Smart Home Forensics: An Exploratory Study on Smart Plug Forensic Analysis

Asif Iqbal [1, 2], Johannes Olegård [1], Ranjana Ghimire [2], Shirin Jamshir [2], Andrii Shalaginov [3]
[1] School of Electrical Engineering and Computer Science (EECS), KTH Royal Institute of Technology, Stockholm, Sweden
[2] Department of Computer and Systems Sciences (DSV), Stockholm University, Stockholm, Sweden
[3] Norwegian University of Science and Technology, Gjøvik, Norway
asif.iqbal@ee.kth.se, jolegard@kth.se, {ragh9926, shja3302}@student.su.se, andrii.shalaginov@ntnu.no

2020 IEEE International Conference on Big Data (Big Data) | 978-1-7281-6251-5/20/$31.00 ©2020 IEEE | DOI: 10.1109/BigData50022.2020.9378183

Abstract—**Connectivity as a whole and the Internet of Things (IoT) has influenced a great many things in the past decade. Among those, the most prominent is our daily life routines, which have increasingly started to depend on technology. A Smart Home, being a central part, has gained more importance from a forensic perspective since it affects many lives and can be an easy target for cybercrimes. In this work in progress paper, we explore the feasibility of conducting forensic analysis on different Smart Plugs and what sort of challenges are encountered in such a forensic investigation. We also review current related work for forensic analysis of Smart Plugs.**

Keywords— Smart Plug, Smart Outlet, Smart Home, Cyber Forensics, Digital Forensics, Forensic analysis, Threat analysis, IoT, Smart Things

I. INTRODUCTION

Smart which used to be merely a word to describe human traits; has outsmarted itself and conquered the digital world. The word smart in technology started with the introduction of the term, Internet of Things (IoT) in 1999 by Kevin Ashton [1]. There are many variations of definition for IoT but in general, it is anything and everything that can connect to the internet whether wired or wireless which includes devices like a Smart Plug, Smart Light, Smart Thermostat, Smart Refrigerator, and many more [2]. IoT is not limited to connecting to the internet, it can be interconnected devices as well. This very connection is what adds more value to IoT and has started to shape our lives to provide us with ease. IoT has provided us with central control of devices in the palm of our hands. The strong monitoring and reactive nature of IoT have made sure that the automation is seamless. All these features have aided in increasing the efficiency of the work which in turn has saved us time [3].

The very thing that made the devices smart has made them vulnerable as well [3]. Around 300,000 Internet Routers, cameras, printers, and DVRs were infected with Mirai botnet in 2016 [4]. The attack was distributed denial of service which started with exploiting a weak password [5]. Capturing human activities via a Smart Camera, spy camera has been going on for years. In 2019, an information security worker found a hidden camera attached within a router which was used to spy, disrupting their privacy [6]. In the same year, a 14-year-old hacker attacked 4,000 IoT devices via malware called Silex whose main targets were IoT running in Unix and Linux with known or guessable default passwords [7]. Similarly, Nokia saw a 400% increase in hacks to Smartphone and IoT devices [8]. After hacking Tesco Bank's mobile application, thieves have gotten away with 2.5 million euros in 2017 [8]. The number of new ways to exploit IoT devices outranks the number of devices introduced every year. With the existence of such exploits, the potential of criminal activities is bound to increase and hence forensics has become essential [9].

One of such IoT implementations is a Smart Plug. Before understanding the Smart Plug, let us go through what actual plugs are and their purpose. The plug is the device that creates an electric connection between the appliance and the mains, acting as a mediator for power lines and devices. The connection is not possible directly, the plug needs to be turned on for the electric transfer, which is manual work and the reason behind creating a smart one. Thus, a Smart Plug is simply the plug with some automation. The main reason behind calling it smart is because it allows the scheduling of appliances from the Smartphone [7], [10].

The Smart Plug itself is connected to the internet, making it vulnerable to attacks, however, the more serious problem is being able to reach the connected devices making them vulnerable as well. Earlier in 2018, the MacAfee team was able to access network devices by creating buffer overflow in Smart Plug [8], [10]. One simple slip and all the household devices get hacked. The higher the risk, the higher the chances of getting attacked. This paper focuses on performing forensic analysis in the Smart Plug, i.e., gathering all the artifacts that can be acquired from a Smart Plug via forensic analysis; whether it be of Smart Plug or the devices plugged on those.

II. RELATED WORK

Ling et al. [11] outline the vulnerabilities and exploits of Smart Plugs and accompanying Smartphone applications by creating a physical Man-in-the-middle (MITM) using USB wireless adapters. They discuss the defense strategies by capturing network traffic and reverse-engineering the Smart Plug firmware and protocol. Zhou et al. [12] use reverse engineering to find serious flaws like unauthorized device login and flawed device synchronization in Philips Smart Plug, Wi-Fi bulb, SmartThings, KASA, MIJIA, and Joylink. Wang et al. [13] study the existing issues with TP-Link Kasa HS100 Smart Plugs and concludes that the away mode lacks randomness and attackers can detect whether it is away mode or the owners are away, thus making it susceptible for attacks. Besides proving the not so intelligent away mode of Smart Plugs, they insight that by decreasing predictability, coordinating plug activity, and personalized per user, the away-mode vulnerability could be resolved. Halterman [14] finds the TP-Link protocols in TP-Link HS100 Smart Plugs, LB100 Smart Bulb, and LB100 Smart Bulb are vulnerable against snooping and spoofing.

