Bridging Devices and Apps: A Joint Analysis of IoT Privacy and Communication

Carlotta Tagliaro[1], Martina Komsic[1], Gianluca Anselmi[2], Anna Maria Mandalari[2], and Martina Lindorfer[1]

[1] TU Wien, Vienna, Austria
{carlotta,martina.komsic,martina}@seclab.wien
[2] University College London, London, UK
{gianluca.anselmi.22,a.mandalari}@ucl.ac.uk

Abstract. Internet of Things (IoT) devices automate everyday tasks, from managing home security systems to optimizing energy usage. Their mobile companion apps often act as a user interface and provide (remote) control for convenience's sake. However, both devices and apps can collect and transmit significant amounts of personal data.
In this paper, we perform a joint measurement study of IoT devices and their companion apps, focusing on privacy implications. We develop an approach for automated experimentation with devices and their apps to understand the extent of data sharing and how network settings (LAN vs. WAN) affect data transmission. Performing over 2,040 experiments with 34 IoT devices, our findings reveal that while a small percentage of devices directly contact third parties, the majority of companion apps do so, highlighting their significant role in posing privacy risks in the IoT ecosystem. We further compare our observed data flows with privacy policies and data-access responses, and find critical GDPR compliance issues. Our study highlights the importance of increased transparency and enforcement of data protection across components of the ecosystem.

Keywords: Internet of Things · Mobile companion apps · Network traffic analysis · Privacy · Tracking and analytics

1 Introduction

Internet of Things (IoT) devices have entered our homes and are helping us with everyday tasks, ranging from vacuuming the floors to smart alarm systems. To do so, they collect, process, and store a conspicuous amount of sensitive data about the surrounding environment and their users. Making matters more complex is that IoT devices are not the only players in this ecosystem. Since devices often lack user interfaces and controls, mobile companion apps provide this functionality and can operate the devices remotely, even when they are not in their proximity. To this end, they mediate the communication between the devices and (cloud) endpoints. Consequently, companion apps also process and store sensitive information collected from the devices and their endpoints. This

crucial role in the IoT ecosystem not only threatens users' privacy but also the correct functioning of the devices themselves.

To observe whether information leaks occur, prior work has analyzed network traffic generated by IoT devices to identify *event-based* patterns and communication with third parties [25, 31, 46]. Some identified advertisers and trackers are specific to the IoT ecosystem [25, 38, 48], in contrast to the mobile app ecosystem, where advertisements (ads) and user targeting have long been an active field of research [35, 39, 47]. In the mobile app space, users might reasonably assume that they "get what they pay for," i.e., that paid apps offer more privacy protections than free apps, which might not always be the case [19]. In the IoT ecosystem, consumers already pay significant sums for devices but still need to install companion apps to unlock their full functionality. Thus, there is an assumption that the business model should center on selling devices rather than users' data. However, related studies showed that these apps and even devices themselves include ads and tracking [25, 38, 48].

We perform the first large-scale comprehensive investigation of the *joint analysis of app-device behavior* between companion apps and IoT devices, i.e., we correlate app and device traffic collected under identical conditions and compare outcomes across the wide area network (WAN), i.e., over the Internet, vs. the local area network (LAN), i.e., the users' local home wireless network (Wi-Fi). With this joint perspective, we can quantify not only *who leaks*, but also *how architectural decisions*, such as app mediation versus direct device communication, amplify or mitigate privacy exposure. We define *local communication* as data exchanged solely within private address spaces (RFC 1918 [37]) without involving remote cloud servers. We use the term *tracking & analytics endpoints* to denote remote network destinations associated with advertising, analytics, or user-behavior profiling services (e.g., Google Analytics, Braze).

First, we analyze whether there are systematic differences in the endpoints accessed by devices and apps. We also investigate whether apps tend to contact more analytics services and trackers than devices do, and if performing specific actions on the device (e.g., turning a device on/off) triggers such requests.

Second, we study communication patterns across network settings. The European Union Agency for Cybersecurity (ENISA) [12] and the National Institute of Standards and Technology (NIST) [30] recommend that local communication should occur without routing traffic through cloud endpoints. We study whether this holds by comparing network traffic when apps and devices are connected to different networks (WAN) or to the same network (LAN).

Third, we investigate which information the companion apps share. They not only help control the devices but can also enrich the information the devices collect through their sensors and provide to the app. In turn, sharing this data with third parties, such as advertisers and trackers, can severely impact users' privacy. Thus, we look for Personally Identifiable Information (PII) and any IDs that can identify a user, determining whether they are reused across devices and endpoints. Finally, we also check for inconsistent behaviors by comparing the information we collect with what the vendors claim in their privacy policies.

In summary, we answer the following research questions throughout our study:

[**RQ1**] **How do devices and apps communicate?** We develop a rigorous, reproducible methodology to explore the functionality of IoT devices and record the traffic generated by the interaction of 34 devices *in conjunction with* their companion apps. We show that while only three devices contact tracking & analytics endpoints (8.82%), 25 companion apps (89.29%) do so. Thus, we identify apps as the main culprit in introducing tracking in the IoT ecosystem.

[**RQ2**] **How does the communication change in the LAN?** We compare the communication between the IoT device and its companion app on the LAN versus the WAN to identify behaviors that may violate best practices. We observe local communication in 33 devices (97.06%), but only 19 devices (55.88%) generate more traffic to local endpoints in the LAN. Local communication can reduce exposure to remote endpoints, but only when actually used and secured.

[**RQ3**] **Do IoT devices' data collection and sharing practices align with their privacy policies, and what specific data is collected and shared?** We examine the devices' privacy policies to assess compliance with relevant regulations. We identify one case in which the same advertiser ID is shared across three devices from three vendors and three endpoints, enabling aggregated user tracking. When we requested copies of the collected data from vendors, we received only 15 responses out of 31 requests. All remaining vendors violate European data protection laws. Additionally, we found inconsistent data-sharing practices across six of the 15 devices for which we obtained data.

Artifacts. To foster future research on this topic we make our code and data publicly available at `https://github.com/SafeNetIoT/iot-vs-mobile`.

2 Methodology

2.1 IoT Testbed

We build our IoT testbed on top of the `Mon(IoT)r` [25, 38] infrastructure in the UK. As Figure 1 illustrates, the testbed consists of: (1) two Android phones, connected to the Internet through the local or remote access point (AP), with the companion apps installed and controllable via Android Debug Bridge (ADB); (2) 34 IoT devices under test, connected via the local AP; (3) a local AP that provides IP connectivity from the ISP to the IoT devices and phones and captures all network traffic; (4) a remote AP that provides IP connectivity from the ISP only to the phone for WAN experiments.

❶ **Mobile Phones.** We use two rooted Google Pixel 3A (Android 12) with `frida-server` (v16.1.1) [16] and `PCAPdroid` [32] installed. PCAPdroid is an open-source app that allows filtering traffic by app package ID, effectively removing background traffic generated by the OS. We modified it to run with root privileges at all times, without requiring confirmation, to facilitate automation.

