A Haystack Full of Needles: Scalable Detection of IoT Devices in the Wild

Said Jawad Saidi
Max Planck Institute for Informatics

Anna Maria Mandalari
Imperial College London

Roman Kolcun
Imperial College London

Hamed Haddadi
Imperial College London

Daniel J. Dubois
Northeastern University

David Choffnes
Northeastern University

Georgios Smaragdakis
TU Berlin
Max Planck Institute for Informatics

Anja Feldmann
Max Planck Institute for Informatics/Saarland University

ABSTRACT

Consumer Internet of Things (IoT) devices are extremely popular, providing users with rich and diverse functionalities, from voice assistants to home appliances. These functionalities often come with significant privacy and security risks, with notable recent large-scale coordinated global attacks disrupting large service providers. Thus, an important first step to address these risks is to know *what* IoT devices are *where* in a network. While some limited solutions exist, a key question is whether device discovery can be done by Internet service providers that only see sampled flow statistics. In particular, it is challenging for an ISP to efficiently and effectively track and trace activity from IoT devices deployed by its millions of subscribers—all with sampled network data.

In this paper, we develop and evaluate a scalable methodology to accurately detect and monitor IoT devices at subscriber lines with limited, highly sampled data in-the-wild. Our findings indicate that millions of IoT devices are detectable and identifiable within hours, both at a major ISP as well as an IXP, using *passive*, sparsely *sampled* network flow headers. Our methodology is able to detect devices from more than 77% of the studied IoT manufacturers, including popular devices such as smart speakers. While our methodology is effective for providing network analytics, it also highlights significant privacy consequences.

CCS CONCEPTS

• **Security and privacy** → *Network security*; • **Networks** → *Network monitoring*; **Public Internet**; **Network measurement**.

KEYWORDS

Internet of Things, IoT detection, IoT secuirty and privacy, Internet Measurement

IMC '20, October 27–29, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-8138-3/20/10…$15.00
https://doi.org/10.1145/3419394.3423650

ACM Reference Format:
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 *ACM Internet Measurement Conference (IMC '20), October 27–29, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3419394.3423650

1 INTRODUCTION

The number of IoT devices deployed within homes is increasing rapidly. It is estimated that at the end of 2019, more than 9.5 billion IoT devices were active, and the IoT population will increase to 20 billion by 2025 [1]. Such devices include virtual assistants, smart home control, cameras, and smart TVs. While users deploy some IoT devices explicitly, they are often unaware of the security threats and privacy consequences of using such devices [2]. Major Internet Service Providers (ISPs) are developing strategies for dealing with the large-scale coordinated attacks from these devices.

Existing solutions focus on instrumenting testbeds or home environments to collect and analyze full packet captures [3–5], local search for IoT anomalies [6, 7], active measurements [8, 9], or data from antivirus companies running scan campaigns from users homes [7]. In isolation, these data sources do not provide enough insights for preventing network-wide attacks from IoT devices [10]. Detecting IoT devices from an ISP can help to identify suspicious traffic and what devices are common among the subscriber lines generating that traffic.

In this paper, we present a methodology for detecting home IoT devices in-the-wild at an ISP, and an Internet Exchange Point (IXP), by relying on passive, sampled network traces and active probing experiments. We build on the insight that IoT devices typically rely on backend infrastructure hosted on the cloud to offer their services. While contacting such infrastructure, they expose information, including their traffic destinations, even when a device is not in use [4]. One of the challenges of detecting IoT devices at scale is the *poor availability and low granularity* of data sources. The available data is often in the form of centrally-collected aggregate and sampled data (e.g., NetFlow [11], IPFIX traces [12]). Thus, we need a methodology that (a) does not rely on payload and (b) handles sparsely sampled data.

Another challenge is *traffic patterns diversity*, across IoT devices and their services.[1] We note that some devices, e.g., cameras, will generate significant continuous traffic; others, e.g., plugs, can be expected to be mainly passive unless used. Moreover, many devices offer the same service, e.g., the Alexa voice assistant [13] is available on several brands of smart speakers as well as on Amazon Fire TV devices. Here, the traffic patterns may depend on the service rather than the specific IoT device. Some services rely on dedicated backend infrastructures, while others may use shared ones, e.g., CDNs. Thus, we need a methodology that identifies which IoT services are detectable from the traffic and then identifies a unique traffic pattern for each IoT device and associated services.

Our key insight is that we can address these challenges by focusing our analysis only on the types of destinations contacted by IoT devices. Even with sparsely sampled data, the set of servers contacted by an IoT device over time can form a reasonably unique signature that is revealed in as little as a few hours. However, this approach has limitations, for example we cannot use it to detect devices or services that use a shared infrastructure with unrelated services (e.g., CDNs).

To understand the detectability of IoT devices in the above-mentioned environment, we focus on the possible communication patterns of end-user IoT services and the types of destinations they contact. Figure 1 shows three possible communication patterns on top of a typical network topology. This includes three households, an ISP, as well as a dedicated infrastructure and a CDN that hosts multiple servers. Device *A* is deployed by two subscribers, and only contacts one server in the dedicated infrastructure. Device *B* is deployed by a single subscriber and contacts both a dedicated server, as well as a CDN server. Device *C* is deployed by two subscribers and contacts only CDN servers. We observe that, using NetFlow traces at the ISP edge, it is possible to identify subscriber lines hosting devices of type *A* and *B*. Devices of type *C* are harder to detect given the sampling rates and header-only nature of NetFlow.

In this paper, we use a unique testbed and dataset to build a methodology for detecting and monitoring IoT devices at scale (see Figure 2). We first use controlled experiments, where we tunnel the traffic of two IoT testbeds with 96 IoT devices to an ISP. This provides us with ground truth IoT traffic within this ISP (Section 2). We confirm the visibility of the ground truth IoT traffic using the NetFlow ISP data (Section 3). Next, we identify backend infrastructures for many IoT services, from the observed ISP IoT traffic (Section 4). We augment this base information with data from DNS queries, web certificates, and banners. Next, we use the traffic signatures to identify broadband subscriber lines using IoT services at the ISP, as well as an IXP (Section 6). Finally, we discuss our results, their significance, and limitations in Section 7, related work (Section 8), and conclude with a summary in Section 9.

Our main contributions are as follows:

- We develop a methodology for identifying IoT devices, by classifying domains and IP addresses of the backend infrastructure. To this end we derive distinct signatures, in terms of IP/domain/port destinations, to recognize IoT devices. With our signatures we were able to recognize the presence of devices from 31 out of 40 manufacturers in our testbed.[2]
- We show that it is possible to detect the presence of IoT devices at subscriber lines, using sparsely sampled flow captures from a large residential ISP, and a major IXP, even if the device is idle, i.e., not in active use. Specifically, we were able to recognize that 20% of 15 million subscriber lines used at least one of the 56 different IoT products in our testbed.
- We highlight that our technique scales, is accurate, and can identify millions of IoT devices within minutes, in a non-intrusive way from passive, sampled data. In the case of the ISP, we were able to detect the presence of devices from 72% of our target manufacturers within 1 hour, sometimes minutes.

Based on our findings, we also discuss why some IoT devices are faster to detect, how to hide an IoT service, as well as how the detectability can be used to improve IoT services and network troubleshooting.

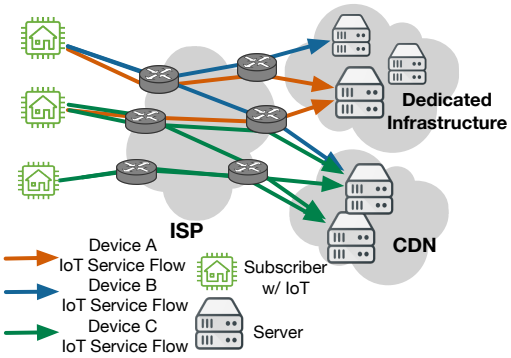


Figure 1: Simplified IoT communication patterns.

2 IOT – CONTROLLED EXPERIMENTS

We need ground truth traffic from IoT devices, as observed both in a testbed and in the wild, for developing and testing our methodology. In this section, we describe our data collection strategy (see point (1) of Figure 2).

2.1 Network Setting

We utilize two *vantage points*, namely a large European ISP, and a major European IXP.

ISP (ISP-VP). The ISP is a large residential ISP that offers Internet services to over 15 million broadband subscriber lines. The ISP uses NetFlow [11] to monitor the traffic flows at all border routers in its network, using a consistent sampling rate across all routers. Figure 3 shows where NetFlow data is collected.

IXP (IXP-VP). The IXP facilitates traffic exchange between its members. At this point, it has more than 800 members, including international, with peak traffic exceeding 8 Tbps. The IXP uses IPFIX [12] to collect traffic data across its switching fabric at a consistent sampling rate, which is an order of magnitude lower than the one used at the ISP. Figure 4 illustrates where the IPFIX data is collected.

[1] Here we refer to IoT services as the set of protocols and destinations that are part of the operations of an IoT device.

[2] To foster further research in the area of IoT privacy and security, we make all the signatures available at https://moniotrlab.ccis.neu.edu/imc20/

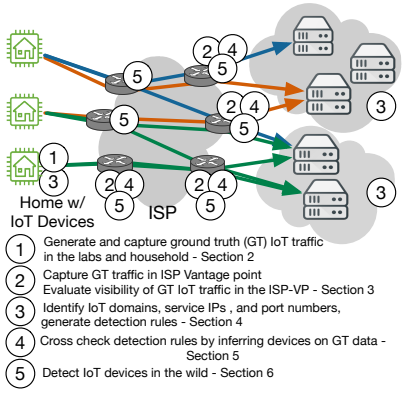


Figure 2: General methodology overview.

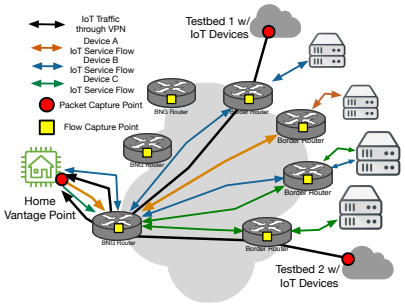


Figure 3: ISP setup & flow collection points.

