SIDDANGOUDA HOSAMANI. International Journal of Engineering Research and Applications
www.ijera.com
ISSN: 2248-9622, Vol. 13, Issue 7, June 2015, pp 243-251

RESEARCH ARTICLE OPEN ACCESS

ZLeaks: Zigbee-based Smart Home Passive Inference Attacks

SIDDANGOUDA HOSAMANI
SELECTION GRADE LECTURER
sidduhosamani@yahoo.com
ELECTRONICS & COMMUNICATION ENGG. DEPARTMENT
GOVERNMENT POLYTECHNIC, BELGAUM - 590001

ABSTRACT
Zigbee is a widely used wireless Internet of Things (IoT) protocol because of its low power consumption. We investigate the Zigbee protocol's security features. In this paper, we introduce ZLeaks, a tool that can detect and categorize leaks of information about home devices and events. zibbee transmission 2) using the device's periodic reporting pattern and interval 3) by inferring a single application layer (APL) command from the event's traffic. An adversary may use this information to deduce user patterns or ascertain whether the smart home is susceptible to intrusion. We tested ZLeaks in three environments (managed RF shield, live smart-home IoT lab, and third-party Zigbee captures) with a total of 19 distinct Zigbee devices spanning many categories and 5 prominent smart hubs.

Using a command inference approach, we were able to i) correctly identify 83.6% of unknown events and devices (without a-priori device signatures) in a public capture, ii) automatically extract a device's reporting signatures, iii) correctly identify 99.8% of known devices using the reporting signatures, and iv) correctly identify 91.2% of APL commands in the public capture. In a nutshell, we emphasize the tension that exists between privacy protections and the development of a low-power, low-cost wireless network. We have also made the research community friendly ZLeaks tool public.

Keywords*: Zigbee, IoT, Device identification, Passive inference*

I. INTRODUCTION

While smart home gadgets (light bulbs, outlets, sensors, etc.) provide convenient wireless access to home management functions, they also expose consumers to serious privacy concerns. Intercepting the IP traffic of a smart home has been shown in previous research to provide information on the house's gadgets [3], events [5], and users' routines [7]. Because the attacker has to discover a weakness in order to collect the user's IP network traffic (say, by acquiring root access to the home router), these types of assaults are difficult to carry out in practice. However, there is a simple privacy violation attack, which involves only sniffing the wireless protocol (e.g., Zigbee) broadcasts that are mistakenly broadcasted to up to hundreds of feet by Internet of Things (IoT) devices. Researchers have recently shown that an attacker may still identify events using a-priori device signatures [7,8] and deduce a few encrypted Zigbee (Network layer) instructions by abusing the payload lengths [9]. This is even though IoT traffic is encrypted to avoid eavesdropping.

In this paper, we examine the security features of Zigbee [10], a widely used wireless protocol for Internet of Things devices like the Amazon Echo Plus, Samsung SmartThings, and Philips Hue, to name a few. Zigbee continues to remain the preferred option of device makers, with over 500 new Zigbee-certified devices being released in 2014 alone, and approximately four billion Zigbee chipsets estimated to be sold by 2015 [11].

Identification of Hardware and Events through Inferred APL Command*:* In this paper, we show that every device's event traffic contains at least one APL command (such as Door Lock/Unlock) that uniquely identifies the triggered event (i.e., lock/unlock) and the functioning device type (i.e., door lock). All such APL instructions are inadvertently disclosed in the Zigbee Cluster Library (ZCL) standard [12]. To identify a given Zigbee device, we combine the manufacturer's

identification retrieved from the Organizationally Unique Identifier (OUI) of the device's MAC address with an inference of a single functionality-specific APL command in the encrypted event data. This method can recognize both known and unknown events and gadgets, unlike previous efforts [7].