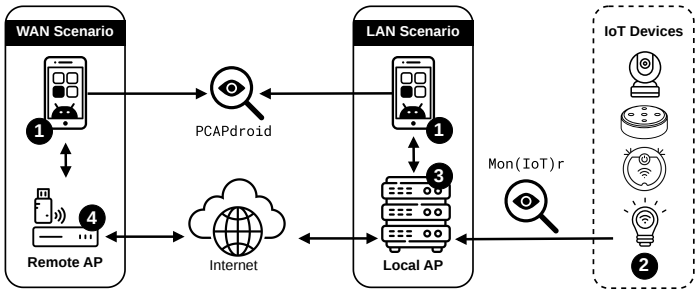

Fig. 1. IoT Testbed Setup. We use two access points (APs) to test different network configurations, one local (LAN) where apps and devices can communicate in the same network, and one remote (WAN) where they have to communicate over the Internet.

❷ **IoT Devices.** Table 5 (in the Appendix) provides the list of the 34 devices we test, along with the specific functionality we evaluate for each device. All devices support Wi-Fi connectivity and have a corresponding companion app (which we install on both phones). The devices were collected by scraping the Amazon top-seller lists for smart devices and selected based on availability and category coverage (e.g., camera, light bulb, vacuum cleaner, doorbell, etc.).

❸ **Local AP.** We use a machine running Ubuntu 22.04 with `frida-tools` (v16.1.1) [16] and `mitmproxy` (v10.0.0) [28] to provide connectivity for the IoT devices. We configure the AP with `Mon(IoT)r` [38] to capture network traffic. The AP gathers network traffic from connected devices, organizing the traffic by device, creating targeted experiments, and marking them with keywords.

❹ **Remote AP.** We utilize another machine running Ubuntu 22.04 on a separate network from our primary AP. We install the same software on this machine. It provides Internet connectivity to phones via a USB Wi-Fi adapter, enabling testing of IoT behavior in a WAN scenario where IoT devices and phones are connected to different networks and communicate via the Internet.

2.2 Test Setup & Interactions

We designed automated experiments with the companion apps interacting with our IoT devices, focusing on a replicable procedure to avoid manual effort. We only manually test the apps once, but can re-run the interaction many times, similar to IoTrigger/IoTrimmer [25].

We document all manual and automated interactions to ensure replicability. We first explore each companion app manually to identify core functionalities (e.g., power toggle, camera view, or playback). We record the action sequence once via screenshots and coordinate logging, and automatically replay it through ADB scripts for all subsequent runs to guarantee consistent inputs across experiments. Two researchers verify that each replay reached the intended app state and produced consistent network traces, checking against silent failures.

As most traffic is encrypted, we run a second round of experiments with `mitmproxy` [28] in transparent mode and `frida` [16] to bypass certificate pinning and machine-in-the-middle (MITM) traffic. This gives us plaintext access to analyzable app traffic, but not for encrypted device payloads. This means there is an asymmetry between our visibility into app and device traffic: we fully observe endpoints and plaintext payloads for apps; for devices, we measure destinations, timing, bytes, and protocols, but not encrypted payload contents. However, addressing this limitation would require IoT firmware modifications and installing custom certificates, which does not scale and risks bricking devices.

We repeat each interaction 20 times with a 5-minute timeout between tests, parallelize experiments on two Android phones, and conduct all analyses between January and February 2024 in the UK, setting the region to the European Economic Area (EEA) when available.

2.3 Data Augmentation

Reverse DNS Lookup. We check for unmatched IPs, i.e., IPs for which we cannot find the resolving hostname in the recorded DNS responses. We run reverse DNS queries against VirusTotal [49] and Domaintools [11] to gather historical data. In case of a hit, for each IP, we look for full domain matches between the hostnames returned by the queried services and the endpoints we previously obtained through DNS responses at device granularity, i.e., we only check if the returned endpoint matched one of the hostnames we found for the specific device(s) contacting the IP. If we cannot find a full match, we match the Top-Level Domain (TLD). Two researchers independently verify the results and resolve conflicts. In some cases, we cannot find a reasonable match; thus, we leave the IP as "unresolved" (48 and 27 IPs for apps and devices, respectively).

Categorization of Endpoints. We start with the approach by Ren et al. [38, 39], which flagged endpoints as first, support, or third party based on popular ad blocking lists. We additionally employ Exodus [14] and other well-known tracking blocklists [1, 10]. Note that for this analysis we remove unresolved IPs. Two researchers independently check uncertain cases using domain ownership and service descriptions. Conflicts are resolved by a third researcher. For example, if an organization offers marketing insights into customer activities, we flag its endpoint as "Analytics" (e.g., Google Analytics). Overall, we use three labels: (1) *First Party* for vendor-controlled endpoints, (2) *Support Party* for auxiliary services such as login, cloud hosting, crash reporting, or content delivery, (3) *Tracking & Analytics* for advertising, analytics, or user-behavior profiling.

3 RQ1: App vs. Device Traffic Analysis

We answer RQ1 by assessing whether companion apps and devices communicate differently and provide an overview in Table 1. We base our results on the traffic captured during our experiments in the WAN scenario.

Table 1. Distribution of Endpoint Categories. We list the number (#) of apps and devices contacting at least one endpoint per category, along with remote and local IP addresses that we could not associate with a domain name.

	Companion Apps		**IoT Devices**	
	Endpoints	# Apps	Endpoints	# Devices
First Party Domains	106 (27.82%)	28 (100.00%)	54 (25.35%)	18 (52.94%)
Support Party Domains	161 (42.26%)	25 (89.29%)	87 (40.84%)	25 (73.53%)
Tracking & Analytics Domains	58 (15.22%)	25 (89.29%)	5 (2.35%)	3 (8.82%)
Remote IP Addresses	48 (12.60%)	11 (39.29%)	27 (12.68%)	11 (32.35%)
Local Addresses	8 (2.10%)	28 (100.00%)	40 (18.78%)	32 (94.12%)
Total	381	28	213	34

Overall, we collect 677 unique endpoints for the apps and 217 for the devices. Initially, we find 212 (31.31%) and 70 (32.26%) unmatched IP addresses in the traffic. After running our reverse DNS analysis (see Section 2), only 48 (7.09%) and 27 (3.23%) IPs remain "unmatched" for apps and devices, respectively. As a result, we identify 381 unique endpoints contacted by the companion apps and 213 by the devices. On average, each app contacts 19.97 endpoints (σ = 12.07; maximum of 53 endpoints contacted by the app of the *SwitchBot* hub) versus 8.59 endpoints per device ($\sigma = 8.32$), i.e., 2.3× more endpoints per app. *Geree* (a doorbell) is the device that contacts the most endpoints, 37 unique ones. However, in general, devices tend to contact fewer endpoints, possibly because of their simpler software and fewer functionalities. By contrast, apps are often more feature-rich, providing account management and social media sharing, thereby increasing the number of endpoint destinations. This aligns with our overall results summarized in Table 1, where apps also contact more *Support* and *Tracking & Analytics* domains.

Upon further investigation, we find that among the 381 endpoints in app traffic, only eight (2.10%) are local IPs, compared to 40 (18.78%) among the 213 IoT device endpoints. Devices contact more local IPs, as also discussed by Girish et al. [18], because they use local protocols to discover, connect to, and manage other IoT devices. In practice, this includes discovery and control traffic (e.g., mDNS, local HTTP, or MQTT) that remains within the home network.

