Enhancing IoT Privacy: Why DNS-over-HTTPS Alone Falls Short?

Samuel Pélissier, Gianluca Anselmi, Abhishek Kumar Mishra, Anna Maria Mandalari, Mathieu Cunche

Enhancing IoT Privacy: Why DNS-over-HTTPS Alone Falls Short?

Samuel Pélissier
INSA-Lyon, Inria
University of Lyon, CITI Lab.
Lyon, France

Gianluca Anselmi
University College London
London, United Kingdom

Abhishek Kumar Mishra
INSA-Lyon, Inria
University of Lyon, CITI Lab.
Lyon, France

Anna Maria Mandalari
University College London
London, United Kingdom

Mathieu Cunche
INSA-Lyon, Inria
University of Lyon, CITI Lab.
Lyon, France

***Abstract*—Recent years have seen widespread adoption of consumer Internet of Things (IoT) devices, offering diverse benefits to end-users, from smart homes to healthcare monitoring, but raising serious privacy concerns. To address this, securing efforts, such as encrypting DNS, have been proposed.**

In this paper, we study the effectiveness of such measures in the specific context of ensuring IoT privacy. We introduce a device identification attack against DNS-over-HTTPS-enabled IoT devices. We conduct more than 25,000 automated experiments across 6 public DNS resolvers and find that the proposed attack can identify devices via DNS-over-HTTPS (DoH) traffic with a 0.98 balanced accuracy. We point out padding as a mitigation technique that reduces identification by a significant 33%. Additionally, we find that half of the evaluated DNS resolvers do not adhere to the relevant specification, substantially compromising user privacy.

***Index Terms*—IoT, Privacy, DNS, DoH, Device Identification**

I. Introduction

In the domestic environment, the proliferation and diversity of consumer Internet of Things (IoT) devices have surged, embedding "smart" appliances into various aspects of daily life, including entertainment, home automation, and healthcare. Despite their convenience, these devices are notorious for their poor security, posing significant risks to users and the broader infrastructure [1]. One of the primary privacy concerns is the potential for personal information to be inferred from passive observation of network traffic, leading to the identification, tracking, and profiling of users' activities, preferences, and behaviors [2]–[4]. This information leakage not only expands the attack surface but also increases the likelihood of security breaches or malicious activities [1].

Encrypted traffic for IoT devices is not enough to ensure privacy [5]. Despite encryption, recent research has explored traffic-based IoT device identification through various network activities, with DNS traffic emerging as a particularly effective method to identify devices quickly and accurately based on a limited number of DNS interactions [6]–[8].

There have been broad efforts to advocate for encrypted DNS in IoT environments to enhance privacy [9]–[12]. DNS-over-HTTPS (DoH) has been introduced as a measure to protect the confidentiality of DNS activities by encrypting DNS queries, potentially disrupting traditional DNS-based identification techniques [13].

However, in this paper, we demonstrate that DoH alone is insufficient for safeguarding user privacy in the context of IoT. To illustrate potential privacy breaches, we propose and execute a device identification attack against DoH-enabled IoT devices. We reveal that associated metadata can still yield an accuracy of 0.98 in identifying IoT devices, unlike other works that utilize clear text DNS names [8] or multiple network layers [14], [15]. Through the collection and analysis of a diverse DNS dataset from numerous real IoT devices, we show that even with DoH, a high degree of device identification accuracy (0.93 balanced accuracy) can be achieved using only two messages. These findings underscore that while encrypted DNS is a critical step toward securing IoT environments, it is not sufficient on its own.

The contributions of this paper are three-fold:

1) For IoT devices using DoH, we propose and execute a successful device identification attack that achieves high accuracy at scale using just the first few seconds of traffic.
2) We assess padding as a potential countermeasure and show that it significantly reduces device identification.
3) Finally, we expose a significant privacy breach, finding that some resolvers do not follow padding specifications. This vulnerability can be generalized to other technologies using misconfigured resolvers (e.g., web browsers) and can go undetected by users.

The remainder of the paper is structured as follows. First, we introduce the required background (Section II) and formulate our threat model (Section III). Next, we present the IoT testbed used to collect our dataset (Section IV) before detailing the identification attack (Section V). Then, we detail our evaluation methodology (Section VI) and demonstrate that DoH alone is not enough to protect IoT privacy (Section VII). We proceed to propose methods for improving DoH protection (Section VII-D). Finally, we discuss how these enhancements

could be implemented and the privacy implications of our findings (Section VIII).

The code of our tool and the data collected in our experiments are publicly available at: https://github.com/SafeNetIoT/doh_iot.

II. Background

In this section, we provide a brief overview of DNS for IoT devices and DNS-over-HTTPS.

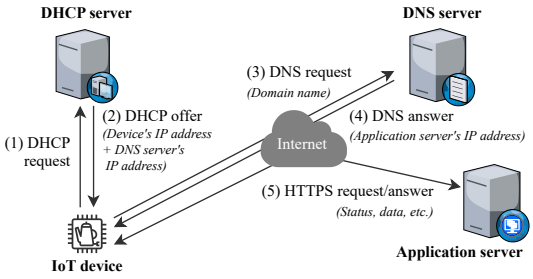

Fig. 1. Example consumer IoT setup architecture.

