Received: 6 November 2018 | Revised: 19 May 2019 | Accepted: 14 August 2019

DOI: 10.1002/widm.1337

OVERVIEW

WIREs DATA MINING AND KNOWLEDGE DISCOVERY WILEY

Behavioral fingerprinting of Internet-of-Things devices

Bruhadeshwar Bezawada[1] | **Indrakshi Ray**[1] | **Indrajit Ray**[1,2]

[1]Department of Computer Science, Colorado State University, Fort Collins, Colorado

[2]National Science Foundation, Alexandria, Virginia

Correspondence
Indrakshi Ray, Department of Computer Science, Colorado State University, Fort Collins, CO.
Email: indrakshi.ray@colostate.edu

Present address
Bruhadeshwar Bezawada, Mahindra École Centrale, Hyderabad, India.

Funding information
Air Force Research Laboratory; CableLabs; Furuno Electric Company; National Science Foundation, Grant/Award Number: CNS 1650573; SecureNok; U.S. Department of Energy, Grant/Award Number: DE-NE0008571

Abstract

Rapid advances in the Internet-of-Things (IoT) domain have led to the development of several useful and interesting devices that have enhanced the quality of home living and industrial automation. The vulnerabilities in the IoT devices have rendered them susceptible to compromise and forgery. The problem of device authentication, that is, the question of whether a device's identity is what it claims to be, is still an open problem. Device fingerprinting seems to be a promising authentication mechanism. Device fingerprinting profiles a device based on information available about the device and generate a robust, verifiable and unique identity for the device. Existing approaches for device fingerprinting may not be feasible or cost-effective for the IoT domain due to the resource constraints and heterogeneity of the IoT devices. Due to resource and cost constraints, behavioral fingerprinting provides promising directions for fingerprinting IoT devices. Behavioral fingerprinting allows security researchers to understand the behavioral profile of a device and to establish some guidelines regarding the device operations. In this article, we discuss existing approaches for behavioral fingerprinting of devices in general and evaluate their applicability for IoT devices. Furthermore, we discuss potential approaches for fingerprinting IoT devices and give an overview of some of the preliminary attempts to fingerprint IoT devices. We conclude by highlighting the future research directions for fingerprinting in the IoT domain.

This article is categorized under:
Application Areas > Science and Technology
Application Areas > Internet
Technologies > Machine Learning
Application Areas > Industry Specific Applications

KEYWORDS
fingerprinting, device behavior, IoT devices

1 | INTRODUCTION TO DEVICE FINGERPRINTING

Recent technological advances in computing and hardware have led to the proliferation of autonomous embedded networked devices in modern homes and industrial production systems. An example of such a device is a smart bulb, which can be controlled by a smart phone. A more complex device is a health monitoring system, which reports a patient's health statistics in real-time to the concerned health-care professionals. Networks comprising of such devices are covered under the umbrella term "Internet-of-Things" (IoT) (Greenough, 2016).

A sample home ecosystem is shown in Figure 1, which comprises of several IoT devices. Naturally, the security of IoT devices is important for service reliability as well as the stability of the host network. These devices have been the target of many attacks (Senrio, 2016), such as cryptographic key replacement, cloning, replay, and wormhole all of which essentially try to replace/forge a legitimate device. For instance, a subverted device or a malicious replacement/clone authenticates itself as a legitimate device to steal confidential information of users or to cause large scale disruption of the network (Krebs, 2017). Therefore, identification and authentication of these devices is a critical security requirement as only valid devices should be allowed to operate inside the host network. Although, network identifiers like IP addresses, MAC addresses, ports numbers, and so on have been used for identifying devices, these are "soft" identifiers and can be spoofed easily. The use of cryptographic keys is an alternative authentication mechanism wherein a device presents a cryptographically validated identity to authenticate itself. However, cryptographic approaches for IoT suffer due to poor key management issues, and resource constrained devices, which give limited flexibility to the developers. Also, there is a lack of consensus in the IoT community regarding the key management standards that are to be uniformly adopted. For instance, most devices present self-signed public-key certificates, which any attacker can easily generate, and there are no standard practices for checking for expired certificates.

Fingerprinting techniques (Lippmann, Fried, Piwowarski, & Streilein, 2003) attempt to solve this problem by creating robust immutable identities, similar to biometric *fingerprints*,[1] for network devices. The general approach for generating a device fingerprint is by observing specific information from the devices, extracting features from this information and encoding these features in a suitable format. The information is related to one or more device components such as operating system, device drivers, clocks, radio circuitry, and protocol implementations. The intuition is that such device fingerprints are intrinsically bound to a particular device or class of devices and hence, act as authentic identifiers for the device or class of devices. A device fingerprint is considered to be easy to verify but difficult to forge. As a simple illustration, when compared to a soft identifier like IP address, which can be easily spoofed, a device fingerprint is composed of multiple pieces of information, possibly using transformations like fast Fourier transform (FFT) or discrete wavelet transform (DWT) making it relatively difficult to spoof. The fingerprinting approach ensures that an attacker's complexity increases manifold and forces the attacker to exert more effort, both computational as well as monetary, when attempting to forge the device fingerprint. As a concrete illustration, consider the physical layer radiofrequency transmissions of a device that are affected by the imperfections/inconsistencies in the radio circuitry when compared with a collection of similar devices. These imperfections manifest themselves in the transmissions of the device and can be extracted as the device's fingerprint. The outcome of this approach is that, for an attacker, forging this information is challenging as the imperfections in the radio circuitry are a product of the manufacturing process and replicating imperfections of a given device is highly improbable. Another illustration of fingerprinting is the use of location dependent information such as received signal strength (RSS) or channel state information at receiver (CSIR), which depends on the relative location of a given device from the fingerprinting device. Any external or remotely located malicious device attempting to authenticate itself as a valid device will be detected and reported.

The IoT domain requires a fresh outlook with respect to fingerprinting due to the complex and heterogeneous nature of the devices involved. For instance, it is highly improbable to assume the availability of a generic device capable of capturing physical layer transmissions from every possible IoT device. But, it may be feasible to observe the network layer *behavior* of IoT devices and use this information, augmenting it with other features like clock-skews, for performing *behavioral fingerprinting*. Broadly speaking, behavioral fingerprinting attempts to fingerprint a device based on its data transmissions, which are indicative of the device behavior or functionality. A malicious or compromised device typically exhibits disruptive behavior that differs from the network behavior of a legitimate device. Behavioral fingerprinting has gained much importance as

FIGURE 1 A typical home Internet-of-Things (IoT) network

internet service providers (ISPs) are attempting to utilize software defined networking based solutions (Hafeez, Antikainen, Ding, & Tarkoma, 2018; Miettinen et al., 2017) to: (a) enforce automatic network access control based on device usage patterns (Roux, Alata, Auriol, Nicomette, & Kaâniche, 2017), (b) detect anomalous devices, based on misbehavior or misconfiguration (Barrera, Molloy, & Huang, 2017; Hall, Barbeau, & Kranakis, 2006), (c) detect malicious or compromised devices, those that have been infected with a malware or are under remote control by attackers (Krebs, 2017), and (d) check for device theft or replacement by unauthorized sources. Research in the industrial IoT domain is focusing on understanding the collective behavior of these devices in an attempt to: (a) predict faulty scenarios before they happen, (Formby, Srinivasan, Leonard, Rogers, & Beyah, 2016) and (b) monitor for the onset of stealthy attacks (Chen, 2010).

In this overview article, we cover previous fingerprinting approaches, and highlight the challenges involved in extending these approaches to the IoT domain. In particular, our survey focuses on the following key questions and how existing work answers one or more of these questions:

- What is the necessity for fingerprinting IoT devices?
- What information is useful for fingerprinting IoT devices? What is the correlation between the information and the type of fingerprint: device versus device-type and behavioral versus nonbehavioral.
- What are the challenges in collecting this information? What is the effect of environment or noise on this information?
- Can this information be manipulated by attackers?