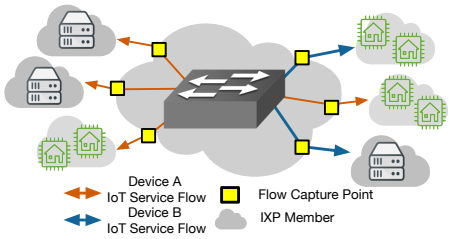


Figure 4: IXP setup & flow collection points.

Ethical considerations ISP/IXP. Neither the ISP nor the IXP flow data contain any payload data, thus no user information. We distinguish user IPs from server IPs and anonymize by hashing all user IPs, following the method described in [5]. The address space of the ISP residential users is known. We call an IP a server IP if it receives or transmits traffic on well-known ports or if it belongs to ASes of cloud or CDN providers. The ports include, e.g., web ports (80, 443, 8080), NTP (123), DNS (53). Moreover, we do not have any specific user activity and can only access and report aggregated statistics in accordance with the policies of the ISP and IXP.

Subscriber line (Home-VP) Network setup. In order to ingest ground truth traffic into the network, we need privileged access to a *home subscriber line*. For this, we use the ISP-VP, but rather than deploying all IoT devices directly within the home, we placed a VPN endpoint with an IP out of the /28 subscriber's prefix and used it to ingest IoT traffic tunneled to the server from two IoT testbeds, one in Europe, one in the US, see Figure 3. The measurement points within the ISP will also capture this traffic. We simply excluded this traffic from our dataset, as the VPN tunnel endpoints are known to us and for each experiment we use the default DNS server for the ISP. Importantly, since the /28 prefix is used explicitly for our experiments, there was no other network activity other than that of the IoT devices.

Ethical considerations–Home-VP setting. With the cooperation of the ISP, we were able to use a reserved /28 allocated to this specific subscriber line (Home-VP) (with signed explicit consent) out of a /22 prefix reserved for residential users. Thus, the analysis in this paper only considers traffic explicitly ingested by the ground truth experiments and does not involve any user-generated traffic.

2.2 Ground Truth Traffic Setting

The IoT testbeds used here consist of 96 devices from 40 vendors. We selected the devices to provide diversity within and between different categories: surveillance, smart hubs, home automation, video, audio, and appliances. Most of these are among the most popular devices, according to Amazon, in their respective region. Our testbed includes multiple instances of the same device (56 different products), so that we can see the destinations that each product contacts in different locations. For a list of the IoT devices and the category of each device, we refer to Table 1. We redirect all IoT traffic to the Home-VP within the ISP, and we capture all the traffic generated by the IoT devices (see (1) in Figure 2).

Category	Device Name
Surveillance	Amcrest Cam, Blink Cam, Blink Hub, Icsee Doorbell, Lefun Cam, Luohe Cam, Microseven Cam, Reolink Cam, Ring Doorbell, Ubell Doorbell, Wansview Cam, Yi Cam, ZModo Doorbell
Smart Hubs	Insteon, Lightify, Philips Hue, Sengled, Smartthings, SwitchBot, Wink 2, Xiaomi
Home Automation	D-Link Mov Sensor, Flux Bulb, Honeywell T-stat, Magichome Strip, Meross Door Opener, Nest T-stat, Philips Bulb, Smartlife Bulb, Smartlife Remote, TP-Link Bulb, TP-Link Plug, WeMo Plug, Xiaomi Strip, Xiaomi Plug
Video	Apple TV, Fire TV, LG TV, Roku TV, Samsung TV
Audio	Allure with Alexa, Echo Dot, Echo Spot, Echo Plus, Google Home Mini, Google Home
Appliances	Anova Sousvide, Appkettle, GE Microwave, Netatmo Weather, Samsung Dryer (idle), Samsung Fridge (idle), Smarter Brewer, Smarter Coffee Machine, Smarter iKettle, Xiaomi Rice Cooker

Table 1: IoT devices under test. *idle* indicates that we capture the traffic just for idle periods because the experiments could not be automated.

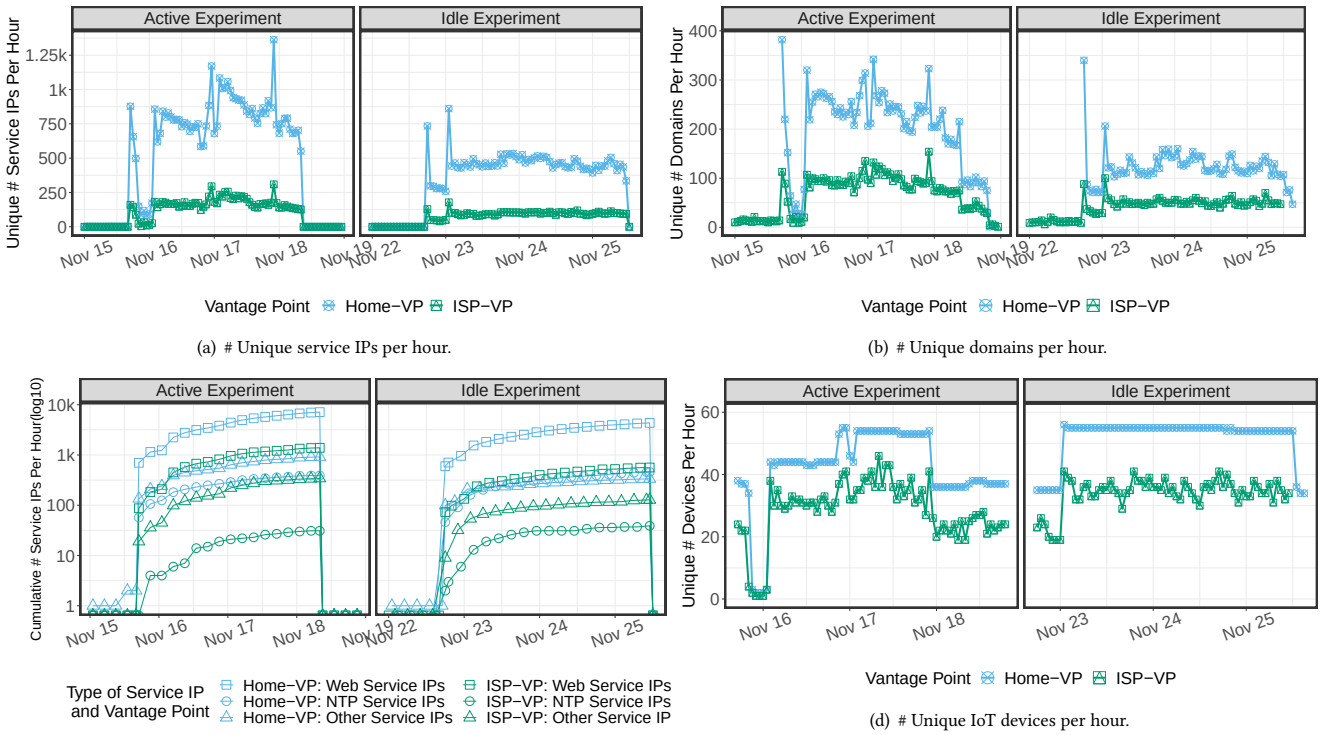
Most of the selected IoT devices are controlled using either a voice interface provided by a voice assistant (such as Amazon Alexa) or via a smartphone companion application. We use the voice interface to automate active experiments by producing voice commands using a Google Voice synthesizer. For IoT devices that support a companion app, we use Android smartphones, and we rely on the Monkey Application Exerciser for Android Studio [14] for automating simulated interactions between the user and the IoT device.

2.3 Active and Idle IoT Experiments

Our experiments can be classified into *idle* and *active* experiments.

Idle experiments. We define as *idle* the experiments during which the devices are just connected to the Internet without being actively used. We generate idle traffic for three days (November 23rd-25th, 2019) from both testbeds.

Active experiments. We define as *active* the experiments involving automated interactions. We perform two types of automated interactions, each one repeated multiple times: (i) *power interactions*, since in a previous study [4] it was reported that many IoT devices generate significant traffic when they are powered off and



(a) # Unique service IPs per hour.

(b) # Unique domains per hour.

(c) Cumulative service IPs per port.

(d) # Unique IoT devices per hour.

Figure 5: Home-VP vs. ISP-VP.

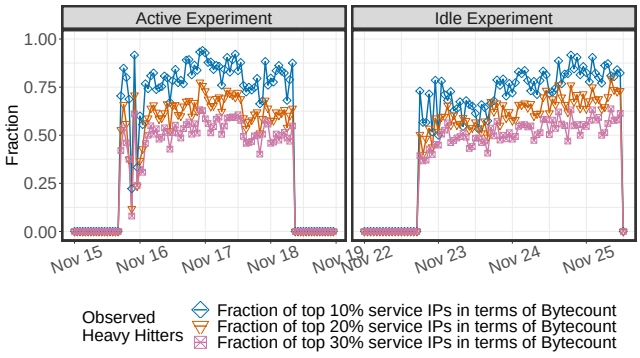


Figure 6: Fraction of observed ISP-VP vs. Home-VP per hour for popular servers (heavy hitters).

on. We manage the power status of the devices through several TP-Link smart plugs that we can control programmatically, followed by two minutes of traffic capture; (ii) *functional interactions*, by automatically controlling the main functionality of the devices (i.e., the act of switching on/off the light for a smart bulb) via voice (either directly or through a smart speaker) or via a companion app running on a separate network with respect to the IoT device (to force the communication to happen over the Internet rather than locally). Unfortunately, some interactions for some devices cannot easily be automated (devices with *idle* in Table 1). For these devices, we consider only idle experiments. In total, we perform 9,810 active experiments between November 15th and 18th, 2019.

3 IOT TRAFFIC – VISIBILITY

In this section, we aim to understand (i) to which extent the IoT related traffic of a *single subscriber line* reaches a diverse set of servers in the Internet, and (ii) whether the low sampling rate of NetFlow limits the subscriber/device visibility. For this, we rely on the ground truth traffic for the Home-VP. More specifically, we monitor the IoT traffic at both vantage points: the Home-VP, as well as the border routers of the ISP-VP (see (1) and (2) of Figure 2).

