Position paper: A systematic framework for categorising IoT device fingerprinting mechanisms

Poonam Yadav
University of York, UK

Angelo Feraudo
University of York, UK

Budi Arief
University of Kent, UK

Siamak F. Shahandashti
University of York, UK

Vassilios G. Vassilakis
University of York, UK

ABSTRACT

The popularity of the Internet of Things (IoT) devices makes it increasingly important to be able to fingerprint them, for example in order to detect if there are misbehaving or even malicious IoT devices in one's network. However, there are many challenges faced in the task of fingerprinting IoT devices, mainly due to the huge variety of the devices involved. At the same time, the task can potentially be improved by applying machine learning techniques for better accuracy and efficiency. The aim of this paper is to provide a systematic categorisation of machine learning augmented techniques that can be used for fingerprinting IoT devices. This can serve as a baseline for comparing various IoT fingerprinting mechanisms, so that network administrators can choose one or more mechanisms that are appropriate for monitoring and maintaining their network. We carried out an extensive literature review of existing papers on fingerprinting IoT devices – paying close attention to those with machine learning features. This is followed by an extraction of important and comparable features among the mechanisms outlined in those papers. As a result, we came up with a key set of terminologies that are relevant both in the fingerprinting context and in the IoT domain. This enabled us to construct a framework called ***IDWork***, which can be used for categorising existing IoT fingerprinting mechanisms in a way that will facilitate a coherent and fair comparison of these mechanisms. We found that the majority of the IoT fingerprinting mechanisms take a passive approach – mainly through network sniffing – instead of being intrusive and interactive with the device of interest. Additionally, a significant number of the surveyed mechanisms employ both static and dynamic approaches, in order to benefit from complementary features that can be more robust against certain attacks such as spoofing and replay attacks.

CCS CONCEPTS

• **Security and privacy → Biometrics**; • **Networks →** *Layering*; • **Computer systems organization →** *Sensor networks*; • **Computing methodologies → Machine learning approaches**; Modeling methodologies; • **General and reference →** *Measurement*.

KEYWORDS

Internet of Things (IoT), Fingerprinting, Machine Learning, Survey, Framework, Device Identification, Network Traffic Analysis.

ACM Reference Format:
Poonam Yadav, Angelo Feraudo, Budi Arief, Siamak F. Shahandashti, and Vassilios G. Vassilakis. 2020. Position paper: A systematic framework for categorising IoT device fingerprinting mechanisms. In *The 2nd International Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things (AIChallengeIoT '20), November 16–19, 2020, Virtual Event, Japan.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3417313.3429384

1 INTRODUCTION

Device fingerprinting is a process of identifying a device or device type based on its unique intrinsic or behavioural properties [18, 21, 50]. Device fingerprinting is very popular in internet-connected general purpose computing devices to track user behaviour and application usage. Some of the interesting applications include browser fingerprinting for web analytics, user tracking, fraud detection and accountability [9, 23, 27] and have gained a significant interests from cyber-security community. While it is clear that device fingerprinting can bring a lot of benefits – especially for providing automated and customisable user experience – there are also concerns that it can pose security and privacy risks [29, 39]. On the other hand, it has been suggested that device fingerprinting can also be used to help improve the security of smart home automation [24].

There are three main properties that need to be satisfied in order to achieve effective fingerprinting of devices [50]:

- *unique identity property*: the device fingerprint has to be unique for the device;
- *integrity property*: the fingerprints must be impossible – or at least, difficult – to forge;
- *reproducibility property*: the features used in the fingerprinting process must be stable, especially in the presence of environment changes and mobility.

The increased prevalence of Internet of Things (IoT) devices makes the task of fingerprinting devices more challenging. To start with, there is a great variety of IoT devices available on the market, and there are many different ways for them to operate. These pose a challenge in creating a generic mechanism that can perform accurate fingerprinting of all IoT devices. Furthermore, there are some fundamental differences between IoT devices and general computers (for which, more mature fingerprinting mechanisms have been developed). For example, IoT devices do not have many standard browser-based applications, which means many standard

AIChallengeIoT '20, November 16–19, 2020, Virtual Event, Japan

ACM ISBN 978-1-4503-8134-5/20/11…$15.00
https://doi.org/10.1145/3417313.3429384

fingerprinting mechanisms will not work for IoT devices. Moreover, many IoT devices do not have a standard Graphical User Interface (GUI) and they might even work autonomously in pervasive environments without user's direct control.

Due to resource constraints and insecure designs, IoT devices are prone to be involved in cyber-attacks, ranging from being the target [22, 45] to being exploited to create a botnet to mount a massive Distributed Denial of Service attack [3, 16]. Therefore, it is necessary and desirable to be able to automatically detect whether certain IoT devices might be vulnerable or could be exploited in cyber-attacks. The automatic device identification is one of the core requirements for building a secure IoT ecosystem, including cyber-attack and anomaly detection systems and automatic management and control.