We discuss behavioral and nonbehavioral fingerprinting of IoT devices and outline some future research directions in this problem space. Based on existing approaches and evidence of attacks (Krebs, 2017; Senrio, 2016), we emphasize that behavioral fingerprinting of IoT devices is an important and critical requirement in home and industrial environments.

2 | OVERVIEW OF DEVICE FINGERPRINTING

In this section, we first outline the threat model being addressed by device fingerprinting. Next, we describe the general approach for device fingerprinting and distinguish between two important approaches for fingerprinting devices: nonbehavioral and behavioral. Finally, we give an outline of existing nonbehavioral and behavioral fingerprinting approaches.

2.1 | Threat model

One major threat is device forgery, cloning or impersonation wherein an attacker possibly steals cryptographic credentials of a legitimate device. For example, the injection of rogue access points (APs) in local area networks (Jana & Kasera, 2010), which intercept and surreptitiously decrypt user communication, is a major threat to the privacy of user data. A device can be programmed to exhibit similar network traffic level features, such as vendor specific banners, and packet headers. or physical hardware characteristics, such as signal strength, Wi-Fi probe scans, and link layer frame headers of a legitimate device (Dabbagh & Saad, 2018). The extent to which an attacker succeeds in this depends entirely on the information base and the capabilities of the attacker. This adversarial possibility highlights the necessity for strong authentication measures that are not necessarily cryptographic in nature. A second major threat is the compromise of a legitimate device by an attacker and making it behave in a malicious manner. The firmware and software found on IoT devices have been found to have numerous vulnerabilities that have been exploited by attackers to launch serious attacks like data theft, and Denial-of-Service (Krebs, 2017). The broadcast nature of the wireless medium in IoT networks exacerbates any such threats and results in major disruptions to user services. Other attacks like sybil (Douceur, 2002) and wormhole attacks (Buttyán, Dóra, & Vajda, 2005) found in cooperative wireless networks such as sensor and mesh networks are major threats to cooperative IoT networks as well.

2.2 | Importance of fingerprinting devices

Fingerprinting devices is an important and effective approach for solving the problem of strong device identification for autonomous distributed devices that are under the threat of forgery or replacement. The process of fingerprinting is analogous to multifactor authentication, except that instead of using multiple temporally separated factors, a fingerprinting algorithm uses multiple distinct features of importance to generate a fingerprint. Depending on the features of choice, subverting a fingerprinting algorithm[2] could be almost impossible, especially features like radiometric features (Danev & Capkun, 2009) or clock-skews (Moon, Skelly, & Towsley, 1999). However, the more robust a fingerprinting algorithm is, it is likely to be more

expensive or inapplicable for a given application scenario. Therefore, the key technical challenge in fingerprinting is to arrive at the best possible fingerprinting algorithm that balances security and cost.

In general, fingerprinting comes in two flavors: device and device-type fingerprinting. Device fingerprinting refers to the fingerprinting of a particular device instance, for example, the D-Link camera located at the entrance of a home. Whereas device-type fingerprinting refers to fingerprinting the "type" or class of any device instance of particular device model, for example, any Philips Hue smart bulb. The key difference is that, a device fingerprint is guaranteed to be unique to only that particular device instance, whereas, the device-type fingerprint is common across all device instances of that device-type. If the device-type fingerprint were to be augmented with device specific features like clock-skews, which represents variability in clocks across devices, or other radiometric information, we will be able to get a device fingerprint for a given device. Most of the literature in the past has used these terms interchangeably where the distinction is made according to the context of the fingerprinting. Specifically, physical layer approaches (Danev & Capkun, 2009; Hall et al., 2006; Tugnait & Kim, 2010), link-layer based approaches (Choong, Cho, Tan, & Lee, 2008; Corbett, Beyah, & Copeland, 2008a, 2008b; Franklin & McCoy, 2006; Jana & Kasera, 2010) usually perform device-fingerprinting, and even few data-layer approaches (Brik, Banerjee, Gruteser, & Oh, 2008; Kohno, Broido, & Claffy, 2005; Moon et al., 1999) perform device fingerprinting. While approaches in Bezawada et al. (2018), Formby et al. (2016), François, Abdelnur, State, and Festor (2009, 2010), Gao, Corbett, and Beyah (2010), Miettinen et al. (2017), Radhakrishnan, Uluagac, and Beyah (2015), Siby, Maiti, and Tippenhauer (2017), and Sivanathan et al. (2017, 2018) consider device-type fingerprinting, partly or solely. In this overview article, the context of fingerprinting is usually made clear and hence, we will use the term device fingerprinting to refer to both approaches.

Our choice of fingerprinting approaches in this article is mainly based on the information collected and utilized by the respective approaches. The nature of information and the type of analysis performed determines the quality of fingerprinting. For instance, the use of *clock-skews*, as shown in Kohno et al. (2005) and Moon et al. (1999) allows us to uniquely identify a computing device or in other words, fingerprint the device. We followed a chronological path, starting with physical layer fingerprinting based on radio transmissions, progressing through other metrics and finally to network traffic based fingerprinting, while covering a representative approach across the different class of approaches. A second important criteria for our choice was based on the semantics of the fingerprinting approach, that is, behavioral versus nonbehavioral fingerprinting. However, the nature of information collected by an approach is quite indicative of the type of fingerprinting being attempted and therefore, this aspect was implicitly in our first choice criteria.

2.3 | General architecture of device fingerprinting

In a general sense, there are four key aspects for fingerprinting devices: information capture, feature extraction, fingerprint generation/registration and fingerprint recognition algorithms as shown in Figure 2.

In the first step, the fingerprinting device or tool is strategically placed to capture the relevant information required to fingerprint a device, for example, if the information required is transient radio emissions (Danev & Capkun, 2009) from the device, then a radio receiver is placed within the transmission range of the device.

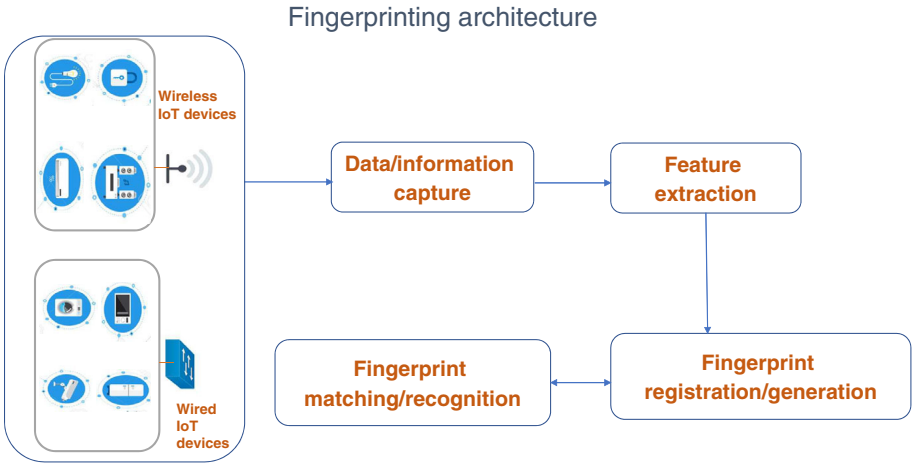

FIGURE 2 General fingerprinting architecture. IoT, Internet-of-Things

In the second step, the relevant features essential to represent the device fingerprint are either recorded from the information or inferred with help of transformations like FFT (Danev & Capkun, 2009) or DWT (Hall et al., 2006). The selection of these features is a key challenge for the fingerprinting process and can vary significantly from device to device.