To far, no research has addressed the extraordinarily difficult task of inferring functionality-specific APL instructions. This is because a hundred different generic APL commands share an awful lot of information with functionality-specific ones. The limited number of manufacturer-configurable APL commands prevents us from utilizing our previous NWK command inference technique [9], which relied only on payload length, packet direction, and radius (hops) to infer APL instructions. By consulting the recommended frame structure [12], we catalogue all APL commands whose payload lengths coincide with those of functionality-specific APL commands and their response instructions (if any), such as a door unlock request and its potential responses. Inference rules for each functionality-specific APL command are built using differences in traffic metadata and the device's logical nature (electricity-powered or battery-powered).

Device Detection via Analysis of Report Frequency: Zigbee gadgets will send data to the central device at regular intervals. We use reporting frequencies and trends to identify individual devices. The Zigbee network has little user traffic, therefore this method is ideal for locating a known vulnerable device that has not been fixed. This method can identify devices even when no event is triggered, unlike previous research [7,8] that only analyse Zigbee data produced due to event occurrence. Because each device's power consumption is unique depending on its communication pattern and hardware, changing the reporting period to meet the minimum 2-year battery life requirement for certification is not a simple task [13].

Using the ZLeaks tool to automatically identify events and devices: To streamline the identification processes, we programmed them into ZLeaks [14], a complete privacy analysis tool for the Zigbee protocol. ZLeaks uses the Zigbee data as input to passively identify smart home events and devices. It also can automatically extract reporting signatures from devices.

We conducted an experimental evaluation of ZLeaks on the largest device set used in privacy analysis of the Zigbee protocol, which consisted of 27 commercial off-the-shelf Zigbee devices, including 19 that had not been used in previous studies. These devices included five popular smart hubs (SmartThings, Amazon Echo Plus, Philips Hue, OSRAM Lightify, and Sengled). Both an RF shielded environment and a real-world smart home "Mon (IoT)r Lab" [15] with numerous IoT and non-IoT networks were used for the studies. In addition, we checked our results against capture files posted by other parties on the Wireshark [16] and Crawdad [17] forums. According to our findings, ZLeaks was able to correctly identify events and devices with 83.6% accuracy using inferred APL instructions, and devices with 99.8% accuracy utilizing reporting patterns. Moreover, utilising our command inference rules, we successfully inferred functionality-specific APL instructions from a publicly available Zigbee capture with an accuracy of 91.2%.

II. BACKGROUND AND MOTIVATION

2.1: Zigbee Overview : For battery-operated applications in smart ecosystems like smart homes and industries, Zigbee is a popular low-cost, low-power, wireless protocol. Zigbee uses the physical (PHY) and medium access control (MAC) layers specified by the IEEE standard for low data-rate wireless personal area networking (PAN). The standard data rate for commercial Zigbee devices is 250 kbps, and they operate in the 2.4 GHz spectrum (which is split into 16 channels, spaced 5 MHz apart). Some Zigbee gadgets can use the 784 MHz, 868 MHz, and 915 MHz bands that don't need a licence to use.

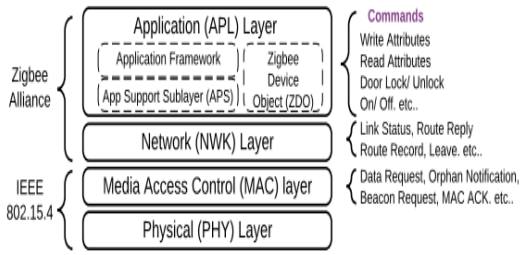

Figure No. 1: Zigbee's Protocol Stack comprises of PHY, MAC, NWK and APL layers.

2.2: Network Architecture: Zigbee is compatible with both centralized and decentralized network topologies. In centralized networks, the Zigbee coordinator (ZC), the Zigbee router (ZR), and the Zigbee end-device (ZED) are all considered separate logical devices. ZEDs are suitable for battery-powered devices (e.g., sensors, door locks) since they do not route traffic and may sleep to preserve energy. As intermediaries, ZRs direct data between nodes and hold on to messages destined for ZEDs until they are needed. One ZC is assigned to each Zigbee network, and it is this ZC's job to create the network, distribute network IDs, and assign logical addresses. To verify new nodes and hand out keys, ZC also serves as a central authority. ZRs and ZCs are powered devices (light bulbs, smart hubs, etc.) that remain awake for the duration of the network. In addition, Zigbee allows for connections to be made in a tree, mesh, or star topology. MAC address randomization is not supported by Zigbee. During device pairing, the ZC takes the 64-bit MAC (extended) address supplied to each Zigbee node by the manufacturer and converts it to a 16-bit network (logical) address. The extended address is used for authentication, whereas the logical address is utilized for routing.