Various device fingerprinting mechanisms have been proposed in the last few years. However, not all of these mechanisms are suitable for the IoT domain. Many IoT device fingerprinting mechanisms are only suitable for specific use cases or tailored to certain requirements, making it challenging to choose a correct fingerprinting mechanism that will be appropriate for a new use case, for example. This shortcoming is the motivation behind our paper. In this paper, we explore and collate existing IoT fingerprinting mechanisms – especially those that leverage Machine Learning (ML) techniques – and present a holistic view and terminologies used in the fingerprinting context, which can be used for further research and development.

Contributions. The key contributions of our paper are:

- The compilation of a key set of fingerprinting terminologies.
- The identification of important features to be included for achieving effective and accurate fingerprinting of IoT devices.
- The construction of ***IDWork***: a systematic framework for categorising IoT device fingerprinting mechanisms, which can be used for comparing and selecting suitable fingerprinting mechanism(s) for an IoT application.

The rest of the paper is organised as follows. Section 2 provides some background and related work, while Section 3 outlines the methodology we followed in our research. Section 4 represents the core of our work, giving an overview of the ***IDWork*** framework, along with the key terminologies and our results. Finally, Section 5 concludes our paper and provides some ideas for future work.

2 RELATED WORK

Several papers have discussed various fingerprinting mechanisms, although they are not necessarily dedicated to IoT devices [15, 49, 50]. Cunche et al. [10] looked into device fingerprinting based on monitoring wireless probes that a device may make, based on the preferred network (access points) stored on that device. The main concern of the paper was privacy infringement, for example by exploiting information contained in the fingerprint to infer social links between device owners. Spooren et al. [46] provides a critical assessment of device fingerprinting for risk-based authentication. In particular, they pointed out that device fingerprinting carries a lot of similarity among mobile devices, making this approach less reliable for risk assessment and step-up authentication.

Ferrag et al. [14] looked at human physiological and behavioural features in their investigation into factors that might hinder biometrics models' development and deployment on a large scale for authenticating IoT devices. They classified related survey papers based on deployment scope, focus biometric area, threat models, countermeasures, as well as the ML algorithms and Data Mining methods used by existing authentication and authorisation schemes for IoT devices. The paper also listed a set of biofeatures that can be used for biometric authentication of IoT devices, including gaze gestures, electrocardiogram, keystroke dynamics, fingerprint, ear shape and hand geometry [14]. They focus only on biofeatures, so other traits (such as network characteristics and device information) are not considered.

Skowron et al. [45] study the information leakage exposed by traffic fingerprinting attacks. They use features of statistical network flows and ML. Hence, this approach is effective even when the IoT traffic is encrypted. This approach relies on decision trees (CART classifier) and heuristically creates random forests out of 100 trees. The proposed approach aims at both device identification as well as the detection of anomalous user activities. The latter is based on features such as packet size, packet inter-arrival time, and transmission rate.

Hamad et al. [21] perform IoT device identification using traffic characteristics (network flows), based on real-time devices connected to an IP network. A passive behavioural fingerprinting approach is used, while device classification is based on features extracted from both packet header and payload. These include IP address, packet size, and other traffic related features. The authors investigated different supervised learning classifiers such as ABOOST, LDA, KNN, Decision Tree, Naive Bayesian, and SVM Random Forest (RF), with RF showing the best performance.

3 METHODS

Between March and June 2020, we carried out a systematic review of relevant papers that have been published on various venues, including USENIX, ACM, IEEE, Nature, ScienceDirect, MDPI, Springer, Elsevier, and Hindawi. We also utilised Google Scholar for comparison and for augmenting our search. The following keywords were used to gather the initial set of papers: "IoT", "fingerprinting", "device identification", "device authentication", "device authorisation", "traffic filtering", "anomaly detection".

We then divided up these papers among ourselves in order to analyse them and to extract key features of each paper. In order to be able to compare these papers fairly and consistently, we constructed a framework we call ***IDWork***, as outlined in Section 4.2. At various stages, we also performed synchronisation checks among ourselves to make sure the process is robust and consistent.

Further papers were added to the list between July and September 2020, mostly as a result of following up cited papers by those in the initial main list. The same analysis and extraction process using ***IDWork*** were applied to these additional papers to ensure consistency.

Our process came up with a final list of 31 articles from the 142 papers that we analysed carefully. The papers in our final list are shown in Table 1, which is constructed by populating a table as a result of applying our ***IDWork*** framework.