Bettayeb et al. [15] propose a structure of testbeds for scanning vulnerabilities in different IoT devices. The testbed was designed to scan ports, network activities, and perform a man-in-the-middle attack between IoT/router and IoT/appliance. To detect anomalies, profile devices, and collect evidence, Brotsis et al. [16] propose a blockchain solution called Smart gateway agent. The solution works by sending the collected evidence from anomaly detection during device profiling to the evidence database.

Al-Masri et al. [17], Alharbi et al. [18], and Goudbeek et al. [19] propose various methods for capturing evidence while performing forensic analysis of IoT devices. During the forensic analysis of refrigerator and IoT sensors, Al-Masri et al. implement a fog computing paradigm in fog nodes, this very fog node could be used for extraction of interaction between devices and fog-node gateway. Similarly, Alharbi et al. propose a framework to collect digital evidence and record all logs of the connected smart devices and work as a central control for those devices. Goudbeek et al. collect multiple artifacts by conducting case studies in detecting human movements, identifying digital evidence in a Smart Home environment, and artifact detection in HAS-like environments. Hariri et al. [20] test Telldus Tellstick attack by exploiting heartbeats of IoT devices. The experiment was performed on Swann Smart Home Security Kit, D-Link Home Security Kit, D-Link camera, Panasonic Home Monitoring and Control Kit, Telldus Smart Home Start-up Kit, and Samsung SmartThings.

Awasthi et al. [21] conduct a forensics investigation of an “Almond+” Smart Hub. They extract the full file system of the Smart Hub by using ADB after rooting, and the commercial “Cellebrite UFED” tool. Kim et al. [22] propose and conduct forensic analysis on five different smart devices, Google Nest hub, Google Duo, TP-Link, Samsung SmartThings Classic, and Samsung SmartThings. The forensics was conducted in four steps, Device activation, device enrollment in-app, device function identification, and experiments with the device. Kim et al.'s experiments were able to collect activity logs, user credentials, and locations via forensic analysis. By analyzing network information, Do et al. [23] establish the connected devices and device status upon a forensic investigation of a Smart Home system including a Smart Light and Smart Plug.

Clinton et al. [24] extract owner ID, shopping list from forensic analysis of Amazon echo by running a custom operating system on Amazon Echo hardware by locating UART, JTAG, and MultimediaCard (MMC) pins.Vargas [25] shows privacy issues related to IoT devices and discovers artifacts in Amazon Echo Dot and Google mini after the forensic analysis. Tristan et al. [26] extract user information from Google Mini even when the user is not logged in which suggested Google Home has higher security flaws than Amazon Echo. In the comparison of Amazon Echo and Google Mini, Vargas shows that both products had Weak authentication, and both had major privacy issues in a synthetic voice.

Krueger et al. [27] and Nieto [28] use various methods to extract information from Amazon Alexa. Krueger extracts voice recording of previous calls, Bluetooth connected devices, user information, and card payment details from API call evaluation while Nieto works to provide all the information including user activity from Amazon Alexa into a user-friendly front end. Yıldırım [29] compares Google Home and Amazon Alexa by performing forensic analysis. The analysis suggested that for each action done via Google Home and Amazon Alexa, the evidence is left in the activity log along with timestamp and voice recording. Google Home was found to store an additional level of records in the activity log beside the Amazon Alexa, they are being the location of the device as well as the device type.

Chi et al. [30] propose a framework for data collection among multiple IoT devices including Smart Home and Smart Bed. The data collection was accessible via mobile devices making it easier to access. By loading the data to the MySQL database, the data could be seen in the front end through SQL queries.

There have been various proposals for upgrading the available Smart Plugs, most of them are focused on measuring energy consumption [31], [32], [33], [34], [35], [36], [37] Gomes et al. [31] add an extra feature in the plug to work in parallel with other Smart Plugs ensuring an optimized resource distribution. Dhaou [33] prototypes a Smart Plug with communication protocol using Arduino and Zigbee to demonstrate the demand management for low cost and interoperability. Heo et al. [35] introduce priority levels on Smart Plugs to support load balancing based on preserved energy in a photovoltaic generator. Morsali et al. [36] propose priority levels based on analyses of peak hours to prioritize pushing the load to specific Smart Plug via device recognition. Based on the power consumption history, Altaf et al. [37] classify appliances into six different categories and uses that classifier for online k-means clustering of unknown events and classifying them.

Ahmed et al. [38] analyze Smart Plugs for low power consumption and righted that wireless/Bluetooth sensors provided better data and Smart Plugs using Zigbee outperformed oscilloscopes with lower power consumption. The study concludes that Zigbee microcontroller sends data to the Home Energy management system with higher accuracy compared to an oscilloscope, higher accuracy data means better management of power consumption. Horvat et al. [39] provide a similar proposal that proposes Bluetooth as a data transferring method in Smart Plugs. The study moves further ahead to include a secure simple pairing mechanism for communication to remove Man-in-the-middle attacks.

Ridi et al. [40] propose two new protocols via smart signal processing in Smart Plugs for device recognition. The protocol, Dynamic Intersession, and Dynamic Unseen Appliance lay on the ground that each electronic device has its digital signature, whose database is made available for further research as well. The Machine learning approach was used for device identification, the devices were classified using a test-train set and k-fold approach. Suryadeva et al. [41] suggest that Smart Plug out wins Smart Meters in Energy Management, Device Identification, Device scheduling, Device control, Occupancy detection, Standby power kill, Indoor positioning, Thermal protection, and Overload protection.