We first focus on the number of IP addresses that are contacted in each hour during the idle and the active experiments by the IoT devices, as stated in Section 2.3. We explicitly exclude DNS traffic, since it is not IoT-specific. From Figure 5(a), we see that during the active experiments, the IoT devices contact between 500 and 1,300 service IPs per hour when monitored at the Home-VP. Due to sampling, not all of this traffic is visible at the ISP-VP. We define *service IPs* as the sets of IPs associated with the backend infrastructures that support the IoT services. Indeed, the number of observed service IPs per hour in the ISP-VP decreases to an average of 16%. Overall, during our idle experiments, the total number of contacted service IPs is lower, but the average percentage of observed service IPs remained at 16.5%.

The spikes in the active experiments are partially due to power and the functional interactions. This can be seen on the idle experiments, where the spike indicates the action of starting the device (only at the beginning). Note that these spikes are also visible in the sampled ISP NetFlow data.

At first glance, 16% sounds like a very small percentage. However, we note that the visibility of popular service IPs is significantly

high. Figure 6 shows the fraction of service IPs that are visible for the servers contacted the most, according to byte count. For the top 10% of the service IPs, more than 75% are visible, rising up to 90% during some experiments. For less popular service IPs, e.g., the top 20% and top 30%, the visibility is only reduced to 70% and 60% in the active experiment, and a bit lower for the idle experiment.

If we consider the entire period of our experiments, the percentage of visible service IPs is more than 34% and 28% for idle and active experiments. Overall, at the daily level, more than 95% of service IPs are visible for the top 20%. Although we cannot observe *all* IoT devices activity at the ISP-VP, a significant subset is visible.

While any specific service IP may not matter that much for an IoT service, its communications with a server domain name that may be hosted on multiple service IPs is essential. From the Home-VP, we know which service IPs correspond to which domain. Thus, we can determine which observed service IPs at the ISP-VP belong to which domain. This information is relevant for our methodology because in the ISP NetFlow data only IPs are visible. Figure 5(b) shows the number of observed Fully Qulified Domain Names (FQDNs, we will refer to them as domains or domain names for the rest of the paper) at the Home-VP and the ISP-VP. Many domains are hosted at multiple service IPs, hence we see that the number of observed service IPs is higher than the number of observed domains.

Figure 5(d) shows the number of observed IoT devices per hour from the ground truth IoT traffic. We observe a device when at least one packet from that device is seen within an hour. Note, For active mode, the experiments on devices from Testbed 1 (see figure 3), are initiated after Testbed 2. Therefore, all devices are not active during the same period. The average percentages of devices visible at ISP-VP, during active and idle experiments are 67% and 64% respectively.

Next, we separate the observable network activity by ports. More specifically, we consider Web Services (ports 443, 80, 8080), NTP services (port 123), and other services (the rest of the ports), and we show the cumulative number of service IPs contacted. The resulting plot, Figure 5(c), shows that (i) the trend of observable service IPs at the Home-VP is mirrored at the ISP-VP, even when different services are considered, and (ii) the number of service IPs converges over time.

We also checked if any of the traffic from the Home-VP is visible at the IXP. However, neither during the active, nor during the idle experiments, we observe traffic at the IXP. This is expected as the ISP is not a member of the IXP. Rather it peers directly (via private interconnects) with a large number of content and cloud providers as well as other networks.

In summary, our analysis of the ground truth IoT traffic shows that, despite the low sampling of NetFlow, popular domains, service IPs, and ports of a *single* subscriber line (the Home-VP) are visible at the ISP.

4 IOT DEVICE DETECTION METHODOLOGY

In this section, we outline our methodology for the detection of IoT devices in-the-wild. IoT services typically rely on a backend support infrastructure (see Figure 1) for user interactions. From our ground truth experiments, we noticed that this backend infrastructure is often also used for keep-alives, heartbeats, updates, maintenance, storage, and synchronization. This observation is consistent with previous works [4, 15].

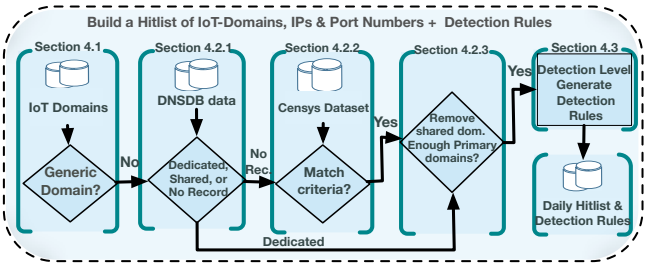


Figure 7: IoT Traffic detection methodology overview.

We focus on identifying which Internet backend infrastructure is supporting *each* of the IoT devices that we deployed in our testbeds (see (3) in Figure 2). When we refer to Internet backend infrastructure, we use two different abstractions: (i) sets of IP addresses/ports combinations as observable from the Internet vantage points, and (ii) sets of DNS domains. We focus also on domains because they are the primary indirect way for the devices to access their backend infrastructure. While domain names are typically part of the permanent programming of the devices, IP addresses are discovered during DNS resolution, and may change over time.

A naive approach for identifying the backend infrastructure would be to use the ground truth traffic to identify which domains, and as a consequence, which service IPs are being contacted by each device. However this is not sufficient for the following reasons:

Limited relevance of some domains: Not all domains are essential to support the services, or are useful for classification; for example, some domains may be used for advertisements or generic services, e.g., `time.microsoft.com` or `wikipedia.org`, see Section 4.1.

Limited visibility of IP addresses: Since the ground truth data is captured at a single subscriber line only and DNS to IP mapping is rather dynamic, just looking at this traffic is not sufficient, see Section 4.2.1.

Usage of shared infrastructure: Not all IoT services are supported by a dedicated backend infrastructure. Some rely on shared ones, such as CDNs. In the former case they can still have dedicated IP addresses; in the latter cases they use shared IP addresses, see Section 4.2.1.

Churn: DNS domain to IP address mappings are dynamic, see Section 4.2.1.

Common programming APIs: Multiple IoT services may use the same common programming API or may be used by different manufacturers; as a result, they often rely on the same infrastructure. This is the case for relatively generic IoT services such as Alexa voice service. While this IoT service is available on dedicated devices, e.g., Amazon Echo, it can also be integrated into third-party hardware, e.g., fridges and alarm clocks [13]. We cannot easily distinguish these from network traffic observations.

Below we tackle these challenges one by one. The outcome is an IoT dictionary that contains mappings for individual IoT services to sets of domains, IP addresses, and ports. Based on IoT services, we generate rules for IoT device detection. For an overview of the resulting methodology, see Figure 7.

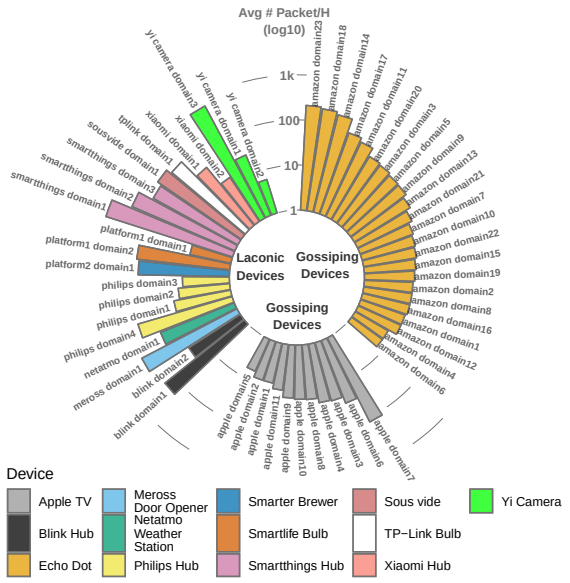


Figure 8: Home-VP: Circular bar plot of average # of packets/hour per domain (log y-scale). The domains belong to 13 IoT devices and separated into three groups: one for laconic and two for gossiping devices (Echo Dot and Apple TV).

4.1 Classifying IoT Domains

The amount and frequency of network traffic that an IoT device exchanges with its backend infrastructure varies from device to device, depending on the complexity of its services, its implementation specifics, and the usage of the device. This is highlighted in Figure 8, where we show the average number of packets per device and per domain (using a log y-scale) for 13 different devices (subset of devices) in their idle mode. The first observation is that most devices are supported by their own set of domains and for many IoT services, this is a small set containing less than 10 domains. We refer to these as *small domain sets* as they correspond to *laconic* devices. Other devices *gossip* and have *sizable domain sets*. Figure 8 shows the domains of two example gossip devices (Apple TV in gray and Echo Dot in orange) and several laconic devices (rest of the colors). Having a sizable domain set often indicates the usage of a larger infrastructure, which may not be dedicated to a specific IoT service. We find that most of these domains are mapped via CNAMEs to other domains. For the two gossiping examples considered in Figure 8, the domains of Echo Dot are mostly mapped to its own infrastructure. However, the ones of Apple TV are mainly mapped to a CDN—in this case, Akamai—that offers a variety of services.

Based on these observations from our ground truth data, we classify the domains as follows:

IoT-Specific domains. Grouped into (i) *Primary* domains: registered to an IoT device manufacturer or an IoT service operator; and (ii) *Support* domains: that are not necessarily registered to IoT device manufacturers or service operators, but offering complementary services for IoT devices, i.e., *samsung-*.whisk.com* for Samsung Fridges, here whisk.com is a service that provides food recipes and images of food.

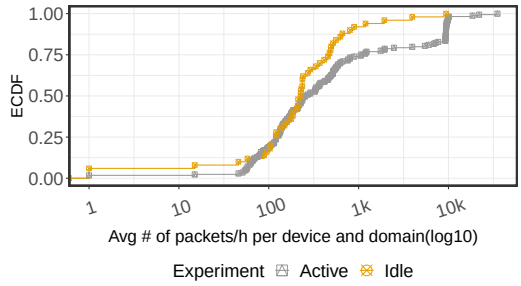


Figure 9: Home-VP: ECDF of average # of packets/hour for all IoT-Specific domains, per device, (idle and active experiments).

