 bytes on wire (1260 bits), 160 bytes captured (1260 bits) Ethernet II, Src: 56:21:41:78:Fe:c2 156:21:41:78:Fe:c2), But: TP-Link,90:e8:61 (d0:86:c2:09:e8:61) Internet Protocol Wernise 4, Src: 10.42.0.282, Dat: 18.42.0.126 Treasmission Cartel Protocol, Sc: Port: 48776, Bat Pre1: 9992, Seg: 1, Ack: 1, Len: 106	une-our, Javascript Deject wotar						
Sthermet II, Sri: SS:21:41:78:76:23 (SS:21:41:70:fe:c3), Dat: TP-Link_00:08:61 (c0:85:63:09:08:61) Internet Protocol Hermion 4, Sri: D.42.0.282, Dat: 18.42.0.126 Internetision Central Protocol, Sri Port 4870, DSI Pert: 9009, Seg: 1, Ack: 1, Len: 106							
Sthermet II, Sri: SS:21:41:78:76:23 (SS:21:41:70:fe:c3), Dat: TP-Link_00:08:61 (c0:85:63:09:08:61) Internet Protocol Hermion 4, Sri: D.42.0.282, Dat: 18.42.0.126 Internetision Central Protocol, Sri Port 4870, DSI Pert: 9009, Seg: 1, Ack: 1, Len: 106							
ithernet II, Sri: 56:21:41:78:fe:23 (56:21:44:70:fe:23), But: TP-Link_00:e0:61 (20:86:23:09:e0:61) ntermet Proincel Wermion 4, Sri: 10.42.0.282, Dut: 18.42.0.126 remeninsion Control Protocol, Sri Port: 44370, DSI Port: 1999, Seg: 1, Ack: 1, Len: 106							
Internet Proiscol Warsion 4, Src: 10.42.0.282, Dut: 18.42.0.126 Transmission Control Protocol, Src Port: 48370, Est Part: 9999, Sog: 1, Ack: 1, Len: 106							
Transmission Control Protocol, Src Port: 48378, Est Port: 9999, Seg: 1, Ack: 1, Len: 105			9:e8:61)				
Transmission Cantral Pretocol, Src Port: 463/0, BSt Part: 9999, Seg: 1, ACK: 1, Len: 105							
				9999, Sed: 1, ACK: 1, DEN:	ort: 463/6, Est Peri	met Pretocol, Srt P	PROFILING SHORT NO
wwsicrist Object Natarian					1000		
a Th 22 63 67 66 74 65 78 74 22 3e Th 22 75 67 73 {*contex 1*. 17 6 25 65 22 3e 25 3e 35 63 3e 65 3e 62 2d 3e row*:*46							

Figure 4: Wireshark analysis using custom LUA dissector

Now able to decipher the commands, set out to enumerated each function that was invoked using the smartphone app, and the resulting cleartext JSON message. See Annex: A Enumeration of TP-Link Smart Home Protocol Commands.

Finally, an end-to-end demonstration of just how weak security (essentially non-existent) measures are, invoked each function of the smart plug using Python. See Annex: B Programmatic Control using Python Script.

Conclusion

The TP-Link (Kasa) Smart Plug demonstrates the delicate balance the industry currently faces towards providing consumer convenient connected technology (i.e., IoT) while in a safe and secure manner. As this paper shows, the security concerns are real.

On one end of the spectrum IoT manufacturers are under aggressive pressure to push products to market with the hope of soaking up as much market share as possible, taking paths of least resistance to "just get the job done". On the other end, consumers, small and medium sized businesses are the ones left exposed due to under-baked products.

What deems an appropriate level of "safe and secure", what frameworks or standards could be

applied without burdening technology companies with unrealistic overheads while providing consumer comprehendible advice. Mr Farrell linked (Home Affairs (2021)) to the thought-provoking initiative the Australian government is actively working through, to address these exact issues.