We also find that apps and devices share 94 unique endpoints, which represents a relatively low share overall. On average, apps and devices share only 2.76 endpoints ($\sigma = 6.31$), with 19 app-device pairs not sharing any endpoints. If we consider IPs, then apps and devices share 124 unique IPs, with an average of 4 per pair ($\sigma = 7.64$). While we expect apps to interact with more services, as discussed above, we do not anticipate that devices would utilize different backend infrastructures to communicate with the cloud. The reason could be that subdomains are reserved for companion apps or devices, separating the app infrastructure from the device infrastructure.

3.1 Country Distribution

Despite our testbed being located in Europe, most endpoints that apps and devices contact are US-based, with 115 (30.75%) and 44 (25.29%), respectively.

Regarding the endpoint location of devices, if we exclude those located in the US, all the others are within European borders, except for one in Canada. Instead, for apps, the fifth-most-recurring country is China, with 14 (3.74%) endpoints. While five are Alibaba-hosted endpoints, seven are flagged by denylists [1] as tracking and analytics. Sending data to tracking services in countries with more lax data protection regulations can pose serious privacy risks.

When looking at the amount of traffic, the country where most data is to directed for apps and devices is the UK, with 1.21 GB and 102.76 MB, respectively. The US only ranks fourth for both, with 130.45 MB and 8.26 MB, and is preceded only by European countries. These numbers highlight the importance of a multidimensional analysis and demonstrate that, despite most endpoints being US-based, more data is stored within EEA jurisdictions.

3.2 Endpoint Categories

We find that while all apps contact at least one *First Party* endpoint, only half of the devices do so (52.94%), mostly relying on *Support Party* endpoints. Relying on *First Party* endpoints would not only be more efficient but also provide greater privacy, as users' data is less likely to be shared with multiple external parties. Organizations have full control over how data is managed, stored, and secured when using *First Party* endpoints. From the users' perspective, identifying the accountable party is also easier in the event of a data breach.

Only three devices (8.82%) contact *Tracking & Analytics* domains: two Amazon and one Google device contacting in-house metrics and analytics endpoints. The *Echo v4* leads with three. Companion apps, by contrast, introduce the most tracking in the ecosystem. Apps often provide additional features, sometimes from third parties, and process more data, making them more appealing to trackers. We find analytics services in 25 apps (89.29%, 58 endpoints - 15.22% of the total). On average, apps contact 4.03 tracking endpoints ($\sigma = 3.89$). The *SwitchBot* app contains the most tracking endpoints, with 16. Apps also share more data in total with such endpoints, 35 MB, instead of devices, 694 kB.

To determine whether a correlation exists to actions triggered by devices, we compute Cumulative Distribution Function (CDF) plots correlating time with the number of bytes sent to *Tracking & Analytics* endpoints. In eight app-device cases, corresponding to six apps (21.43%), traffic to these endpoints increases rapidly while we interact with the device/app, then plateaus when our interaction ends. By analyzing the traffic we capture with `Frida` and `mitmproxy`, we can identify what we refer to as *event-based* tracking. With that, we mean instances where we find payload values in the traffic that we associate with the activity we perform on the app at that specific moment, e.g., turning a device on/off. Some concrete examples are *"Dashboard Slide Tapped"* or *"Tag Event Video Subscribe Cam Plus Clicked."* To confirm our findings, we perform additional experiments without any interaction and record traffic for one hour on each device, where we initially find *event-based* tracking. Indeed, these second captures confirm that tracking endpoints receive traffic primarily when actively interacting with the app, and that *event-based* payloads appear only during that stage.

This type of analytics and tracking poses significant risks to users' privacy. Services that receive such information can analyze users' interaction behaviors and infer critical information to better tailor content and advertising. Additionally, four out of the six apps in question rely on third-party services for this purpose. As a result, vendors and their customers must trust third-party services to handle, store, and process their data, raising serious regulatory and security concerns. This risk is already a reality: in August 2025, a US jury found Meta guilty of violating wiretap laws after the company collected personal information and button taps from a period-tracking app and used that data in its advertising system [4]. This behavior is similar to the bursts of user-event data that our study observes apps sending to third-party trackers.

Our results indicate a predominance of app-based tracking, which contrasts with the findings of Jakaria et al. [21]: their devices connect to more third-party endpoints than to first- and support-party ones. Several factors may explain this discrepancy: (1) their dataset of 25,123 IoT devices leverages IoT Inspector. This monitoring tool allows users to analyze their devices' traffic and send anonymous data to the IoT Inspector server. This dataset also includes gaming consoles and smart TVs, which are more likely to interact with tracking endpoints. (2) Their endpoint classification significantly differs from ours. After classifying first- and support-party endpoints using NLP, they categorize all other endpoints as third-party, e.g., endpoints that "may represent any third-party app or skill for high-end devices such as voice assistance or smart TV." Instead, we aim for a more nuanced understanding of these third parties. (3) Regional differences can also be a crucial factor, as most of their devices are based in the US.

4 RQ2: LAN vs. WAN Scenarios

We compare traffic when phones and devices are on different networks (WAN) or on the same network (LAN). We expect them to prefer local communication to avoid relying on cloud providers, thereby conserving resources and speeding up communication. Here, we observe 520 app endpoints and 245 device endpoints, corresponding to a 23.19% decrease app-side, but a 12.90% increase device-side. We observe only a slight increase in local IP addresses contacted by devices (i.e., from 40 to 47), suggesting that the new endpoints mostly belong to LAN discovery or control paths rather than an increased dependency on remote services.

4.1 Local Communication

We flag traffic from and to a local IP as local communication and count the number of bytes sent to local IPs. We find local communication in 33 devices (97.06%). Nineteen devices (55.88%) show increased local communication in the LAN scenario, signaling that traffic (e.g., commands) is not passing through cloud endpoints. Apart from the *Withings* blood pressure monitor, the remaining 13 devices (38.23%) have the same amount of local communication in both scenarios. Figure 2 shows the ratio of remote communication for each device in

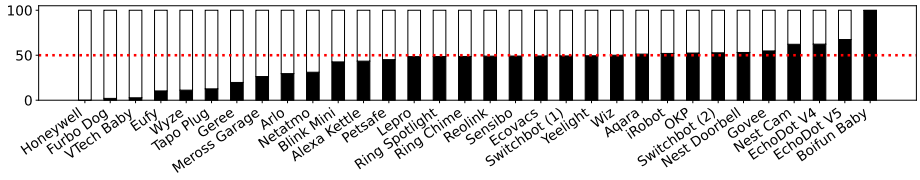

Fig. 2. Communication with Remote Endpoints. We show the normalized ratio of remote communication for each device in the LAN ■ or WAN □ scenarios. The red dotted line indicates the 50% ratio; i.e., the device has the same amount of remote communication in both. All devices, except the *Boifun* baby monitor, either have the same amount of remote communication in both scenarios or more in the WAN one.

both scenarios. All devices, except the *Boifun* baby monitor, either have the same amount of remote communication or have more in the WAN scenario.

Keeping traffic within the local network rather than routing it over the Internet not only improves the user experience by decreasing latency and providing bandwidth savings, it also enhances security and privacy: attackers outside the network cannot intercept the data, and data is not shared with potentially untrusted or insecure cloud servers [45]. From a regulatory perspective, local communication may simplify compliance efforts. Local links also let devices keep working if the Internet connection goes down, matching ENISA's guidance that "essential features should continue to work with a loss of communications" [12].