Generic domains. Domains registered to generic service providers that are heavily used by non-IoT devices as well, e.g., *netflix.com*, *wikipedia.org*, and public NTP servers.

We classify each domain name from our idle and active experiments using pattern matching, manual inspection, and by visiting their websites and those of the device manufacturers. Since the *Generic* domains cover non-IoT traffic, we do not further consider them. Rather, we focus on the *IoT-Specific* domains. As a result, we classify 415 out of the 524 domains as *Primary* and 19 as *Support* domains.

Next, we explore the volume of traffic that the IoT devices exchange with all domains. Figure 9 shows the ECDF of the average number of packets per hour per domain for all IoT-Specific domains for both the idle and the active experiments. First, we note that almost all devices and domains , except for one device in its idle mode, are exchanging at least 100 packets per hour, and this *may not suffice for detecting* them in any given hour in the wild due to sampling. However, during the active experiments, we see that some domains are only used when the device is *active* or other domains receive significantly more traffic, up to and exceeding 10K packets, which *may suffice for detection*. These latter domains may be ideal candidates for detecting such devices in the wild.

4.2 Identifying Dedicated Infrastructures

Once we have a list of IoT-Specific domains (FQDNs) with their associated service IP addresses and port mappings from the ground truth experiments, we need to understand whether they have a shared or dedicated backend infrastructure. The reason is that, if we want to identify IoT services and consequently IoT devices in the wild by using network traces such as NetFlow, we can only observe standard network level features such as src/dst IP and port numbers without packet payload. Therefore, if a service IP belongs to a shared infrastructure such as a CDN or a generic web hosting service, this service IP can serve many domains, and it is impossible for us to exactly know which domain was actually contacted. To this end, the purpose of this section is two-fold. First, to expand the candidate service IPs beyond those directly observed in the ground truth experiments (to mitigate that we are focusing on a single subscriber line). Second, to classify domains into those that use backend services hosted on dedicated infrastructure service IPs vs. those that rely on shared infrastructure service IPs. We do this by relying on DNSDB [16], Censys [9], and applying additional filters.

4.2.1 From IoT-Specific Domains to Service IPs: DNSDB. We use IoT-Specific domains to identify the backend infrastructure that is hosting them. To this end, we leverage the technique in [17], and use these domain names to identify all associated service IPs on which these domains are hosted during the time period of our experiments. We use both the ground truth experiments, and external DNS databases, including DNSDB [18]. We found that the specific IP addresses mapping to specific domains can change often. However, DNSDB provides information for all domains served by an IP address in a given time period and vice versa, hence it mitigates the issues caused by this churn. DNSDB also provides all records, including CNAMEs that may have been returned in the DNS response, for a given domain. Thus, we use DNSDB to check if a service IP address is *exclusively* used for a specific IoT service, or if it hosts additional domains. We say a service IP is *exclusively* used if it only serves domains from a single "second-level" domain (SLD) and its CNAMEs. However, we note that the CNAMEs may not involve the same second-level domain. Let us consider an example: the domain `devA.com` is mapped via a chain of CNAMEs such as `devA-VM.ec2compute.amazonaws.com` to IP `a.b.c.d`. This IP only reverse maps to `devA-VM.ec2compute.amazonaws.com` and its associated CNAME `devA.com`. Since this is the only CNAME associated with the IP, we may consider this IP a direct mapping for the domain. Yet, at the same time, we find support that public IP addresses assigned to a cloud resource such as a virtual machine in AWS EC2, that is occupied by a tenant, is not shared with other tenants unless the current resource is released. This is a popular service offered by multiple platforms [19–21]. Let us consider a second example: domain `devB.com`. It may use the Akamai CDN. Thus, the domain `devB.com` is a CNAME for `devB.com.akadns.net`. This domain then maps to IP `a.b.c.d`. However, in this case, many other domains, e.g., `anothersite.com.akadns.net`, also map to this IP. Thus, we may conclude that this domain is hosted on a *shared* infrastructure.

Once we understand if an IP is exclusively used for a specific IoT service, we can also classify the domains as either using a *dedicated* or *shared* infrastructure. For the former, all service IPs have to be dedicated to this domain for all days, otherwise we presume that the domain relies on a shared infrastructure.

Once we apply this methodology to all 434 domain names, we find that 217 are hosted on dedicated service IPs, while 202 are relying on a shared backend infrastructure. For 15 of the domains we did not have sufficient information in DNSDB. We handle them in the next step.

4.2.2 From IoT-Specific Domains to Service IPs: Censys. Among the reasons that DNSDB may not suffice for mapping some domains to service IPs is that (a) frequent remapping of domains to IPs or, (b) missing data since the requests for the domains may not have been recorded by DNSDB, which intercepts requests for a subset of the DNS hierarchy. To overcome this limitation, we rely on the certificate and banner datasets from Censys [9], to infer the ownership of the domains and the corresponding IPs, as long as these are using HTTPS. For example, we did not find any record for the domain `c.devE.com` in the DNSDB dataset. We then check if device *E* uses `HTTPS` to communicate with this domain. This allows us to query for all service IPs that potentially offer the same web certificate as the hosts in this domain. For a certificate to be associated with a domain, we require that the domain name and the *Name* field entry in the certificate match at least the SLD or higher, i.e. the Name field of the certificates matches the pattern `c.devE.com` or `*.devE.com` and that there is no other *Subject Alternative Name (SAN)* in the certificate. Next, we query the Censys dataset for all IPs with the same certificate and `HTTPS` banner checksum for the domain from our ground truth dataset within the same period. This allows us to identify data for 8 out of 15 of the domains which belong to 5 devices.

4.2.3 Removal of Shared IoT Backend Infrastructures. In the last step of our methodology we filter out devices that use shared backend infrastructures. We find that Google Home, Google Home Mini, Apple TV, and Lefun camera, all have a shared backend infrastructure. For LG TV, we are left with only one out of 4 domains; for Wemo Plug and Wink-hub, we could not identify sufficient information. Because of this, we have excluded these devices from further consideration.

The result forms our *daily* list of dedicated IoT services, along with their associated domains, service IPs and port combinations.

4.3 IoT Services to Device Detection Rules

Once we identified the set of IoT services that can be monitored, we generate the rules for detecting IoT devices. Depending on the set of IoT services contacted by the devices we can generate device detection rules at three granularity levels: (i) Platform-level, (ii) Manufacturer-level, and (iii) Product-level, from the most coarse-grained to the most fine-grained, respectively. In this section, first, we show how we determine the detection level for each device. Then, we explain how we generate the detection rules for each IoT device for the detection level that can be supported.

4.3.1 Determining IoT Detection Level.

Platform-level: Some manufacturers use off-the-shelf firmware, or outsource their backend infrastructure to IoT platform solution companies such as Tuya [22], electricimp [23], AWS IoT Platform [24]. These IoT platforms can have several customers/manufacturers that rely on their infrastructure. Therefore, we may not be able to distinguish between different manufacturers from their network traffic.

Manufacturer-level: The majority of our studied IoT services rely on dedicated backend infrastructures that are operated by the manufacturers themselves. We also observe that many manufacturers rely on similar APIs and backend infrastructures to support their different products and services. This makes distinguishing individual IoT products from their network traffic more challenging.

Product-level: This is the most fine-grained detection level, where we are able to distinguish between different products of a manufacturer, e.g., Samsung TV, or Amazon Echo vs. Amazon Fire TV. For detection at the product level, we underline the importance of side information about the purpose associated with a domain. With this information, we can improve our classification accuracy. For example, for Alexa Enabled devices, the domain `avs-alexa.*.amazon.com` is critical, as it is the base URL for the Alexa Voice Service API [13] (shown in Figure 8 as amazon domain23). Other examples are the

Samsung devices that use the domain `samsungotn.net` to check for firmware updates [25].

Additionally, some advanced services of the devices often require additional backend support from manufacturers. These may then contact *additional* domains. By considering more specific features (domains), the capabilities to distinguish products increases. We leverage these specialized features e.g., to distinguish Amazon Fire TV, which contacts significantly more domains than other Amazon products, e.g., Echo Dot.

4.3.2 Generation of Detection Rules. For any of our three levels of detection, we require that a subscriber contacts at least one IP/port combination associated with a Primary domain of the IoT service, to claim detectability of IoT activity at the subscriber. However, if there are many domains, requiring only one such activity may not have enough evidence. For example, by monitoring a single domain we can detect all Alexa Enabled devices, but this service can be integrated into third party hardware as well. Therefore, in order to detect products manufactured by Amazon, e.g., Amazon Echo, it is essential to monitor additional domains that are contacted by the Amazon Echo devices. For this, we introduce the *detection threshold D*. If an IoT service has N IoT-Specific domains, we require to observe traffic involving k IP/port combinations that are associated with $max(1, \lfloor D \times N \rfloor)$ of the N domains. To determine an appropriate value for this threshold, we rely on our ground truth dataset, see Section 5.