2.3: Zigbee Protocol Stack: The Zigbee standard [10] specifies both the Network and Application layers' capabilities. The Network layer oversees everything from creating and managing networks to allocating IP addresses. The 12 NWK commands include things like "Link Status," "Route Record," "Route Reply," and more.

The Application Support (APS) sublayer, the Zigbee Device Object (ZDO), and the Application Framework are all parts of Zigbee's Application layer. ZDO implements the device in one of the three logical roles (ZC, ZR, or ZED), while the APS sublayer is responsible for keeping track of binding tables and address mappings. For end-manufacturers, the application framework provides pre-defined profiles (like home automation and health care) and functional areas called clusters (like lighting and security). In general, APL commands are either domain-specific (like Read Attributes) or generic (like Report Attributes, etc.).

III. SYSTEM AND THREAT MODELS

In this scenario, a smart hub (ZC) is coupled with several widely used Zigbee devices (ZRs and ZEDs), as shown in Figure 2. The IP gateway is linked to the hub so that the cloud and the user's smart app may get status updates from the devices. The residents of a smart house go about their daily lives, with the added convenience of being able to manage their home's technology from anywhere via a mobile app. We presume a passive attacker is listening in on Zigbee communications from within wireless range of the target network using a wireless Zigbee sniffer. We utilize a TI CC2531 Zigbee sniffer [8] with an omnidirectional antenna to pick up signals from Zigbee nodes up to 20 meters away. A Zigbee sniffer may be implanted nearby, allowing an attacker to remotely monitor communications without needing access to the smart app or even being present in the smart house itself.

Without knowing the network or link keys, the QR code of the device, or specific events like device pairing or rejoining, an attacker can passively identify the events and devices by analyzing the captured Zigbee traffic and applying either command inference or periodic reporting patterns. We anticipate a completely functional Zigbee network with all subject devices (such as door locks, light bulbs, electrical outlets, and other sensors) already set up and commissioned. A simple familiarity with the Zigbee standard is all that's needed for an assault. There's no need to gather individual device event signatures. Attackers only require the devices' reporting signatures if they need to single out a particular device in the target residence when no one is really using it.

Challenges: Zigbee employs the AES-128 algorithm, which has been shown to inhibit eavesdropping because of its confusion and dispersion qualities. An adversary may infer NWK frames by employing the already-existing technique for inferring NWK commands [9], which considers the payload size, the radius, and the logical device type that is being actively determined. Unfortunately, the information about events and devices is hidden away in APL instructions where the distance between them is negligible. Furthermore, there are over a hundred APL commands, the majority of which have no fixed payload lengths [10] like the 12 NWK commands.

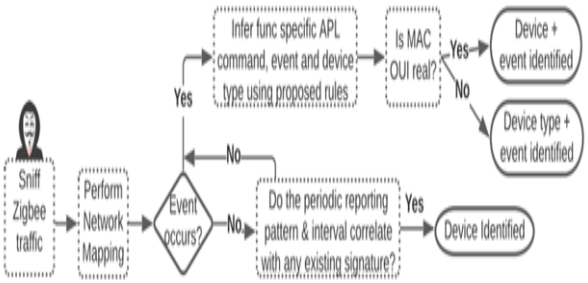

Figure No. 2: When an event happens, the inferred APL command is combined with the device's MAC address to reveal both the device and the occurrence. In the absence of an event, the device may be determined by correlating its periodic signature. If it doesn't work, you'll need to anticipate something.

This may be altered by the manufacturer (like Report Attributes or Read Attributes). Therefore, with each possible payload length, there are several overlaps. Due to these limitations, the currently used method [9] for inferring APL instructions is inadequate.