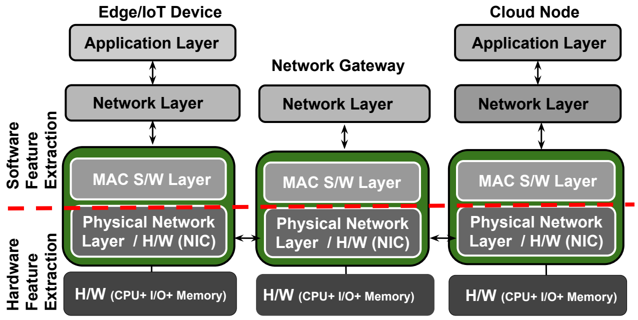

Figure 1: Edge/IoT device network end-to-end components

4 OVERVIEW OF IDWORK

For building ***IDWork***, we systematically reviewed the literature and recent state-of-the-art work to understand different terminologies used in IoT fingerprinting mechanisms. We simplified the presentation of IoT end-to-end ecosystem, as shown in Figure 1. The Edge/IoT device – showing the TCP/IP networking stack – is partitioned horizontally. The upper partition represents *software feature extraction*, which is composed of application, networking and Medium Access (MAC) layers. The lower partition represents *hardware feature extraction* comprising of two sub-layers - the first layer (upper) represents the features extracted from the Physical Network Layer along with the firmware and hardware (Networking Interface responsible for Link/Physical layer communication) layers and the second layer (lower) represents the hardware features which explicitly use the hardware device. Similarly, same layer partition is performed at Network Gateway and Cloud Node.

4.1 Fingerprinting terminology

In this section, we explain the key terminology that is important to grasp, before we define the ***IDWork*** framework.

Passive vs Active Fingerprinting: In passive fingerprinting, we collect information produced by a device and create an identification pattern by only observing the data coming from the device, i.e., no interaction with the device is carried out. In active fingerprinting, we instigate the target device to produce useful information, e.g. by making the device emit particular signals (at the physical-layer), or by producing packets which require a specific response from the device (at the network-layer). Thus, the difference between the two methods is that the former uses a sniffer to capture and analyse traffic, but it does not send traffic to the target [52], while the latter sends queries to the target and analyses the response.

Static Feature vs Dynamic Feature Fingerprinting: A static feature fingerprinting includes only features that do not usually change over the time (e.g. MAC address), while dynamic feature fingerprinting uses dynamic features that can change over time such as inter-arrival time associated with data flow.

Adaptive vs Fixed Fingerprinting Algorithms: An "adaptive" fingerprinting approach uses an algorithm that changes in response to certain conditions observed during the fingerprinting process. On the other hand, when the fingerprinting process always uses the same (deterministic) algorithm with pre-determined and constant parameters for all cases observed, we can consider that as "fixed".

Hardware Feature vs Software Feature Fingerprinting: The former approach uses features that are extracted using special Physical Unclonable Function (PUF) circuits to capture hardware-intrinsic properties. The latter approach uses behavioural software properties, which could be found in the network traffic generated by the IoT device.

Rule-based vs ML-based Fingerprinting: In rule-based approaches, the fingerprinting criteria are mathematically-formalised in the form of *if-then-else* rules. Such rules are often defined by thresholds and used to create fingerprints by capturing the correlation between the features/parameters. In ML-based fingerprinting, an ML model is created using the input features/parameters, and trained on the data to learn possible data correlations for generalisation.

Device vs Network vs Cloud Level: Fingerprinting approaches can act on different levels. Usually in the case of device-level fingerprinting, the approach generates a device signature which relies on its hardware characteristics, e.g. Radio Signal or clock skew. When the approach analyses network traffic – i.e. there is an additional entity within the network that monitor the traffic to produce device pattern – we refer to it as network-level fingerprinting. It is even possible that fingerprinting procedures are applied externally to a network so that they can be executed on multiple networks. We refer to this case as cloud-level figerprinting.

White-box vs Black-box Fingerprinting: White-box fingerprinting is possible when we can directly access a device's firmware source code and then build a dynamic model of that device [28]. Black-box fingerprinting exploits the interaction between different layers (e.g. application layer and transport layer) to build devices' fingerprints.

Unique Device Identification vs Type Identification vs Class Identification: Fingerprinting approaches can have different outputs depending on the designer's goal. In particular they can produce: a unique device identifier, device model or device class (devices with similar properties).

Supervised vs Unsupervised ML-based Fingerprinting: Supervised learning involves labeled data, which means that a prior knowledge about the classification of the learning data is provided. Conversely, unsupervised learning involves unlabeled data, so the ML goal is to infer a suitable classification of the data involved as well as classifying the data.

Radio vs MAC vs Network vs Application Layer: Radio fingerprinting exploits the unique characteristics in the radio signal emitted by a device. MAC fingerprinting exploits the characteristics of the data frames produced by a device (e.g. probe request in Wi-Fi). In network fingerprinting, the network packet parameters are used to build an identification pattern. Application fingerprinting approaches typically gather information to find out the device's services and operating system.

Open-world vs Close-world Evaluation: Open-world refers to any approach that is able to identify IoT devices within a larger set of devices not only restricted to IoT devices. Closed-world is when identification is evaluated on data that is restricted to only IoT devices.