4.2 Network Protocols

HTTPS is first in both scenarios. We see more use of its insecure counterpart, HTTP, in the LAN scenario (9 vs. 14), which weakens confidentiality even when data remains local. Message Queuing Telemetry Transport (MQTT) is second, with 20 endpoints communicating on its secure port (8883). In contrast, we find only 10 instances of it in our WAN scenario. Multicast DNS (mDNS) ranks third in the LAN scenario, with 15 devices, but appears only once in the WAN scenario. Although HTTPS remains at the top for device traffic, we find BOOTP client/server to be second and third, respectively, in both scenarios. These consist of bootstrapping protocols that automatically assign an IP address to a network device. We also observe repeated flows on TCP port 9100 for *Boifun* and *Geree*, directed to `turnsvr*.mearicloud.com`. Packet inspection indicates these correspond to TURN-relayed audio/video sessions (not printing), suggesting that some camera backends repurpose uncommon ports to improve reachability through home networks. Again, for devices, we observe more instances of MQTT over TLS (port 8883) in the LAN scenario, suggesting that some communication patterns have changed, possibly relying more on local communication. HTTP is also more common. While the LAN may seem safer than the Internet, the increased use of HTTP might pose more risks to users. An attacker who compromises another device on the same network could intercept traffic between IoT devices and apps, potentially capturing credentials or other PII.

5 RQ3: PII + ID Sharing & Compliance

Mobile apps are known for sharing PII of different nature, e.g., geolocation, age, and gender, possibly but not limited to for analytics and tracking purposes [7, 35, 52]. To study the relationship between IoT devices and companion apps, we inspect payloads and API calls. Because device-side traffic is encrypted, we can only inspect the *plaintext* packets and cannot find any PII. For readability, in the following we discuss each finding based on the *device name* rather than the app ID (e.g., package name) of the corresponding companion app. Further note that multiple devices from a single vendor can share one companion app but expose distinct UI paths, triggering different request sequences.

5.1 PII Sharing

To extract PII from the collected traffic, we employ a combination of regular expressions and keyword analysis. We define as PII (1) *Device Identifiers*, e.g., Mobile Equipment Identifier (IMEI), Advertiser ID, MAC address, Service Set Identifier (SSID), (2) *User Identifiers*, which serve to identify a specific user, such as name, date of birth, email address, phone numbers, (3) *Location*, e.g., geographical coordinates and IPs, and (4) *Credentials*, i.e., username and password. We adopt a similar list to Ren et al.'s [40] and enrich it with online resources [26]. Since field names are not standardized (e.g., `name`, `firstname`, or `surname`), we use regexes to capture variants and standard formats such as IP or MAC addresses. We acknowledge that our approach might miss some values if field names differ significantly from the PII list we identify or are not included in it. We manually validate our findings against the *PetSafe* and *Meross* devices to refine our regular expressions and also verify the extracted information to eliminate false positives.

Table 2 summarizes our results. Fourteen devices (41.18%) adopt the same behavior in both scenarios. Overall PII sharing increases in the LAN setting. For example, the *Tapo P110* app sends geolocation data to their *First Party* endpoints in this scenario. Several other devices also send more PII to remote endpoints, for instance, *Arlo*, *PetSafe*, *Nest Doorbell*, *Eufy*, *OKP*, and *Tapo P110*, rather than keeping it local. Only *Meross* increases PII shared with a local endpoint (`10.13.0.12`) in the LAN scenario. In our experiments, cameras and doorbells leak the most data: *Boifun*, *Geree*, *Wyze*, and *Blink* disclose multiple identifiers such as email addresses, user names, MAC or SSID, and embed them in query strings, which enables persistent cross-session tracking, even after app restarts. Several apps (*Honeywell*, *Netatmo*, *Wyze*) also upload detailed crash logs that include stack traces and library versions.

We also recover four passwords shared with *First Party* endpoints; two are sent in plaintext, and two are encrypted. While manually checking the data, we find API keys in the payloads from seven apps. Among them, three transmit the API key as a URL parameter. Although TLS protects data in transit, placing an API key in the query string remains unsafe. Servers and reverse proxies usually log the entire URL, and analytics or crash-reporting may also capture it.

Table 2. PII Sharing in LAN and WAN Scenarios. We list the number (#) of apps and fraction (%) of devices for which we extracted at least one type of PII. We see increased sharing of PII in the LAN scenario.

PII	WAN Scenario		LAN Scenario		Increase
	# Apps	% Devices	# Apps	% Devices	WAN→LAN
MAC	6	17.65%	8	23.53%	+5.88%
SSID	3	8.82%	5	14.71%	+5.89%
Email Address	15	44.12%	18	52.94%)	+8.82%
Name	5	14.71%	7	20.59%	+5.88%
Geolocation	5	14.71%	6	17.65%)	+2.94%
IP	7	20.59%	9	26.47%	+5.88%
Username	3	8.82%	3	8.82%	+0.00%
Password	2	5.88%	4	11.76%	+5.88%
Encrypted Data	8	23.53%	8	23.53%	+0.00%
No PII Found	10	29.42%	7	20.59%	-8.83%
No Data Extracted	6	17.65%	6	17.65%	–

After collecting all app-shared PII, we examine what they share with *Tracking & Analytics* endpoints. Because most payloads are encrypted, we cannot inspect their contents. Nonetheless, the *Furbo Dog* app sends base64-encoded data, along with the user's email, to a *Tracking & Analytics* endpoint. Generally, more apps leak PII to analytics endpoints in the LAN scenario, which aligns with our earlier findings that more apps share PII in this scenario.

5.2 ID Analysis

To identify standard identifiers, we use the Android app `Device ID` (v1.2.1) [9], which reports the Android ID, Android Device ID, Google Service Framework ID, and Advertising ID. We also retrieve the IMEI numbers of both phones and search for all identifiers in the captured traffic. The only matching identifier is the Advertising ID, reported in the first entry of Table 3.

However, we notice a broad use of non-standard identifiers. We develop a regex to extract potential IDs from network requests and compare them across apps and endpoints to detect sharing or reuse, which can support data aggregation and user profiling. As before, we manually remove false positives.

We classify an ID match as interesting if (1) the same ID is transmitted to more than one endpoint, or (2) two or more unique apps share the same ID. In one case, the same ID is shared across devices from different manufacturers and endpoints, including analytics ones, as shown in Table 3. We also see apps sharing user IDs across different endpoints, including *Tracking & Analytics* ones as for *Wyze*. The shared ID that occurs across *Wiz*, *Furbo*, and *Honeywell* is rarely the only PII shared. *Furbo* pairs this ID with user identifiers and context, such as email addresses, customer IDs, device/app IDs, and even the "limit ad tracking" flag. *Honeywell* includes the same ID in Localytics with rich telemetry events (e.g., "Thermostat - Mode Changed" with old/new values). *Wyze* uses a unique UID across Braze and RudderStack that is associated with session starts and ends, as well as screen views, making it actionable.

Table 3. ID Sharing Across Devices and Endpoints. One ID may appear under different names. We distinguish two cases: (1) an ID shared by different apps *and* endpoints, or (2) one app sharing the ID with *multiple* endpoints.