The incremental frame sequence numbers roll back after 256 in the unencrypted IEEE 802.15.4 frames in Zigbee traffic, making it extremely difficult to trace the communicating nodes. This is because the frequently exchanged IEEE 802.15.4 ACK does not mention the network or MAC address for the source or destination. In addition, current research projects use a-priori event signatures to identify events [7,8]. Users occur less often in practice, especially at night. While in this mode, no sensitive data is sent between the devices and the hub, just periodic reports are exchanged. Therefore, there are still open difficulties for the attacker, such as recognizing a device without event signatures or when there are no events at all.

3.1: Identify the Events and Devices: We begin by excluding any APL instructions from the event's communication that were not destined for the intended logical address (say, 0xabcd).

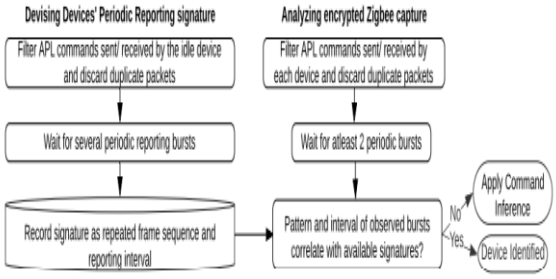

Figure No. 3: Strategy to identify devices using periodic reporting patterns.

and toss out any packets that are duplicates. The functionality-specific command is often the first APL command in an event burst; hence, we also ignore event bursts that do not include any frame with target payload lengths (11- 17 bytes) in the first half of the burst. We check Table 1 to see whether the APL command, event, and device type correspond to the specified payload length. The device is finally identified by combining the manufacturer's identity determined from the MAC OUI (e.g., PhilipL) with other information. Keep in mind that the MAC OUI is more concerned with the actual manufacturer of the device than it is with the manufacturer of the system-on-chip (SOC), such as SiliconL. Basically, without the Network key or event signatures, we may passively recognize unknown occurrences and devices in the target functional domains (light bulbs, wall outlets, door locks, and sensors).

3.2: Device Detection through Interval Reporting

ZC receives monthly updates from Zigbee devices detailing their health (battery life, firmware changes, etc.). Manufacturers must adjust the periodic reporting frequency and certain frame properties to ensure their products meet the Zigbee certification criterion of a minimum 2-year battery life [13]. We can create distinctive device fingerprints and identify devices even when no event happens (for example, during office hours) because to the variations in reporting patterns and intervals. Reporting bursts, in contrast to event bursts, do not immediately indicate the identity of the device and lack any functionality-specific APL command.

IV. EXPERIMENTAL SETUP AND RESULTS

4.1: Using the ZLeaks Tool to Automate Passive Inference Attacks

To simplify the inference attacks shown in Figure 3, we built a Python command line programme called ZLeaks. ZLeaks analyses a Zigbee PCAP capture to find out what devices and events are present in the network. With a single command, an attacker in range of the target network may install and execute ZLeaks on a laptop or embedded board like a Raspberry Pi. ZLeaks leverages the Pyshark library [15] to parse necessary frame information (such as payload length, logical kinds of nodes, etc.) into a temporary CSV file for analysis, and then executes all APL instructions found in the collected data. Then, ZLeaks use the suggested APL inference rules (Section 3.3) or the available reporting signatures (Section 3.4) to try to identify events and devices. Keep in mind that the attacker may use ZLeaks Signature Extractor to automatically extract the reporting signatures of a sleeping Zigbee device.

4.2: Experimental Setup

Based on Amazon's popularity and manufacturer variety, we chose a device set of 27 commercial off-the-shelf Zigbee devices (including lamps, locks, outlets, and other sensors). Of the total of 27, 19 were distinct devices, while the remaining 8 were duplicates that were obtained and evaluated from a separate source to maintain reliability in the assessment findings for a given device and model. In addition, we employed 11 distinct devices to develop our inference technique, but we reserved 8 of them as "unknown devices" for the assessment phase. At least one device from each functional area was included. Table 3 and Table 4 list the known and unknown devices, respectively, for convenience. The SmartThings and Amazon Echo Plus hubs were used for the manufacturer-agnostic universal testing, whereas the Philips Hue Bridge 2.1, Sengled Z02-hub, and Lithify Gateway hubs were used for the vendor-specific testing. When it comes to testing the Zigbee protocol, this is the largest collection of Zigbee devices available.