Network Packet vs Flow-based Features: A fingerprinting approach that relies on network traffic can use packet-based or flow-based features, or both. Packet-based features use the content of individual packet payloads and headers, whereas flow-based features are based on temporal features of multiple packets coming from the same device, e.g. packet flow direction, inter-arrival time and inter-packet length [33].

Network Packet Header vs Deep Packet Based Features: When fingerprinting involves packet payload we refer to it as using deep-packet features. Otherwise, when only packet header parameters are used to build an identification pattern, we refer to it as using packet header features.

Encrypted vs Unencrypted Network Traffic: Some fingerprinting approaches do not need access to the packet payload, i.e. they can work on both encrypted and unencrypted packets. Conversely, others are designed to work on encrypted and unencrypted packet payloads, such as the algorithm proposed by Robyns et al. [40] which exploits per-bit entropy analysis (MAC address randomization). Furthermore, some approaches are able to extract the features required only if the payload is not encrypted.

Real Devices vs Simulated: Fingerprinting approaches based on deep-learning algorithms require a large amount of data to properly identify devices. Moreover, budget constraints do not allow for a large number of devices to produce an exhaustive dataset for evaluation purposes. Thus, tools to simulate IoT devices are used (e.g. Node-RED [25]). In this case, traffic flows and most of the important features of typical categories of IoT devices – such as fridges or washing machines – can be simulated and used to build datasets. On the other hand, if real devices are used, the results will be more representative of real-world scenarios, but the budgetary requirements are higher as well.

Testbed vs Real World Evaluation: Fingerprinting approaches are developed either on testbeds or in real-world environments. The latter approaches provide additional resilience and deployment credibility for the fingerprinting algorithm.

4.2 IDWork Framework

The construction of ***IDWork*** started with an understanding of the basic fingerprinting creation and verification workflow, as shown in Figure 2. The fingerprinting process consists of three steps: (1) Fingerprint template creation and storage, (2) Live fingerprint creation, and (3) Fingerprint verification. We analysed different considerations under each step. The task of initial template creation is a one-time process. The live fingerprint creation process may or may not follow the same approach or steps; however, the general approach follows these two sub-steps every time: feature extraction from the raw input features, and fingerprint generation. In the fingerprint template creation, the final step involving template storage and access mechanism is essential. In our work, we have considered different options under each sub-step (as shown in Figure 2), that differentiates one fingerprinting mechanism from another. Some of these options are either implementation-related or real-time deployment-related and they need to be considered, analysed and accessed before deciding an appropriate fingerprinting process suitable for an IoT application. Some of the important options we explored are:

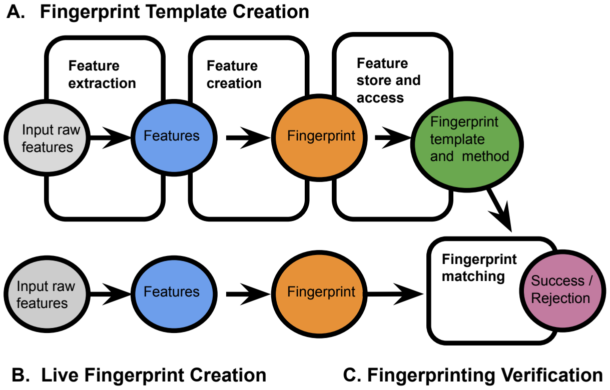

Figure 2: IoT device fingerprinting workflow.

- Does fingerprint feature extraction, or creation mechanism need device access to gather fingerprinting raw input?
- Does fingerprint feature extraction, or creation mechanism need invasive mechanisms?
- Does fingerprint feature extraction, or creation mechanism need additional hardware?
- What is the fingerprint feature extraction and creation, or what is the storage location in the IoT end-to-end system, for example, on the device, at a network gateway or on a cloud server?
- What are the security, integrity, anti-tempering considerations when storing and accessing the pre-created fingerprints?
- What is the computational complexity of individual steps?

After careful consideration, we picked seven labels categorised under three broad steps: *fingerprinting methods*, *fingerprinting input features* and *fingerprinting output*. We studied two categorisations under *fingerprinting methods*, namely Passive (P)/Active (A) fingerprinting and Static (S)/Dynamic (D) fingerprinting. Under *fingerprinting input features*, we considered categorisation using TCP/IP networking stack label and we used three choices: MAC/Network/Application Layers under this label. We further analysed these layers with two sub-labels with these choices: Packet/Dataframe/Flow and Header/Payload. We also explored the mechanisms which use these sub-labels for fingerprint creation, and we

Table 1: The classification of existing IoT fingerprinting mechanisms based on seven key categories