Next step, in the fingerprint generation/registration, the fingerprint generation step encodes the features from the previous step and records them against the device's identifier. Depending the type of fingerprint recognition algorithm used, this step stores the identified fingerprints differently.

Final step, the fingerprint recognition algorithm validates or "reidentifies" a device's run-time fingerprint against the stored fingerprints with help of a similarity measurement technique. Supervised and unsupervised machine learning based classification algorithms (Jain, Duin, & Mao, 2000) have been popular for fingerprint recognition. Supervised algorithms generally perform "white-listing" of known devices where the learning phase involves training the machine learning classifier on the available data of the device and use the trained classifier information to reidentify a known device. Unsupervised algorithms have been used in scenarios where a general profiling of devices is performed without any prior information about the available devices. In both approaches, with an invalid or unknown fingerprint is reported by the fingerprinting tool for further analysis and inspection.

3 | CHALLENGES IN IOT DEVICE FINGERPRINTING

Danev, Zanetti, and Capkun (2012) and Xu, Zheng, Saad, and Han (2016) identify some important properties for any fingerprinting approach: *universality*, features that any considered device should have; *uniqueness*, each device should have unique fingerprint; *permanence*, the fingerprints should not vary over time and *collectability*, the necessary information required for fingerprinting should be available with existing equipment. Keeping these properties in mind, we outline why most of the existing approaches may not be suitable for fingerprinting IoT devices.[3]

First, existing physical layer based radiometric identification techniques (Brik et al., 2008; Hall et al., 2006; Tugnait & Kim, 2010; Yu & Sadler, 2011) require expensive signal capturing equipment that needs to be attuned to the channel frequencies, and further, performing signal processing using FFT or DWT requires necessary hardware and software capabilities. Now, from the perspective of fingerprinting, collectability would be a major issue for IoT devices since IoT devices operate in all known radio communication spectrum with standards like 802.15.4 based Zigbee, ISA100.11a, WirelessHART, MiWi, SNAP, Bluetooth, Wi-Fi, Ethernet, LPWAN, LoRaWAN, RFID, and 3GPP. Second, techniques that analyze the behavior of individual protocols or characterize the implementation of device drivers are not feasible either because of the broad range of protocols used by IoT devices and any technique that works for one protocol would have to be reengineered for other protocols. The wide range of protocols in IoT devices, such as REST, HTTP/2, SOAP, MQTT, CoAP, STOMP, XMPP-IoT, AMQP, DDS, LWM2M, 6LowPAN, 6TiSCH, RPL, IPv4/v6, mDNS, and DNS-SD confirm this aspect. Third, approaches (Bratus, Cornelius, Kotz, & Peebles, 2008; Choong et al., 2008; Corbett et al., 2008a, 2008b) that analyze link layer scanning techniques may not be feasible as it may not be possible to place the fingerprinting tool between the device and the AP, say, in home IoT environments. These limitations cover an almost exhaustive range of existing fingerprinting approaches and hence, provide the necessary motivation of developing novel approaches for fingerprinting IoT devices.

In the following sections, we will describe various fingerprinting approaches starting with fingerprinting approaches for general computing devices and then describe approaches for fingerprinting IoT devices. We mainly focus on two types of fingerprinting approaches: nonbehavioral and behavioral, which exist for both general computing devices and IoT devices. We highlight the key differences between fingerprinting general computing devices and IoT devices and show that there is need for new approaches for fingerprinting IoT devices. We begin with nonbehavioral fingerprinting approaches for general computing devices and work our way to behavioral fingerprinting approaches for IoT devices.

4 | NONBEHAVIORAL DEVICE FINGERPRINTING

Our usage of the term "nonbehavioral" fingerprinting is mainly to characterize the class of fingerprinting approaches that do not use the device's functionality or data communication for fingerprinting. Specifically, these approaches do not consider the "behavioral" aspects of the device such as responding to requests or sizes of the messages exchanges, and so on.

4.1 | Fingerprinting general computing devices

In this section, we consider nonbehavioral fingerprinting approaches of general computing devices like PCs, laptops, PDAs, and so on, that were prevalent in literature prior to the IoT revolution.

Radiometric analysis is one such approach used by many earlier researchers (Brik et al., 2008; Danev & Capkun, 2009; Hall et al., 2006) to fingerprint general purpose wired and wireless devices. For instance, the transient signal phase during the turn-on period of a device contains sufficient distinctive information, such as frequency and phase offsets, to fingerprint a device with good accuracy. In Hall et al. (2006), the fingerprinting tool samples the transient portion of the signal and applies the DWT technique to transform the signal into the frequency domain and extracts features like the normalized DWT coefficients, power per signal segment, and normalized DWT coefficients by levels. Danev and Capkun (2009) used transient signal analysis for fingerprinting 802.15.4 CC2420 radio transceivers for wireless sensor nodes Using FFT spectra based features of the sampled transient signal of a device, the authors were able show unique sensor identification with a high accuracy of 99%. Typically, the transient signal reflects the hardware profile of the device and therefore, is very difficult to forge due to inevitable inconsistencies in manufacturing the concerned hardware.

Several other approaches (Tugnait & Kim, 2010; Yu & Sadler, 2011) explored location dependent features like RSS and CSIR, which depend on the relative distance from the receiver and the device. Location dependent features suffer from issues like device mobility and signal noise in the vicinity of the receiver, that is, the device performing the fingerprinting. While an attacker can take advantage of these factors to subvert or disrupt a fingerprint, the attacker still needs to be in the physical vicinity of the fingerprinter, which may or may not always be possible.

Franklin and McCoy (2006) looked at link layer frame features and characterized the probing behavior in terms of cycles, that is, the probing behavior of a particular device repeats in cycles. Using a machine learning based model with the help of a binning model for time-deltas of the probing cycles, the authors were able to fingerprint the device drivers on the wireless network interface card (NIC) of the device. Strictly speaking, this approach performs fingerprinting of a specific type of device. Subverting this approach is possible by injecting fake link layer frames to confuse the fingerprinter, an approach known as adversarial classification (or sampling) (Dalvi, Domingos, Mausam, Sanghai, & Verma, 2004). Similarly, the approaches in Choong et al. (2008) and Corbett et al. (2008a) used the timing analysis of the probe request frames during the active scanning phase of the device's wireless NIC card for generating the device fingerprint. However, similar to the approach in Franklin and McCoy (2006), these approaches are also susceptible to adversarial classification attacks. In closing the discussion, we note that, most of radiometric and link layer based approaches are not directly applicable for fingerprinting IoT devices as the fingerprinter needs to be in the physical proximity of the device, which may not be always possible.

Another key feature that researchers (Jana & Kasera, 2010; Kohno et al., 2005; Moon et al., 1999) have focused on is the clock variability, that is, the skew of the clocks between any two devices. Clock skews can be measured with the help of TCP time-stamps added by the device or time synchronization information found in 802.11 frames (Kohno et al., 2005) or the packet delays (Moon et al., 1999) as observed from the device being fingerprinted. One important application of this approach was to detect rogue APs (Jana & Kasera, 2010) that are introduced into the network to steal confidential data of the users. Most of these features are extracted passively from the transmissions of the device and do not depend on the device's data communication or intended functionality. Subverting these approaches is possible by actively injecting fake packets to disrupt the measurements being taken by the fingerprinter. However, an attacker needs to be able to do this each time the device is being fingerprinted, which makes this approach quite robust to attacks.