	Devices	ID Names	Endpoints

Shared across apps *and* endpoints	*Wiz*
	Furbo Dog
	Honeywell
		`advertiser_id`
		`gcadid`
		`advertising_id`
			graph.facebook.com
			api.getblueshift.com
			analytics.localytics.com

	Geree *Boifun*	`id`	graph.facebook.com
Same app, one ID, multiple endpoints	*Honeywell*	`caid` `aid`	honeywelllyric.helpshift.com analytics.localytics.com
	Wyze	`UID`	ssl.kaptcha.com dxp.kaxsdc.com
	Wyze	`userId` `user_id`	wyze.dataplane.rudderstack.com sdk.iad-03.braze.com

	Furbo Dog	`customer_id`
		`user_id`
		`furbo_account_id`
			api.getblueshift.com
			sdk.iad-03.braze.com
			crashlyticsreports-pa.googleapis.com
			firebaseremoteconfig.googleapis.com

5.3 Compliance with Privacy Policies

We investigate whether the transmitted data aligns with privacy policies by checking whether (1) policies disclose all collected data and (2) tracking and analytics activities, including involved parties, are clearly identified. Here, we evaluate 31 of the 34 devices and their apps. For the remaining three, we cannot intercept analyzable traffic, and they do not contact *Tracking & Analytics* endpoints, preventing us from assessing policy alignment.

We first collect privacy policies for every companion app from the Google Play Store, noting that most are region-agnostic. Because our experiments take place in Europe, we manually look for region-specific GDPR versions of each policy (e.g., changing the language tag to "en-eu") *and*, where available, a UK version ("en-gb"). In some cases, the companion apps embed the policies or redirect to different URLs. If two policies are present, we verify their consistency; if we find discrepancies, we prioritize the in-app policy. We notice that all apps, except two, provide only general privacy policies that apply uniformly across all products. Notably, *Aqara* and *Honeywell* are the only manufacturers to offer device-specific privacy policies. General privacy policies are suboptimal since different IoT devices collect and transmit varying data types.

All but three companion apps (89.29%) display their privacy policies in the Google Play Store, and the same is true for in-app ones. Notably, the *iRobot* app includes an outdated policy from 2018, while the Google Play Store lists a more recent version. Five apps (17.85%) lack an EEA-specific policy and only offer a global one. Given the varying privacy regulations worldwide, a single policy may pose legal risks for companies that handle data across different regions.

We find that 29.42% of apps do not share any PII, either because (1) traffic remains encrypted, (2) we cannot bypass all certificate pinning, (3) the triggered functionality does not transmit PII, or (4) the app does not need PII to work. Among the inspected apps, six (17.14%) transmit data that is not direct PII but can be associated with users, including IP addresses, SSIDs, and MAC addresses.

Nineteen apps (61.29%) transmit clear PII, such as email addresses and names. Six of the 22 apps that share some data do not fully disclose their collection practices. For instance, *Geree* transmits a MAC address and SSID, while *PetSafe* collects email addresses without disclosing this practice.

Twenty-two apps (70.97%) indicate that they transmit data to tracking endpoints without specifying the exact endpoints or the type of data being shared. We find clear information about the third parties involved in data sharing in only seven cases (22.58%). Additionally, two apps (6.45%) do not disclose any data sharing practices in their privacy policies, even though our network traffic analysis shows they communicate with tracking endpoints.

We exercise GDPR access rights [17] by requesting all personal data held by the organizations, allowing us to compare the data they share with the PII and information we extract from analyzable traffic. We aim to identify discrepancies and evaluate compliance with the organization's privacy policies. We extract the email addresses of the Data Protection Officers (DPOs) from the policies. For *Boifun*, *Blink*, and *OKP*, we found neither a DPO email address nor an alternative data access portal. There is no direct contact with a DPO for Google and Amazon devices, but data should be requested via a specific form. In all other cases, we send an email to the DPO in April 2024. We send 31 requests and receive 24 replies: one automatic reply, eight incomplete follow-ups where data was promised but not provided, and 15 (62.50%) usable exports. Seven vendors did not respond after more than six months, although GDPR guidance expects responses within one month [13].

We compare the extracted data with vendor exports and policies to identify information collected beyond what vendors disclose. We find inconsistencies in six of the 15 apps for which we receive data. The *Arlo* camera transmits the SSID without disclosure. Similarly, we find the MAC address in the *Honeywell* thermostat's network traffic but not in the policy. Although we did not find any PII in the traffic of the two *Ring* devices, we find discrepancies in the data export and the privacy policy: the export includes the MAC address and phone information (model, OS, and name), which are not specified in the privacy policy.

6 Discussion

Taxonomy of Problematic Behavior. We organized the issues we identified into four categories (see Table 4), highlighting the main ways in which apps and devices expose users to risk: tracking, privacy policy mismatches, reliance on the cloud, and the reuse of identifiers across services. These categories correspond directly to our measurements: RQ1 explains why apps dominate *Tracking & Analytics*, RQ2 addresses cloud-locked behavior even in the LAN, and RQ3 reveals inconsistent disclosures and cross-domain identifier reuse.

Some apps contact four or more tracking endpoints *and* leak PII (e.g., *iRobot*, *Furbo*). They send data to several analytics and advertising services, in addition to first-party services. In other cases, data handling practices do not align with privacy policies (e.g., *Petsafe* sharing an email address even though the policy

Table 4. Taxonomy of Identified Issues. We observe excessive tracking, privacy policy mismatches, over-reliance on the cloud, and reuse of identifiers across services.

Issue	Device/Vendor
Tracker Heavy	*iRobot*, *Furbo*, *Echo 4/5*, *Wyze*
Policy Inconsistent	*Petsafe*, *Boifun*, *Arlo*, *Eufy*, *Geree*, *Honeywell*
Cloud Locked	*Nest Doorbell/Cam*, *Ecovacs*, *Ring Chime*, *Ring Spotlight*, *Arlo*
ID Reuse	*Echo 4/5*, *Wyze*, *Furbo*, *Meross*, *Honeywell*, *SwitchBot 1/2*

does not mention it). Some vendors share the same ID across multiple domains (e.g., *Wyze* 's `userId` reused across `rudderstack.com` and `braze.com`), facilitating linking of user activity across services. Finally, devices route >70% of their traffic through remote endpoints even when the phone is on the same LAN (e.g., *Ecovacs*, *Arlo*). Heavy use of the cloud increases latency, risks the exposure of sensitive data, and introduces dependencies to platforms and providers in different jurisdictions [15, 41].

Behavior per Device Type. Cameras exhibit issues in several categories. Hubs and assistants typically appear in the ID-reuse group, such as *SwitchBot* and *Echo*. Some vendors fall into more than one category: *Arlo* is both policy-inconsistent and cloud-locked; *Wyze* and *Echo* are tracker-heavy and also reuse IDs; *Honeywell* is both policy-inconsistent and reuses IDs. The tracker-heavy group contains four vendors, while the other three groups six each, suggesting that policy gaps, cloud dependence, and ID reuse are more common than tracking. However, the mix of device types within each group suggests that the behavior originates in vendor design choices and backend integrations, rather than a specific product type.