Passive/ Active (Cat 1)	Static/ Dynamic (Cat 2)	MAC/ Network/ Application (Cat 3)	Packet/ DataFrame/ Flow (Cat 4)	Header/ Payload (Cat 5)	Rule-based/ ML-based (Cat 6)	Class/ Type/ Unique (Cat 7)	Papers	With respect to the first two main features
P	S	N	P	H	RB	C+T	[19]	Passive and Static
P	S	N	P	H	ML+RB	C	[11][1]	
P	S	N	P+F	H	RB	T	[41]	
P	S	N	F	H	ML	C	[2]	
P	S	M+N	P	H+P	ML	C+T	[6]	
P	D	N	F	-	ML	C+T	[17]	Passive and Dynamic
P	D	N	P+F	H+P	ML	C	[47]	
P	D	M	DF	H+P	RB	U	[43]	
P	D	M+N	P	H+P	ML	C[2]	[48]	
P	D	M+N+A	P+F	H+P	ML+RB	U	[44]	
A	S	N	P	H+P	ML	C+T	[52]	Active and Static
A	D	M	DF	P	ML	-	[7]	Active and Dynamic
A	D	A	P	H+P	RB	C+T	[13]	
P	S+D	N	P	P	ML	T	[4]	Employing a combination of approaches
P	S+D	N	P	H+P	ML	C	[35]	
P	S+D	N	F	H	ML	U	[32]	
P	S+D	N	P	H	ML	C	[5]	
P	S+D	N	P+F	P	ML	C	[36]	
P	S+D	N	F	P	ML	U	[26]	
P	S+D	N	P	H+P	RB	C+T	[20]	
P	S+D	N	P	H	ML	U	[8]	
P	S+D	N	P+F	H	ML	C	[34]	
P	S+D	N	P+F	H	ML	-	[37]	
P	S+D	N	P+F	H+P	ML	C	[30]	
P	S+D	N	P+F	H+P	ML	C	[12]	
P	S+D	M+N+A	P+F	H+P	ML	C	[1]	
P	S+D	N	P+F	H+P	ML	C	[42]	
P	S+D	N	P+F	H+P	RB	C	[51]	
A	S+D	-	-	-	RB	C	[31][3]	
P+A	S+D	N	P[4]	H+P	ML	C	[29]	
P+A	D	N	F	H	ML	U	[38]	

categorise them as Rule-based or ML-based. We classify *fingerprint output* into three categories: Class, Type and Unique.

As a summary, in this work we present seven important features (as shown in Table 1), which broadly allow for a systematic and consistent way for classifying IoT fingerprinting mechanisms into logical categories. In total, there are potentially 432 exhaustive combinations, but certain combinations are more prevalent than others. We perceive these popular categories as the more promising approaches that one should take in their effort to achieve a meaningful fingerprinting exercise of IoT devices.

5 CONCLUSION

IoT fingerprinting has become an important research area, due to the increased prevalence of IoT devices. Fingerprinting mechanisms serve as a key component in a network administrator's effort to identify and categorise IoT devices, in order to be able to observe and manage IoT devices in their network properly, especially in relation to pinpointing potential causes of security problems.

There are many existing IoT fingerprinting mechanisms available, but it is not easy to choose a suitable mechanism for one's network, because there is currently no consistent framework for analysing these mechanisms. This is a gap that our paper aims to address. Firstly, we compiled a list of key terminologies that are essential in understanding and analysing IoT fingerprinting mechanisms. From there, we carefully constructed a framework called ***IDWork***, which provides a frame of reference for a fair and consistent comparison of these mechanisms. And finally, we demonstrated the usefulness of our framework by populating a table with

[1] They use an ML algorithm only for evaluation purposes

[2] Their mechanism can also identify unique device events

[3] A Physical Unclonable Function (PUF) is being used rather than any network traffic based approaches

[4] They also use Flow in the shape of inter-arrival time

some example mechanisms. We mainly focused on the mechanisms that use Machine Learning techniques. However, there are several mechanisms employing Rule-based techniques that are worth mentioning.

There are several key insights that came up from our research. We found that the majority of existing IoT fingerprinting mechanisms use a passive fingerprinting approach. This means a less intrusive approach is generally favoured. Furthermore, a dynamic approach – or a combination of both static and dynamic approaches – is very popular, quite possibly due to the need to fulfill a liveness property to minimise the risk of stale data or replay attacks. On the other hand, the least common mechanism seems to be based on a combination of active and static approach. This could be because such a combination might be more prone to a spoofing attack.

While we endeavoured to be as thorough and exhaustive as we can in our research, we are aware that there are some limitations in our work. For example, there are seven categories that we mainly consider in our framework, as presented in Table 1. However, it is possible that there are other categories that need to be considered in more detail. Furthermore, our current classification is mostly based on the software-related features of the IoT fingerprinting mechanisms. It would be more complete if hardware-related features – in particular, leveraging the Physical Unclonable Function (PUF) features – are also considered.