A. DNS in consumer-grade IoT

Whether for data transmission or for fetching software updates, consumer IoT devices require communicating with remote hosts to operate. These exchanges are performed over IP, and the hosts are generally identified by a domain name. One of the first actions of a newly powered on IoT device is thus to use DNS to retrieve the IP address of a remote host. As seen in Figure 1, in a typical domestic environment, the device receives the DNS resolver address from a DHCP server operating on the local network.[1] Then, the device requests IP addresses of remote application servers by sending their *domain name* to the DNS resolver. IoT devices typically send multiple DNS requests on the first minute of DNS traffic following their first connection to fetch new updates and report their states to the various application servers [7]. Hence, focusing on the first few minutes of communication provides a good way of identifying devices.

B. DNS-over-HTTPS

In DNS, requests and responses are transmitted in clear-text, exposing host names and IP addresses to any eavesdropper on the path. In an effort to protect users' privacy, the confidentiality of DNS messages' content can be secured using an encryption layer. Two main approaches have been proposed: DNS-over-HTTPS (DoH) [17] or DNS-over-TLS (DoT) [18], which respectively encapsulate DNS traffic in HTTPS and Transport Layer Security (TLS). From a privacy perspective, DoH is preferred over DoT, among other reasons because its features (in particular the port) are similar to generic HTTPS traffic, hiding DoH in a larger quantity of traffic [19].[2]

[1]In some rare cases, a DNS server is hard-coded in the device [16].

[2]In this paper, we present the investigation with DoH only due to space constraints, but we observe similar findings with DoT.

DoH is currently supported by several major DNS actors such as Cloudflare and Google, and implemented in all major web browsers. DoH maintains the request/response model of DNS but uses HTTP requests transmitted over a TLS-secured channel. Clients query DNS resolvers by opening an HTTPS session and transmitting DNS requests within HTTP POST or GET requests. For both the request and the response, the DNS payload is encoded in binary format as in the original DNS.

III. Threat model

To estimate the privacy protection offered by DoH, we consider an identification attack based on the threat model illustrated in Figure 2.

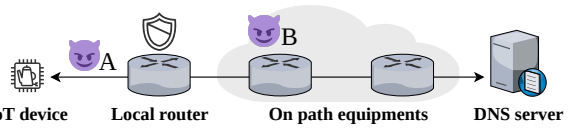

Fig. 2. Threat model, with an attacker A in the local network, and an external attacker B on path.

Target. The target is any user of a consumer IoT device communicating via DoH.

Adversary. The adversary is any party that can eavesdrop on the DoH traffic between the IoT device and the resolver. In practice, we consider two possible locations. First, an adversary A can monitor communications on the *local network*. For instance, eavesdroppers could have access to a malicious IoT device sniffing the surrounding traffic [20], or a compromised local router.

Second, we consider an adversary B outside the local network, *positioned between the local router and DNS resolver*, who tracks devices via uniquely assigned IPv6 addresses. This scenario is relevant and plausible for several reasons. The widespread adoption of IPv6 allows direct addressing of devices behind a router, enabling one-to-one mapping without relying on NAT [21]. To prevent tracking, both parts of the IPv6 address can be dynamically updated. However, this process typically occurs every 24 hours by default, which still allows an eavesdropper to track devices within this time frame [22]. Moreover, several studies have shown that correctly implementing IPv6 address rotation is complex, and some vulnerabilities may persist, leading to robust tracking through this protocol [21], [23].

Furthermore, we assume the adversary only leverages DoH traces. The rationale behind our approach is twofold: 1) we evaluate the privacy impact of DoH alone, demonstrating its sufficiency for conducting an accurate attack, and 2) other information such as MAC addresses may change over time [24], or may not be available based on the adversary position in the network.

Threat. The adversary is able to accurately identify which IoT devices are used in a local network, such as a household.

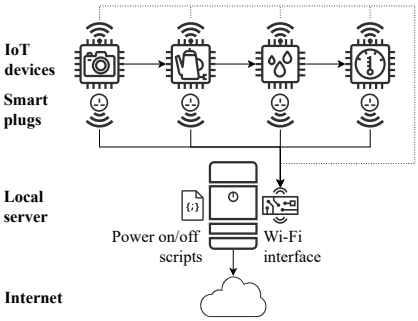

Fig. 3. Testbed overview.

IV. IoT Testbed

To capture DoH traffic from IoT devices, we use the testbed shown in Figure 3. It consists of: (*i*) a *server* providing IP connectivity to smart plugs and IoT devices, managing and recording all of their network activity, (*ii*) a total of 34 *IoT devices*, plugged into (*iii*) a set of *smart plugs*, and orchestrated by (*iv*) a set of *power on/off scripts* (see Section VI-A).

IoT Devices. We consider 34 consumer IoT devices that offer a large variety across several categories and are representative of a typical smart home network: Appliance (4), Baby Monitor (2), Camera (5), Doorbell (4), Hub (2), Light (6), Pet (2), Plug (1), Medical (1), Sensor (2) and Speaker (5). The complete list of IoT devices is available in Table I.

TABLE I
IoT devices present in the testbed.