Privacy Implications. Our measurements indicate not only the presence of data transmissions but also their significant impact on privacy. For instance, we found that some companion apps transmit PII, such as email addresses, SSIDs, and MAC addresses, to third-party analytics services, including Localytics and Braze. The *Furbo Dog* and *Honeywell* apps send unique user identifiers and *event-based* telemetry to remote domains outside the EEA, thereby creating persistent behavioral profiles. These findings highlight that companion apps serve as intermediaries between home devices and advertising ecosystems, collecting sensitive data that goes beyond operational necessities. These cross-service data flows facilitate user tracking across multiple vendors, raising serious concerns about privacy and GDPR compliance. Although privacy policies are meant to inform users, they often do not align with the actual practices of the devices and apps. These policies can be misleading, implying that data is handled securely and with user consent. In reality, however, the devices may engage in extensive tracking and data sharing. Alarmingly, some vendors do not provide users access to their data even when requested. Essentially, users are frequently unaware of how their data is used, and privacy policies do not always accurately represent the behavior of IoT devices. This situation emphasizes the need for transparency and regulatory enforcement to safeguard user privacy within the IoT ecosystem.

Threats to Validity. Our visibility into device payloads is limited because decrypting device-side TLS would require firmware modification and custom

CAs, which does not scale and risks bricking devices. A more in-depth analysis of unencrypted payloads, looking for API patterns and PII leakage, could reveal more differences or similarities between the apps' traffic and the device.

For the endpoint classification distinguishing between *First Party*, *Support Party*, and *Tracking & Analytics* is not trivial. We manually verified uncertain cases and consulted available information, including whether domain owners had websites outlining the purposes of their services. To support reproducibility, we make our labeled endpoint list available as part of our artifacts.

We conducted all our experiments in a controlled environment, with no real-user interaction. Instead, we record and replay the functionality ourselves. This might limit our exploration of apps and devices, preventing us from triggering all the traffic they can generate and potentially missing *Tracking & Analytics* endpoints. However, our setting ensures that no ethical boundaries are crossed.

We only analyze interactions between devices and their Android companion apps. We do not consider other platforms, such as iOS, which has already been shown to exhibit different behaviors [22], including in aspects related to network and IoT functionality [34, 42]. We also only analyze the default configuration of the apps. Different analytics permissions or privacy settings might change the traffic generated by the apps and devices. Finally, our analysis dates to early 2024, and the IoT ecosystem is dynamic. Vendors may have changed their practices since then, and new devices and apps may exhibit different behaviors.

Responsible Disclosure. We began our responsible disclosure process in August 2024 to inform vendors of discrepancies between their products' behavior and their policies. If we could not find a policy or a vendor did not respond to our data access request within the 30-day timeframe, we have informed the relevant national Data Protection Authorities.

7 Related Work

IoT Security and Privacy. Recent work investigated IoT companion apps [6, 8, 36, 38] looking for security and privacy issues with a combination of static and dynamic program analysis. Focusing on the interplay between companion app and devices, related work has for example used static program analysis to extract embedded sensitive information in companion apps, such as API keys and passwords [50, 52], and to characterize potential IoT device behavior, including local and remote communication with cloud backends, at scale [43]. Zhou et al. [51] studied the interactions among IoT devices, cloud services, and apps using state machines and identified invalid transitions that can lead to device hijacking. Nevertheless, most authors relied solely on the code of companion apps to enhance scalability. No studies focus on the distinction between device- and app-generated traffic, which endpoints are contacted, who is the main culprit in introducing tracking and analytics, and how the interaction changes when the mobile phone (on which the companion app is installed) and the device are in different network scenarios.

Tracking in the IoT Ecosystem. Several studies focused on identifying IoT device activities (e.g., turning on/off) by looking at network-level traffic as well as packet signatures [2, 23, 24, 27, 31, 33, 46]. Related work has also explored IoT-generated traffic to identify what role trackers and advertisers play in the ecosystem [25, 31, 46], highlighting how some are specific to the IoT domain and are missed by most general denylists, which mostly focus on the mobile app ecosystem [38, 44, 48]. In contrast to prior single-sided analyses, we jointly examine device- and app-generated traffic under identical conditions, enabling correlation across layers that prior work could not. This joint perspective is essential because companion apps often trigger device events that silently propagate to remote endpoints, amplifying privacy exposure across entities.

Privacy Policies. A long line of research looked into automatically parsing and understanding privacy policies [20]. Andow et al. [3] show that across 13,796 applications and their privacy policies, 42.4% are incorrect or omit to disclose privacy-sensitive data flows. Further, Bui et al. [5] demonstrated inconsistent behaviors between trackers and the stated policies. Focusing on IoT companion app policies, Nan et al. [29] found that among their 6,208 companion app dataset, 1,973 apps expose user data without proper disclosure. We also investigate the role of devices and how they enrich the data shared by companion apps. In fact, only two manufacturers provide a device-specific policy.

8 Conclusion

We examined the intricate dynamics between IoT devices and their companion apps, focusing on the privacy implications of their communication practices. We reveal that apps play a significant role in introducing tracking in the IoT ecosystem, often transmitting data to external analytics and tracking endpoints. We also demonstrate the importance of local communication, which, when properly secured, can mitigate privacy risks by reducing dependence on cloud services. Furthermore, our analysis of privacy policies and data sharing practices uncovered widespread inconsistencies and potential violations of data protection laws. Our results are consistent across multiple device types and manufacturers.

Vendors should adopt a LAN-by-default approach, minimize the use of third-party tracking and analytics services, and ensure that data flows align with privacy policies. Based on our findings, we argue there is a need for improved privacy standards, greater transparency, and stricter regulatory oversight in the development and deployment of the IoT ecosystem.

Acknowledgments. This work is based on research supported by netidee (Internet Stiftung Austria), the Vienna Science and Technology Fund (WWTF) and the City of Vienna [Grant ID: 10.47379/ICT19056], the Austrian Science Fund (FWF) [Grant ID: 10.55776/F8515-N], and SBA Research (SBA-K1 NGC), a COMET Center within the COMET – Competence Centers for Excellent Technologies Programme and funded by BMIMI, BMWET, and the federal state of Vienna. The COMET Programme is managed by FFG.

Table 5. Overview of Tested IoT Devices and Results. We report (1) if all tests were successful without and with `frida` (●=Fully Successful, ◐=`frida` Failed); (2) if the devices share PII (✔=Yes, ✖=No) and if they declare this in their privacy policy (✔=Yes, ✖=No, ✔=Partially); (3) if the vendor provides a means to request user data (✔=Yes, ✖=No); (5) if the vendor replied and shared the data (✔=Yes, ✖=Replied but not shared); (6) if the data in the export matches our findings or the privacy policy (✔=Yes, ✔=Partially). We highlight problematic findings in red.