An interesting direction for future work is to look at the distribution and impact of IoT fingerprinting mechanisms and see if any significant patterns emerge in terms of mechanisms that are more popular or more effective, and how such patterns change over time.

Further work will also need to be carried out to validate our framework. This could be done by utilising publicly available datasets (provided by some IoT fingerprinting mechanisms) in an experiment to classify real-world IoT devices in a live setting. This will allow for the accuracy of existing mechanisms to be calculated, enabling a fairer comparison of these mechanisms to be performed. Achieving this will enable system administrators to justify their choices with regard to which IoT fingerprinting mechanism(s) they would like to employ in their network.

ACKNOWLEDGEMENT

Yadav and Feraudo are supported in part by "Data Negotiability in Multi-Mode Communication Networks" project funded by EPSRC grant EP/R045178/1.

REFERENCES

[1] A. Aksoy and M. H. Gunes. 2019. Automated IoT Device Identification using Network Traffic. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*. 1–7.

[2] S. Aneja, N. Aneja, and M. S. Islam. 2018. IoT Device Fingerprint using Deep Learning. In *2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS)*. 174–179.

[3] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J. Alex Halderman, Luca Invernizzi, Michalis Kallitsis, Deepak Kumar, Chaz Lever, Zane Ma, Joshua Mason, Damian Menscher, Chad Seaman, Nick Sullivan, Kurt Thomas, and Yi Zhou. 2017. Understanding the Mirai Botnet. In *26th USENIX Security Symposium, USENIX Security 2017, Vancouver, BC, Canada, August 16-18, 2017.* 1093–1110. https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis

[4] Noah Apthorpe, Dillon Reisman, and Nick Feamster. 2017. Closing the blinds: Four strategies for protecting smart home privacy from network observers. *arXiv preprint arXiv:1705.06809* (2017).

[5] L. Bai, L. Yao, S. S. Kanhere, X. Wang, and Z. Yang. 2018. Automatic Device Classification from Network Traffic Streams of Internet of Things. In *2018 IEEE 43rd Conference on Local Computer Networks (LCN)*. 1–9.

[6] Bruhadeshwar Bezawada, Maalvika Bachani, Jordan Peterson, Hossein Shirazi, Indrakshi Ray, and Indrajit Ray. 2018. Behavioral Fingerprinting of IoT Devices. In *Proceedings of the 2018 Workshop on Attacks and Solutions in Hardware Security (ASHES '18)*. Association for Computing Machinery, New York, NY, USA, 41–50. https://doi.org/10.1145/3266444.3266452

[7] Sergey Bratus, Cory Cornelius, David Kotz, and Daniel Peebles. 2008. Active behavioral fingerprinting of wireless devices. In *Proceedings of the first ACM conference on Wireless network security*. 56–61.

[8] Anat Bremler-Barr, Haim Levy, and Zohar Yakhini. 2019. IoT or NoT: Identifying IoT Devices in a ShortTime Scale. *arXiv preprint arXiv:1910.05647* (2019).

[9] A. Crabtree, T. Lodge, and J. et al. Colley. 2018. Building accountability into the Internet of Things: the IoT Databox model. *Journal of Reliable Intelligent Environments* 4 (2018), 39–55.

[10] Mathieu Cunche, Mohamed-Ali Kaafar, and Roksana Boreli. 2014. Linking wireless devices using information contained in Wi-Fi probe requests. *Pervasive and Mobile Computing* 11 (2014), 56–69.

[11] Bharat Atul Desai, Dinil Mon Divakaran, Ido Nevat, Gareth W Peter, and Mohan Gurusamy. 2019. A feature-ranking framework for IoT device classification. In *2019 11th International Conference on Communication Systems & Networks (COMSNETS)*. IEEE, 64–71.

[12] B. A. Desai, D. M. Divakaran, I. Nevat, G. W. Peter, and M. Gurusamy. 2019. A feature-ranking framework for IoT device classification. In *2019 11th International Conference on Communication Systems Networks (COMSNETS)*. 64–71.

[13] Xuan Feng, Qiang Li, Haining Wang, and Limin Sun. 2018. Acquisitional Rule-Based Engine for Discovering Internet-of-Thing Devices. In *Proceedings of the 27th USENIX Conference on Security Symposium (SEC'18)*. USENIX Association, USA, 327–341.

[14] Mohamed Amine Ferrag, Leandros Maglaras, and Abdelouahid Derhab. 2019. Authentication and authorization for mobile IoT devices using biofeatures: Recent advances and future trends. *Security and Communication Networks* 2019 (2019).

[15] Jason Franklin, Damon McCoy, Parisa Tabriz, Vicentiu Neagoe, J Van Randwyk, and Douglas Sicker. 2006. Passive Data Link Layer 802.11 Wireless Device Driver Fingerprinting.. In *USENIX Security Symposium*, Vol. 3. 16–89.