Category	Device name (Device ID)
Appliance	Alexa Swan Kettle (1), Coffee Maker Lavazza (7), Cosori Air Fryer (8), Meross Garage Door (20)
Baby Monitor	Boifun Baby (5), VTech Baby Camera (30)
Camera	Arlo Camera Pro4 (3), Blink Mini Camera (4), Google Nest Camera (13), SimpliCam (27), Wyze Cam Pan v2 (33)
Doorbell	Eufy Chime (11), Google Nest Doorbell (14), Reolink Doorbell (24), Ring Chime Pro (25)
Hub	Aqara HubM2 (2), Google Nest Hub (15)
Light	Govee Strip Light (16), Lepro Bulb (18), Lifx Mini (19), NanoLeaf Triangles (21), Wiz Bulb (32), Yeelight Bulb (34)
Medical	Withings Sleep Analyser (31)
Pet	Furbo Dog Camera (12), Petsafe Feeder (23à
Plug	Tapo Plug (29)
Sensor	Netatmo Weather Station (22), Sensibo Sky Sensor (26)
Speaker	Bose Speaker (6), Echodot4 (9), Echodot5 (10), Homepod (17), Sonos Speaker (28)

DNS resolvers. While DoH is standardized, small differences can be observed on the wire depending on the resolver [13]. To diversify our approach and avoid relying solely on a single implementation, we select 6 public DNS resolvers based on their popularity due to their longevity [13] and, for some, their default integration in Firefox and Chromium: Google, Cloudflare, Quad9, CleanBrowsing, NextDNS, and AdGuard.

V. DoH-based identification attack

To handle the high volume of DoH requests in our experiments, we employ machine learning for device identification. We begin by selecting effective features, showcase their extraction, and then discuss our selected models.

A. Features selection

Encrypting traffic with DoH reduces available data, but does not completely hide the `length` nor the time between two DNS requests (inter-arrival times, `IAT`). Our intuition is that such information varies per device and is discriminative enough to identify them. This differs from prior methods using clear-text DNS messages to access complete domain names directly [7], [8].

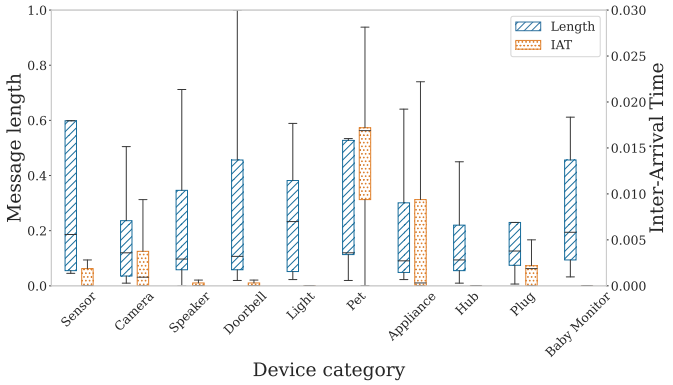

Fig. 4. Discriminative behavior of `length` and `IAT`.

For all devices in our dataset, we normalize message `length` and `IAT` of DNS messages and study their respective distribution. When looking at the size of messages in Figure 4, there is significant variance across device categories. For instance, plugs and hubs have smaller message `length` than sensors. Similarly, appliances show higher `IAT` values compared to speakers and doorbells. `IAT` values remain low for most device categories, hinting at relatively less effectiveness for identification than message `length`. Hence, we select both the `length` and `IAT` of DNS messages as features.

B. Features extraction

For each power cycle, we collect device-specific DoH traffic metadata. More precisely, we extract TLS Application Data `length`[3], and the `IAT` between DNS requests.

`IAT` and `length` raw features are saved as vectors alongside the corresponding descriptive statistics: mean, variance, standard deviation, skewness, and kurtosis. These synthetic representations have proven effective in forming fingerprints of network traffic [6], [25]. Such a broad approach enables the machine learning model to automatically select the most pertinent data representation.

[3]We disregard generic TCP handshake and TLS negotiation, focusing on the request with the domain name and the response from the DNS resolver.

C. Model selection

The goal of the paper is not to propose a new machine-learning architecture for device identification, but rather to show that DoH is not perfect for IoT privacy protection.

We consider various well-known multi-class machine learning methods: Neural Network[4], Random Forest, K-Nearest Neighbors, Complement Naive Bayes, Logistic Regression, C-Support Vector (SVC): linear, one-vs-one, one-vs-the-rest.

We first split the dataset into an 80:20 ratio between training and held-out data. Using Halving Random Search with cross-validation [26], we efficiently explore a grid of hyper-parameters and split the training data into 5 subsets of 80:20 proportion to prevent overfitting [27]. After finding the best hyperparameters based on the balanced accuracy, we select the best machine learning method, train the model using the initial 80% of the dataset, and validate it against the held-out 20%. Doing so avoids any test snooping [27] and shows the model correctly generalizes for unseen data.

VI. Evaluation Methodology

In this section, we present our data collection methodology as well as the experimentation details.

A. Dataset collection

1) Power experiments: To maximize the capture of DNS requests from IoT devices for our deployed testbed (see Section IV), we perform a power cycle during each experiment. A single run consists of 5 minutes of network traffic captured via the Mon(IoT)r tool [4], and 2 minutes of timeout to ensure proper device shutdown.

We repeat the process 50 times daily for 15 days to ensure coverage of diverse time periods for a longitudinal analysis. This extended duration generates a robust dataset of 25,500 on-off experiments.

2) Encrypting DNS traffic: IoT devices do not support DoH yet, requiring us to encrypt the clear-text DNS requests they send. One original clear-text DNS request generates multiple DoH requests, each for different resolvers and mitigation (cf. Section VII-D3). The high number of requests may prompt a resolver to block them if not slightly delayed. Such constraints make it difficult to introduce a proxy converting on-the-fly clear-text requests to DoH.