Using the structural equation model, Ghazal et al. [42] validate that Smart Plug with energy consumption information allowed consumers to reduce per-capita energy consumption. Using the Zigbee protocol, the experiment was conducted by dividing the Smart Plugs into two categories, master with coordination capability and slave with power measurement capability.

III. EXPERIMENT AND OBSERVATION

In this paper, five IoT devices and their corresponding Android apps were investigated for digital forensic evidence.

Forensics is what law enforcement uses to identify, collect, preserve, and analyze evidence of the crime. Digital forensics is the part of forensics that involves information technology. An overview of the digital forensic process followed is given in Figure 1. Even though there already exists a well-established process for collecting digital evidence as a part of Computer Crime Investigation, one also needs to consider so-called Digital Forensics Readiness as was defined by Rowlingson [43]. Another need that arose in this area is keeping the Incident Response and Digital Forensics together as a part of the organizational strategy [44]. While Smart Plug is a small resource-constrained device, it can provide access to valuable digital evidence if the approach strategy has been properly chosen and best practices are followed. There exist numerous examples of IoT forensics readiness demonstrating sources of relevant data in the Edge environment [45]. Therefore, the goal of the experimental phase is to highlight the value of the data from Smart Plugs in overall crime investigation.

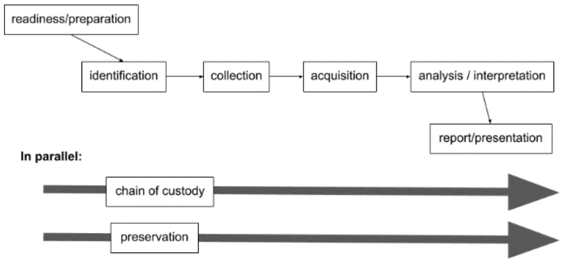

Figure 1 Overview of a Digital Forensic process

The goal was to investigate what evidence remains from actions taken in the app. For example, if an app was used to turn a Smart Plug "ON" at a certain time, can evidence of that action be found on the Smartphone or the device itself? The following are the research questions investigated resulting in the contributions of this paper.

Research questions:

- Whether the digital evidence is stored on the Smartphone and IoT devices?
- How evidence can be captured on the network during the experiment?
- What digital forensic challenges are encountered?

The experiment proceeded as follows. First various "activities" were conducted in each of the Applications (apps). The purpose of this phase is to leave evidence. Next, forensic copies for all the devices included in the experiment were made. Finally, the copies were analyzed to find evidence of the activities.

Digital forensic academic research attempts to document and solve various "challenges" in digital forensics [46], [47], [48]. An overview of these challenges is given in Figure 2. At the end of the experiment, the challenges encountered during the experiments were compared to those from the literature.

A. *Design specification*

The devices under investigation were:

Table 1 Devices under investigation

Brand	**Model**	**Device type**
D-Link	DSP-W115	Smart Plug
TP-link	HS100	Smart Plug
Telldus	TellStick ZNet Lite V2	IoT gateway
Telldus	TZWP-102	Smart Plug
Amazon	HD34Bx	Smart Plug
LG	Nexus 5	Android Smartphone (without SIM-card)

The devices used as part of the investigation were:

- A Dell laptop running Ubuntu.
- A Lenovo laptop running Windows 10.
- A D-link "DWA-140" Wi-Fi USB adapter.
- A Tilgin "HG2332_EU" Wi-Fi router. The router is internet-connected.
- The router runs a 2.4GHz Wi-Fi network that will be referred to as the home Wi-Fi network in the experiments.
- A Velleman VMA-440 USB-to-UART adapter.

B. *Software*

The software installed on the Dell laptop were:

Table 2 Installed Softwares on the Dell laptop

aapt v0.2-27.0.1	wireshark 2.6.10-1~ubuntu18.04.0
ADB 29.0.6-6198805	create_ap git commit 048ca4ecf68a50d56ab934d713a4993bcce49552
apktool 2.3.4-1~18.04	python-leveldb 0~svn68-3build3 (python3 library)
bash 4.4.18-2ubuntu1.2	softScheck/tplink-smartplug (wireshark dissector) git commit dcf978b970356c3edd941583d277612182381f2c
binwalk 2.1.1-16	esptool 2.8 (installed using pip)
ettercap 0.8.2	libleveldb (libleveldb1v5) 1.20-2
SQLite 3.22.0-1ubuntu0.3	nc (netcat-openbsd) 1.187-1ubuntu0.1
hostapd 2.6	javaobj 0.1.0 (python3 library)
kpartx 0.7.4-2ubuntu3	md5sum (coreutils) 8.28-1ubuntu1

nmap 7.60	cmp (coreutils) 8.28-1ubuntu1
python 3.6.7-1~18.04	dd (coreutils) 8.28-1ubuntu1
screen 4.06.02	tshark 2.6.10-1~ubuntu18.04.0
ubuntu 18.04	xxd 2:8.0.1453-1ubuntu1.3

The software installed on the Smartphone were:

- Android 6.0.1 “Marshmallow”.
- Mydlink v1.11.0
- Kasa v2.21.0.924
- Telldus live v3.12.21
- Alexa v2.2.329454.0
- Supersu v2.82-SR5 (app version)
- F-droid 1.7.1.
- Busybox 1.30.1 (app version)
- Mozilla Firefox v68.8.0

The software installed on the Windows laptop were:

- chromecacheview 2.15
- cf-auto-root cfar_lge_nexus-5_hammerhead_m4b30z_t1. 2016-11-03 M4B30Z (3437181).