In Table 1, we highlight the distinct set of approaches, the features used and the fingerprint recognition algorithms utilized by these approaches while showing the classification accuracy achieved.[4]

TABLE 1 Overview of nonbehavioral approaches

Method	Features	Similarity method	Device population size	Accuracy (%)
Franklin and McCoy (2006)	NIC probe times	Bayesian classifier	17	77–96
Hall et al. (2006)	DWT features	Hotelling's T2 classifiers	30	87–100
Choong et al. (2008)	Probe times	Mann–Whitney *U*-test	3 laptops and 3 NICs	86–100
Danev and Capkun (2009)	FFT spectra	Mahalanobis matching	50	99
Kohno et al. (2005)	TCP/ICMP time stamps	Linear programming	1–3 devices	N/A
Jana and Kasera (2010)	Time synchronization function	Linear programming	2 laptops	99

Abbreviations: DWT, discrete wavelet transform, FFT, fast Fourier transform; NIC, network interface card.

4.2 | FINGERPRINTING IoT DEVICES

Based on the knowledge base of fingerprinting general devices, several researchers have developed fingerprinting approaches for IoT devices based on physical and link layer characteristics.

Dalai and Jena (2017) have considered the wireless probe requests to extract different features. Their main focus was to identify and extract features that cannot be forged. They used correlation-based feature selection measure to evaluate subsets of features that are correlated to classification but exhibit relative independence from each other. After extracting these features, they further analyzed which of these features are prone to manipulation by attackers/environment or exhibit variability and narrowed down to 19 features out of total of 53 fields in the 802.11 probe fields. These are mainly related to the management features of the devices and are difficult to forge. They used different similarity metrics for measuring the closeness of feature vectors such as Cosine, Euclidean, and Jacquard distance. The mean, standard deviation and energy of the feature vectors are treated as the signature of the device. The similarity to the signature is measured using the distance similarity metrics. They tested their approach on 300 device types and report accuracy of 95%.

Oser, Kargl, and Lüders (2018) use clock characteristics and timestamp based features to identify devices. They considered 51 device models and 562 devices. Popular features they considered are: increase over all consecutive timestamp values, largest increase of two consecutive timestamp values mean increase over all consecutive timestamp offsets, smallest increase of two consecutive timestamp values, timestamp overflow occurred, sum over all consecutive timestamp offsets, last timestamp in a scan period, largest timestamp in a scan period, median over all timestamps, smallest timestamp in scan period, and so on. For each device, 576 timestamps is one scan period and they considered several scan periods for training and testing with a 97% precision in their learning model.

Features like interarrival times (IAT) and clock skews might work as it may be possible to universally extract them from most devices with minimal packet capture equipment. However, these features exhibit variability that does not guarantee permanence when measuring IAT in congested networks, and collectability (Section 3) when measuring clock-skews with the help of protocol specific options that may or may not be universally available. However, since all IoT devices use some amount of packet data for communication, the aspect of *universality* (Section 3), is achievable if the fingerprinting features can be modeled on the network traffic. Therefore, existing IoT fingerprinting approaches have been based on the features extracted from network traffic and more recent approaches have considered behavioral modeling of IoT devices.

5 | BEHAVIORAL FINGERPRINTING OF DEVICES

The behavioral fingerprinting approaches focus on the behavioral aspects of a device that can be observed and measured. For instance, a device communicating with DNS servers at regular intervals is a behavioral trait. For many devices, the set of protocols utilized, the patterns of request-response found in the protocols, the periodicity of messages, the sizes of the messages exchanged, and so on, are treated as behavioral aspects. Clearly, the behavioral traits of a device serve as good metrics for fingerprinting and have been used by network researchers to develop useful fingerprinting approaches in the past. General computing devices exhibit patterns specific to their configurations, that is, the applications installed on the device or the services being provided by the device. Whereas IoT devices exhibit patterns specific to their functionality and user interactions with the device. Moreover, IoT devices, such as motion sensors, may also respond to environmental factors and exhibit irregular communication patterns. These differences are sufficient to demonstrate that fingerprinting approaches for general purpose devices may not be directly applicable to IoT devices and some level of adaptation may be required. In the following, we will first describe and compare behavioral fingerprinting approaches for general purpose devices and then describe fingerprinting approaches for IoT devices.

5.1 | Fingerprinting general computing devices

In this section, we consider the behavioral fingerprinting approaches used for general purpose computing devices. The behavioral fingerprinting approaches use features that depend on the data communication of the device.

Brik et al. (2008) describe PARADIS, a fingerprinting framework for wireless NICs using data modulation dependent features such as the frequency error, SYNC correlation, I/Q origin offset, magnitude and phase errors. The authors considered the base band signals for this purpose and used support vector machine based classification to achieve good fingerprinting accuracy. Broadly speaking, behavioral fingerprinting is likely to produce better fingerprints due to the significant behavioral

variation among the devices. While it is possible to subvert this approach with adversarial classification, the likelihood is lesser due to the number of factors that an adversary has to consider.

Corbett et al. (2008a) use spectral analysis to identify the type of wireless NIC by considering rate switching, a vaguely specified mechanism required by the 802.11 standard, that is, implemented in the hardware and software of the wireless NIC. As rate switching affects the transmission patterns of a wireless stream, it is a good feature to model for fingerprinting. The authors use spectral analysis to analyze the periodicity embedded in the wireless traffic caused by rate switching to generate the device fingerprints. The difficulty of subverting this approach is similar to subverting the transient signal based approaches as the periodicity of wireless traffic depends on the specific hardware and software, which are difficult to replicate exactly.

In Bratus et al. (2008), using a tool called BAFFLE,[5] the authors have collected responses in the IEEE 802.11 MAC header fields to nonstandard events, as these are not well specified in the 802.11 standard and depend from vendor to vendor. From this data, the fingerprinting features extracted were the combinations of frame type, sub type and frame control flags that are not specified in the standard or unlikely to occur in normal device operations. However, active fingerprinting is not suitable in many scenarios as it disrupts the functioning of the device being fingerprinted and moreover subverting this approach is relatively easier as it only requires injecting fake link layer frames.

In François et al. (2009, 2010) considered the variations in the implementation of specific protocols like the session initiation protocol across devices as a feature vector. The feature vector for each device is generated by parsing the protocol messages to generate the corresponding syntax trees for the 27 different types of messages. By determining abnormal sequences in these feature vectors, it is possible to fingerprint a device or a particular vendor. These approaches are based on the behavior of the protocol and subverting them requires the adversary to behave exactly the same way as a legitimate device.

Gao et al. (2010) used TCP or UDP packet IAT to characterize an AP wherein the feature vector of a given AP is generated using several packet traces and the values are stored in fixed sized bins. The authors used wavelet transforms and circular cross-correlation to perform the similarity matching of the generated feature vectors with results reaching near 100% accuracy. GTID (Radhakrishnan et al., 2015) is another approach where the authors used fine-grained IAT analysis by observing the packet traces of a given device and using the binning approach coupled with supervised machine learning algorithms. Both these approaches are susceptible to inconsistencies in delay measurements or due to delay introduced by network switches.

For cyber-physical systems or industrial-control systems, such as power generation or natural gas plants, in Formby et al. (2016), the authors used two distinct approaches. In the first approach, they modeled the workload of a particular device by considering the time taken by devices to acknowledge, specifically to send back the TCP acknowledgements, commands from other components of the system. The timing measurements give a decent estimate about a device and its work profile, which enables the fingerprinter to validate at a latter time that the device is behaving within its normal profile. The second approach involved considering the time take by a device for individual control operations, that is, by considering the specific set of operations that can be performed by this device and measuring the time taken by the device to complete these operations. As seen in earlier approaches, due to imperfections in hardware, two similar devices tend to display dissimilarities in such measurements and hence, provide valid feature vectors to fingerprint each device. For the application scenario considered, these approaches are difficult to subvert and any attacks are usually performed in a stealth mode as done in the famous Stuxnet attack (Chen, 2010). We provide an overview of the different methods and their features in Table 2.