Instead, our solution involves two steps: capturing the clear-text traffic of IoT devices and replaying DNS requests using DoH from the same vantage point. To maintain consistency, the DNS requests are replayed at the same time they were originally sent, minimizing discrepancies like DNS cache state differences. Contrary to other steps in our pipeline, the replay is only done once due to its time-intensive nature.

B. Handling dataset imbalance

Each device has its communication pattern: some generate dozens of DNS requests every time they are switched on, while others may only send a few queries from time to time. To address dataset imbalance, we apply random oversampling on the training set, while keeping the evaluation set imbalanced to reflect real-world conditions. The performance metric is the *balanced* accuracy, calculating the average recall for each class, addressing dataset imbalance [27].

[4]We use the same layers as stated in previous works [7], only updating the last layer to match the output classes.

C. Experimental details

Unless specified otherwise, we average model performances across all DNS resolvers and a single day of replay. We manually confirm consistency across multiple days and choose a random day as the baseline.

The machine learning pipeline is then run 15 times, each with a different seed initializing the Pseudo-Random Numbers Generators (PRNGs).

VII. Results

In this section, we present the performance analysis of device identification over our 34 devices dataset. We start to examine the accuracy of the device identification attack. Afterward, we discuss and analyze several enhancements that potentially improve DoH's privacy protection capabilities.

A. Comparison of machine learning methods

Table II showcases the performance of all machine learning methods initially tested. More specifically, it reports the median value of averaged balanced accuracy over all cross-validations. We further note that the *same* features and datasets are used to compare each method. Random Forest yields the best results, reaching $\sim$0.97 balanced accuracy, well above values obtained by other methods, including a similar experimental setup by Thompson et al. based on a Neural Network [7] ($\sim$0.89 only). This aligns with previous empirical research, where Random Forest consistently outperformed other classifiers in extensive experiments with varied datasets [28]. Thus, we select the Random Forest method for the remaining results.

TABLE II
Performance of machine learning methods during Halving Random Search Cross-Validation.

Machine learning method	Balanced accuracy
Random Forest	0.9684
Neural Network [7]	0.8931
SVC (linear)	0.8861
Logistic Regression	0.8485
K-Nearest Neighbors	0.8462
SVC (one-vs-the-rest)	0.8433
SVC (one-vs-one)	0.8392
Complement Naive Bayes	0.5816

B. Device identification attack

Figure 5 shows the confusion matrix for all devices, with predictions on the y-axis and actual devices on the x-axis, normalized to correctly take into account the different number of occurrences for each device. Anything outside of the diagonal is a misclassification.

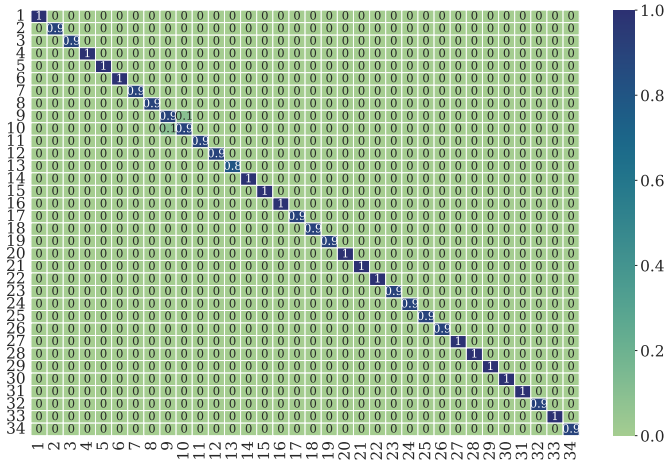

Fig. 5. Devices (corresponding device IDs) confusion matrix.

False positives are minimal, yielding a balanced accuracy of ∼0.97 across devices. Interestingly, three devices sold by the same company (Google's Nest Camera (13), Nest Doorbell (14), and Nest Hub (15)) are correctly classified. Likewise, the two Echo Dot devices (9) and (10) only differ in version (4 and 5) while showing distinguishable enough traffic.

As seen in Figure 6, one message is enough to identify the devices with a balanced accuracy of ∼0.92. We obtain similar results when looking at the time window of listening instead of the absolute number of DNS requests. We notice that just within 1 second after the switch-on, the same balanced accuracy is achieved, allowing attackers to promptly assess the targeted network's content.

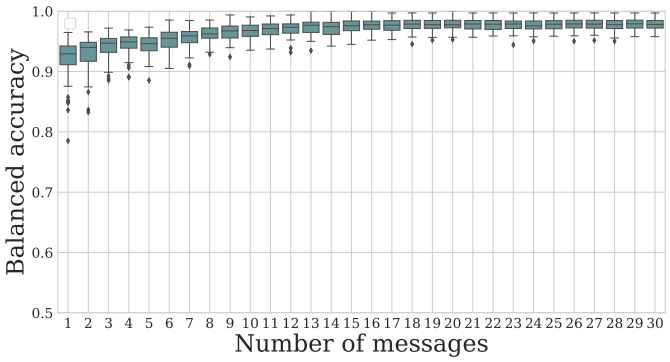

Fig. 6. BA vs. number of requests after power cycle.