	IoT Device	Scenario		PII		DPO	DPO	PII
		WAN	LAN	Shared	Declared	Email	Reply	Match
Camera	Wyze	●	●	✔	✔	✔	no reply	–
	Arlo	◐	●	✔	✔	✔	✔	✔
	Google Nest	●	●	✖	–	✔	✔	✔
	Furbo Dog	●	●	✔	✔	✔	✖	–
	Ring Spotlight	●	●	✖	–	✔	✔	✔
	Boifun Baby	◐	●	✔	✔	✖	–	–
	Vtech Baby	◐	◐	✖	–	✔	✖	–
	Blink Mini	◐	◐	✖	–	✖	–	–
Lightbulb	Wiz	●	●	✖	–	✔	✖	–
	Lepro	●	●	–	–	✔	no reply	–
	Govee	●	●	✔	✔	✔	✖	–
	Yeelight	◐	◐	✔	✔	✔	no reply	–
Robovac	Eufy 30C	◐	◐	✔	✔	✔	✖	–
	iRobot Roomba	●	●	✔	✔	✔	✖	–
	ECOVACS N8	◐	◐	✖	–	✔	✔	✔
	OKP K2P	●	●	✔	✔	✖	–	–
Doorbell	Reolink	●	●	–	–	✔	✖	–
	Google Nest	●	●	✔	✔	✔	✔	✔
	Ring Chime	●	●	✖	–	✔	✔	✔
	Geree	●	●	✔	✔	✔	no reply	–
Hub/Assistant	SwitchBot Mini (1)	●	◐	✔	✔	✔	✔	✔
	SwitchBot Mini (2)	●	●	✔	✔	✔	✔	✔
	Aqara M2	●	●	✖	–	✔	no reply	–
	Sensibo	◐	◐	–	–	✔	no reply	–
	EchoDot 4	●	●	✔	✔	✔	✔	✔
	EchoDot 5	●	●	✔	✔	✔	✔	✔
Appliance (Generic)	Netatmo Weather	●	●	✔	✔	✔	✖	–
	Honeywell Thermostat	●	●	✔	✔	✔	✔	✔
	Meross Garage	●	●	✔	✔	✔	no reply	–
	Alexa Kettle	●	●	✔	✔	✔	✔	✔
	Tapo P110	●	●	✔	✔	✔	✔	✔
	Withings BPM	●	●	✔	✔	✔	✔	✔
	PetSafe Feeder	●	●	✔	✔	✔	no reply	–
	Google Nest Wi-Fi	●	●	✔	–	✔	✔	✔

References

[1] *1Hosts*. URL: https://github.com/badmojr/1Hosts.

[2] D. Ahmed, A. Das, and F. Zaffar. "Analyzing the Feasibility and Generalizability of Fingerprinting Internet of Things Devices". In: *PETS*. 2022. DOI: 10.2478/popets-2022-0057.

[3] B. Andow, S. Y. Mahmud, J. Whitaker, W. Enck, B. Reaves, K. Singh, and S. Egelman. "Actions Speak Louder than Words: Entity-Sensitive Privacy Policy and Data Flow Analysis with PoliCheck". In: *USENIX Security*. 2020.

[4] J. Brodkin. *Jury finds Meta broke wiretap law by collecting data from period-tracker app*. 2025. URL: https://arstechnica.com/tech-policy/2025/08/jury-finds-meta-broke-wiretap-law-by-collecting-data-from-period-tracker-app/.

[5] D. Bui, B. Tang, and K. G. Shin. "Do Opt-Outs Really Opt Me Out?" In: *ACM CCS*. 2022. DOI: 10.1145/3548606.3560574.

[6] J. Chen, W. Diao, Q. Zhao, C. Zuo, Z. Lin, X. Wang, W. C. Lau, M. Sun, R. Yang, and K. Zhang. "IoTFuzzer: Discovering Memory Corruptions in IoT Through App-based Fuzzing". In: *NDSS*. 2018. DOI: 10.14722/ndss.2018.23159.

[7] A. Continella, Y. Fratantonio, M. Lindorfer, A. Puccetti, A. Zand, C. Kruegel, and G. Vigna. "Obfuscation-Resilient Privacy Leak Detection for Mobile Apps Through Differential Analysis". In: *NDSS*. 2017. DOI: 10.14722/ndss.2017.23465.

[8] M. J. Davino, L. Melo, H. Lu, M. d'Amorim, and A. Prakash. "A Study of Vulnerability Analysis of Popular Smart Devices Through Their Companion Apps". In: *IEEE SafeThings Workshop*. 2019. DOI: 10.1109/SPW.2019.00042.

[9] *Device ID*. URL: https://play.google.com/store/apps/details?id=tw.reh.deviceid.

[10] *DNS Blocklists*. URL: https://github.com/hagezi/dns-blocklists.

[11] DomainTools. *Reverse IP*. 2024. URL: https://reverseip.domaintools.com/.

[12] ENISA. *IoT Security Standards Gap Analysis*. URL: https://enisa.europa.eu/publications/iot-security-standards-gap-analysis/.

[13] European Commission. *Respect individuals' rights*. URL: https://edpb.europa.eu/sme-data-protection-guide/respect-individuals-rights_en.

[14] Exodus. *Trackers*. URL: https://reports.exodus-privacy.eu.org/.

[15] T. Fiebig, S. Gürses, C. H. Gañán, E. Kotkam, F. Kuipers, M. Lindorfer, M. Prisse, and T. Sari. "Heads in the Clouds? Measuring Universities' Migration to Public Clouds: Implications for Privacy & Academic Freedom". In: *PETS*. 2023. DOI: 10.56553/popets-2023-0044.

[16] *Frida*. URL: https://frida.re/.

[17] *General Data Protection Regulation*. 2022. URL: https://gdpr-info.eu/.

[18] A. Girish, T. Hu, V. Prakash, D. J. Dubois, S. Matic, D. Y. Huang, S. Egelman, J. Reardon, J. Tapiador, D. Choffnes, and N. Vallina-Rodriguez. "In the Room Where It Happens: Characterizing Local Communication and Threats in Smart Homes". In: *IMC*. 2023. DOI: 10.1145/3618257.3624830.

[19] C. Han, I. Reyes, Á. Feal, J. Reardon, P. Wijesekera, N. Vallina-Rodriguez, A. Elazari, K. A. Bamberger, and S. Egelman. "The Price is (Not) Right: Comparing Privacy in Free and Paid Apps". In: *PETS*. 2020. DOI: 10.2478/popets-2020-0050.

[20] H. Hosseini, M. Degeling, C. Utz, and T. Hupperich. "Unifying Privacy Policy Detection". In: *PETS*. 2021. DOI: 10.2478/popets-2021-0081.

[21] M. Jakaria, D. Yuxing Huang, and A. Das. "Connecting the Dots: Tracing Data Endpoints in IoT Devices". In: *PETS*. 2024. DOI: 10.56553/popets-2024-0090.

[22] K. Kollnig, A. Shuba, R. Binns, M. Van Kleek, and N. Shadbolt. "Are iPhones Really Better for Privacy? A Comparative Study of iOS and Android Apps". In: *PETS*. 2022. DOI: 10.2478/popets-2022-0033.

[23] S. Lazzaro, V. De Angelis, A. M. Mandalari, and F. Buccafurri. "Is Your Kettle Smarter Than a Hacker? A Scalable Tool for Assessing Replay Attack Vulnerabilities on Consumer IoT Devices". In: *IEEE PerCom*. 2024. DOI: 10.1109/PerCom59722.2024.10494466.

[24] F. Loi, A. Sivanathan, H. H. Gharakheili, A. Radford, and V. Sivaraman. "Systematically Evaluating Security and Privacy for Consumer IoT Devices". In: *IoTS&P Workshop*. 2017. DOI: 10.1145/3139937.3139938.