We tested three different environments to see how well ZLeaks detection approaches worked: RF shield; By partnering numerous devices with each hub at once, we were able to i) examine how devices react to event triggers to develop command inference rules, ii) gather data on the device's reporting behavior, and iii) conduct a controlled review of ZLeaks. As shown in Figure 6, ZC was put within the RF shield and linked to the gateway so that he would have constant access to the Internet.

An SMA connection ran from the laptop (outside the shield) to a cheap TI CC2531 wireless Zigbee sniffer [18], which was linked to a standard omnidirectional antenna (within the shield) to intercept the Zigbee transmission.

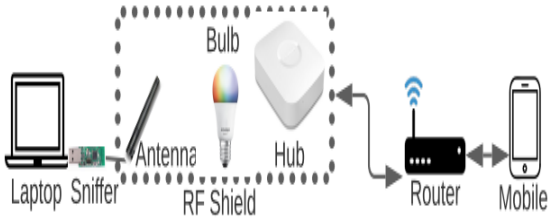

Figure No. 4: The Zigbee Device Analysis Experimental Setup: The antenna of the sniffer (housed inside the RF shield) is linked to the laptop and the TI CC2531 sniffer (housed without the shield) via a SMA connection.

IoT "Living Lab": The "Mon (IoT)r Lab" at Northeastern University [15] is a realistic loud IoT lab with more than a hundred smart devices already linked via several wireless networks, in addition to a number of non-IoT networks.

To demonstrate that ZLeaks is not reliant on an assessment testbed, device set, or unknown devices, we utilized Zigbee captures from the public domain, including those found on the i) Wireshark forum [16] and ii) Prior captures [9] accessible on Crawdad [17]. Using the Network keys that came with the capture files, we double-checked the findings. Neither capture included enough examples of the periodic reporting strategy to analyze, just event bursts.

V. EVALUATION METRICS

ZLeaks was assessed in three different ways: 1) based on inferred APL instructions; 2) based on event and device type retrieved from APL commands; and 3) based on correlated periodic reporting patterns. Parameters 1 and 3 were analyzed using the standard accuracy measures. Since the outcomes of an inferred APL command are

the same for both events and devices, we used the suggested Device Score method to evaluate parameter 2.

Conventional Measurement: The rates of accurate and missing (or out-of-order) observations are referred to as the True Positive Rate (TPR) and the False Negative Rate (FNR), respectively. Because there are no erroneously classified observations (False Positives; FPs) or properly classified ones (True Negatives; TNs; evaluation findings), we define accuracy as the ratio of correctly inferred observations to the total number of observations, and we get this as:

$$TPR\ (recall) = \frac{TP}{TP\ +\ FN}$$

$$FNR\ = \frac{FN}{TP\ +\ FN}$$

$$Accuracy\ = \frac{TP\ +\ TN}{TP\ +\ TN\ +\ FP\ +\ FN}$$

Device Score (or simply Score) is a metric used to quantify how much data can be gleaned from an inferred APL command and device OUI. The Score is a weighted sum of the attributes device type (DT), event type (ET), and manufacturer's identification (M), as shown in Table 2.

$$Score\ =\ M\ +\ DT\ +\ ET$$

Attributes	Score	Example
Manufacturer (M)	0 = SOC OUI	SiliconL, Ember, TexasIns, NordiacSE ..
	1 = Real MAC OUI	PhilipsL, OSRAM, SmartThi, Zhejiang ..
Device Type (DT)	0 = Unidentified	-
	1 = Uncertain	door lock or bulb (different commands)
	1.5 = Indistinct	either outlet or bulb? (same command)
	2 = Identified	Outlet, door lock, motion sensor, bulb ..
Event Type (ET)	0 = Unidentified	-
	1 = Uncertain	lock/unlock or on/off (different commands)
	1.5 = Indistinct	either door lock or unlock? (same commands)
	2 = Identified	motion detected, color change, etc ..