The high balanced accuracy reached using just one DNS request questions the usefulness of the `IAT` (requiring at least 2 requests). In practice, we find that `length`-based features yield a 0.98 balanced accuracy when used alone versus 0.79 for `IAT`-based features across all resolvers. This can be explained by multiple factors. First, as seen in Section V-A, `IAT` values are generally more concentrated than `length`. Second, the number of DNS requests is low (a median of 5 per power cycle) which reduces the number of available `IAT` values. Third, the `IAT` is subject to random variations both in the time the DNS resolver takes to answer and the one a device requires to handle incoming information.

TABLE III
IMPACT OF THE RESOLVERS ON PERFORMANCE.

Resolver	Balanced accuracy
AdGuard	0.9820
Google	0.9788
Quad9	0.9781
Cloudflare	0.9778
NextDNS	0.9768
CleanBrowsing	0.9765

Finally, we investigate the identification performance with respect to various DNS resolvers, illustrated in Table III. There is no significant difference between DNS resolvers. All values are stable, and such a tendency is verified for all the other results. Without any other mitigation (see Section VII-D3), the identification attack works regardless of the resolver.

C. Attack persistence

A well-known phenomenon in machine learning is the degradation of the model over time. As features continuously drift away from their values at training, the initial model yields worse and worse results [29]. To evaluate the decrease of performance of the model, we test it against new, unseen data collected in the following days after training the model of reference. Figure 7 shows our model maintains a mean balanced accuracy of over ∼0.91 across all resolvers over 15 days, with day 0 used a reference.

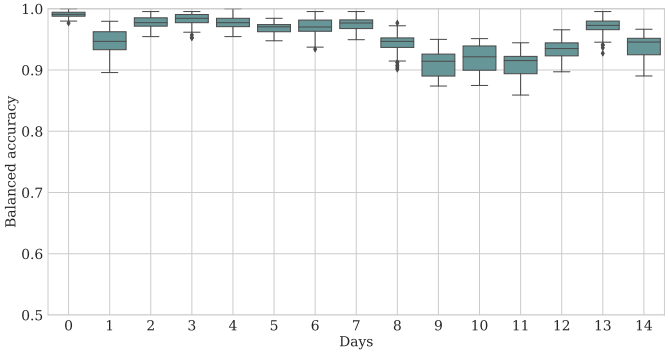

Fig. 7. Performance evolution over 15 days.

D. Improving DoH protection

The above results show that our device identification attack using DoH is potent and persistent, forcing us to examine additional privacy-protecting measures. However, the options to mitigate the attack are limited due to the low number of raw features (`IAT` and `length`).

1) Delaying messages: Reducing the relevance of `IAT` can be done by introducing delays between messages [30]. However, this approach is challenging for multiple reasons. First, the `IAT` is closely linked to the application logic. For

instance, a device might first contact an update server and then request the IP of an application server to upload data. The presence and timing of the second request depend on the first one. Second, IoT devices often rely on real-time applications that cannot tolerate delays. Deploying a network middleware to introduce delays could result in timeouts, potentially leading to denial-of-service.

Moreover, detecting delay-tolerant requests would require analyzing each underlying application, which is impractical. For these reasons, we do not further explore delaying messages as a privacy-preserving technique.

2) Merging DNS requests: Another way of impacting `IAT` and `length` is to send multiple DNS requests in one message. By default, DNS queries correspond to a single resource, creating a one-to-one relationship between domain names and observed packets. Merging all requests into a single packet could hide which exact domains are queried. While the original DNS RFC [31] suggested supporting multiple resources in one message, under-specification (e.g., handling partial resource availability) led to poor resolver support. Despite standardization attempts [32], [33], adoption was never achieved. Other ideas, like forwarding queries in a mesh network [34], also lack implementation. Therefore, merging queries is currently out of scope but might be worth exploring in future work.

3) Padded DNS: The DNS request `length` can be partially hidden by padding before encryption using the EDNS(0) Padding Option [35]. According to RFC 8467, clients should choose the "closest multiple of 128 octets" [36]. It also presents "Random-Block-Length Padding", randomly selecting a block length to pad with. Other solutions, like padding up to the maximal length or drawing from a known distribution, result in excessive overhead or are impractical in diverse IoT networks [30].

We implement two padding strategies by setting the EDNS(0) option in the DNS request and we repeat the methodology of replaying requests: padding to the closest 128-byte block, or the closest block between $[128, 256, 384, 512]$ bytes chosen randomly. We train the models of each resolver using only the `length` of messages and compare the results to models not using any padding. We also study a hypothetical perfect protection, where the `length` is not leaking any information, and an attacker can only leverage the `IAT`.

Figure 8 demonstrates significant reductions in balanced accuracy for Cloudflare, Google, and AdGuard with both strategies. For instance, using padding with Cloudflare results in a $\sim$33% decrease. Interestingly, Random-Block-Length Padding incurs a considerable overhead (125% additional bytes on the wire compared to 67% for 128-byte block padding) but does not improve results. A straightforward padding strategy achieves comparable levels to the ideal scenario where only the `IAT` is available.

Despite this, we note discrepancies between resolvers: Quad9, CleanBrowsing, and NextDNS show minimal impact from the padding. To investigate further, we train new models using only lengths from request messages (IoT devices to resolvers) and others from answer messages (resolvers to IoT devices). As anticipated, request messages are properly padded, resulting in a significant drop in balanced accuracy for all resolvers (from 0.97 to 0.64 on average). While RFC 7830 mandates padding DNS responses when the corresponding query includes the Padding option [35], compliance appears inconsistent across resolvers.