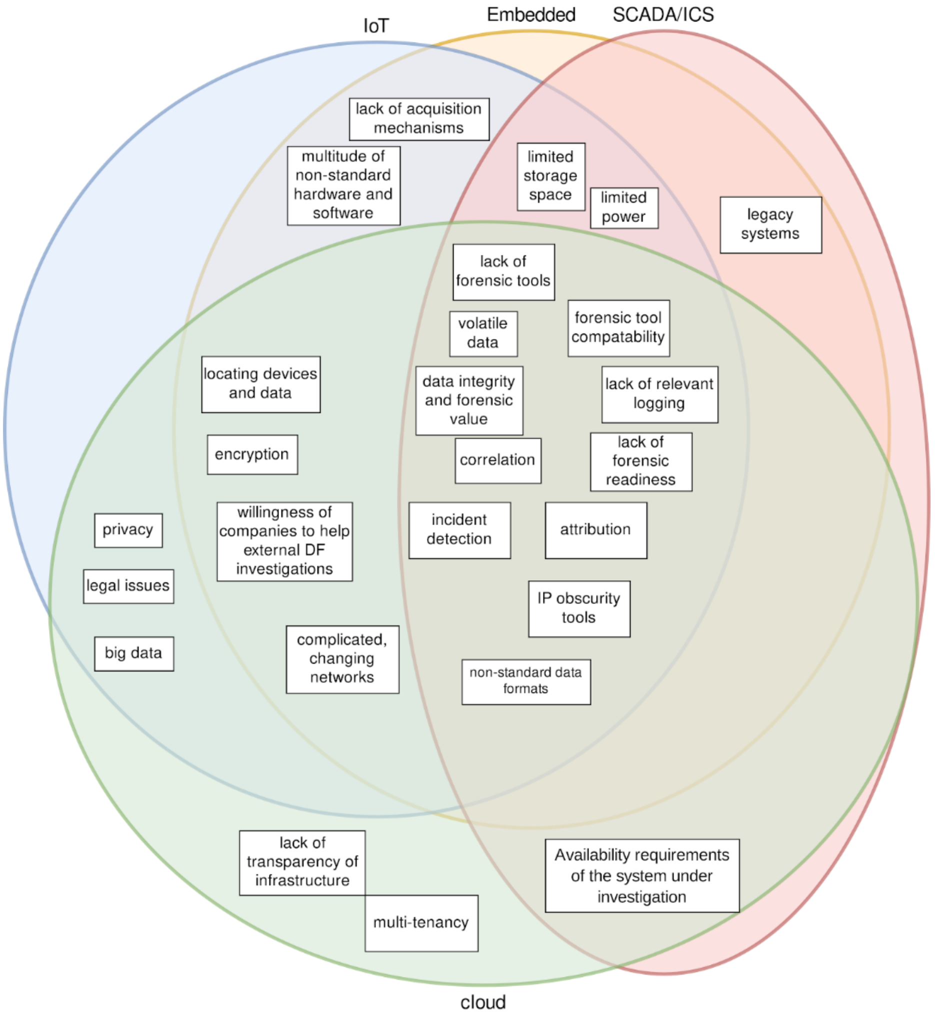
IoT
Embedded
SCADA/ICS
lack of acquisition mechanisms
multitude of non-standard hardware and software
limited storage space
limited power
legacy systems
lack of forensic tools
volatile data
forensic tool compatability
locating devices and data
data integrity and forensic value
lack of relevant logging
encryption
correlation
lack of forensic readiness
privacy
willingness of companies to help external DF investigations
incident detection
attribution
legal issues
IP obscurity tools
big data
complicated, changing networks
non-standard data formats
lack of transparency of infrastructure
multi-tenancy
Availability requirements of the system under investigation
cloud

Figure 2 Overview of Digital Forensic challenges

C. Activities

The activities include e.g., creating an account, logging in, configuring the corresponding IoT device, and remote controlling the IoT device. The specific activities varied slightly between apps depending on the model of the IoT device. All Smart Plugs could be turned “ON” and “OFF” remotely and most Smart Plugs supported a “scheduling” feature of some kind.

In addition to the activities, the IoT devices were also scanned for open ports using the “Nmap” tool. During the activities, the network traffic was recorded using the “Wireshark” tool. Depending on the IoT device, this required various network setups.

D. Network recording setups

To reliably capture network traffic, the traffic must pass through the machine running Wireshark -- in this case the Ubuntu laptop. In most cases, the D-Link “DWA-140” Wi-Fi USB adapter was used to turn the laptop into a MITM router. Figures 3-5 illustrate some of the necessary setups.

Most of the IoT devices have a “configuration phase” where the IoT device creates a small Wi-Fi network, through which the app can configure the device. The configuration involves giving the IoT device the login credentials to an internet-connected Wi-Fi network. Figures 4 and 5 show two different capturing setups for the configuration phase. Some apps would not work with the setup in Figure 4 (using the Wi-Fi USB adapter) and it was necessary to use the “Ettercap” tool (Figure 5). Ettercap was used to perform a spoofing attack, where the Smart Plug and Smartphone are tricked into routing IPv4-based traffic through the laptop.