TABLE 2 Overview of behavioral approaches

Method	Features	Similarity method	Device population size	Accuracy (%)
Brik et al. (2008)	Transient I/Q features	SVM	138	99
Corbett et al. (2008a)	Rate switching	N/A	6	N/A
Bratus et al. (2008)	Frame control flags	Decision trees	40–90	N/A
François et al. (2009)	Syntax trees	Support vector clustering	40	80
François et al. (2010)	Syntax trees	SVM	6	90–99
Gao et al. (2010)	IAT	Circular cross-correlation	6	99–100
Radhakrishnan et al. (2015)	IAT	Artificial neural network	37	83–96
Formby et al. (2016)	Timing analysis	Naive Bayes	130	92–99

Abbreviations: IAT, interarrival times; SVM, support vector machine.

5.2 | BEHAVIORAL FINGERPRINTING OF IoT DEVICES

Behavioral fingerprinting approaches (Bezawada et al., 2018; Dabbagh & Saad, 2018; Hafeez et al., 2018; Meidan et al., 2017; Roux et al., 2017; Siby et al., 2017; Sivanathan et al., 2017, 2018; Thangavelu, Divakaran, Sairam, Bhunia, & Gurusamy, 2018; Yang, Li, & Sun, 2019) for fingerprinting IoT devices have been proposed recently. Siby et al. (2017) describe IoTScanner, an architecture that passively observes network traffic at the link layer, and analyzes this traffic using frame header information during specific observation time windows. Their main goal is to distinguish actively scanning IP cameras from other noncamera devices. This work is more concerned with discerning the distinct devices and their presence based on the traffic patterns observed during the traffic capture time window. A shortcoming of this approach is that two identical device-types could be classified as two different device-types due to the variations in traffic generated during traffic capture time window. This approach is useful for network mapping at a high level, but performing this analysis periodically can be cumbersome.

Meidan et al. (2017) focused on detecting unauthorized IoT devices in the network. They examined two kinds of attacks: untargeted, where an IoT device has already been infected by a malware due to cross-contamination, and targeted, where a malware has been intentionally injected into a device. Based on this model, the authors consider white-listing and black-listing of IoT devices. They used only TCP/IP traffic data collected over a long period of time for classification of nine IoT device types. These features were extracted from TCP/IP sessions. Since TCP/IP sessions typically correspond to behavior of the device, this approach essentially examines the behavior of the IoT device type. Sample IoT devices were refrigerator, watches and baby monitors. They used 274 features extracted from application, transport and network layers. The popular features were based on time-to live (TTL) such as minimum TTL, first quartile TTL, and average TTL, and byte ratio between bytes sent and received. For classification, one of the device types was left out as unauthorized and the training performed on the remaining device types, with each device type as a single class. They treat the problem as a multiclass classification problem and train a Random Forest classifier from the extracted TCP/IP session data. The unauthorized device type detection accuracy was 94–100% and their device identification rate was 97% on an average. The main issue with this scheme is that the TTL values constitute soft information, that is, information that is vulnerable to forging. An adversary with the knowledge of these patterns, can create a device that impersonates these values.

Roux et al. (2017) focus on detecting potential attacks in smart places (e.g., smart homes) by detecting deviations from legitimate communication behavior, in particular at the physical layer. Specifically, their approach attempts to address device usage in the following scenarios: anomalous location, unused wireless technology, unusual period of day, and unusual pattern of usage. The proposed solution is based on the profiling and monitoring of the radio signal strength indication (RSSI) associated to the wireless transmissions of the connected objects. A machine learning neural network algorithm is used to characterize legitimate communications and to identify suspicious scenarios. However, as with most approaches that attempt to perform profiling based on physical characteristics, this approach is vulnerable to noise in the environment and also, the requirement of being the proximity of the devices is difficult to satisfy in all environments.

Dabbagh and Saad (2018) consider the following two adversarial threats: (a) adversaries that are able to emulate security keys, device addresses, and transmitted data type, and (b) replicate the object's software, such as object security keys and object network address, and also, clone the legitimate object's device-specific information such as transmission speed, signal strength, processing speed, and operating temperature and humidity. They make an observation that, one of the features that emulation attackers cannot replicate are the environmental changes that pertain the environment that surrounds IoT objects, such as changes in temperature, humidity, wind, physical displacement, or any physical changes affecting IoT objects. Based on this observation, they design a fingerprinting approach that uses features from a three-tiered space and model the fingerprint as being impacted by the noise from the neighborhood of the device. First, at the object level, the features are CPU load, clock skew, memory usage, and temperature of the object and/or surroundings, among others Second, the features that the monitoring objects collect are signal strength, signal spectral features and packet arrival times. The third tier of features is collected at the IoT server side by measuring the traffic properties of objects and frequency of received packets. The authors create a training fingerprint for a given object using these features and use the Bhattacharya distance (Kailath, 1967) for measuring similarity. They have estimated the impact on the fingerprint by the environment using a linear noise model that rotates and translates the fingerprint. The noise matrices are calculated by considering the unique neighborhood of each IoT device and using the reference fingerprints of those objects along with singular value decomposition to solve for the two transformation matrices. To compute similarity, the reverse transformations are applied to any estimated fingerprints. The validation of the technique was done on 25 RFID tags. This technique combines both behavioral and nonbehavioral fingerprinting along with estimating the environmental effects.

However, many applications require remote fingerprinting of devices, which may not allow estimating object level features as described in this work. Moreover, this technique places overhead in estimating a sample fingerprint during the testing phase. Also, the fingerprinting tools needs to have intimate knowledge of the object's neighborhood in order for this scheme to be effective.

Bai, Yao, Kanhere, Wang, and Yang (2018) use long-short-term-memory–convolutional neural network cascade model to classify IoT devices based on their observed network behavior. First, they collect network data of devices that may correspond to various network configuration data like DNS queries, user activity data like Google Home commands, and background device-to-server data.

For extracting features, based on fixed time period value T, the packets are segmented into chronological sequences based and arranged according as per increasing order of timestamps. A further division of packets is made into control data, which consists of ARP, DNS packets, and so on and user data, which consists of HTTP, TCP, UDP packets, and so on corresponding to user activities. Some sample control data features are time stamp, packet length, protocol, and so on the counts of packets for different protocols like DNS, ARP, NTP, total packets, control packets, received packets, transmitted packets, user packet count, user packet length average, user packet length peak, and so on. Also, features include packet length statistics such as maximum, minimum, mean, sum, standard deviation, variance, skewness and kurtosis, as computed over a particular sequence of packets. They utilized network traffic from 15 devices that included Amazon Echo from Hubs, Belkin Wemo Switch, TP-Link Smart Plug from Switches&Triggers, Pix photo frame from Electronics and Withing Smart Baby Monitor, Netatmo Welcome, and Samsung Smart Camera from Cameras, for training. Data from six other distinct, but similar categorical devices was used for testing, which represents unseen data. Unlike most existing approaches, they used multiclass labeling algorithm and achieve 80% accuracy. But, this approach is mostly focused on categorizing data into specific IoT device-type category and not necessarily into specific device-type. As such, this approach can be utilized as a preliminary step for device-type fingerprinting.

Thangavelu et al. (2018) describe a decentralized fingerprinting approach, DEFT, that is, unique approach compared the other existing approaches. In this approach, the DEFT control logic resides in the ISP network, and controls a set of gateways for fingerprinting IoT devices. The session data is collected at gateways and features are extracted. The feature training is done by the controller and the models are provided to the gateways. The gateways use the models to separate the traffic for sessions where a session is marked by a time interval. Sessions classified with low probability are different sessions and are used for error correction and further retraining the model. The aggregate features are: number of DNS queries, DNS packet count, TLS packet count, TLS packet length, HTTP packet stats, protocol specific data for different protocols like SSDP/QUIC/MQTT/STUN/NTP/BOOTP, and so on for a total 111 features. The features are clustered to generate signatures common to a single category and therefore can detect unknown devices. They use Euclidean distance, z-score, and k-means clustering. The validation was performed over 16 devices that included: Echo dot, Smart remote, Camera, Smart Socket, Chromecast, Hue Light, Smart Bulb, and so on achieving 97% accuracy for known devices and 70% for unknown devices.