Table 1 – Score Table for Evaluating Command Inference Approach

Score may be visualized as a light switch that, when activated, sends an 11-byte, functionality-specific APL instruction from ZC to ZED. Inferring all characteristics correctly yields a perfect Score of 5, whereas failing to infer any attributes at all results in a Score of 0. Lock/unlock and on/off are the two options shown in Table 1. If the two instructions are identical, then DT and ET are both set to 1 in Table 2. The DT (lights/outlets) and ET (on/off) values are both 1.5 for the on/off command, however the DT (locks) and ET (both) values are both 2 for the Lock/unlock command.

VI. INFERRED APL COMMAND FOR DEVICE AND EVENT DETECTION

6.1: Controlled Evaluation in RF Shield: Inside the RF shield, we randomly created events while pairing all suitable devices with a single hub at once. ZLeaks extrapolated the APL commands and MAC OUIs for each device's capability from the captured data to identify the devices and events that had been triggered. For a given device-event combination (say, color change for Sengled bulb), the inferred APL command and MAC OUI are always the same, hence the Score is always the same for every event prompt no matter the hub. Table 3 only includes one report for each piece of equipment because of this. Unlike binary events (on/off), we may easily infer the occurrence of more nuanced events (colour change, motion detection, etc.). The Philips bulb is an exception to this rule since it interprets on and off as separate instructions. In addition, we were able to distinguish between different sensors using only a single Zone Status command and their unique behavioral consistency criteria (see Table 1). In conclusion, the Score relies on recognizing the APL command and the MAC OUI revealing the actual manufacturer, such as PhilipsL (Philips), SmartThi/ Samjin (SmartThings), Ledvance (OSRAM), Zhejiang (Sengled), Jennic (Aqara), etc. With an average Score of 4.3 out of 5, ZLeaks was able to detect every single device (or an extraction success rate of 86.3%).

6.2: Realistic Evaluation in an IoT Device: Then, we transferred the gadgets, the hubs, and the 8 invisible gadgets to the Internet of Things laboratory. Once again, we had all the devices connect to the same hub at once, and we'd set off random triggers.

Device (Model)	Event	OUI	Command (#)	M	DT	ET	Score
Philips Hue Color Bulb (LCA-003)	Off	PhilipsL	Off with effect	1	2	2	5
	On	PhilipsL	On/off: On	1	2	2	5
	Color change	PhilipsL	Color Control	1	2	2	5
	Dim	PhilipsL	Level Control	1	2	2	5
Sengled Color Bulb (E11-N1EA)	Color change	Zhejiang	Color Control	1	2	2	5
	Dim	Zhejiang	Level Control	1	2	2	5
	On/off	Zhejiang	On/Off	1	1.5	1.5	4
Sengled White Bulb G14	On/Off	Zhejiang	On/Off	1	1.5	1.5	4
Centralite Outlet (Mini)	On/Off	siliconL	On/Off	0	1.5	1.5	3
Sonoff Outlet (S31 Lite)	On/Off	texasIns	On/Off	0	1.5	1.5	3
SMT Outlet (US-2)	On/Off	Smartthi	On/Off	1	1.5	1.5	4
SMT Motion sensor (IM)	Motion	Smartthi	Zone Status (1*)	1	2	2	5
SMT Multisensor (250)	Open/close	samjin	Zone Status (1)	1	2	1.5	4.5
Ecolink Water Sensor	water leak	ember	Zone Status (2)	0	2	2	4
Ecolink Sound Sensor	Sound	ember	Zone Status (3)	0	2	2	4
Yale Door lock (YRD226)	Lock/unlock	ember	Lock/Unlock	0	2	1.5	3.5