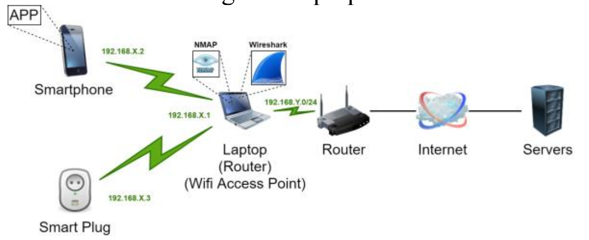

Figure 3 MITM network recording setup with internet access

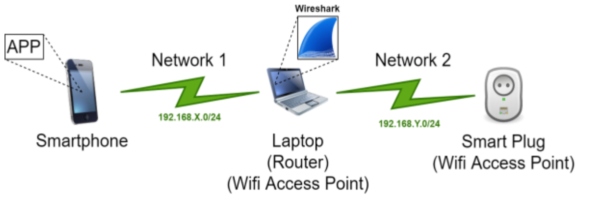

Figure 4 MITM network recording setup during the configuration phase.

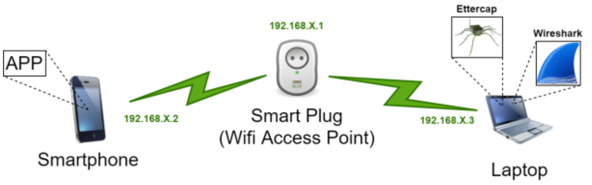

Figure 5 MITM network recording setup using the “Ettercap” tool, during the configuration phase.

E. Acquisition and Analysis

The Smartphone acquisition process consisted of using the “ADB” tool and a USB-cable to copy the special file “/dev/block/mmcblk0” from the device as it is running [49], [50]. To do so the Smartphone had to be “rooted” (essentially the administration privileges on the Smartphone were unlocked) before the experiment. The rooting process typically involves factory resetting the Smartphone, which is why the rooting had to be done before the “activities” phase, or else the evidence would have been deleted. An alternative method would consist of finding a privilege escalation exploit that would circumvent the factory reset but this was deemed to be beyond the scope of this work.

The analysis of the Smartphone forensic copy consisted of mounting the file system partitions of interest (primarily “/data”) using the “kpartx” tool. Each app has a corresponding directory in “/data/data” (e.g., “/data/data/com.dlink.mydlinkunified”), where app-specific files are stored. For each app, each file in the corresponding directory was analyzed using the appropriate tools. For example, each app has a “databases”-directory where SQLite-databases are stored. These files were analyzed using the “SQLite” tool (or using the python sqlite module). All apps also have a google chrome-based HTTP cache which can be analyzed using the “chromecacheview” tool. When necessary, python scripts were written to further analyze the data obtained. For example, some SQLite databases contained JSON-snippets. The JSON data was decoded recursively using Python. Python was also used to interpret timestamps.

Only limited evidence of the activities was found on the Smartphone. The evidence recovered from the Smartphone is summarized in Table 3. Here “N/A” (meaning: Not Applicable) means that the activity was not performed for that app due to the app lacking the necessary feature. The relevant evidence was primarily stored in SQLite-databases. Some of the challenges encountered included the apps not storing logs of certain events (or at least not storing the logs on the Smartphone). Encryption was an issue when analyzing both files and network traffic, especially for the Alexa app. It was possible to access the encryption keys in “/misc/keystore”, but no attempt was made in this work to decrypt the data. The decryption would require partially reverse engineering the apps and this was deemed out of scope. Only the D-Link app and the Telldus app did not use the Android keystore. The Kasa app primarily encrypted individual pieces of information (for example GPS coordinates, IP addresses and passwords) while the Amazon app encrypted full files (such as “databases/comms.db”).

The port scanning results are shown in Table 4. The table also includes observations based on the recorded network traffic. Most of the traffic was encrypted using Transport Layer Security (TLS).

Port 80/TCP (HTTP) on the D-Link Smart Plug can be used to gain basic information about the Smart Plug (such as model and firmware version). The username is “Admin”, and the password is printed on the back the Smart Plug (the “P/N” field) [51]. Port 8080/TCP (TLS) is used to initially configure the Smart Plug and is also used to command the Smart Plug from the Local Area Network (LAN).

Port 30303/UDP is used to discover the Telldus gateway on the LAN and port 80/TCP (HTTP) can be used to command the gateway from the LAN. The telldus gateway also has an SSH backdoor (22/TCP) with unknown login credentials.

Port 9999/UDP is used to discover the TP-link Smart Plug on the LAN. Both port 9999/TCP and 9999/UDP are used to command the Smart Plug from the LAN. The protocol is also used to initially configure the Smart Plug. The “tplink-Smart Plug” wireshark dissector tool by softScheck can be used to deobfuscate the protocol (which is not encrypted). This can be used to discover the time the Smart Plug is activated/deactivated from the LAN.

Table 3 Smartphone-based evidence of activities

	Timestamp recovered in app files			
Activity	**Mydlink**	**Kasa**	**Telldus live**	**Alexa**
Register an account.	No.	**Yes. (databases/iot.1.db [sqlite])**	No.	No.
Log in to the account.	No.	No.	No.	No.
Configure the Smart Plug.	**Yes, the time the QR-code was scanned. (databases/barcode_scan_history.db [sqlite])**	No.	No.	No.
Activate and deactivate the Smart Plug a few times.	No.	No.	**Yes, each event is logged. (databases/tellduslocalstorage.db [sqlite])**	No.
Create a schedule.	No.	No.	No.	No.
Configure two device groups.	No.	**Yes. (databases/iot.1.db [sqlite])**	N/A	No.
Activate (or deactivate) the two “device groups” alternatively a few times.	**Yes, the last use of a “one-tap” is stored. (shared_prefs/${USER_EMAIL}.xml [text/python])**	No.	N/A	No.
Schedule “away-mode”	N/A	No.	N/A	N/A
Configure the Telldus gateway	N/A	N/A	No.	N/A
Use the telldus live “check local control” feature.	N/A	N/A	No.	N/A