We start with 96 devices in our testbeds. We have multiple copies of a same device deployed in different continents. This reduces the set of devices to 56 unique products. Of these, many are from the same manufacturer, e.g., a Xiaomi rice cooker, a Xiaomi plug, and a Xiaomi light bulb. Since these devices are often supported by the same backend infrastructure of the manufacturer, the list of domains has significant overlap and often fully overlaps. In our methodology we can detect 3 different IoT platforms, the coarsest level, as 4 of our products rely on them. Moreover, we generated rules for the detection of 29 IoT devices at the manufacturer level. We had a diverse range of products from Amazon and Samsung in our testbed that allowed us an in-depth analysis, and cross-examination of domains contacted by different products. Therefore, for devices using Alexa voice service (i.e., Alexa Enabled), and for Samsung IoT devices, we detect the former at the platform level and the latter at the manufacturer level. For Alexa Enabled and Samsung IoT devices, we compared the domains across different devices and obtained enough side information about the purpose of their domains that allowed us to further divide each of them into two subclasses at more fine grained levels. For this, we defined a hierarchy, namely Amazon products, and Fire TV, under Alexa Enabled devices. Amazon products are detected at manufacturer level, and include products such as Amazon Echo family and is superclass of Fire TV. We identified 33 additional domains, besides the Alexa voice service domain, that were contacted by Amazon products. Moreover, Fire TV contacts up to 67 domains (34 more domains than Amazon products). This allows us to establish its subclass, at product level, under Amazon products. Using side information [25] and comparing the set of domains across different Samsung products, we monitor 14 domains in total, but only one domain is important to detect Samsung IoT devices with Samsung

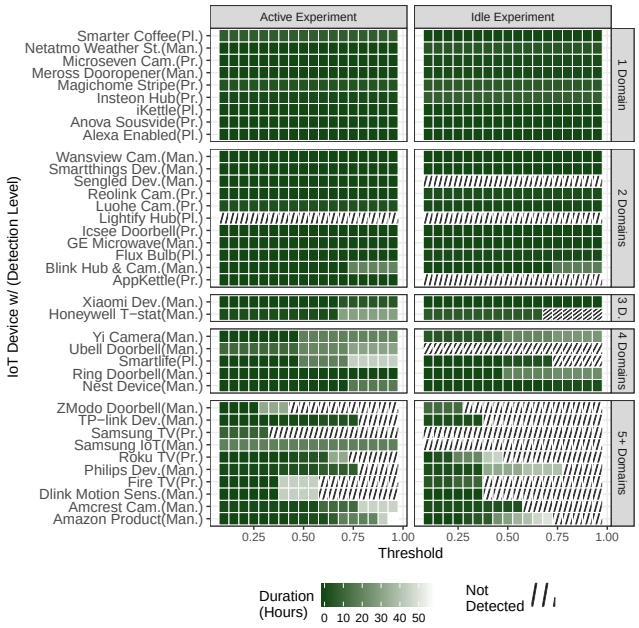


Figure 10: Home-VP: Time to detect IoT (per threshold).

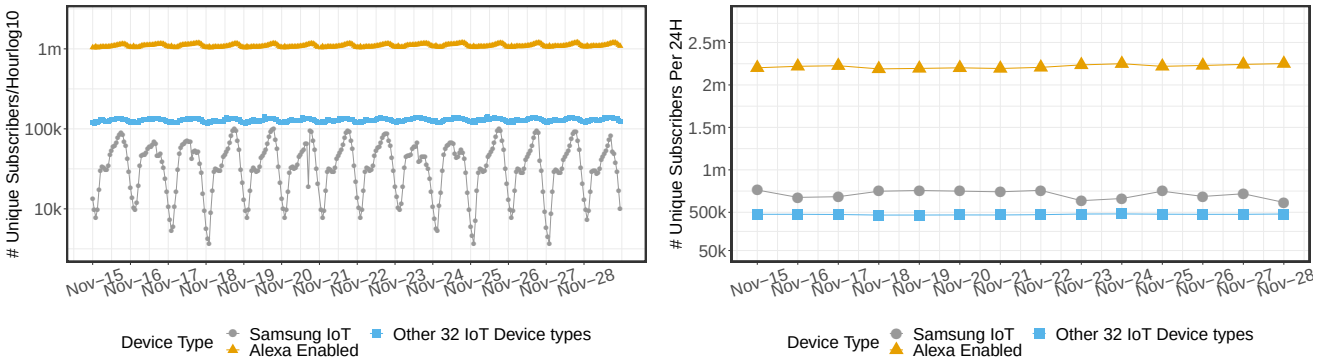
firmware (these include a broad range of products, such as fridges, washing machines and TVs). Samsung TVs contact 16 additional domains that are not used by any of the other Samsung devices in our testbed.

Using the above methodology, except for the devices listed in section 4.2.3, we generated detections rules at different levels for our testbed devices. We generated rules for the detection of 20 manufacturers, and 11 products that amounts to the 77% of manufacturers in our testbeds. We generate rules for 4 *unique* IoT platforms by monitoring 1 to 4 domains (2 platforms were contacted by 4 devices, we report them separately). Finally, for 11 products we consider between 1 to 67 domains. For a detailed number of domains per IoT device see Figure10.

5 METHODOLOGY: CROSSCHECK

We use our ground truth dataset to check how long it takes for our methodology (applied to the sampled flow data from the ISP) to detect the presence of the IoT devices for the idle and the active experiments (see (4) of Figure 2). For this, we report the time that it takes to detect an IoT device that is hosted in our ground truth subscriber line when it is in active mode (Figure 10 left) and idle mode (Figure 10 right). We only include the ones that are detectable with our methodology, i.e., those that do not rely exclusively on shared infrastructures. We also annotate the device name with its detection levels: Platform (Pl.), Manufacturer (Man.), and Product level (Pr.).

On average, by requiring the evidence of at least 40% of domains, we are able to detect 72/93/96% of IoT devices that are detectable at manufacturer or product level within 1/24/72 hours in the active mode. Even in idle mode their the percentage is 40/73/76% with 1/24/72 hours. For the devices detectable only at product level (Pr.), with the same required evidence, we detected 63/81/90% of them within the 1/24/72 hours respectively, in active mode. Note,



(a) Per Hour. (b) Per Day.

Figure 11: ISP: Per Hour, Subscriber lines with IoT activity (Alexa Enabled, Samsung IoT, and others).

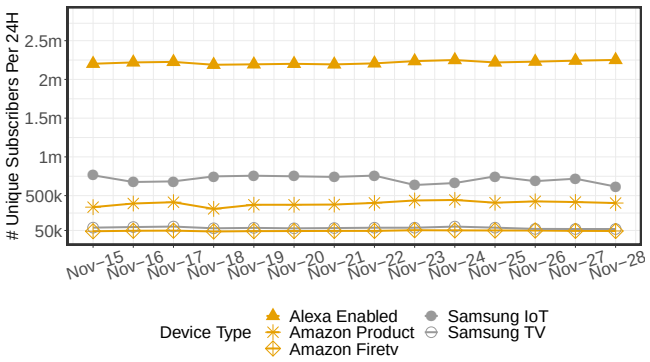


Figure 12: ISP: Drill down for Amazon and Samsung IoT devices–per day.

we are using the sampled ISP data. Indeed, popular products such as Amazon products (i.e., Echo Dot, Echo Spot) can be almost instantly detected. This is a significant finding and underlines that it is possible to use sampled flow data within an ISP to accurately detect the presence of a specific IoT product within a subscriber line, despite differences in activity and IP churn due to operational requirements.

A closer look reveals that, in general, it takes longer to detect an idle IoT device in comparison to when it is active. This is not surprising, as most IoT devices show more network activity in active mode. However, this does not mean that the increase will occur across all of the services contacted by a device, since there are exceptions that take longer to detect even in active mode, e.g., SmartLife, and Nest.

Figure 10 also contains information regarding the number of monitored domains per IoT device with their detection level. For 9 IoT devices, a single domain is considered. For the others, we consider many more (up to 67). A *threshold* determines the fraction of domains for which we require evidence of network traffic to claim detection. To understand the impact of such threshold on detection time, we variate its value from 0.1 to 1 and show the corresponding detection times. Note, for IoT devices where we consider only one domain, the variation of the threshold does not change the detection time, as we always require evidence of at least one domain. Overall, we note that a larger threshold can increase the detection time, and some IoT devices may no longer be detectable. However, it may also increase the false positive rate. We crosscheck possible false positives by running another experiment where we only enable a small subset of IoT devices. We then apply our detection methodology to these traces and do not identify any devices that are not explicitly part of the experiment. We also try to avoid false positives by ensuring that the domain sets per device differ.

Regarding detectability, we notice that 6 IoT devices could not be detected even after the entire duration of our idle experiments. A closer investigation shows that for 5 of these, the frequency of traffic is so small that their likelihood of detection is very low. Indeed, for this specific time period, they were invisible in the NetFlow data. This highlights that in order to be able to confidently detect a device, the device have to either exchange enough packets with the targeted domains or the sampling rate shall be increased. For Samsung TV, we require to observe enough domains to confirm the presence of a Samsung IoT device, before moving forward with detection. Thus, if we do not see enough Samsung IoT domains, then we do not claim the detection of Samsung TVs. Nevertheless, the results look very promising for us to attempt on detecting deployed IoT devices in the wild.

6 RESULTS: IOT IN THE WILD

In this section, we apply our methodology for detecting IoT activity in the ISP and IXP data (see (5) in Figure 2). For this we focus on the two weeks in which we collected the data from the ground truth experiments to obtain up-to-date mappings of domains to IPs.

6.1 Ethical Considerations and Privacy Implications

Applying our methodology to traffic data from ISPs and IXPs may raise ethical concerns as it may be considered as analyzing customer activities. However, this is not the goal of this paper. The goal here is to showcase that it is possible to detect and map the penetration of IoT device usage. As such, this study is not about subscribers' device activities, instead it is about detection capabilities and aggregated usage. Thus, we report on percentages of subscriber lines where we can observe IoT related activity. Indeed, we are unable to trace IoT activity back to individuals as the raw data was anonymized

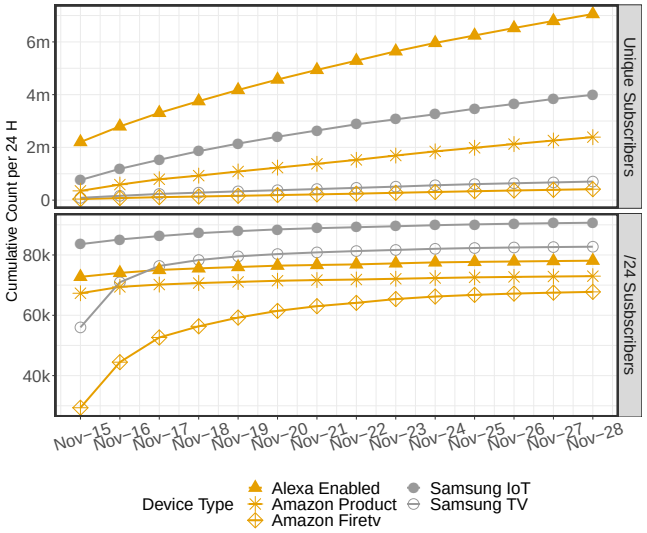


Figure 13: ISP: Cumulative # of subscriber lines resp. /24s with daily IoT activity across two weeks.