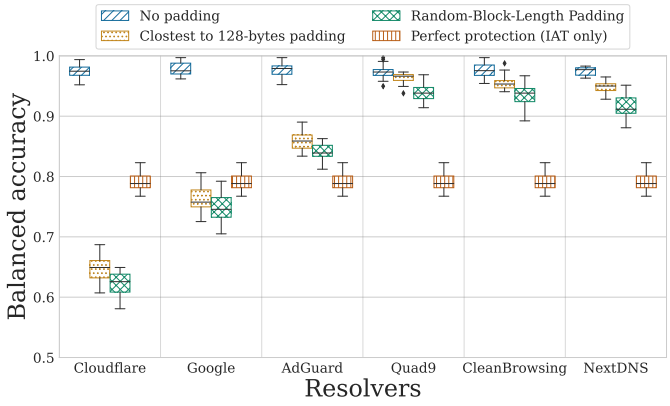

Fig. 8. Impact of padding strategies.

Upon analyzing the traffic captures, we discover the following padding behaviors: Cloudflare and Google pad up to a fixed value, respectively 707 encrypted bytes exactly and $\sim$850 encrypted bytes; AdGuard roughly matches the request padding, CleanBrowsing nearly never pads the answer, based on unknown heuristics; NextDNS and Quad9 send an empty ENDS(0) option containing no padding, virtually adding a few bytes but producing a similar length than the original. These findings align with Figure 8: models utilizing both request and answer can leverage unaffected answer information to correctly identify IoT devices.

VIII. Discussion

In this section, we examine how padding can be implemented according to our threat model and technical requirements. We also assess the impact of this approach on privacy and discuss its limitations.

A. Implementing padding mitigation

If an adversary gains direct access to the local network (e.g., via a compromised device or the router itself, as shown by adversary A in Figure 2), the only feasible solution is to deploy padded DoH directly on the IoT devices. However, implementing such countermeasures on a large scale within IoT devices presents significant challenges, and relying on widespread manufacturer updates is impractical.[5]

For an eavesdropper outside the local network (adversary B in Figure 2), we recommend deploying a middlebox at the router level to act as a local DNS resolver, relaying clear-text DNS requests as padded DoH. Similar to our experimental

[5]The challenge of updating existing IoT devices highlights the importance of implementing mitigations directly at the manufacturer level, anticipating new deployments, or upgrading existing devices.

setup, IoT devices are typically assigned a default DNS server via the *Domain Name Server* DHCP option [37]. For this approach to work, devices must accept DHCP-assigned DNS resolvers. In our experiments, 3-12% of devices did not support this[6], which aligns with values reported in previous studies [16]. If devices ignore DHCP-assigned DNS resolvers, a man-in-the-middle approach could be used to intercept clear-text DNS and forward them as padded DoH. This method could be implemented directly in routers, for example, through an extension of the open-source software OpenWRT [38].

B. Implications for privacy

While DoH alone is a starting point, it is ultimately insufficient to fully protect privacy as we obtain a 0.98 identification accuracy despite encryption. We also show that although mitigations are feasible for both manufacturers and end-users, their effectiveness depends on DNS clients and resolvers adhering to the standard, which is not currently the case.

Moreover, this issue impacts well-behaved clients that correctly send padded DoH requests and thus expect maximum protection. We again highlight that DNS answers do not show any form of warning when padding is asked for but ultimately missing. Detecting this misconfiguration requires detailed network trace analysis, or an automated client-side check of DNS options, as it cannot be easily identified otherwise.

As DoH is increasingly deployed in critical user-facing applications like web browsers and IoT devices, it is essential to ensure the highest level of privacy protection and provide clear mechanisms to inform users about any missing features.

C. Limitations and future works

Our dataset includes 34 devices, which is more than previous works [7], [16]. While reaching internet-scale in a lab is not feasible and performance might differ in larger-scale settings, our method adeptly distinguishes *similar* devices (see Section VII-B), implying robustness in broader deployments. Additionally, we [will] open source our code and dataset for other people to test their own setup.

Initial identification happens within minutes of device activation, leveraging increased request rate post-switch-on [7]. Early traffic ensures consistent device fingerprinting but long-term monitoring efficacy warrants further exploration.

Our threat model assumes that the DNS-over-HTTPS traffic is already identified among other encrypted communications. In practice, this traffic would be included in other HTTPS traffic. Nevertheless, DNS traffic can be easily identified thanks to the IP address of the destination (especially for popular resolvers) or by leveraging the specific features of messages [13]. While some IoT devices may bypass our attack by using hard-coded IP addresses, this approach is not recommended and unobserved in our dataset.

[6]We cannot confirm the resolver used by Google devices, as our DHCP server advertises 8.8.8.8 and 8.8.4.4 as DNS resolvers.

D. Responsible disclosure

The padding misconfiguration was disclosed to relevant DNS resolvers. Quad9 clarified that their front-end, dnsdist, currently does not support padding.[7]

IX. Related works

To the best of our knowledge, no other work has analyzed the potential of DoH in protecting IoT privacy. Table IV summarizes studies addressing generic privacy leakages in DNS, both through clear-text and DoH.

TABLE IV
SUMMARY OF RELATED WORKS ON DNS-BASED PRIVACY LEAKAGE.