[25] A. M. Mandalari, D. J. Dubois, R. Kolcun, M. T. Paracha, H. Haddadi, and D. Choffnes. "Blocking Without Breaking: Identification and Mitigation of Non-Essential IoT Traffic". In: *PETS*. 2021. DOI: 10.2478/popets-2021-0075.

[26] Matomo Analytics. *Personally identifiable information guide: a list of PII examples*. 2024. URL: https://matomo.org/personally-identifiable-information-guide-list-of-pii-examples/.

[27] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A.-R. Sadeghi, and S. Tarkoma. "IoT SENTINEL: Automated Device-Type Identification for Security Enforcement in IoT". In: *IEEE ICDCS*. 2017. DOI: 10.1109/ICDCS.2017.283.

[28] *mitmproxy*. URL: https://mitmproxy.org/.

[29] Y. Nan, X. Wang, L. Xing, X. Liao, R. Wu, J. Wu, Y. Zhang, and X. Wang. "Are You Spying on Me? Large-Scale Analysis on IoT Data Exposure through Companion Apps". In: *USENIX Security*. 2023. DOI: 10.5555/3620237.3620610.

[30] NIST. *Cybersecurity for IoT Program*. 2021. URL: https://nist.gov/itl/applied-cybersecurity/nist-cybersecurity-iot-program/consumer-iot-cybersecurity.

[31] T. OConnor, R. Mohamed, M. Miettinen, W. Enck, B. Reaves, and A.-R. Sadeghi. "HomeSnitch: Behavior Transparency and Control for Smart Home IoT Devices". In: *ACM WiSEC*. 2019. DOI: 10.1145/3317549.3323409.

[32] *PCAPdroid*. URL: https://github.com/emanuele-f/PCAPdroid.

[33] R. Perdisci, T. Papastergiou, O. Alrawi, and M. Antonakakis. "IoTFinder: Efficient Large-Scale Identification of IoT Devices via Passive DNS Traffic Analysis". In: *IEEE EuroS&P*. 2020. DOI: 10.1109/EuroSP48549.2020.00037.

[34] A. Pradeep, M. T. Paracha, P. Bhowmick, A. Davanian, A. Razaghpanah, T. Chung, M. Lindorfer, N. Vallina-Rodriguez, D. Levin, and D. Choffnes. "A Comparative Analysis of Certificate Pinning in Android & iOS". In: *IMC*. 2022. DOI: 10.1145/3517745.3561439.

[35] A. Razaghpanah, R. Nithyanand, N. Vallina-Rodriguez, S. Sundaresan, M. Allman, C. Kreibich, and P. Gill. "Apps, Trackers, Privacy, and Regulators: A Global Study of the Mobile Tracking Ecosystem". In: *NDSS*. 2018. DOI: 10.14722/ndss.2018.23009.

[36] N. Redini, A. Continella, D. Das, G. De Pasquale, N. Spahn, A. Machiry, A. Bianchi, C. Kruegel, and G. Vigna. "DIANE: Identifying Fuzzing Triggers in Apps to Generate Under-constrained Inputs for IoT Devices". In: *IEEE S&P*. 2021. DOI: 10.1109/SP40001.2021.00066.

[37] Y. Rekhter, B. Moskowitz, and D. Karrenberg. *Address Allocation for Private Internets*. RFC 1918. 1996. URL: https://datatracker.ietf.org/doc/html/rfc1918.

[38] 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: *IMC*. 2019. DOI: 10.1145/3355369.3355577.

[39] J. Ren, M. Lindorfer, D. J. Dubois, A. Rao, D. Choffnes, and N. Vallina-Rodriguez. "Bug Fixes, Improvements,... and Privacy Leaks: A Longitudinal Study of PII Leaks Across Android App Versions". In: *NDSS*. 2018. DOI: 10.14722/ndss.2018.23143.

[40] J. Ren, A. Rao, M. Lindorfer, A. Legout, and D. Choffnes. "ReCon: Revealing and Controlling PII Leaks in Mobile Network Traffic". In: *MobiSys*. 2016. DOI: 10.1145/2906388.2906392.

[41] S. J. Saidi, S. Matic, O. Gasser, G. Smaragdakis, and A. Feldmann. "Deep Dive into the IoT Backend Ecosystem". In: *IMC*. 2022. DOI: 10.1145/3517745.3561431.

[42] D. Schmidt, A. Ponticello, M. Steinböck, K. Krombholz, and M. Lindorfer. "Analyzing the iOS Local Network Permission from a Technical and User Perspective". In: *IEEE S&P*. 2025. DOI: 10.1109/SP61157.2025.00045.

[43] D. Schmidt, C. Tagliaro, K. Borgolte, and M. Lindorfer. "IoTFlow: Inferring IoT Device Behavior at Scale through Static Mobile Companion App Analysis". In: *ACM CCS*. 2023. DOI: 10.1145/3576915.3623211.

[44] C. Tagliaro, F. Hahn, R. Sepe, A. Aceti, and M. Lindorfer. "I Still Know What You Watched Last Sunday: Security and Privacy of the HbbTV Protocol in the European Smart TV Landscape". In: *NDSS*. 2023. DOI: 10.14722/ndss.2023.24102.

[45] C. Tagliaro, M. Komsic, A. Continella, K. Borgolte, and M. Lindorfer. "Large-Scale Security Analysis of Real-World Backend Deployments Speaking IoT-Focused Protocols". In: *RAID*. 2024. DOI: 10.1145/3678890.3678899.

[46] R. Trimananda, J. Varmarken, A. Markopoulou, and B. Demsky. "Packet-Level Signatures for Smart Home Devices". In: *NDSS*. 2020. DOI: 10.14722/ndss.2020.24097.

[47] N. Vallina-Rodriguez, J. Shah, A. Finamore, Y. Grunenberger, K. Papagiannaki, H. Haddadi, and J. Crowcroft. "Breaking for Commercials: Characterizing Mobile Advertising". In: *IMC*. 2012. DOI: 10.1145/2398776.2398812.

[48] J. Varmarken, H. Le, A. Shuba, Z. Shafiq, and A. Markopoulou. "The TV is Smart and Full of Trackers: Measuring Smart TV Advertising and Tracking". In: *PETS*. 2020. DOI: 10.2478/popets-2020-0021.

[49] VirusTotal. *Get an IP Address Report*. URL: https://virustotal.readme.io/reference/ip-info.

[50] X. Wang, Y. Sun, S. Nanda, and X. Wang. "Looking from the Mirror: Evaluating IoT Device Security through Mobile Companion Apps". In: *USENIX Security*. 2019. DOI: 10.5555/3361338.3361418.

[51] W. Zhou, Y. Jia, Y. Yao, L. Zhu, L. Guan, Y. Mao, P. Liu, and Y. Zhang. "Discovering and Understanding the Security Hazards in the Interactions between IoT Devices, Mobile Apps, and Clouds on Smart Home Platforms". In: *USENIX Security*. 2019. DOI: 10.5555/3361338.3361417.

[52] C. Zuo, Z. Lin, and Y. Zhang. "Why Does Your Data Leak? Uncovering the Data Leakage in Cloud from Mobile Apps". In: *IEEE S&P*. 2019. DOI: 10.1109/SP.2019.00009.