[16] Josh Fruhlinger. [n.d.]. The Mirai botnet explained: How teen scammers and CCTV cameras almost brought down the internet. https://www.csoonline.com/article/3258748/the-mirai-botnet-explained-how-teen-scammers-and-cctv-cameras-almost-brought-down-the-internet.html.

[17] Tianbo Gu and Prasant Mohapatra. 2018. Bf-iot: Securing the iot networks via fingerprinting-based device authentication. In *2018 IEEE 15Th international conference on mobile ad hoc and sensor systems (MASS)*. IEEE, 254–262.

[18] Xiaodan Gu, Ming Yang, Yiting Zhang, Peilong Pan, and Zhen Ling. 2018. Fingerprinting Network Entities Based on Traffic Analysis in High-Speed Network Environment. *Security and Communication Networks* 2018 (2018).

[19] Hang Guo and John Heidemann. 2018. Ip-based iot device detection. In *Proceedings of the 2018 Workshop on IoT Security and Privacy*. 36–42.

[20] I. Hafeez, M. Antikainen, A. Y. Ding, and S. Tarkoma. 2020. IoT-KEEPER: Detecting Malicious IoT Network Activity Using Online Traffic Analysis at the Edge. *IEEE Transactions on Network and Service Management* 17, 1 (2020), 45–59.

[21] Salma Abdalla Hamad, Wei Emma Zhang, Quan Z Sheng, and Surya Nepal. 2019. IoT Device Identification via Network-Flow Based Fingerprinting and Learning. In *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE, 103–111.

[22] Ali Hariri, Nicolas Giannelos, and Budi Arief. 2019. Selective Forwarding Attack on IoT Home Security Kits. In *Computer Security*. Springer, 360–373.

[23] Thomas Hupperich, Davide Maiorca, Marc Kührer, Thorsten Holz, and Giorgio Giacinto. 2015. On the Robustness of Mobile Device Fingerprinting: Can Mobile Users Escape Modern Web-Tracking Mechanisms?. In *Proceedings of the 31st Annual Computer Security Applications Conference (ACSAC 2015)*. Association for Computing Machinery, New York, NY, USA, 191–200. https://doi.org/10.1145/2818000.2818032

[24] Arun Cyril Jose, Reza Malekian, and Ning Ye. 2016. Improving home automation security; integrating device fingerprinting into smart home. *IEEE Access* 4 (2016), 5776–5787.

[25] Nickolaos Koroniotis, Nour Moustafa, Elena Sitnikova, and Benjamin Turnbull. 2019. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. *Future Generation Computer Systems* 100 (2019), 779–796.

[26] Jaidip Kotak and Yuval Elovici. 2020. IoT Device Identification Using Deep Learning. In *13th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2020)*. Springer International Publishing, New York, NY, USA, 76–86.

[27] Pierre Laperdrix, Nataliia Bielova, Benoit Baudry, and Gildas Avoine. 2020. Browser Fingerprinting: A Survey. *ACM Trans. Web* 14, 2, Article 8 (April 2020), 33 pages. https://doi.org/10.1145/3386040

[28] Q. Li, X. Feng, R. Wang, Z. Li, and L. Sun. 2018. Towards Fine-grained Fingerprinting of Firmware in Online Embedded Devices. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*. 2537–2545.

[29] Antonio Mangino, Morteza Safaei Pour, and Elias Bou-Harb. [n.d.]. Internet-scale Insecurity of Consumer Internet of Things: An Empirical Measurements Perspective. *ACM Transactions on Management Information Systems (TMIS)* ([n. d.]).

[30] Samuel Marchal, Markus Miettinen, Thien Duc Nguyen, Ahmad-Reza Sadeghi, and N Asokan. 2019. AuDI: Toward Autonomous IoT Device-Type Identification Using Periodic Communication. *IEEE Journal on Selected Areas in Communications* 37, 6 (2019), 1402–1412.

[31] Cédric Marchand, Lilian Bossuet, Ugo Mureddu, Nathalie Bochard, Abdelkarim Cherkaoui, and Viktor Fischer. 2017. Implementation and characterization of a physical unclonable function for IoT: a case study with the TERO-PUF. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 37, 1 (2017), 97–109.

[32] Yair Meidan, Michael Bohadana, Asaf Shabtai, Martin Ochoa, Nils Ole Tippenhauer, Juan Davis Guarnizo, and Yuval Elovici. 2017. Detection of Unauthorized IoT Devices Using Machine Learning Techniques. arXiv:cs.CR/1709.04647

[33] N. Moustafa and J. Slay. 2015. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *2015 Military Communications and Information Systems Conference (MilCIS)*. 1–6.

[34] Nizar Msadek, Ridha Soua, and Thomas Engel. 2019. IoT Device Fingerprinting: Machine Learning based Encrypted Traffic Analysis. In *2019 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 1–8.

[35] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A. Sadeghi. 2019. DÏoT: A Federated Self-learning Anomaly Detection System for IoT. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. 756–767.