Traffic type \ Protocol	Clear-text DNS	DoH
Web browsing	[39]	[30], [40]
IoT	[7], [8]	**Our work**

First, clear-text DNS itself offers no privacy by design. For instance, Guha and Francis [39] were able to identify and track users via geo-located IP addresses generated by their browsing habits. Focusing on IoT traffic, clear-text DNS has been shown to be exploitable for device identification, thus posing a risk to privacy. Perdisci *et al*. [8] used the query URLs of a device's DNS requests to reliably identify devices. Using the same feature, Thompson *et al.* [7] focused on the first 5 minutes after start-up, arguing that most communications occur during this time window. Unlike these works, DoH does not allow direct access to clear-text queries.

Second, while encrypting web traffic with DoH protects DNS content, it does not completely prevent attacks. Siby *et al.* [40] analyzed DoH traces and demonstrated that website identification is still possible based on packet length and direction. Their work was extended by Bushart and Rossow [30], who added timing to improve accuracy and discussed padding as a countermeasure, while intentionally ignoring direction. Our work combines both approaches, leveraging packet length and timing. Additionally, we find that the direction is relevant for IoT devices and discover that multiple DNS resolvers do not adhere to current specifications, due to this feature.

Multiple studies have shown that IoT traffic is distinguishable from web browsing, particularly based on communication timings and domain name lengths [8], [16]. With the recent push for encrypted DNS adoption in IoT networks [9]–[12], potential differences between traditional and IoT traffic prompt us to analyze the impact of DoH in this context.

Finally, numerous methods of IoT device identification based on other protocols have been studied recently. For instance, Meidan *et al*. [41] identified device brands and models with 99% accuracy using IP addresses and ports. Other works achieve similar results by leveraging features from various protocols [14], [15]. These studies demonstrate that DNS is not the sole source of privacy leakage in IoT networks. However, contrary to our work, they are either

[7]https://github.com/PowerDNS/pdns/issues/10018

confined to local networks, rely on volatile information such as IP addresses, or depend on *unencrypted* DNS.

X. Conclusion

DNS-over-HTTPS in consumer IoT devices has been rightly advocated to improve users' privacy. However, we introduce an attack that reliably identifies devices using only metadata, such as message length and timing.

Using only DoH traffic, our method reliably classifies IoT devices with a 0.98 balanced accuracy. It correctly identifies individual devices from the same manufacturer or even two devices using different versions, no matter the DNS resolver targeted. We propose to enhance DoH with padding which yields a $\sim$33% decrease in accuracy. We also reveal that not all DNS resolvers pad their answers, nullifying the mitigation and putting the users' privacy at risk. Through these findings, we stress that the privacy of IoT devices cannot be guaranteed by DoH alone, calling for additional measures.

To support further research, all software and data we produced as part of this work are publicly available at https://github.com/SafeNetIoT/doh_iot.

Acknowledgment

This work has been supported by the ANR-BMBF PIVOT project (ANR-20-CYAL-0002), H2020 SPARTA project and the INSA-Lyon SPIE ICS IoT Chair.

References

[1] D. Kumar, K. Shen, B. Case, D. Garg, G. Alperovich, D. Kuznetsov, R. Gupta, and Z. Durumeric. All Things Considered: An Analysis of {IoT} Devices on Home Networks. 2019.

[2] A. Acar, H. Fereidooni, T. Abera, A.K. Sikder, M. Miettinen, H. Aksu, M. Conti, A. Sadeghi, and S. Uluagac. Peek-a-boo: i see your smart home activities, even encrypted! In *WiSec*, 2020.

[3] R. Houser, Z. Li, C. Cotton, and H. Wang. An investigation on information leakage of DNS over TLS. In *CoNEXT*, 2019.

[4] J. Ren, D. Dubois, D. Choffnes, A. M. Mandalari, R. Kolcun, and H. Haddadi. Information exposure from consumer iot devices: A multidimensional, network-informed measurement approach. In *IMC*, 2019.

[5] Noah Apthorpe, Dillon Reisman, and Nick Feamster. A smart home is no castle: Privacy vulnerabilities of encrypted iot traffic. *arXiv preprint arXiv:1705.06805*, 2017.

[6] H. Aksu, A. S. Uluagac, and E. S. Bentley. Identification of Wearable Devices with Bluetooth. *IEEE T-SUSC*, 2021.

[7] O. Thompson, A. M. Mandalari, and H. Haddadi. Rapid IoT Device Identification at the Edge. In *Proceedings of the 2nd ACM International Workshop on Distributed Machine Learning*, 2021.

[8] R. Perdisci, T. Papastergiou, O. Alrawi, and M. Antonakakis. IoTFinder: Efficient Large-Scale Identification of IoT Devices via Passive DNS Traffic Analysis. In *2020 IEEE EuroS&P*, 2020.

[9] Noah Apthorpe, Dillon Reisman, and Nick Feamster. A Smart Home is No Castle: Privacy Vulnerabilities of Encrypted IoT Traffic, May 2017.

[10] C. Hesselman, M. Kaeo, L. Chapin, K. Claffy, M. Seiden, D. McPherson, D. Piscitello, A. McConachie, T. April, J. Latour, and R. Rasmussen. The DNS in IoT: Opportunities, Risks, and Challenges. *IEEE Internet Computing*, 2020.

[11] M. S. Lenders, C. Amsüss, C. Gündogan, M. Nawrocki, T. C. Schmidt, and M. Wählisch. Securing Name Resolution in the IoT: DNS over CoAP. *ACM on Networking*, (CoNEXT), 2023.

[12] Zhiwei Yan and Jong-Hyouk Lee. The road to DNS privacy. *Future Generation Computer Systems*, 112:604–611, November 2020.