Table 2: Experimental Design: Inferred APL Commands for Detecting Devices and Events. SmartThings (SMT), Manufacturer (M), Device Type (DT), Event Type (ET), and Burst Repeat (*) after a few seconds are all defined below.

and ZLeaks was used to analyse the data. Table 3 shows that the Scores of the identified devices remained consistent even in the presence of noise. Table 4 displays experimental data regarding mysterious gadgets. ZLeaks successfully identified previously unknown devices by their unique MAC OUIs and other identifying characteristics, such as the fact that the Sengled bulb's colour was changing. ZLeaks, overall, was able to correctly identify 83.6% of devices and events (with a Score of 4.2 out of 5). We find that the functionality-specific APL command stays the same and may be used to successfully identify any unknown device with a single event trigger, despite devices presenting diverse event signatures across multiple hubs.

6.3 - Device Detection through Interval Reporting: We used a radio frequency (RF) shield to conduct a controlled evaluation in which we linked all known devices with a single hub at once and then left them in the idle state for at least three hours. To adequately assess the two primary features—reproducibility and uniqueness of periodic signatures—36 and 18 reporting patterns, respectively, were generated by devices reporting the characteristics every 5 or 10 minutes, respectively. The outcomes of this experiment are summarized in Table 1, with s, m, and h representing second, minute, and hour reporting periods, respectively. Several devices displayed several reporting patterns (battery, temperature, etc.), whereas a smaller number of devices showed varying numbers of reporting patterns across hubs (SMT vs. Sonoff outlet, for example). In essence, this assisted in determining both the gadget and the intelligent hub from the secret data exchange. With a TPR of 0.998 and a FNR of 0.002, it is clear that the periodic signatures were distinguishable and stable throughout time, with the exception of a single instance in which the Centralite output and SMT Multisensor displayed two out-of-order packets that went unidentified.

VII. DISCUSSION AND RELATED WORK

7.1: Data Leaks and Their Consequences for Security

The security of a smart home depends on the owner's familiarity with the gadgets inside it and the state those devices might be in (for example, door open, light off). A potential intruder may use this data to learn about the household's wealth and the best times to break in. Moreover, an adversary may utilize the Common Vulnerabilities and Exposures (CVE) database [13] to locate and exploit vulnerabilities in the unpatched devices that have been detected. The compromised equipment may be used as a weapon in DDoS assaults, IoT botnet creation, or malware dissemination. A side-channel attack is another method an attacker might employ to take control of the susceptible hub [2]. From a commercial standpoint, the exposed data may provide invaluable insights into consumer behavior for Zigbee device makers. This data may be sold to marketers for use in targeted campaigns, used to monitor users around the web, or factored into the development of new products. In conclusion, our research offers substantial insights into possible information leakages at the very point of origin.

7.2: Potential Countermeasures

ZLeaks proves that unencrypted information (MAC OUI, frame, and payload lengths) is crucial for pinpointing Zigbee network instructions, events, and devices with a certain set of capabilities. For low-power Zigbee devices, the transmission overhead and increased power consumption of exponential padding [7] are not worth the benefits of concealing payload lengths. We propose utilizing the reserved field in the Zigbee security header to indicate the number of padded

bytes (for example, 0, 1, 2, or 3 bytes) in each payload. With this method, the Zigbee orders cannot be distinguished from one other even if they use the identical APL command but one of four possible payload sizes. Second, Zigbee Alliance might require the chipset manufacturer's identification to be used as the MAC OUI. As a result, the typical Score determined by the APL inference method for unknown devices drops from 4.0 (80%) to 3.1 (62%).

VIII. CONCLUSION

This research showed how the Zigbee protocol's focus on maximizing battery life has undermined the legal notion of privacy in connected dwellings. Using a cheap wireless Zigbee sniffer (TI CC2531), we introduced ZLeaks [14], a privacy analysis tool that utilises two inference approaches to show how readily a passive eavesdropper may detect in-home devices and events from the encrypted data. The ZLeaks command inference approach recognized unknown events and devices with 83.6% accuracy, without employing event signatures, in an examination done on a comprehensive collection of 19 distinct Zigbee devices and 5 smart hubs. Furthermore, ZLeaks' periodic reporting approach accurately recognized known devices (with zero user input) 99.8 percent of the time. Finally, we tested our command inference rules on an external capture file and found that regardless of the secret keys, we were 91.2% accurate in identifying APL commands that were particular to a certain capability. We draw the conclusion that the suggested inference attacks are unavoidable without fundamental modifications to the Zigbee protocol's architecture.