[36] Jorge Ortiz, Catherine Crawford, and Franck Le. 2019. DeviceMien: Network Device Behavior Modeling for Identifying Unknown IoT Devices. In *Proceedings of the International Conference on Internet of Things Design and Implementation (IoTDI '19)*. Association for Computing Machinery, New York, NY, USA, 106–117. https://doi.org/10.1145/3302505.3310073

[37] Gaetano Pellegrino, Qin Lin, Christian Hammerschmidt, and Sicco Verwer. 2017. Learning behavioral fingerprints from netflows using timed automata. In *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 308–316.

[38] S. V. Radhakrishnan, A. S. Uluagac, and R. Beyah. 2015. GTID: A Technique for Physical Device and Device Type Fingerprinting. *IEEE Transactions on Dependable and Secure Computing* 12, 5 (2015), 519–532.

[39] Jingjing Ren, Daniel J Dubois, David Choffnes, Anna Maria Mandalari, Roman Kolcun, and Hamed Haddadi. 2019. Information exposure from consumer iot devices: A multidimensional, network-informed measurement approach. In *Proceedings of the Internet Measurement Conference*. 267–279.

[40] Pieter Robyns, Bram Bonné, Peter Quax, and Wim Lamotte. 2017. Noncooperative 802.11 mac layer fingerprinting and tracking of mobile devices. *Security and Communication Networks* 2017 (2017).

[41] Said Jawad Saidi, Anna Maria Mandalari, Roman Kolcun, Hamed Haddadi, Daniel J. Dubois, David Choffnes, Georgios Smaragdakis, and Anja Feldmann. 2020. A Haystack Full of Needles: Scalable Detection of IoT Devices in the Wild. In *Internet Measurement Conference (IMC '20)*. ACM.

[42] Ola Salman, Imad H. Elhajj, Ali Chehab, and Ayman Kayssi. [n.d.]. A machine learning based framework for IoT device identification and abnormal traffic detection. *Transactions on Emerging Telecommunications Technologies* n/a, n/a ([n. d.]), e3743. https://doi.org/10.1002/ett.3743 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/ett.3743

[43] Sandra Siby, Rajib Ranjan Maiti, and Nils Ole Tippenhauer. 2017. IoTScanner: Detecting Privacy Threats in IoT Neighborhoods. In *Proceedings of the 3rd ACM International Workshop on IoT Privacy, Trust, and Security (IoTPTS '17)*. Association for Computing Machinery, New York, NY, USA, 23–30. https://doi.org/10.1145/3055245.3055253

[44] Arunan Sivanathan, Daniel Sherratt, Hassan Habibi Gharakheili, Adam Radford, Chamith Wijenayake, Arun Vishwanath, and Vijay Sivaraman. 2017. Characterizing and classifying IoT traffic in smart cities and campuses. In *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 559–564.

[45] Monika Skowron, Artur Janicki, and Wojciech Mazurczyk. 2020. Traffic Fingerprinting Attacks on Internet of Things Using Machine Learning. *IEEE Access* 8 (2020), 20386–20400.

[46] Jan Spooren, Davy Preuveneers, and Wouter Joosen. 2015. Mobile device fingerprinting considered harmful for risk-based authentication. In *Proceedings of the Eighth European Workshop on System Security*. 1–6.

[47] Vijayanand Thangavelu, Dinil Mon Divakaran, Rishi Sairam, Suman Sankar Bhunia, and Mohan Gurusamy. 2018. Deft: A distributed iot fingerprinting technique. *IEEE Internet of Things Journal* 6, 1 (2018), 940–952.

[48] Rahmadi Trimananda, Janus Varmarken, Athina Markopoulou, and Brian Demsky. 2019. PingPong: Packet-Level Signatures for Smart Home Device Events. *arXiv preprint arXiv:1907.11797* (2019).

[49] S. Wang, J. Wang, and Z. Yu. 2018. Privacy-Preserving Authentication in Wireless IoT: Applications, Approaches, and Challenges. *IEEE Wireless Communications* 25, 6 (2018), 60–67.

[50] Qiang Xu, Rong Zheng, Walid Saad, and Zhu Han. 2015. Device fingerprinting in wireless networks: Challenges and opportunities. *IEEE Communications Surveys & Tutorials* 18, 1 (2015), 94–104.

[51] Poonam Yadav, Qi Li, Richard Mortier, and Anthony Brown. 2019. Network Service Dependencies in Commodity Internet-of-things Devices. In *Proceedings of the International Conference on Internet of Things Design and Implementation (IoTDI '19)*. ACM, New York, NY, USA, 202–212. https://doi.org/10.1145/3302505.3310082

[52] Kai Yang, Qiang Li, and Limin Sun. 2019. Towards automatic fingerprinting of IoT devices in the cyberspace. *Computer Networks* 148 (2019), 318–327.