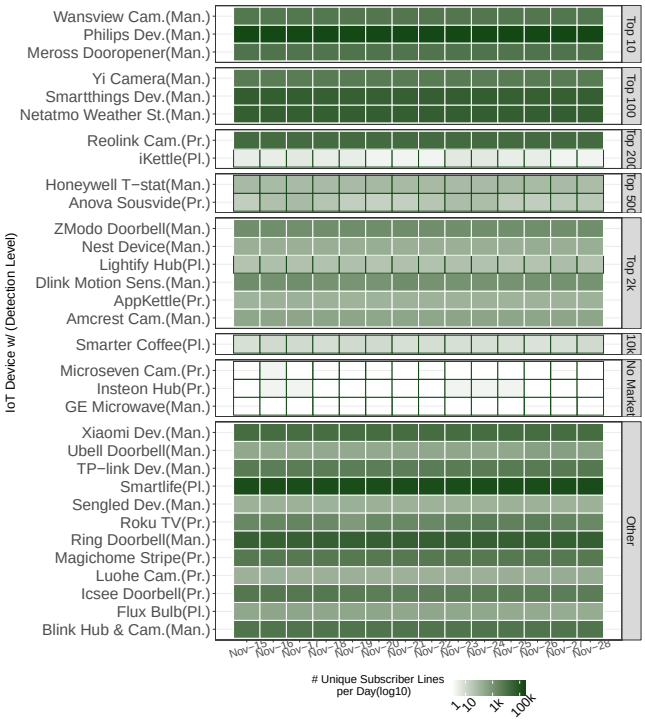


Figure 14: ISP: Drill down of IoT activity for 32 different IoT device types with their popularity in the ISPs country.

as per recommendations by [5] and never left our collaborators' premises. Moreover, we do not analyze any data that is not related to the detection of IoT presence, e.g., DNS queries [26], or flows that are not related to IoT backend infrastructures, to eliminate any user Web visit profiling.

6.2 Vantage Point: ISP

IoT related activity in-the-wild. Figure 11 shows the number of ISP subscriber lines for which we detect IoT related activity. The ISP does not operate a carrier-grade NAT. Even if multiple IoT devices are hosted at an ISP subscriber, we count the hosting subscriber only once. Thus, the number of subscribers that host a given IoT device is a lower bound for the number of the given IoT device in the premises of ISP subscribers. Figure 11(a) and Figure 11(b) focus on hourly and daily summaries. Since the top IoT devices detected are Alexa Enabled and Samsung IoT, we show them separately. We see IoT related activity for roughly 20% of the subscriber lines. Our results show a significant penetration of Alexa Enabled devices of roughly 14%. This is slightly more than estimates of national surveys in the country where the ISP operates, stating that the market penetration of Alexa Enabled devices, as of June 2019, is around 12% [27–29]. Yet, these reports cannot capture which devices are in active use at any particular day, e.g., Nov. 2019, contrary to our study. Note, in Figures 11, 12, 14 and 15 we apply our methodology on each time bin independently.

Daily patterns of IoT related activity. By looking at the hourly plots in Figure 11(a), we see some significant daily patterns for Alexa Enabled and Samsung IoT devices. We do not see diurnal patterns for the other 32 IoT device types. Such diurnal patterns are correlated with human activities. Typically, during the day, network activity increases as the users interact with the IoT devices while it decreases during the night when the devices are idle. As detection likelihood is correlated with network activity, the devices detectability also correlates with this diurnal pattern. We note that the patterns for Alexa Enabled does not differ from those for Samsung. The reason is that many of the Alexa Enabled and Samsung IoT (Samsung TVs) class may be used more for entertainment, which is why their activity is higher in the evenings. Samsung IoT devices have a small spike in the mornings before gradually reaching their peak around 18:00 (ISP timezone).

For the drill down for Samsung IoT devices see Figure 12. Even with the presence of a diurnal variation for Alexa Enabled, there is a significant baseline during the night. This is expected as IoT devices often have traffic even when they are idle and are thus detectable. Over the course of a day, the diurnal variation is rather low compared with the typical network activity driven by human activity. This explains the low variance of the observed number of subscriber lines for Alexa Enabled devices.

Aggregation per day. We observed in Section 5 that, while it is often possible to detect Alexa Enabled devices within an hour, the same is not always true for Samsung IoT devices. Therefore, Figure 11(b) reports the same data but this time using an aggregation period of a day.[3] We see that the total number of observed subscriber lines does not change drastically from day to day. However, we also note that the number of subscriber lines with Alexa Enabled devices roughly doubled, while those with Samsung increased by a factor of 6. The reason is that detecting Samsung IoT devices is more challenging because they are contacting their Primary domain less frequently than Alexa Enabled devices. Thus, their detection is heavily helped by the increase in the observation time period. For the other IoT devices we see these effects, whereby the increase is correlated to the expected time for detection. Note, certain Samsung

[3]Most subscriber lines are not subject to new address assignments within a day. Most addresses remain stable as the ISP offers VoIP services.

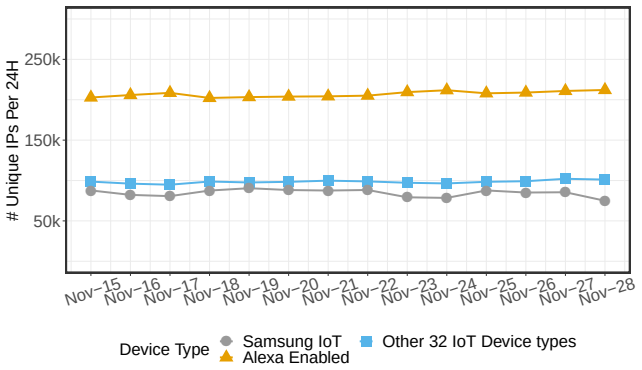


Figure 15: IXP: Number of Samsung IoT, Alexa Enabled, and Other 32 IoT device types IPs observed/day.

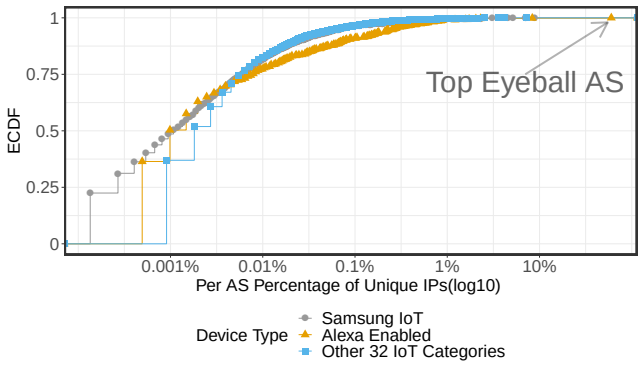


Figure 16: IXP: ECDF of Per-ASN Percentage (# Unique IPs) - Day 15-11-2020.

domains are contacted by both Samsung IoT and Non-IoT devices. In our analysis, we only consider domains that are exclusively contacted by *Samsung IoT* devices. By adding those domains, the number of detected Samsung devices will be increased at least by a factor of two, but this also adds false positives to our results.

Detecting specific devices. So far, we have focused on the superclass of Alexa Enabled and Samsung IoT devices. However, by adding more specialized features, our methodology allows us to further differentiate them. For example, some subsets of domains are only contacted by specific products. Thus, in Figure 12 we show which fraction of the Alexa Enabled IoT devices are confirmed Amazon products and which fraction of these are Fire TVs using a conservative detection threshold of 0.4. For Samsung IoT devices, we show how many of them are Samsung TVs. Again, the number of subscriber lines with such IoT devices is quite constant across days. As expected, the specialized devices only account for a fraction of the devices of both manufacturers.

Subscriber lines churn. While the ISP's overall churn of subscriber line identifier is pretty low (as was also confirmed by the ISP operator), some changes are possible and may bias our results. Possible reasons for such changes are: unplugging/rebooting of the home router, regional outages, or daily re-assignment of IPs for privacy reasons. Yet, as most IoT devices are detectable within a day (recall Section 5), the churn should not bias our results. Still, to check for such artifacts, we move to larger time windows: see the upper panel of Figure 13, which plots the cumulative number of subscriber lines with detected Alexa Enabled and Samsung IoT devices, respectively, for up to two weeks. Here, we see that the fractions increase. However, we may have substantial double counting due to identifier rotation. To underline this conclusion, we consider penetration at the /24 prefix aggregation level, see the lower panel in Figure 13. The penetration lines stabilize smoothly, but at different levels and with different speed. The latter is related to the popularity of an IoT device. If it is already popular, the likelihood of moving from a known to an unknown subscriber line identifier is lower with respect to less popular IoT devices.

Detecting other IoT devices in-the-wild. Figure 14 reports the detected number of the IoT devices that are neither Alexa Enabled nor Samsung IoT. We report them using a heatmap, where each column corresponds to a day and each row to an IoT device annotated with its detection level. The color of each entry shows the number of subscribers lines during that day. Our first observation is that the number of subscriber lines for each device class is very stable across the duration of our study. Next, we point out that our experiments include popular devices from both the European as well as the US market. For a reference, we report the relative popularity of each IoT device in the Amazon ranking for that device, in the country where the ISP operates. If a ranking of a device is not available, we categorize them as "other." Popular devices are more prominent than unpopular ones or the ones that are not available in the country's market. For example, on the one hand there are Philips devices that are popular and in heavy use with more than 100 K subscription lines on a daily basis. On the other hand there is Microseven camera that is not in the country's market. Yet, we can still observe some deployments, these results highlight that our methodology is able to detect both popular and unpopular IoT devices when the domains and associated service IPs that IoT devices visit can be extracted.

6.3 Vantage Point: IXP

Next, we apply our detection methodology at the IXP vantage point. Here, we have to tackle a few additional challenges: First, the sampling rate at the IXP is an order of magnitude lower than at the ISP. Second, the vantage point is in the middle of the network, which means that we have to deal with routing asymmetry and partial visibility of the routes. Third, while the ISP does aggressive spoofing prevention, e.g., with reverse path filtering, this is not possible at the IXP. Spoofing prevention is the responsibility of individual IXP members. Thus, we require TCP traffic to see at least one packet without flags, indicating that a TCP connection was successfully established. While this may reduce visibility, it prevents us from over-estimating the presence of IoT traffic.