[13] Levente Csikor, Himanshu Singh, Min Suk Kang, and Dinil Mon Divakaran. Privacy of DNS-over-HTTPS: Requiem for a Dream? In *EuroS&P 2021*, pages 252–271, Vienna, Austria, September 2021. IEEE.

[14] A. Aksoy and M. H. Gunes. Automated iot device identification using network traffic. In *2019 IEEE International Conference on Communications (ICC)*, 2019.

[15] J. Kotak and Y. Elovici. Iot device identification using deep learning. *CoRR*, abs/2002.11686, 2020.

[16] Y. Wan, K. Xu, F. Wang, and G. Xue. Characterizing and Mining Traffic Patterns of IoT Devices in Edge Networks. *IEEE Transactions on Network Science and Engineering*, 2021.

[17] P. E. Hoffman and P. McManus. DNS Queries over HTTPS (DoH). Request for Comments 8484, IETF, 2018.

[18] Z. Hu, L. Zhu, J. Heidemann, A. Mankin, D. Wessels, and P. E. Hoffman. Specification for DNS over Transport Layer Security (TLS). Request for Comments 7858, IETF, 2016.

[19] Dns over tls vs. dns over https. https://www.cloudflare.com/learning/dns/dns-over-tls/. Accessed: 2023-11-30.

[20] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A. Sadeghi, and S. Tarkoma. Iot sentinel: Automated device-type identification for security enforcement in iot. In *IEEE ICDCS*, 2017.

[21] Said Jawad Saidi, Oliver Gasser, and Georgios Smaragdakis. One Bad Apple Can Spoil Your IPv6 Privacy, March 2022.

[22] David Barrera, Glenn Wurster, and P C van Oorschot. Back to the Future: Revisiting IPv6 Privacy Extensions. 2011.

[23] Erik C. Rye, Robert Beverly, and kc claffy. Follow the Scent: Defeating IPv6 Prefix Rotation Privacy. In *Proceedings of the 21st ACM Internet Measurement Conference*, pages 739–752, November 2021.

[24] Ellis Fenske, Dane Brown, Jeremy Martin, Travis Mayberry, Peter Ryan, and Erik Rye. Three years later: A study of mac address randomization in mobile devices and when it succeeds. *Proceedings on PETs*, 2021.

[25] Randall W. Klein, Michael A. Temple, and Michael J. Mendenhall. Application of wavelet-based RF fingerprinting to enhance wireless network security. *Journal of Communications and Networks*, 2009.

[26] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization, 2018.

[27] D. Arp, E. Quiring, F. Pendlebury, A. Warnecke, F. Pierazzi, C. Wressnegger, L. Cavallaro, and K. Rieck. Dos and Don'ts of Machine Learning in Computer Security. In *31st USENIX Security Symposium*, 2022.

[28] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim. Do we need hundreds of classifiers to solve real world classification problems? *The journal of machine learning research*, 2014.

[29] Roman Kolcun, Diana Andreea Popescu, Vadim Safronov, Poonam Yadav, Anna Maria Mandalari, Richard Mortier, and Hamed Haddadi. Revisiting IoT Device Identification, July 2021.

[30] J. Bushart and C. Rossow. Padding ain't enough: Assessing the privacy guarantees of encrypted DNS. 2020.

[31] Mockapetris. Domain names - implementation and specification. Request for Comments RFC 1035, Internet Engineering Task Force, November 1987.

[32] Paul A. Vixie. Extensions to DNS (EDNS). Internet Draft draft-ietf-dnsind-edns-03, Internet Engineering Task Force, August 1998.

[33] Paul A. Vixie. Extensions to DNS (EDNS1). Internet Draft draft-ietf-dnsext-edns1-03, Internet Engineering Task Force, August 2002.

[34] Oscar Arana, Hector Benítez-Pérez, Javier Gomez, and Miguel Lopez-Guerrero. Never Query Alone: A distributed strategy to protect Internet users from DNS fingerprinting attacks. *Computer Networks*, 199:108445, November 2021.

[35] Alexander Mayrhofer. The EDNS(0) Padding Option. Request for Comments 7830, IETF, 2016.

[36] Alexander Mayrhofer. Padding Policies for Extension Mechanisms for DNS (EDNS(0)). Request for Comments 8467, IETF, 2018.

[37] Ralph Droms and Steve Alexander. DHCP Options and BOOTP Vendor Extensions. Request for Comments 2132, IETF, 1997.

[38] Openwrt. https://openwrt.org/docs/guide-user/services/dns/doh_dnsmasq_https-dns-proxy. Accessed: 2023-11-30.

[39] Saikat Guha and Paul Francis. Identity Trail: Covert Surveillance Using DNS. In Nikita Borisov and Philippe Golle, editors, *Privacy Enhancing Technologies*. Springer Berlin Heidelberg, 2007.

[40] S. Siby, M. Juarez, C. Diaz, N. Vallina-Rodriguez, and C. Troncoso. Encrypted DNS –> Privacy? A Traffic Analysis Perspective. *arXiv:1906.09682 [cs]*, 2019.

[41] Y. Meidan, M. Bohadana, A. Shabtai, J. D. Guarnizo, M. Ochoa, N. 0. Tippenhauer, and Y. Elovici. ProfilIoT: A machine learning approach for IoT device identification based on network traffic analysis. In *Proceedings of the Symposium on Applied Computing*, 2017.