Table 4 Port information

D-Link	80/TCP, 8080/tcp[6], 8081/TCP[2 6], 67/UDP[3 4]
TP-Link	9999/TCP, 9999/UDP[1]
Telldus gateway	22/TCP[2], 80/TCP, 5353/UDP[2 4], 30303/UDP, 42314/UDP[2 4], 60710/UDP[2 4]
Amazon plug	20384/TCP[2 5], 30029/TCP[2 5], 1300/UDP[2 4], 20219/UDP[2 4], 20328/UDP[2 4], 20823/UDP[2 4], 23956/UDP[2 4], 27773/UDP[2 4], 41837/UDP[2 4], 42049/UDP[2 4], 43818/UDP[2 4], 47278/UDP[2 4], 54117/UDP[2 4], 54777/UDP[2 4]

Notes: 1: Not reported by Nmap. 2: Detected by Nmap but not observed in other recorded traffic. 3: Only during the configuration phase. 4: Reported as “open|filtered” as opposed to “open”. 5: Reported as “filtered” as opposed to “open|filtered”. 6. TLS with self-signed certificate.

The primary challenges encountered in the experiment were:

- The Android rooting-process complicates the acquisition. In this case, the challenge was circumvented by rooting before the experiment -- which is not practical outside the experiment.
- Many data formats were encountered, and many separate tools were necessary to decode the data. Some formats were harder to decode and interpret than others, such as: java serialization data with obfuscated class names, Google Protobuf-based files and sqlite journals. Due to the recursive nature of some files, it was necessary to glue multiple tools together using Python and shell script. The tools for some data formats (such as sqlite journals) were not freely and readily available. Other tools were not portable, such as chromecacheview.
- Encrypted files and encrypted network traffic prevented further analysis. One way to circumvent this for the experiment is to inject code into the apps before the experiment to log the data before encryption or after decryption. However, this is not practical outside the experiment.
- Interpreting the data is not straightforward. The code that produced the data must be analyzed. On Android, the compiled code is often obfuscated (for example by minimizing class names), which makes reverse engineering a challenge. Alternatively, an interpretation can be obtained empirically by repeating the experiment. However, without control over the cloud system the original experimental environment can not be fully reproduced.
- If the activities were logged at all, they may have been logged to a cloud service rather than to the device itself.

IV. Conclusion

In this paper, we have reviewed related work around Smart Things in general, especially focusing on Smart Plugs. We compared the challenges faced in conducting forensic investigations with other fields and have provided a summary. We explored five major Smart Plugs to conduct forensic analysis, and as compared to other devices we concluded those provided less forensic evidence. We experimented with the different network setups for capturing forensic data. In the future, we intend to provide a detailed analysis of such arrangements and how each device generates forensic evidence.

V. Acknowledgment

This work has received support from the Swedish Civil Contingencies Agency (MSB) through the research center Resilient Information and Control Systems (RICS).

References

[1] L. Knud, “Why it is called Internet of Things: Definition, history, disambiguation.” https://iot-analytics.com/internet-of-things-definition/ (accessed Oct. 19, 2020).

[2] W. Goddard, “History of IoT: What It Is, How It Works, Where It’s Come From, and Where It’s Going,” *ITChronicles*, Jun. 17, 2019. https://itchronicles.com/iot/history-of-iot-what-it-is-how-it-works-where-its-come-from-and-where-its-going/ (accessed Aug. 19, 2020).

[3] Krishnakumar, “Top IoT devices 2019 You Should Know!,” *Eduonix Blog*, Jul. 31, 2019. https://blog.eduonix.com/internet-of-things/internet-of-things-and-iot-top-devices-2019-know/ (accessed Aug. 20, 2020).

[4] “‘Internet of Things’ cyber risks tackled during INTERPOL Digital Security Challenge.” https://www.interpol.int/ar/1/1/2018/Internet-of-Things-cyber-risks-tackled-during-INTERPOL-Digital-Security-Challenge (accessed Aug. 21, 2020).

[5] M. Antonakakis et al. , “Understanding the Mirai Botnet,” p. 19.

[6] L. O’Donnell, “Airbnb Superhost Secretly Recorded Guests with Hidden Bedroom Camera.” https://threatpost.com/airbnb-hidden-camera-bedroom/144508/ (accessed Oct. 19, 2020).

[7] L. O’Donnell, “Thousands of IoT Devices Bricked By Silex Malware.” https://threatpost.com/thousands-of-iot-devices-bricked-by-silex-malware/146065/ (accessed Aug. 22, 2020).

[8] B. Casey, “IoT Crimes,” Cybercrime Magazine, Feb. 28, 2018. https://cybersecurityventures.com/internet-of-things-hacks/ (accessed Aug. 21, 2020).

[9] A. Shalaginov, I. Kotsiuba, and A. Iqbal, “Cybercrime Investigations in the Era of Smart Applications: Way Forward Through Big Data,” in *2019 IEEE International Conference on Big Data (Big Data)*, Los Angeles, CA, USA, Dec. 2019, pp. 4309–4314, doi: 10.1109/BigData47090.2019.9006596.

[10] Z. Zorz, “Hacking smart plugs to enter business networks,” Help Net Security, Aug. 23, 2018. https://www.helpnetsecurity.com/2018/08/23/hacking-smart-plugs/ (accessed Aug. 25, 2020).