Appendix A: Enumeration of TP-Link Smart Home Protocol Commands

Switch on request:

```
ſ
    "context": { "source": "46a4d58b-6279-432c-ae23-e115c2db8354" },
    "system": { "set_relay_state": { "state": 1 } }
}
Switch off request:
{
    "context": { "source": "46a4d58b-6279-432c-ae23-e115c2db8354" },
    "system": { "set_relay_state": { "state": 0 } }
}
Scheduled "turn off" event:
{
    "schedule": {
        "get_rules": {
            "rule_list": [
                {
                     "id": "689CB20F436CEDEE16F8752FDFD26EBC".
                     "name": "Schedule Rule",
                     "enable": 1,
                    "wday": [1, 1, 1, 1, 1, 1],
                     "stime_opt": 0,
                     "smin": 1245,
                     "sact": 0,
                     "eact": -1,
                     "repeat": 1
                }
            ],
            "version": 2,
            "enable": 1,
            "err code": 0
        }
    }
}
System status response:
{
    "system": {
        "get_sysinfo": {
            "sw_ver": "1.0.17 Build 210506 Rel.075231",
            "hw_ver": "1.0",
```

```
"model": "KP115(AU)",
        "deviceId": "8006D8042A39366BDFDD6E65504D85FF1E64FC77",
        "oemId": "33AF6F102C6172F2E5BD6CEA656C19FF",
        "hwId": "8EDCE7F6CF8231678B90A6730D02BB41",
        "rssi": -34,
        "latitude_i": -354348,
        "longitude i": 1491063,
        "alias": "Kasa smart plug",
        "status": "new",
        "obd_src": "tplink",
        "mic_type": "IOT.SMARTPLUGSWITCH",
        "feature": "TIM:ENE",
        "mac": "C0:06:C3:09:E0:61",
        "updating": 0,
        "led_off": 0,
        "relay_state": 1,
        "on_time": 31,
        "icon_hash": "",
        "dev_name": "Smart Wi-Fi Plug Mini",
        "active_mode": "schedule",
        "next_action": {
            "type": 1,
            "id": "689CB20F436CEDEE16F8752FDFD26EBC",
            "schd sec": 74700,
            "action": 0
        },
        "ntc_state": 0,
        "err_code": 0
    }
}
```

Appendix B: Programmatic Control using Python Script

Switch on

}

```
$ python3 ./tplink_smartplug.py -t 10.42.0.126 -c on
Sent: {"system":{"set_relay_state":{"state":1}}}
Received: {"system":{"set_relay_state":{"err_code":0}}}
```

Switch off

```
$ python3 ./tplink_smartplug.py -t 10.42.0.126 -c off
Sent: {"system":{"set_relay_state":{"state":0}}}
Received: {"system":{"set_relay_state":{"err_code":0}}}
```

Report energy use metrics

```
$ python3 ./tplink_smartplug.py -t 10.42.0.126 -c energy
Sent: {"emeter":{"get_realtime":{}}
Received: {"emeter":{"get_realtime":{"current_ma":0,"voltage_mv":242327,"power_mw":0,"total_wh":0,"
```

Show cloud details

```
$ python3 ./tplink_smartplug.py -t 10.42.0.126 -c cloudinfo
Sent: {"cnCloud":{"get_info":{}}}
Received: {"cnCloud":{"get_info":{"username":"ben@bencode.net","server":"n-devs.tplinkcloud.com","b
```

Run an access point (AP) scan

```
$ python3 ./tplink_smartplug.py -t 10.42.0.126 -c wlanscan
Sent: {"netif":{"get_scaninfo":{"refresh":0}}}
Received: {"netif":{"get_scaninfo":{"ap_list":[],"err_code":0}}}
```

Show the time

```
$ python3 ./tplink_smartplug.py -t 10.42.0.126 -c time
Sent: {"time":{"get_time":{}}
Received: {"time":{"get_time":{"year":2021,"month":9,"mday":9,"hour":21,"min":9,"sec":42,"err_code"
```