Shahid, Blanc, Zhang, and Debar (2018) use IAT of packets as features to profile IoT devices. They use a fixed parameter N for the number of packets from which the features are to be extracted. The features generated are: the size of first N packets sent, size of first N packets received, the $N-1$ packet IAT for sent packets and $N-1$ packet IAT for received packets. They used standard classification techniques like Random Forest, Decision tree, and so on to achieve an accuracy range of 90–90%, for a device space that included D-Link Motion Sensor, Nest Security Camera, TP-Link Smart Bulb, TP-Link Smart Plug, and so on. This approach basically uses IAT, a feature that has worked well in generic fingerprinting approaches (Jana & Kasera, 2010). However, this approach is susceptible to attacker manipulation as IAT is a controllable parameter.

Hafeez et al. (2018) describe a device profiling approach by considering the connections made by a device. While this approach is not necessarily fingerprinting a device, it does create a behavioral profile for a device that can be used for fingerprinting the device. They consider N connections where a connection has the same source, same destination, same service, for feature generation purpose. For all connections they collect statistical features like total number of IP addresses, unique destination IP addresses, source ports, destination ports, connection duration, and so on. They also extract packet counters for counting number of instances of protocol usage for protocols like ARP, LLC, IP, ICMP, EAPoL, TCP, UDP, HTTP, FTP, HTTPS, DHCP, DNS, NTP, and protocol fields like Router Alert, SYN, REJ, Urgent, Padding, and so on. They use c-means clustering for categorizing the data and for anomaly detection with an accuracy range from 74 to 99% for various types of attacks.

Bezawada et al. (2018) model the behavior of an IoT device and generate features based on this model. Specifically, the authors consider the different types of messages sent by each device, the buffer space available in different devices and the nature of information in the messages sent by the devices. The features are extracted by capturing the network traffic

corresponding to a device interaction and training a machine learning classifier against all interactions of a particular device. In addition to packet header features, the authors considered the payload length, the entropy of messages and the TCP window size of each device type being fingerprinted. These features are collectable even if the data is encrypted and hence, suitable for IoT devices. The training was performed on standard classifiers with a five-fold cross-validation. The approach was able to successfully fingerprint 14 device-types with a high identification rate of 98% and an accuracy of 99%.

Sivanathan et al. (2017, 2018) present a holistic approach for fingerprinting while making the data sets public and enabling researchers to experiment on these sets. Essentially, the authors consider profiling of IoT devices in a large network system to enable better understanding of the device behavior. Their network set up has over 28 types of IoT devices in addition to regular computing devices like PCs, laptops and smart phones. The various features studied were: traffic load and signaling patterns, packet size distribution, dominant protocols used and the distribution of active and sleep times. They observed some important behavior of IoT devices such as: (a) IoT activities are short lived, (b) IoT devices exhibit sleep and active profiles where IoT devices wake up frequently and generate network traffic, (c) most of IoT traffic is not encrypted, (d) cloud communication is more for IoT devices and selected If-This-Then-That servers for each IoT devices are observed behavior, (e) DNS queries limited to only few domains, specific to their vendor mostly whereas non-IoT devices look for more DNS domains, and (f) NTP is frequently used to synchronize. For the experiment, the authors collected data for 6 months from the test bed where the data was collected at the university gateway. Then data was then divided into hourly segments to generate the training and testing data points. In each hourly data points, for a given device identified by its unique MAC address, the features extracted were: flow volume, flow duration, average flow rate, device sleep time, number of server ports visited, number of distinct DNS queries, number of NTP queries and number of SSL/TLS cipher-suites used. They used a multistage classifier to perform the classification: the first stage used a subset of nominal features, that is, features not interpreted as numeric values, such as number of ports visited and the second stage used the other real or continuous valued features along with the output of the first stage for training and classification. Their approach resulted in a high accuracy of 99% for fingerprinting. The key limitation of this approach is that amount of data required to fingerprint a particular device, that is, for testing a device's fingerprint at least 1-hour worth of data is necessary. Most approaches in literature expect the fingerprinting process to happen on shorter duration of observed data.

A more recent approach, in Yang et al. (2019), uses features from network, transport and application layer of a device to generate fingerprints. Their approach attempts to automatically extract the device specific information from vendor websites and use this information to effectively fingerprint the devices. At the network layer, the features extracted are, Type of Service, TTL, Do not Fragment; at the transport layer the features extracted are, Port number, Receiver window, Max Segment Size and TCP Options. At the application layer, header and Payload of all application layer protocols: FTP, HTTP, SSH, TELNET, S7 protocol, PCWorx protocol, MELSEC-Q protocol, MODBUS protocol, FINS, GE_SRTP, MOXA NPORT, FOX, HART-IP, BACnet, UPnP, and so on. The authors use web crawlers to extract product/vendor information for effective labeling of the data sets. The authors note that many vendors use specific labels in their application layer protocols that makes it easy to fingerprint. They considered over 67 device types. For training, each packet is labeled and the feature vector extracted. Neural networks are used to train the classifier. Classification is done on a per device type level where the vendor can be anything. Therefore, this approach is more generic than existing approaches in literature. Their precision is close 95%. However, this scheme is vulnerable to manipulation as the authors have only considered vendor specific information and this can be easily forged.

One behavioral aspect of an IoT device is the energy consumption pattern as this pattern is directly related to the amount of processing and communication performed by the device. However, there are considerable difficulties in the collectability aspect of this metric as the fingerprinter needs to be able instrument the device and monitor for the energy consumption levels. There has been some research to model (Martinez, Monton, Vilajosana, & Prades, 2015) and measure (Morin, Maman, Guizzetti, & Duda, 2017) the energy consumption of IoT devices. Martinez et al. (2015) model three key aspects of a device to measure the power consumption: data acquisition, data handling and data communication. These aspects are used to estimate the power usage of an IoT device in different communication scenarios like point-to-point communication, retransmission, radio usage to wake up and sleep, and use of time sychronization protocols for MAC layer transmissions. Morin et al. (2017) study the expected lifetime of IoT devices operating in wireless networks. The authors develop an anlayzer to estimate energy consumption of a given protocol in a given state and the duration of the state. Their analyzer considers the active and inactive states of the radio, microcontroller etc with known power consumption patterns and timing constraints defined by the MAC layers. The outcome of the work was to choose the right MAC parameters to optimize the energy consumption for a given application. However, while this is a promising direction for fingerprinting, making such fine-grained measurements on an IoT device may be difficult or even impossible in several network settings.

6 | FUTURE RESEARCH DIRECTIONS

The area of fingerprinting IoT devices is both challenging and exciting at the same time. One of the future challenges is to be able to fingerprint devices even when they are using encrypted communication. The behavioral modeling is difficult if the device's communication is scrambled. However, even in this scenario it is quite essential for a network administrator to be able to fingerprint the device.

The second challenge is to be able to use the fingerprinting results effectively to perform access control on the network. There needs to be some form of automatic defense framework in place that will generate the necessary access control policies when some suspicious activity is reported in the network. The challenge lies in the successful implementation, acceptable performance and the correctness of such a dynamic access control framework.

The performance of the fingerprinting approach and its impact on the network are key aspects for ensuring a good security profile of the network. Most fingerprinting approaches are passive in nature and do not cause disruptions. However, the periodicity of the fingerprinting might be a cause of concern, if the fingerprinting results are to be used for enforcing run-time security policies.