While the IXP offers network connectivity for every ASes, only a few member ASes are large eyeballs [30]. It is not that surprising that we did not observe any activity of the ground truth experiment, recall Section 3. Still, we are able to detect significant IoT activity. Figure 15 shows the number of IPs for which we detected IoT activity per day for our two-week study period (November 15th-28th, 2019). We are able to detect roughly 90k Samsung devices, 200k Alexa Enabled devices, and more than 100k of other IoT devices. This underlines that our methodology, which is based on domains and generalized observations from a single subscriber

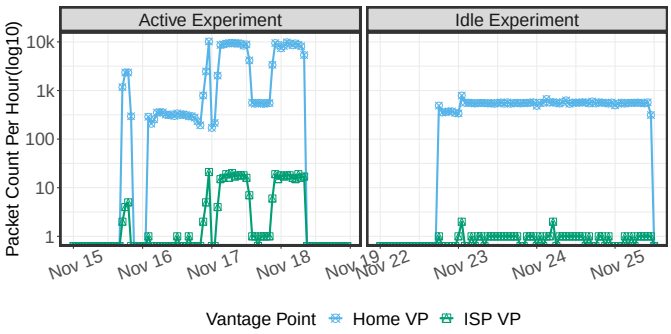


Figure 17: Home-VP/GT Household: Single Alexa Enabled device.

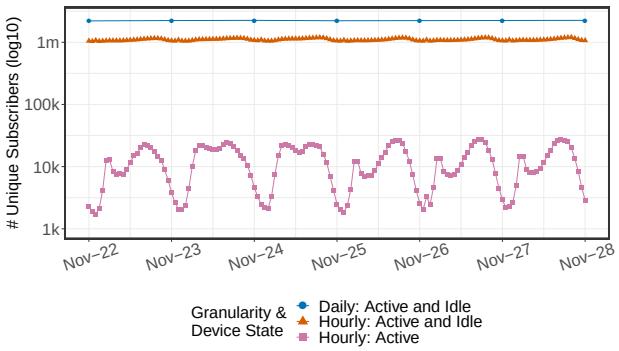


Figure 18: ISP: # Subscribers with active Alexa Enabled/hour.

line, is successful. Most IXP members are non-eyeball networks. As such, we expect that the detected IoT activity is concentrated on these members. Figure 16 shows an ECDF of the distribution of IoT activity per AS for one day (November 15th, 2019) and three IoT device types, namely, Samsung IoT, Alexa Enabled, and the other IoT devices. The distributions are all skewed—a small number of member ASes are responsible for a large fraction of the IoT activity. Manual checks showed that these are all eyeball ASes. Yet, we also see a fairly long tail. This underlines that some IoT devices may not only be used at home (and, thus, send their traffic via a non-eyeball AS).

7 DISCUSSION

7.1 Device Usage Detection

A natural question is whether sampled flow data also allows one to distinguish if an IoT device is in active use. Our results indicate that the answer is positive. First, our ground truth experiments show that for some devices, the domain sets used during the idle experiments differ from those during active experiments. Hence we can use these domains to determine the mode (active/idle) of an IoT device. Second, the amount of traffic also varies depending on the mode. To highlight this, Figure 17 shows the number of observed packets at the Home-VP for a single Alexa Enabled device, as well as the ISP-VP for both modes. Activities cause spikes above 1K at the home vantage points and above 10 at the ISP-VP. These ranges are never reached during the idle experiments.

When using the first insight for, e.g., devices from TP-link (TP-link Dev.), we are able to capture active use for only 3.5% of the devices. The reason is that these are plugs, which have a total traffic volume so low that it limits the detectability due to the low sampling rate at the ISP. When using the second insight for Alexa Enabled devices, we find that we can detect significant activity. Figure 18 shows both the subscriber lines with Alexa-enabled devices per hour, per day as well as the subscriber lines with active Alexa-enabled devices. Based on the above-mentioned observations, we used the threshold of 10 for packet counts per hour to filter out subscribers that actively used Alexa-enabled devices in a given hour. Based on this threshold, we see that the number of actively used devices reaches 27,000 during the day and weekends (November 23rd-24th, 2019), following the diurnal pattern of human activity.

The ability to distinguish active from idle usage of IoT devices in the wild may raise ethical/privacy concerns. However, the goal of this paper is not to analyze user behavior, but rather to point out the privacy concerns associated with having these IoT devices at home [3].

7.2 Potential Security Benefits

The ability to detect IoT services can be used in a constructive manner or even as a service by ISPs. For example, if there are known security problems with an IoT device, the ISP/IXP can block access to certain domains/IP ranges or redirect their traffic to benign servers. The methodology can also be used for troubleshooting, incident investigation, and even incident resolution. For example, an ISP can use our methodology for redirecting the IoT devices traffic to a new backend infrastructure that offers privacy notices or security patches for devices that are no longer supported by their manufacturers.

Moreover, if an IoT device is misbehaving, e.g., if it is involved in network attacks or part of a botnet [31], our methodology can help the ISP/IXP in identifying what devices are common among the subscriber lines with suspicious traffic. Once identified, their owner can be notified in a similar manner, as suggested by [32], and it may be possible to block the attack or the botnet control traffic [33].

7.3 Limitations

Our methodology has some limitations.

Sample devices. We need to have sample devices in order to observe which domains are being contacted.

Superclass detection. We mostly check for false negatives and limitedly for false positives as we only have traffic samples from a subset of IoT devices, but not for all possible IoT devices. If an IoT device relies on a shared backend infrastructure or common IoT APIs, we only detect the superclass, e.g., at the manufacturer level.

Network activity. We rely on the network activity of IoT devices. As such, if the traffic volume is very low detectability decreases, and detection time increases.

Shared infrastructures. We cannot detect IoT services that rely on shared infrastructures. If the IoT devices change their backend infrastructure, e.g., after an update, we may have to update our detection rules too.

7.4 Lessons Learned

Our analysis could be simplified if an ISP/IXP had access to all DNS queries and responses as they do in [34] and [26]. Even having a partial list, e.g., from the local DNS resolver of the ISP, could improve our methodology. Yet, this raises many privacy challenges. An increasing number of end-users rely on technologies like DNS over TLS [35], or public DNS resolvers, e.g., Google DNS, OpenDNS, or Cloudflare DNS, rather than the local ISP DNS server [36]. Yet, this also points to another potential privacy issue—the global data collection and analysis engines at these DNS operators, which can identify IoT devices at scale from the recorded DNS logs using our insights. Capturing DNS data from the network itself would require deep packet inspection and thus, specialized packet capture, which is beyond the scope of this paper.

The subscriber or device detection speed varies depending not only on the device and its traffic intensity, but also on the traffic capture sampling rates. The lower this rate, the more time it may take to detect a specific IoT device. Moreover, identifying the relevant domains for each IoT device does require sanitization, which may involve manual work, e.g., studying manuals, device documentation, vendor web sites, or even programming APIs. Given that we are unable to identify IoT services if they are using shared infrastructures (e.g., CDNs), this also points out a good way to hide IoT services.

7.5 Future Directions

We can use our insights to develop signatures that allow an ISP to identify households that use specific IoT services. If such services are, e.g., subject to security concerns they can use such signatures to notify the corresponding customer of the potential problem and fix. This is also possible if the IoT service is no longer supported or needs end-user manual upgrades, e.g., to mitigate threats. Such signatures may also be used to move from DDoS attacks towards identifying culprits. Our approach is potentially scalable further using MUD profiles [37], where devices will signal to the network what sort of domains, access and network functionality they require to properly function. It is also possible to extend the list of signatures of IoT devices using crowdsourcing [38].

8 RELATED WORK

There have been some recent papers in understanding home IoT traffic patterns and identifying devices based on their signatures, trackers, and network traffic [39]. These approaches often rely on testbed data [4, 40], or tools for the active discovery of the household devices and their network traffic [41]. The authors in [40] use a broad range of network features from packet captures, including domain names to train a machine learning model and detect IoT devices in a lab environment. However, they do not further study the backend infrastructure supporting IoT devices. There have also been a few early attempts at mitigating against these device discoveries using traffic padding [42] or blocking techniques [33].

A number of recent efforts focused on inferring IoT device types from network traffic [6, 43]. In [15] the authors used instrumented home gateways to look at IoT traces from over 200 households in a US city. Their analysis revealed that while the IoT space is fragmented, few popular cloud and DNS services act as a central hub for the majority of the devices and their data.

Generally, many IoT devices periodically connect to specific servers on the Internet. Authors in [26] and [34] proposed a method to identify IoT devices by observing passive DNS traffic and unique IP addresses that the device connects to. Unfortunately, many IoT devices rely on shared infrastructures and often different IoT devices from the same vendor connect to the same servers, therefore detection at the scale of ISP/IXP, based on the IP addresses and port numbers without considering the important role of shared infrastructures, cannot be very reliable.

Complementing the approaches based on testbeds and home gateways, there have been efforts in understanding IoT traffic patterns using data from transit networks [44], though it has been challenging to successfully validate the derived signatures. Similar works relied on specific port numbers [45] that may also be used for specialized industrial IoT systems [46], though the approach used cannot be easily extended to general-purpose IoT devices and smart home systems that utilize popular ports, e.g., 443, 80.

These related works indicate that often, neither data from core networks subject to sampling and middleboxes, nor data from few devices using home gateways or testbeds are enough for rapidly and accurately detecting IoT devices, and understanding their anomalies and misconfigurations [10].

In this paper, for the first time we have complemented detailed ground truth data from testbeds and a particular subscriber, with large-scale data from an ISP and an IXP, to reveal the aggregate behavior of these devices, alongside the ability to isolate and identify specific subscriber devices using sampled data at an ISP.