[11] Z. Ling, J. Luo, Y. Xu, C. Gao, K. Wu, and X. Fu, “Security Vulnerabilities of Internet of Things: A Case Study of the Smart Plug System,” *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1899–1909, Dec. 2017, doi: 10.1109/JIOT.2017.2707465.

[12] W. Zhou *et al.* , “Discovering and Understanding the Security Hazards in the Interactions between IoT Devices, Mobile Apps, and Clouds on Smart Home Platforms,” 2019, pp. 1133–1150, Accessed: Aug. 15, 2020. [Online]. Available: https://www.usenix.org/conference/usenixsecurity19/presentation/zhou.

[13] A. Wang and S. Nirjon, “A False Sense of Home Security — Exposing the Vulnerability in Away Mode of Smart Plugs,” in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Mar. 2019, pp. 316–321, doi: 10.1109/PERCOMW.2019.8730664.

[14] A. Halterman, “Storming the Kasa? Security analysis of TP-Link Kasa smart home devices,” *Creat. Compon.*, Jan. 2019, [Online]. Available: https://lib.dr.iastate.edu/creativecomponents/392.

[15] M. Bettayeb, O. A. Waraga, M. A. Talib, Q. Nasir, and O. Einea, “IoT Testbed Security: Smart Socket and Smart Thermostat,” in *2019 IEEE Conference on Application, Information and Network Security (AINS)*, Nov. 2019, pp. 18–23, doi: 10.1109/AINS47559.2019.8968694.

[16] S. Brotsis *et al.* , “Blockchain Solutions for Forensic Evidence Preservation in IoT Environments,” in *2019 IEEE Conference on Network Softwarization (NetSoft)*, Jun. 2019, pp. 110–114, doi: 10.1109/NETSOFT.2019.8806675.

[17] E. Al-Masri, Y. Bai, and J. Li, “A Fog-Based Digital Forensics Investigation Framework for IoT Systems,” in *2018 IEEE International Conference on Smart Cloud (SmartCloud)*, Sep. 2018, pp. 196–201, doi: 10.1109/SmartCloud.2018.00040.

[18] R. Alharbi and W. H. Allen, “Collection and Analysis of Digital Forensic Data from Devices in the Internet of Things,” in *2019 SoutheastCon*, Apr. 2019, pp. 1–6, doi: 10.1109/SoutheastCon42311.2019.9020349.

[19] A. Goudbeek, K.-K. R. Choo, and N.-A. Le-Khac, “A Forensic Investigation Framework for Smart Home Environment,” in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, New York, NY, USA, Aug. 2018, pp. 1446–1451, doi: 10.1109/TrustCom/BigDataSE.2018.00201.

[20] A. Hariri, N. Giannelos, and B. Arief, “Selective Forwarding Attack on IoT Home Security Kits,” in *Computer Security*, Cham, 2020, pp. 360–373, doi: 10.1007/978-3-030-42048-2_23.

[21] A. Awasthi, H. O. L. Read, K. Xynos, and I. Sutherland, “Welcome pwn: Almond smart home hub forensics,” *Digit. Investig.*, vol. 26, pp. S38–S46, Jul. 2018, doi: 10.1016/j.diin.2018.04.014.

[22] S. Kim, M. Park, S. Lee, and J. Kim, “Smart Home Forensics—Data Analysis of IoT Devices,” *Electronics*, vol. 9, no. 8, Art. no. 8, Aug. 2020, doi: 10.3390/electronics9081215.

[23] Q. Do, B. Martini, and K.-K. R. Choo, “A Data Exfiltration and Remote Exploitation Attack on Consumer 3D Printers,” *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 10, pp. 2174–2186, Oct. 2016, doi: 10.1109/TIFS.2016.2578285.

[24] I. Clinton, L. Cook, and S. Banik, A survey of various methods for analyzing the Amazon Echo (2016). 2018.

[25] S. Vargas, “Privacy Issues and Digital Forensic Analysis For ‘Smart Personal Assistants,’” Thesis, Auckland University of Technology, 2020.

[26] S. Tristan, S. Sharma, and R. Gonzalez, “Alexa/Google Home Forensics,” in *Digital Forensic Education: An Experiential Learning Approach*, X. Zhang and K.-K. R. Choo, Eds. Cham: Springer International Publishing, 2020, pp. 101–121.

[27] C. Krueger and S. McKeown, “Using Amazon Alexa APIs as a Source of Digital Evidence,” in *2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, Jun. 2020, pp. 1–8, doi: 10.1109/CyberSecurity49315.2020.9138849.

[28] A. Nieto, “Becoming JUDAS: Correlating Users and Devices During a Digital Investigation,” *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 3325–3334, 2020, doi: 10.1109/TIFS.2020.2988602.

[29] İ. Yıldırım, E. Bostancı, and M. S. Güzel, “Forensic Analysis with Anti-Forensic Case Studies on Amazon Alexa and Google Assistant Build-In Smart Home Speakers,” in *2019 4th International Conference on Computer Science and Engineering (UBMK)*, Sep. 2019, pp. 1–3, doi: 10.1109/UBMK.2019.8907007.

[30] H. Chi, T. Aderibigbe, and B. C. Granville, “A Framework for IoT Data Acquisition and Forensics Analysis,” in *2018 IEEE International Conference on Big Data (Big Data)*, Dec. 2018, pp. 5142–5146, doi: 10.1109/BigData.2018.8622019.

[31] L. Gomes, F. Sousa, and Z. Vale, “An Intelligent Smart Plug with Shared Knowledge Capabilities,” *Sensors*, vol. 18, no. 11, Art. no. 11, Nov. 2018, doi: 10.3390/s18113961.