List general information (including firmware version)

```
$ python3 ./tplink_smartplug.py -t 10.42.0.126 -c info
Sent: {"system":{"get_sysinfo":{}}}
Received: {"system":{"get_sysinfo":{"sw_ver":"1.0.17 Build 210506 Rel.075231","hw_ver":"1.0","model
```

List active schedules

```
$ python3 ./tplink_smartplug.py -t 10.42.0.126 -c schedule
Sent: {"schedule":{"get_rules":{}}
Received: {"schedule":{"get_rules":{"rule_list":[{"id":"689CB20F436CEDEE16F8752FDFD26EBC","name":"S
```

Appendix C: nmap scan results

```
nmap -v -sV 10.42.0.126
Starting Nmap 7.80 ( https://nmap.org ) at 2021-09-07 21:04 AEST
NSE: Loaded 45 scripts for scanning.
Initiating Ping Scan at 21:04
Scanning 10.42.0.126 [2 ports]
Completed Ping Scan at 21:04, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 21:04
Completed Parallel DNS resolution of 1 host. at 21:04, 0.01s elapsed
Initiating Connect Scan at 21:04
Scanning 10.42.0.126 [1000 ports]
Discovered open port 9999/tcp on 10.42.0.126
Completed Connect Scan at 21:04, 1.84s elapsed (1000 total ports)
Initiating Service scan at 21:04
Scanning 1 service on 10.42.0.126
Completed Service scan at 21:04, 9.11s elapsed (1 service on 1 host)
NSE: Script scanning 10.42.0.126.
Initiating NSE at 21:04
Completed NSE at 21:04, 0.01s elapsed
Nmap scan report for 10.42.0.126
```

Host is up (0.054s latency). Not shown: 999 closed ports PORT STATE SERVICE VERSION 9999/tcp open abyss?

References

- Co, TP-Link Technologies. 2021a. "Kasa Smart Wi-Fi Plug Slim Product Page." September 2021. https://www.tp-link.com/au/home-networking/smart-plug/kp105/.
 - 2021b. "TP-Link Smart Plug Firmware Hs110(US)_V1_151016.zip." September 2021. https://static.tp-link.com/res/down/soft/HS110(US)_V1_151016.zip.
- ———. 2021c. "TP-Link Smart Plugs Kp105 V2 Specifications." September 2021. https://www.tp-link.com/au/home-networking/smart-plug/kp105/#specifications.
- Corporation, MITRE. 2021. "The MITRE ATT&CK Framework." January 2021. https://attack.mit re.org/.
- Foundation, Wikimedia. 2021. "Autokey Cipher." September 2021. https://en.wikipedia.org/wiki/ Autokey_cipher.
- Home Affairs, The Department of. 2021. "Strengthening Australia's Cyber Security Regulations and Incentives." July 2021. https://www.homeaffairs.gov.au/reports-and-publications/submissionsand-discussion-papers/cyber-security-regulations-incentives.
- Junior, Davino Mauro, Luis Melo, Hao Lu, Marcelo d'Amorim, and Atul Prakash. 2019. "A Study of Vulnerability Analysis of Popular Smart Devices Through Their Companion Apps." In 2019 IEEE Security and Privacy Workshops (SPW), 181–86. IEEE.
- Stroetmann, L, and Esser T. 2018a. "Reverse Engineering the TP-Link Hs110." July 2018. https: //www.softscheck.com/en/reverse-engineering-tp-link-hs110/.
- ——. 2018b. "Tplink-Smarthome.lua." July 2018. https://github.com/softScheck/tplink-smartplug/blob/master/tplink-smarthome.lua.
- ——. 2018c. "Tplink_smartplug.py." July 2018. https://github.com/softScheck/tplink-smartplug/blob/master/tplink_smartplug.py.