The accuracy and identification rates of the fingerprinting approaches remain a key issue of contention. At present, except for limited experimental analysis, there is no other method available to test the scalability and reliability of a given classification algorithm. Furthermore, the impact of adversarial sampling, that is, attempts by an adversary to manipulate the training data of the fingerprinting algorithm, need to be factored in as well. This problem remains an open area of research and needs further exploration.

7 | CONCLUSION

In this overview article, we have covered the chronological highlights of the exciting area of device fingerprinting with a focus on IoT devices. Several existing works have looked at the physical layer characteristics to fingerprint devices. However, these approaches are expensive and many of them do not adapt to the current IoT device, which exhibit a vast amount of heterogeneity. We have described several other approaches that considered the data aspect of the device for fingerprinting the device. Some of these approaches are based on timing analysis and are designed to fingerprint specific devices that exhibit a certain behavior, for example, probe scans for an AP. The general nature of IoT devices requires more generic fingerprinting solutions that are cost-effective and provide value. Especially, the fingerprinting techniques for IoT should be geared towards providing a better security profile and detect malicious activity in real-time. We outlined a few future research directions that might give some guidelines to security researchers in this domain.

ACKNOWLEDGMENTS

This work is supported in part by funds from National Science Foundation (NSF) under award no. CNS 1650573, CableLabs, AFRL, Furuno Electric Company, and SecureNok. A gift from CableLabs supported the equipment to carry out this research. In addition, this work is partially supported by the U.S. Department of Energy under award number DE-NE0008571 subcontracted through the Ohio State University. This material is also based upon work performed by Indrajit Ray while serving at the National Science Foundation. Research findings presented here and opinions expressed are solely those of the authors and in no way reflect the opinions the DOE, the NSF or any other federal agencies.

CONFLICT OF INTEREST

The authors have declared no conflicts of interest for this article.

AUTHOR CONTRIBUTIONS

B.B. contributed to investigation-equal, methodology-equal, and writing-original draft-equal. Indrakshi Ray and Indrajit Ray contributed to conceptualization-equal, funding acquisition-equal, investigation-equal, methodology-equal, writing-original draft-equal, writing-review, and editing-equal.

ENDNOTES

[1] According to Wikipedia, its "…an impression left by the friction ridges of a human finger".

[2] We describe this in later sections for specific fingerprinting algorithms in literature.

[3] The polychotomy needs to be noted, IoT devices, wireless devices, desktop devices, embedded devices, etc. all represent different classes of devices wherein IoT devices exhibit the tremendous technological heterogeneity.

[4] Note that these experiments were conducted on different populations of devices and hence, these numbers cannot be used for empirical comparison

[5] Behavioral Active Fingerprinting of Link Layer

ORCID

Indrakshi Ray https://orcid.org/0000-0002-0714-7676

RELATED WIREs ARTICLES

Knowledge discovery for enabling smart Internet of Things: A survey

REFERENCES

Bai, L., Yao, L., Kanhere, S. S., Wang, X., & Yang, Z. (2018). Automatic device classification from network traffic streams of internet of things. *arXiv preprint arXiv:1812.09882*.

Barrera, D., Molloy, I., & Huang, H. (2017). Idiot: Securing the internet of things like it's 1994. *arXiv preprint arXiv:1712.03623*.

Bezawada, B., Bachani, M., Peterson, J., Shirazi, H., Ray, I., & Ray, I. (2018). Behavioral fingerprinting of IoT devices. In *Proceedings of the 2018 workshop on attacks and solutions in hardware security*, *ashes@ccs 2018, Toronto, on, Canada, October 19, 2018* (pp. 41–50). Retrieved from http://doi.acm.org/10.1145/3266444.3266452

Bratus, S., Cornelius, C., Kotz, D., & Peebles, D. (2008). Active behavioral fingerprinting of wireless devices. In *Proceedings of the first ACM conference on wireless network security, WISEC 2008, Alexandria, VA, USA, March 31–April 2, 2008* (pp. 56–61). Retrieved from http://doi.acm.org/10.1145/1352533.1352543

Brik, V., Banerjee, S., Gruteser, M., & Oh, S. (2008). Wireless device identification with radiometric signatures. In *Proceedings of the 14th annual international conference on mobile computing and networking, MOBICOM 2008, San Francisco, California, USA, September 14–19, 2008* (pp. 116–127). Retrieved from http://doi.acm.org/10.1145/1409944.1409959

Buttyán, L., Dóra, L., & Vajda, I. (2005). Statistical wormhole detection in sensor networks. In *Security and privacy in ad-hoc and sensor networks, second European workshop, ESAS 2005, Visegrad, Hungary, July 13–14, 2005, revised selected papers* (pp. 128–141). Retrieved from https://doi.org/10.1007/11601494_11

Chen, T. M. (2010). Stuxnet, the real start of cyber warfare? *IEEE Network*, *24*(6), 2–3. https://doi.org/10.1109/MNET.2010.5634434

Choong, D. L. C., Cho, C. Y., Tan, C. P., & Lee, R. S. (2008). Identifying unique devices through wireless fingerprinting. In *Proceedings of the first ACM conference on wireless network security, WISEC 2008, Alexandria, VA, USA, March 31–April 2, 2008* (pp. 46–55). Retrieved from http://doi.acm.org/10.1145/1352533.1352542

Corbett, C. L., Beyah, R. A., & Copeland, J. A. (2008a). Passive classification of wireless NICs during active scanning. *International Journal of Information Security*, *7*(5), 335–348. https://doi.org/10.1007/s10207-007-0053-7

Corbett, C. L., Beyah, R. A., & Copeland, J. A. (2008b). Passive classification of wireless nics during rate switching. *EURASIP Journal on Wireless Communications and Networking*, *2008*, 32. https://doi.org/10.1155/2008/495070

Dabbagh, Y. S., & Saad, W. (2018). Authentication of everything in the internet of things: Learning and environmental effects. *arXiv preprint arXiv:1805.00969*.

Dalai, A. K., & Jena, S. K. (2017). Wdtf: A technique for wireless device type fingerprinting. *Wireless Personal Communications*, *97*(2), 1911–1928.

Dalvi, N. N., Domingos, P. M., Mausam, Sanghai, S. K., & Verma, D. (2004). Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on knowledge discovery and data mining, Seattle, Washington, USA, August* 22–25, 2004 (pp. 99–108). Retrieved from http://doi.acm.org/10.1145/1014052.1014066

Danev, B., & Capkun, S. (2009). Transient-based identification of wireless sensor nodes. In *Proceedings of the 8th international conference on information processing in sensor networks, IPSN 2009, April 13–16, 2009, San Francisco, California, USA* (pp. 25–36). Retrieved from http://doi.acm.org/10.1145/1602165.1602170

Danev, B., Zanetti, D., & Capkun, S. (2012). On physical-layer identification of wireless devices. *ACM Computing Surveys*, *45*(1), 6:1–6:29. https://doi.org/10.1145/2379776.2379782

Douceur, J. R. (2002). The sybil attack. In *Peer-to-peer systems, first international workshop, IPTPS 2002, Cambridge, MA, USA, March August 7, 2002, revised papers* (pp. 251–260). Retrieved from https://doi.org/10.1007/3-540-45748-8_24

Formby, D., Srinivasan, P., Leonard, A., Rogers, J., & Beyah, R. A. (2016). Who's in control of your control system? Device fingerprinting for cyber-physical systems. In *23rd annual network and distributed system security symposium, NDSS 2016, San Diego, California, USA, February 21–24, 2016*. Retrieved from http://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2017/09/who-control-your-control-system-device-fingerprinting-cyber-physical-systems.pdf

François, J., Abdelnur, H. J., State, R., & Festor, O. (2009). Automated behavioral fingerprinting. In *Proceeding of the 12th raid symposium* (pp. 182–201).