REFERENCES

1.	Marchal, S., Miettinen, M., Nguyen, T., Sadeghi, A.R., Asokan, N.: AuDI: Towards autonomous IoT device-type identification using periodic communications. IEEE Journal on Selected Areas in Communications. (2014)
2.	Meidan, Y., Bohadana, M., Shabtai, A., Guarnizo, J.D., Ochoa, M., Tippenhauer, N.O., Elovici, Y.: Profiliot: A machine learning approach for iot device identification based on network traffic analysis. In: Proceedings of the symposium on applied computing. pp. 506–509. ACM, Morocco (2013)
3.	Miettinen, M., Marchal, S., Hafeez, I., Asokan, N., Sadeghi, A.R., Tarkoma, S.: Iot sentinel: Automated device-type identification for security enforcement in iot. In: 37th International Conference on Distributed Computing Systems. pp. 2177–2184. IEEE, USA (2011)
4.	Pierre Marie Junges, J.F., Festor, O.: Passive inference of user actions through iot gateway encrypted traffic analysis. IEEE Symposium on Integrated Network and Service Management. IEEE, USA (2012)
5.	Trimananda, R., Varmarken, J., Markopoulou, A., Demsky, B.: Packet-level signatures for smart home devices. In: Network and Distributed System Security Symposium, 10(13), 54. NDSS, USA (2010)
6.	Copos, B., Levitt, K., Bishop, M., Rowe, J.: Is anybody home? inferring activity from smart home network traffic. In: IEEE Security and Privacy Workshops (SPW). pp. 245–251. IEEE, USA (2010)
7.	Acar, A., Fereidooni, H., Abera, T., Sikder, A., Miettinen, M., Aksu, H., Conti, M., Sadeghi, A.R., Uluagac, A.S.: Peek-a-boo: I see your smart home activities, even encrypted! In: WiSec 2013 - 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks. ACM, Austria (2011)
8.	Zhang, W., Meng, Y., Liu, Y., Zhang, X., Zhang, Y., Zhu, H.: Homonit: Monitoring smart home apps from encrypted traffic. In: Proceedings of the SIGSAC Conference on Computer and Communications Security. pp. 1074–1088. ACM, Canada (2014)
9.	Akestoridis, D.G., Harishankar, M., Weber, M., Tague, P.: Zigator: Analyzing the security of zigbee-enabled smart homes. In: WiSec 2011 - 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks. ACM, Austria (2010).
10.	Zigbee Alliance: ZigBee Specification, 05-3474-21 edn. (2013)
11.	Zigbee Alliance: Zigbee Cluster Library Specification, 07-5123-06 edn. (2013)
12.	Zigator CRAWDAD dataset CMU, (v. 2012-05-26), https://crawdad.org/cmu/zigbee-smarthome/20200526. Last accessed May 2014

13.	Ronen, E., Shamir, A., Weingarten, A.O., OFlynn, C.: Iot goes nuclear: Creating a zigbee chain reaction. In: IEEE Symposium on Security and Privacy, USA (2013)
14.	Sugawara, T., Cyr, B., Rampazzi, S., Genkin, D., Fu, K.: Light commands: laserbased audio injection attacks on voice-controllable systems. In: 29th USENIX Security Symposium. pp. 2631–2648. USENIX, USA (2011)
15.	Sun, Q., Simon, D.R., Wang, Y.M., Russell, W., Padmanabhan, V.N., Qiu, L.: Statistical identification of encrypted web browsing traffic. In: IEEE Symposium on Security and Privacy. pp. 19–30. IEEE, USA (2002)