9 CONCLUSION

Home IoT devices are already popular, and their usage is expected to grow further. Thus, we need to track their deployment without deep packet inspection or active measurements, both intrusive and unscalable methods for large deployments. Our insight is that many IoT devices contact a small number of domains, and, thus, it is possible to detect such devices at scale from sampled network flow measurements in very large networks, even when they are in idle mode. We show that our method is able to detect millions of such devices in a large ISP and in an IXP that connects hundreds of networks.

Our technique is able to detect 4 IoT platforms, 20 manufacturers and 11 products–both popular and less popular ones–at vendor level and in many cases even at product granularity. While this detection may be useful to understand the penetration of IoT devices at home, it raises concerns about the general detectability of such devices and the corresponding human activity.

In light of our alarming observations, as part of our future work, we would like to investigate how to minimize the harm of potential attacks and surveillance using IoT devices. We also want to use our insights to help ISPs to tackle security and performance problems caused by IoT devices, e.g., by detecting them, redirecting their traffic, or blocking their traffic.

ACKNOWLEDGEMENTS

We thank the anonymous reviewers and our shepherd Kensuke Fukuda for their constructive feedback. This work was supported in part by the European Research Council (ERC) Starting Grant ResolutioNet (ERC-StG-679158), the EPSRC Defence Against Dark Artefacts (EP/R03351X/1), the EPSRC Databox (EP/N028260/1), and the NSF (CNS-1909020).

REFERENCES

[1] IoT Analytics. IoT 2019 in Review: The 10 Most Relevant IoT Developments of the Year. https://iot-analytics.com/iot-2019-in-review/, 2020.
[2] S. Greengard. Deep Insecurities: The Internet of Things Shifts Technology Risk. *Comm. of the ACM*, 62(5), 2019.
[3] D. J. Dubois, R. Kolcun, A. M. Mandalari, M. T. Paracha, D. Choffnes, and H. Haddadi. When Speakers Are All Ears: Characterizing Misactivations of IoT Smart Speakers. In *Privacy Enhancing Technologies Symposium (PETS)*, 2020.
[4] J. Ren, D. J. Dubois, D. Choffnes, A. M. Mandalari, R. Kolcun, and H. Haddadi. Information Exposure From Consumer IoT Devices: A Multidimensional, Network-Informed Measurement Approach. In *ACM IMC*, 2019.
[5] L. F. DeKoven, A. Randall, A. Mirian, G. Akiwate, A. Blume, L.K. Saul, A. Schulman, G.M. Voelker, and S. Savage. Measuring Security Practices and How They Impact Security. In *ACM IMC*, 2019.
[6] S. Marchal, M. Miettinen, T. D. Nguyen, A.-R. Sadeghi, and N. Asokan. AUDI: Towards Autonomous IoT Device-Type Identification using Periodic Communication. *IEEE Journal on Sel. Areas in Comm.*, 37(6), 2019.
[7] D. Kumar and K. Shen and B. Case and D. Garg and G. Alperovich and D. Kuznetsov and R. Gupta and Z. Durumeric. All Things Considered: An Analysis of IoT Devices on Home Networks. In *USENIX Security Symposium*, 2019.
[8] Z. Durumeric, E. Wustrow, and J. A. Halderman. ZMap: Fast Internet-Wide Scanning and its Security Applications. In *USENIX Security Symposium*, 2013.
[9] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. A. Halderman. A Search Engine Backed by Internet-Wide Scanning. In *ACM CCS*, 2015.
[10] H. Haddadi, V. Christophides, R. Teixeira, K. Cho, S. Suzuki, and A. Perrig. Siotome: An edge-isp collaborative architecture for iot security. In *1st International Workshop on Security and Privacy for the Internet-of-Things (IoTSec)*, 2018.
[11] B. Claise. RFC 3954: Cisco Systems NetFlow Services Export Version 9, 2004.
[12] B. Claise, B. Trammell, and P. Aitken. RFC 7011: Specification of the IPFIX Protocol for the Exchange of Flow Information, 2013.
[13] Amazon. Alexa Voice Service Endpoints (accessed 2019-11). https://developer.amazon.com/en-US/docs/alexa/alexa-voice-service/api-overview.html#endpoints.
[14] P. Patel, G. Srinivasan, S. Rahaman, and I. Neamtiu. On the Effectiveness of Random Testing for Android: Or How I Learned to Stop Worrying and Love the Monkey. In *Proceedings of the 13th International Workshop on Automation of Software Test*, 2018.
[15] M. Hammad Mazhar and Z. Shafiq. Characterizing Smart Home IoT Traffic in the Wild. In *ACM/IEEE Conference on Internet of Things Design and Implementation*, 2020.
[16] Farsight Security. DNSDB. https://www.dnsdb.info/, 2017.
[17] C. Iordanou, G. Smaragdakis, I. Poese, and N. Laoutaris. Tracing cross border web tracking. In *ACM IMC*, 2018.
[18] F. Weimer. Passive DNS Replication. In *17th Annual FIRST Conference*, 2005.
[19] Amazon AWS. What is Amazon VPC? (accessed 2019-11). https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html.
[20] Amazon AWS. Public IPv4 addresses and external DNS hostnames (accessed 2019-11). https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-instance-addressing.html#concepts-public-addresses.
[21] Microsoft. Public IP addresses (accessed 2019-11). https://docs.microsoft.com/en-us/azure/virtual-network/virtual-network-ip-addresses-overview-arm#public-ip-addresses.
[22] Tuya Inc. Tuya Platform and Services (accessed 2019-12). https://www.tuya.com/platform.
[23] Electric Imp. Electric imp Platform (accessed 2019-12). https://www.electricimp.com/platform/how-it-works/.
[24] Amazon. AWS IoT Platform (accessed 2019-12). https://aws.amazon.com/iot/.
[25] AuraK, Samsung Community Moderator. Backgroundverbindungen (auch Standby), Datenschutz - in German(accessed 2019-11). https://eu.community.samsung.com/t5/TV/Backgroundverbindungen-auch-Standby-Datenschutz/m-p/625473/highlight/true\#M24445, July 2018.
[26] R. Perdisci, T. Papastergiou, O. Alrawi, and M. Antonakakis. IoTFinder: Efficient Large-Scale Identification of IoT Devices via Passive DNS Traffic Analysis. In *IEEE European Symposium of Security and Privacy*, 2020.
[27] Bitkom e.V. Zukunft der Consumer Technology – 2019 - in German (accessed 2019-11). https://www.bitkom.org/sites/default/files/2019-09/190903_ct_studie_2019_online.pdf, 2019.
[28] IDC. Google Overtakes Amazon to Lead the European Smart Home Market in 1Q19, says IDC (accessed 2019-11). https://www.idc.com/getdoc.jsp?containerId=prEUR145337319, 2019.
[29] Deutsche Welle. Voice Assistants on the rise in Germany (accessed 2019-11). https://www.dw.com/en/voice-assistants-on-the-rise-in-germany/a-45269599.
[30] Amir H Rasti, Nazanin Magharei, Reza Rejaie, and Walter Willinger. Eyeball ASes: from Geography to Connectivity. In *ACM IMC*, 2010.
[31] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou. Understanding the Mirai Botnet. In *USENIX Security Symposium*, 2017.
[32] O. Çetin, C. Gañán, L. Altena, T. Kasama, D. Inoue, K. Tamiya, Y. Tie, K. Yoshioka, and M. van Eeten. Cleaning Up the Internet of Evil Things: Real-World Evidence on ISP and Consumer Efforts to Remove Mirai. In *NDSS*, 2019.
[33] A. M. Mandalari, R. Kolcun, H. Haddadi, D. J. Dubois, and D. Choffnes. Towards Automatic Identification and Blocking of Non-Critical IoT Traffic Destinations. In *IEEE S & P Workshop on Technology and Consumer Protection*, 2020.
[34] H. Guo and J. Heidemann. Detecting IoT Devices in the Internet. *IEEE/ACM Transactions on Networking*, 2020. [to appear].
[35] Google. DNS-over-TLS. https://developers.google.com/speed/public-dns/docs/dns-over-tls, 2020.
[36] F. Chen, R. K. Sitaraman, and M. Torres. End-User Mapping: Next Generation Request Routing for Content Delivery. In *ACM SIGCOMM*, 2015.
[37] E. Lear, R. Droms, and D. Romascanu. RFC 8520: Manufacturer Usage Description Specification, 2019.
[38] D. A. Popescu, V. Safronov, P. Yadav, R. Kolcun, A. M. Mandalari, H. Haddadi, D. McAuley, and R. Mortier. Sensing the IoT network: Ethical capture of domestic IoT network traffic: poster abstract. In *ACM SenSys posters*, 2019.
[39] N. Apthorpe, D. Reisman, and N. Feamster. A Smart Home is No Castle: Privacy Vulnerabilities of Encrypted IoT Traffic. *Data and Algorithmic Transparency Workshop*, 2016.
[40] A. Sivanathan, H. H. Gharakheili, F. Loi, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman. Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics. *IEEE Transactions on Mobile Computing*, 18(8), 2019.
[41] D. Y. Huang, N. Apthorpe, G. Acar, F. Li, and N. Feamster. IoTInspector: Crowdsourcing Labeled Network Traffic from Smart Home Devices at Scale. In *ACM IMWUT / Ubicomp*, 2020.
[42] N. Apthorpe, D. Y. Huang, D. Reisman, A. Narayanan, and N. Feamster. Keeping the Smart Home Private with Smart(er) IoT Traffic Shaping. *Proceedings on Privacy Enhancing Technologies*, 2019.
[43] A. Sivanathan, H. H. Gharakheili, and V. Sivaraman. Inferring IoT Device Types from Network Behavior Using Unsupervised Clustering. In *IEEE Conference on Local Computer Networks (LCN)*, 2019.
[44] G. Hu and K. Fukuda. Toward Detecting IoT Device Traffic in Transit Networks. In *International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 2020.
[45] A. Sivanathan, H. H. Gharakheili, and V. Sivaraman. Can We Classify an IoT Device using TCP Port Scan? In *2018 IEEE International Conference on Information and Automation for Sustainability (ICIAfS)*, pages 1–4, 2018.
[46] M. Nawrocki, T. C. Schmidt, and M Wählisch. Uncovering Vulnerable Industrial Control Systems from the Internet Core. In *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, 2020.