François, J., Abdelnur, H. J., State, R., & Festor, O. (2010). Machine learning techniques for passive network inventory. *IEEE Trans. Network and Service Management*, *7*(4), 244–257.

Franklin, J., & McCoy, D. (2006). Passive data link layer 802.11 wireless device driver fingerprinting. In *Proceedings of the 15th USENIX security symposium, Vancouver, BC, Canada, July 31–August 4, 2006*. Retrieved from https://www.usenix.org/conference/15th-usenix-security-symposium/passive-data-link-layer-80211-wireless-device-driver

Gao, K., Corbett, C. L., & Beyah, R. A. (2010). A passive approach to wireless device fingerprinting. In *Proceedings of the 2010 IEEE/IFIP international conference on dependable systems and networks, DSN 2010, Chicago, IL, USA, June 28–July 1, 2010* (pp. 383–392). Retrieved from https://doi.org/10.1109/DSN.2010.5544294

Greenough, J. (2016). *How the ?Internet of things? Will impact consumers, businesses, and governments in 2016 and beyond. Retrieved from* http://www.businessinsider.com/how-the-internet-of-things-market-will-grow-2014-10?r=DE&IR=T

Hafeez, I., Antikainen, M., Ding, A. Y., & Tarkoma, S. (2018). IoT-keeper: Securing IoT communications in edge networks. *arXiv preprint arXiv: 1810.08415*.

Hall, J., Barbeau, M., & Kranakis, E. (2006). Detecting rogue devices in bluetooth networks using radio frequency fingerprinting. In *Proceedings of the third IASTED international conference on communications and computer networks, Lima, Peru, October June 4, 2006*, (pp. 108–113).

Jain, A. K., Duin, R. P. W., & Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(1), 4–37. https://doi.org/10.1109/34.824819

Jana, S., & Kasera, S. K. (2010). On fast and accurate detection of unauthorized wireless access points using clock skews. *IEEE Transactions on Mobile Computing*, *9*(3), 449–462. https://doi.org/10.1109/TMC.2009.145

Kailath, T. (1967). The divergence and bhattacharyya distance measures in signal selection. *IEEE Transactions on Communication Technology*, *15* (1), 52–60.

Kohno, T., Broido, A., & Claffy, K. C. (2005). Remote physical device fingerprinting. In *2005 IEEE symposium on security and privacy (S&P 2005), Oakland, CA, USA, 8–11 May 2005* (pp. 211–225). Retrieved from https://doi.org/10.1109/SP.2005.18

Krebs, B. (2017, November). *Mirai IoT botnet co-authors plead guilty? Krebs on security*. Retrieved from https://krebsonsecurity.com/tag/mirai-botnet/

Lippmann, R., Fried, D., Piwowarski, K., & Streilein, W. (2003). Passive operating system identification from TCP/IP packet headers. In *Workshop on data mining for computer security* (p. 40).

Martinez, B., Monton, M., Vilajosana, I., & Prades, J. D. (2015). The power of models: Modeling power consumption for iot devices. *IEEE Sensors Journal*, *15*(10), 5777–5789.

Meidan, Y., Bohadana, M., Shabtai, A., Guarnizo, J. D., Ochoa, M., Tippenhauer, N. O., & Elovici, Y. (2017). 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).

Miettinen, M., Marchal, S., Hafeez, I., Asokan, N., Sadeghi, A., & Tarkoma, S. (2017). Iot SENTINEL: automated device-type identification for security enforcement in IoT. In *37th IEEE international conference on distributed computing systems, ICDCS 2017, Atlanta, GA USA, June August 5, 2017* (pp. 2177–2184). Retrieved from https://doi.org/10.1109/ICDCS.2017.283

Moon, S. B., Skelly, P., & Towsley, D. F. (1999). Estimation and removal of clock skew from network delay measurements. In *Proceedings IEEE INFOCOM '99, the conference on computer communications, eighteenth annual joint conference of the IEEE computer and communications societies, the future is now, New York, NY, USA, March 21–25, 1999* (pp. 227–234). Retrieved from https://doi.org/10.1109/INFCOM.1999.749287

Morin, E., Maman, M., Guizzetti, R., & Duda, A. (2017). Comparison of the device lifetime in wireless networks for the internet of things. *IEEE Access*, *5*, 7097–7114.

Oser, P., Kargl, F., & Lüders, S. (2018). Identifying devices of the internet of things using machine learning on clock characteristics. In *International conference on security, privacy and anonymity in computation, communication and storage* (pp. 417–427).

Radhakrishnan, S. V., Uluagac, A. S., & Beyah, R. A. (2015). GTID: A technique for physical device and device type fingerprinting. *IEEE Transactions on Dependable and Secure Computing*, *12*(5), 519–532. https://doi.org/10.1109/TDSC.2014.2369033

Roux, J., Alata, E., Auriol, G., Nicomette, V., & Kaâniche, M. (2017). Toward an intrusion detection approach for IoT based on radio communications profiling. In *2017 13th European dependable computing conference (EDCC)* (pp. 147–150).

Senrio. (2016). *400,000 publicly available IoT devices vulnerable to single flaw*. Retrieved from http://blog.senr.io/blog/400000-publicly-available-iot-devices-vulnerable-to-single-flaw

Shahid, M. R., Blanc, G., Zhang, Z., & Debar, H. (2018). Iot devices recognition through network traffic analysis. In *2018 IEEE international conference on big data (big data)* (pp. 5187–5192).

Siby, S., Maiti, R. R., & Tippenhauer, N. (2017). Iotscanner: Detecting and classifying privacy threats in IoT neighborhoods. *arXiv preprint arXiv: 1701.05007*.

Sivanathan, A., Gharakheili, H. H., Loi, F., Radford, A., Wijenayake, C., Vishwanath, A., & Sivaraman, V. (2018). Classifying IoT devices in smart environments using network traffic characteristics. *IEEE Transactions on Mobile Computing*, *18*, 1745–1759.

Sivanathan, A., Sherratt, D., Gharakheili, H. H., Radford, A., Wijenayake, C., Vishwanath, A., & Sivaraman, V. (2017). Characterizing and classifying IoT traffic in smart cities and campuses. In *2017 IEEE conference on computer communications workshops (Infocom Wkshps)* (pp. 559–564). Washington, DC: IEEE.

Thangavelu, V., Divakaran, D. M., Sairam, R., Bhunia, S. S., & Gurusamy, M. (2018). *Deft: A distributed iot fingerprinting technique*. *IEEE Internet of Things Journal*. Washington, DC: IEEE.

Tugnait, J. K., & Kim, H. (2010). A channel-based hypothesis testing approach to enhance user authentication in wireless networks. In *Second international conference on communication systems and networks, COMSNETS 2010, Bangalore, India, January September 5, 2010* (pp. 1–9). Retrieved from https://doi.org/10.1109/COMSNETS.2010.5432018

Xu, Q., Zheng, R., Saad, W., & Han, Z. (2016). Device fingerprinting in wireless networks: Challenges and opportunities. *IEEE Communications Surveys and Tutorials*, *18*(1), 94–104. https://doi.org/10.1109/COMST.2015.2476338

Yang, K., Li, Q., & Sun, L. (2019). Towards automatic fingerprinting of iot devices in the cyberspace. *Computer Networks*, *148*, 318–327.

Yu, P. L., & Sadler, B. M. (2011). MIMO authentication via deliberate fingerprinting at the physical layer. *IEEE Transactions on Information Forensics and Security*, *6*(3–1), 606–615. https://doi.org/10.1109/TIFS.2011.2134850

How to cite this article: Bezawada B, Ray I, Ray I. Behavioral fingerprinting of Internet-of-Things devices. *WIREs Data Mining Knowl Discov*. 2021;11:e1337. https://doi.org/10.1002/widm.1337