[32] S. Altaf and A. Anand, “Data acquisition and control using Arduino-Android platform: Smart plug,” in 2013 International Conference on Energy Efficient Technologies for Sustainability, Apr. 2013, pp. 241–244, doi: 10.1109/ICEETS.2013.6533389.

[33] I. B. Dhaou, “Smart Plug Design for Demand Side Management Program,” in *2019 4th International Conference on Power Electronics and their Applications (ICPEA)*, Sep. 2019, pp. 1–5, doi: 10.1109/ICPEA1.2019.8911130.

[34] P. Mtshali and F. Khubia, “A Smart Home Energy Management System using Smart Plugs,” in *2019 Conference on Information Communications Technology and Society (ICTAS)*, Mar. 2019, pp. 1–5, doi: 10.1109/ICTAS.2019.8703522.

[35] S. Heo, W.-K. Park, and I. Lee, “Energy management based on communication of smart plugs and inverter for smart home systems,” in *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, Oct. 2017, pp. 810–812, doi: 10.1109/ICTC.2017.8190788.

[36] H. Morsali *et al.* , “Smart plugs for building energy management systems,” in *Iranian Conference on Smart Grids*, May 2012, pp. 1–5.

[37] M. Aftab and C.-K. Chau, “Smart Power Plugs for Efficient Online Classification and Tracking of Appliance Behavior,” in *Proceedings of the 8th Asia-Pacific Workshop on Systems*, New York, NY, USA, Sep. 2017, pp. 1–7, doi: 10.1145/3124680.3124735.

[38] M. S. Ahmed, A. Mohamed, R. Z. Homod, H. Shareef, A. H. Sabry, and K. Bin Khalid, “Smart plug prototype for monitoring electrical appliances in Home Energy Management System,” in *2015 IEEE Student Conference on Research and Development (SCOReD)*, Dec. 2015, pp. 32–36, doi: 10.1109/SCORED.2015.7449348.

[39] I. Horvat, N. Lukac, R. Pavlovic, and D. Starcevic, “Smart plug solution based on bluetooth low energy,” in *2015 IEEE 5th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, Sep. 2015, pp. 435–437, doi: 10.1109/ICCE-Berlin.2015.7391301.

[40] A. Ridi, C. Gisler, and J. Hennebert, “Processing smart plug signals using machine learning,” in *2015 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, Mar. 2015, pp. 75–80, doi: 10.1109/WCNCW.2015.7122532.

[41] N. K. Suryadevara and G. R. Biswal, “Smart Plugs: Paradigms and Applications in the Smart City-and-Smart Grid,” *Energies*, vol. 12, no. 10, Art. no. 10, Jan. 2019, doi: 10.3390/en12101957.

[42] M. Ghazal, M. Akmal, S. Iyanna, and K. Ghoudi, “Smart plugs: Perceived usefulness and satisfaction: Evidence from United Arab Emirates,” *Renew. Sustain. Energy Rev.*, vol. 55, pp. 1248–1259, Mar. 2016, doi: 10.1016/j.rser.2015.07.096.

[43] R. Rowlingson, “A Ten Step Process for Forensic Readiness,” vol. 2, no. 3, p. 28, 2004.

[44] C. Gurkok, “Chapter 10 - Cyber Forensics and Incident Response,” in Managing Information Security (Second Edition), J. R. Vacca, Ed. Boston: Syngress, 2014, pp. 275–311.

[45] A. Shalaginov, A. Iqbal, and J. Olegård, “IoT Digital Forensics Readiness in the Edge: A Roadmap for Acquiring Digital Evidences from Intelligent Smart Applications,” in Edge Computing – EDGE 2020, Cham, 2020, pp. 1–17, doi: 10.1007/978-3-030-59824-2_1.

[46] H. Alobaidli, Q. Nasir, A. Iqbal, and M. Guimaraes, “Challenges of Cloud Log Forensics,” in Proceedings of the SouthEast Conference, New York, NY, USA, Apr. 2017, pp. 227–230, doi: 10.1145/3077286.3077302..

[47] A. Iqbal, M. Ekstedt, and H. Alobaidli, “Digital Forensic Readiness in Critical Infrastructures: A Case of Substation Automation in the Power Sector,” in Digital Forensics and Cyber Crime, Cham, 2018, pp. 117–129, doi: 10.1007/978-3-319-73697-6_9.

[48] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis, “A Survey on the Internet of Things (IoT) Forensics: Challenges, Approaches, and Open Issues,” IEEE Communications Surveys Tutorials, vol. 22, no. 2, pp. 1191–1221, Secondquarter 2020, doi: 10.1109/COMST.2019.2962586.

[49] B. Iqbal, A. Iqbal, M. Guimaraes, K. Khan, and H. A. Obaidli, “Amazon Kindle Fire from a Digital Forensics Perspective,” in 2012 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Oct. 2012, pp. 323–329, doi: 10.1109/CyberC.2012.61.

[50] A. Iqbal, H. Alobaidli, A. Marrington, and I. Baggili, “Amazon Kindle Fire HD Forensics,” in Digital Forensics and Cyber Crime, Cham, 2014, pp. 39–50, doi: 10.1007/978-3-319-14289-0_4.

[51] L. Unnebäck, “Information about the DSP-W115 Smart Plug,” Gist. https://gist.github.com/LinusU/75e0bcbb992b532d6e28b750f4d1597c (accessed Oct. 20, 2020).